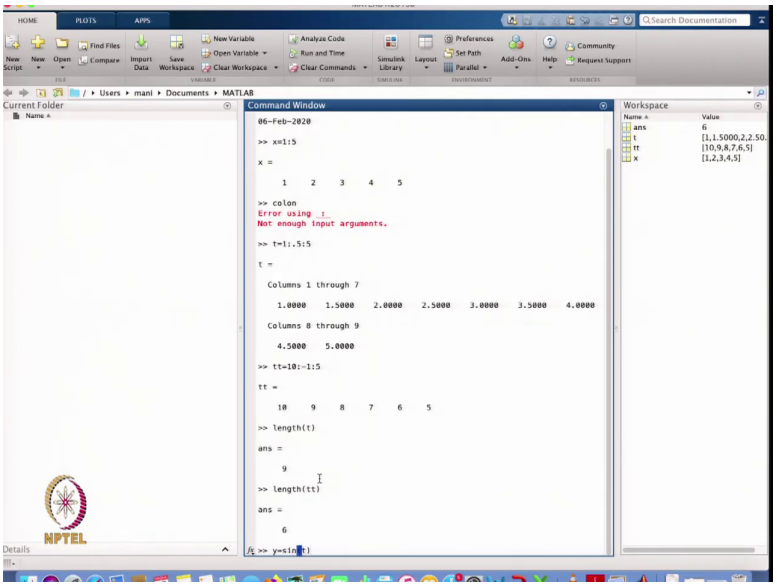
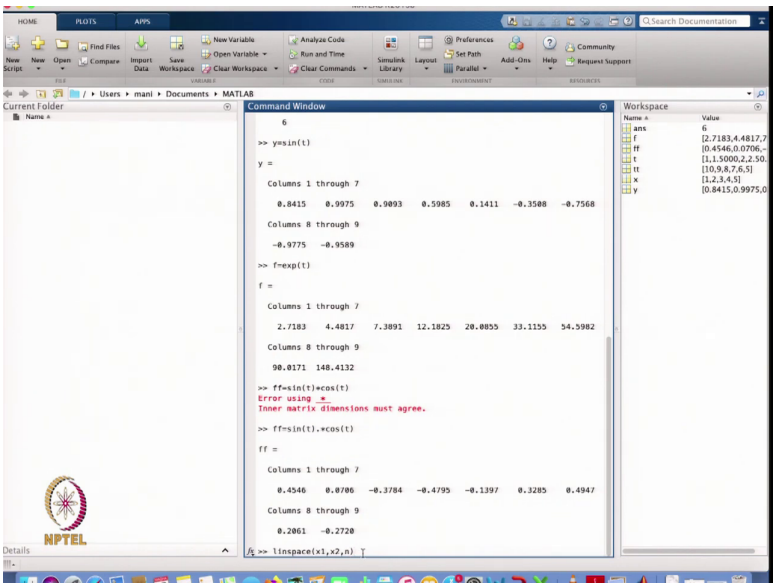
Scientific Computing Using Matlab
Professor. Vivek Aggarwal
Department of Mathematics
Indian Institute of Technology, Delhi
Professor. Mani Mehra
Department of Mathematics
Indian Institute of Technology, Delhi
Lecture No. 02
Plotting of functions in Matlab

Hello viewers, welcome back to this course. So, this is the second lecture of scientific computing using MATLAB. So, in the previous lecture we have just started with some basics of the MATLAB. Then in this lecture also will continue the basics of MATLAB. And today we will talk about how we can define the functions and how we can plot the functions, and then also we will talk about how we can do the symbolic computation. So, let us go to the MATLAB prompt and then start doing the programming there.

(Refer Slide Time: 0:50)





So, this is the MATLAB window. This is a command window, yesterday in the previous lecture, we also discussed this. So in the, now, today we will talk about how we can make the vectors and how we can plot the functions. So, today, let's suppose I write a function, command x is equal to 1:5, so if I put the colon so this is the colon sign, so colon means this is the colon sign. So, whenever I write x=1: 5, it gives you the equi-distributed points from 1 to 5.

So, in this case suppose I write t = 1 : 0.5 : 5, so that gives me the vector starting from 1 upto 5 with the sub interval having a width of 0.5. So, it is starting from 1, 1, 1.5 to 2.5 this one. So, this is used to create a mesh whenever we are plotting some function, because in this case, we are dealing with numerical computation, actually if you see there are two types of computation, numerical computation and the symbolic computation.

So, in the MATLAB, at present we are dealing with a numerical computation, means whenever I want to plot the function, I have to take the grid on x dimension in the x axis and in the similarly I want the vectors value of the function at this grid point, and then we able to plot this function even I can take another variable tt, so I even define the negative increment.

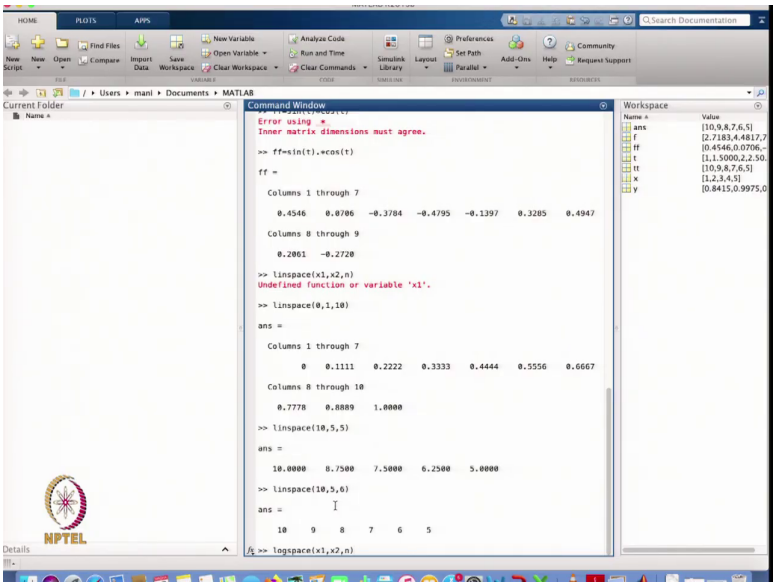
So, suppose I take the negative increment and 5, so it gives me the vector starting from 10 : -1 : 5, so it is 10, 9, 8, 7, 6, 5, and if I want to find the length of that vector, so I can define this command length of t, so it gives you the 9, length of tt that is the 6.

Now, suppose I want to define a vector or a matrix like this one, or suppose I just want to define a function, y = sin(t). So now, I define this function on the t, t is the vector, so that is the value of the function sin(t) given on the t mesh, so that was starting from 1 with the sub interval length 0.5 and going upto 5. So, this is a function y = sin(t), I can define the function on this one or even I can define the function another function f is equal to maybe exponential function exp over t.

So, that is the exponential function, I can define the other function double f = sin(t) cos(t). So, this is the cos(t) and here because these are the two vectors, so I have to define, multiply by the vector product, so, I have to put the dot here. So, now, this is the value, the function with this point starting sin(t) multiplied by the, so this is the vector multiplication and then we multiply with a cos(t) . So, this is the value of the new function.
f=sin(t).*cos(t)

Now, the same thing I can do with the help of another function that is called the linspace So, linspace it gives you the linear spacing between x_1 to x_2 with the n number of points.
linspace(x_1,x_2,n)
So, that is the command for linspace. It means that it is creating the vector starting from x_1 upto x_2 with n numbers of points in between. So, this is the linspace.

(Refer Slide Time: 5:26)



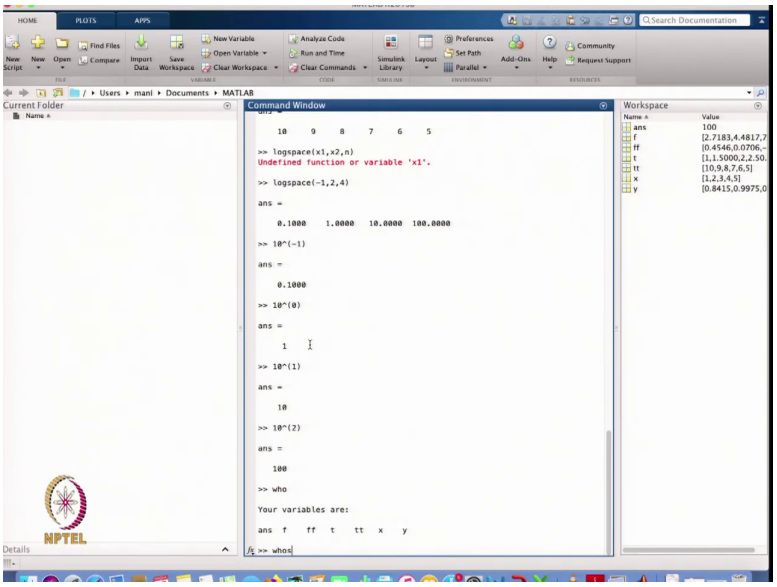
So, now, suppose I define the same command and I write starting from 0 upto 1 and then I give suppose 10, so it gives you the vector with the 10 component starting from 0 and the last element will be 1 with the 10 number of between this the 10 number of mesh points. So, starting from 0 it means, it is distributing the interval within the 9 sub interval and giving me the 10 points.
linspace(0,1,10)

So, similarly I can define, so whatever we have done in the with the with the help of colon, the same thing I can do with the help of linspace maybe I can define starting from 10 and then going upto 5 with the, so in this case I will just define 5, so that is a 5. So, starting from 10 to 5 and then I want to distribute the interval with the 5 points. So, it is 10, 8.7, 7.5, 6. Similarly, I can define the same one with the 6 element, and that is the 6 element, if I go it is 10, 9, 8, 7, 6, 5.

So, whatever the points you define with the starting point and the end point, it will distribute

linearly these points, all the sub points in between or sub all the mesh points in between. Now, I want to define another function that is logspace. So, logspace is also doing the same thing it is starting from x_1 upto x_2 and this is n numbers of points.
logspace(x_1,x_2,n)

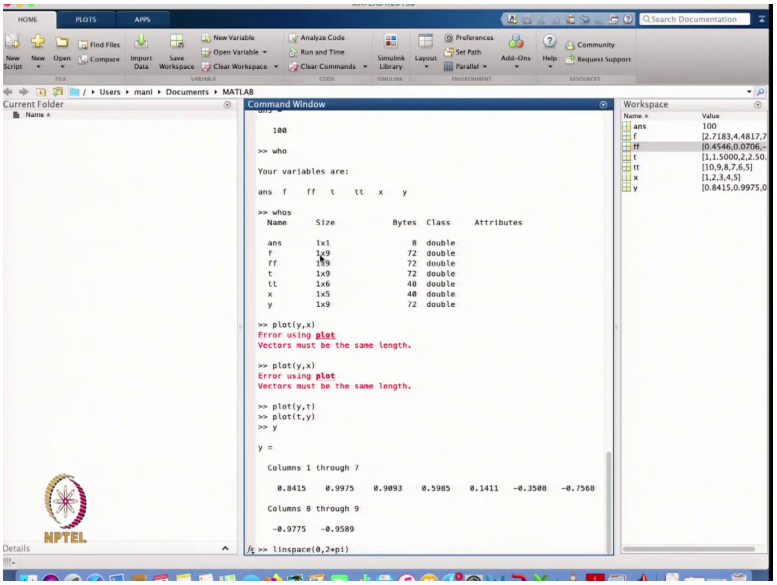
(Refer Slide Time: 7:20)



So, in this case it will distribute the points between the x1 and x2, n number of points with the logspace. So, that is my, so suppose I want to define here, so logspace I just take -1, another one I take 2 and suppose I divide by 4. So, that gives me the value of the logspace starting from -1, so minus 1 means, the first element will be 10^{-1}, so this is the 1 another element will be 10^{0}, so that is the 1, another value is 10^{1}, that is the value 10 and another 10^{2} and that is 100.

So, it is giving you the mesh with the help of logspace and starting from -1 to 2 and distributing the interval from -1 to 2 in the 4, with the 4 elements with the help of the logspace vector, so this is another way, logspace(-1,2,4) but in the linear space we are getting the linear mesh, but this is the log mesh or the non-linear mesh. So, this one I can define.

(Refer Slide Time: 8:38)

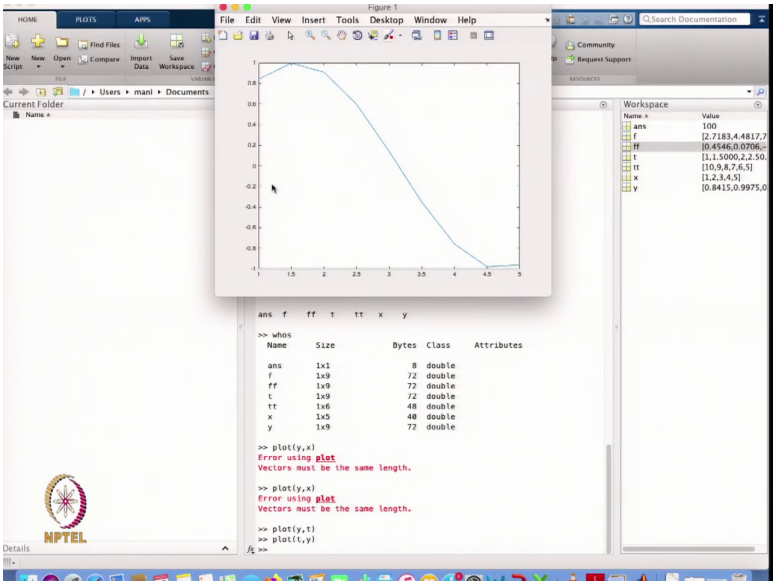


Now, I want to check what are the, my elements here or what are the variables here. So, I can define my who. So, who will give you what are the variables we are using in the workspace. So, this is the workspace we have. I have answers f, ff, t, tt. So, this variable I can show here in the command window by typing the command who and if I write the whos So, it will give you that which, what is the size of this. So, suppose the answer is taking one by one the scalar, it is taking the 8 bytes of the memory.

Similarly, I can define my tt, so it is a vector of the size 1 x 6 means it is a row vector. So, suppose I take this vector, so this shows me that the tt is a row vector with row 1 and the 6 columns and it is taking 48 bytes in the memory and the class is the double Similarly, I can define my y variable. So, y again the vectors a row vector and taking the 72 bytes. So, this one we can define.

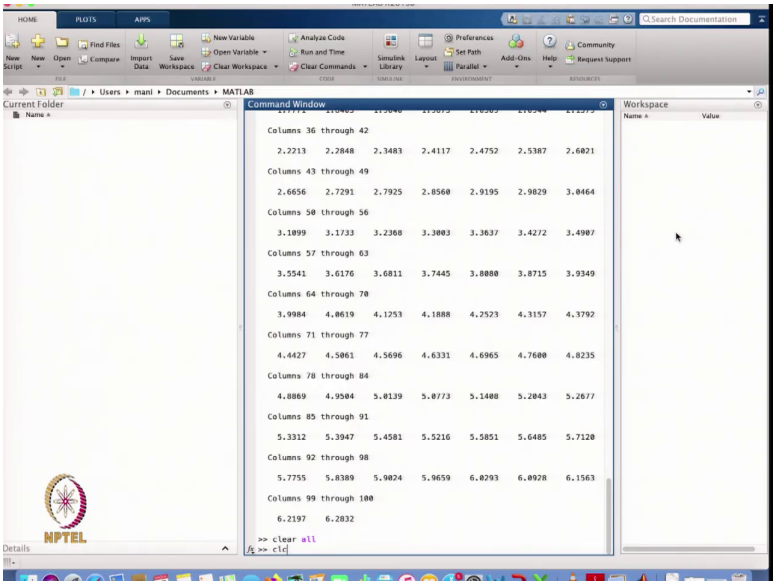
Now, suppose, I want to go a little bit further and I want to plot. So, I want to plot the function y(x) with respect to x. So, suppose I want to plot this one. So, this one, I want to define, what is the y. So, y is here, and it is whenever I define the y I was taking the value of this y over this interval t. So, I want to plot the y with respect to t.

(Refer Slide Time: 10:29)



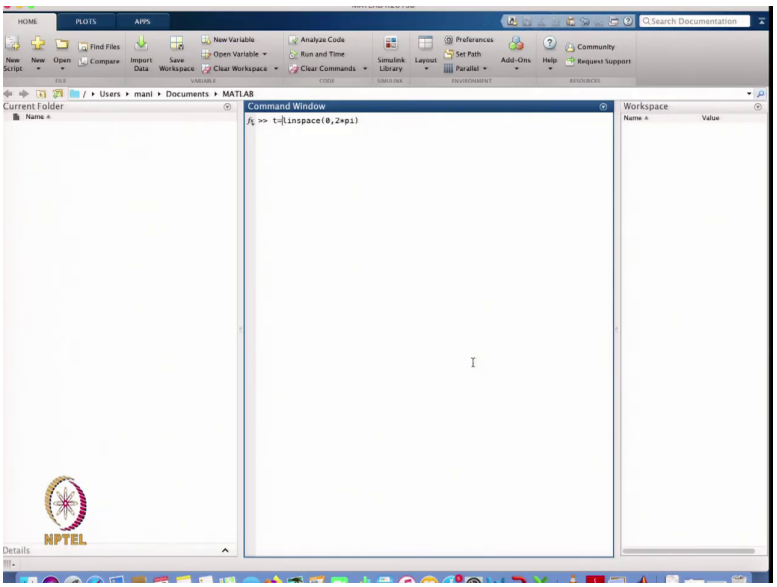
So, this one, so this is the I have to write t, this is the x variable and this is the y that is the plot. So, this is the plot whatever we have defined for y. And, what was my y here? So, y was this vector and I have defined this y when, so this y = sin t. So, we can see that, even I can define a vector linspace from 0 to 2 with, so suppose I do not give any number that how many numbers I want or how many mesh points I want. So, it will automatically distribute the element from 0 to 2 pi into 100 one.

(Refer Slide Time: 11:23)



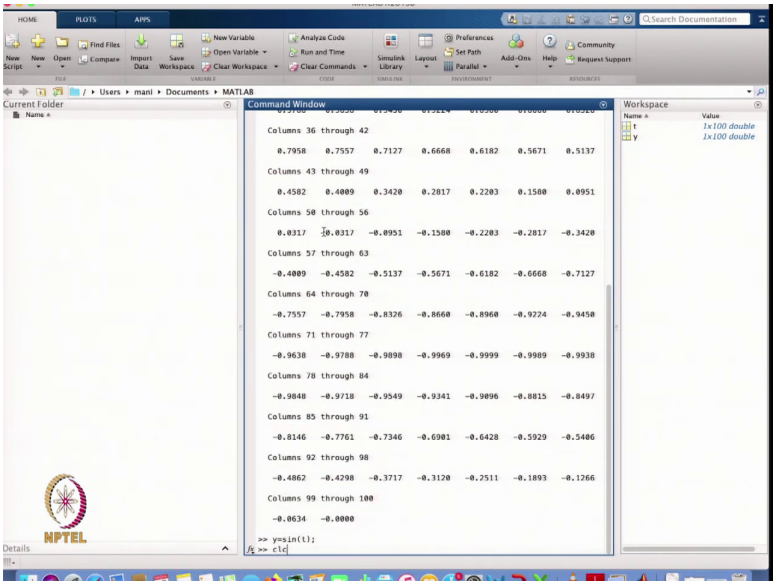
So, these are the 100 elements we are getting. So, that is the new mesh I got and this is I can give the name here maybe I can clear the previous variables. So, clear all, so it will clear all the variables that were available in the memory and I can also write clc to clear the screen.

(Refer Slide Time: 11:46)



Now, I define suppose that the same command I gave in the previous command, so if you want to see that how many commands you have typed in the previous one, then the arrow, the upper arrow, you can just put the upper arrow in your computer or in your computer then this will show you that what are the commands you have recently typed and using this one, then no need to re type the commands again. So, I have typed this command. So, this command I can show here. And let us define this one as t is equal to this.

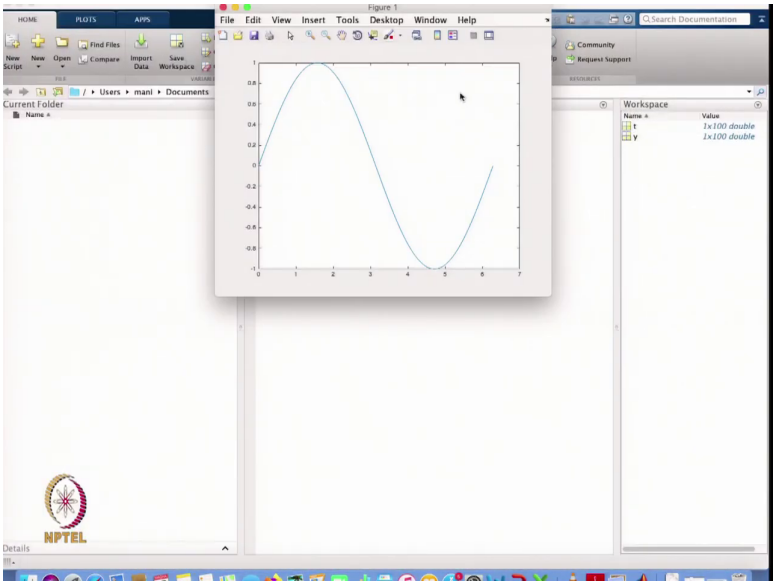
(Refer Slide Time: 12:24)

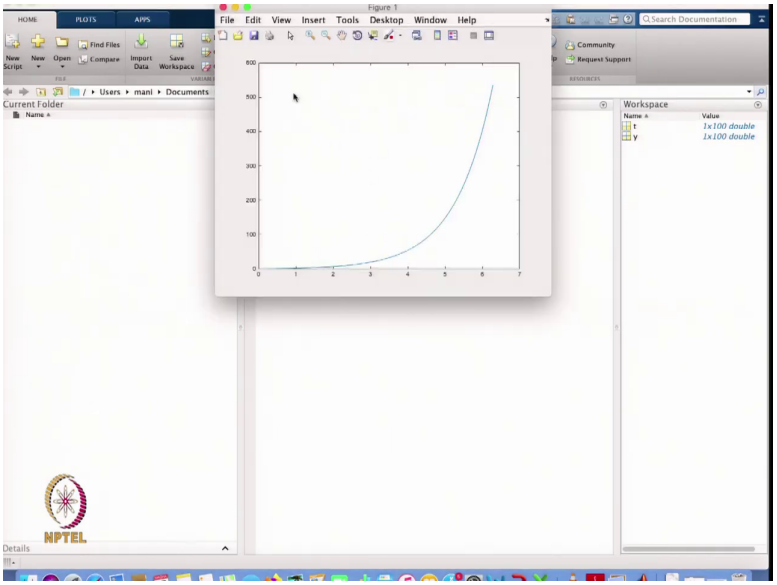


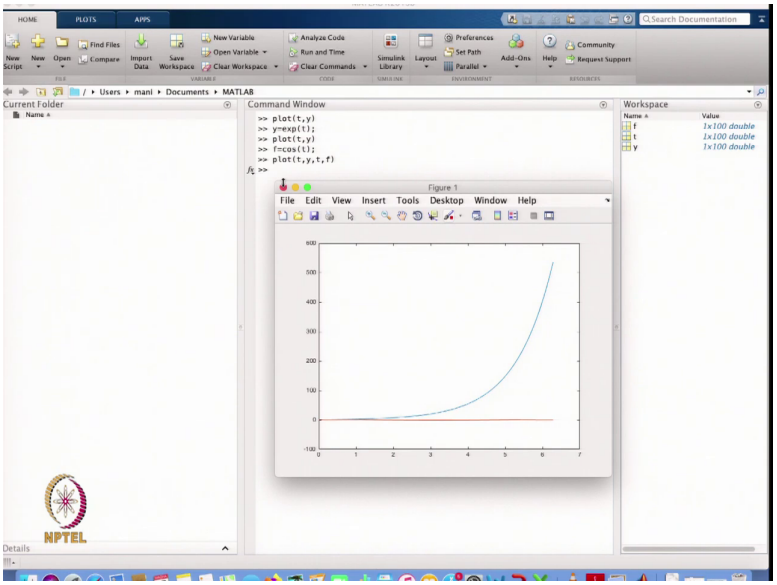
So, that is the elements we are getting even and I do not want to show the results on the screen. So, in this case, I can put the semi colons. So, semi colons as we have discussed earlier also, that semicolon will suppress the output to appear on the command window, but here my t is the variable, I can define that whos, so it is a t variable having a size the row

vector this one and taking this much of memory. Now, I define the function y = sin t. So, that is my value of the function, in this case.

(Refer Slide Time: 13:07)





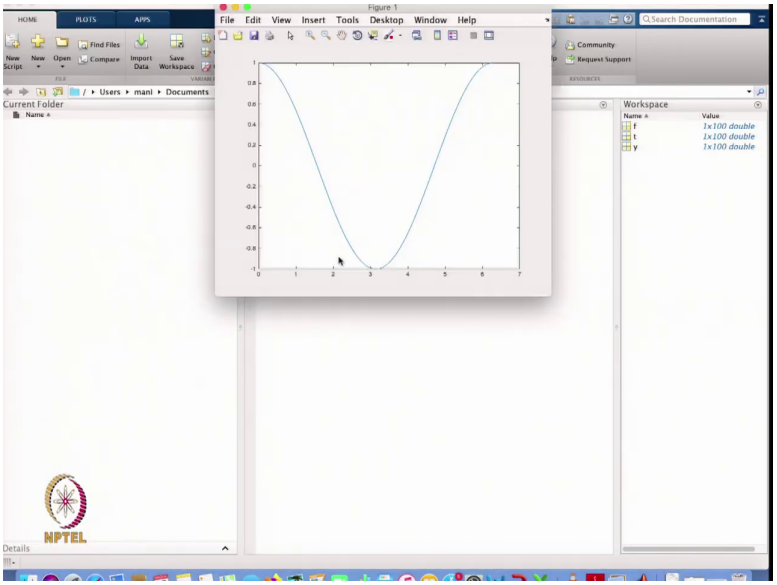


Now, I can plot. So, if I plot this one with t and y, so that is the plot we are getting. So, you can see that this is a sin wave starting from 0 to 2π, even I can define the another function maybe y is equal to exp(t) this one then I can plot this function. So, this is the exponential function just now we have defined or maybe suppose I take sin t. So, I define another function f = cos t, so this is my cos t.

Now, I want to plot. So, in this case, if I want to plot two functions together, that is also possible. So, I will write t, but they should be of the same length, so t, y then t, f. So, this one I plot. And, that is the plot. So, you can see that one is the plot for cos function and another one is the plot for exponential. So, this is the exponential by the blue color and red is giving you the plot for the f cos t, but here it looks like that this function is a straight line, because here the magnitude of the exponential function is very high.

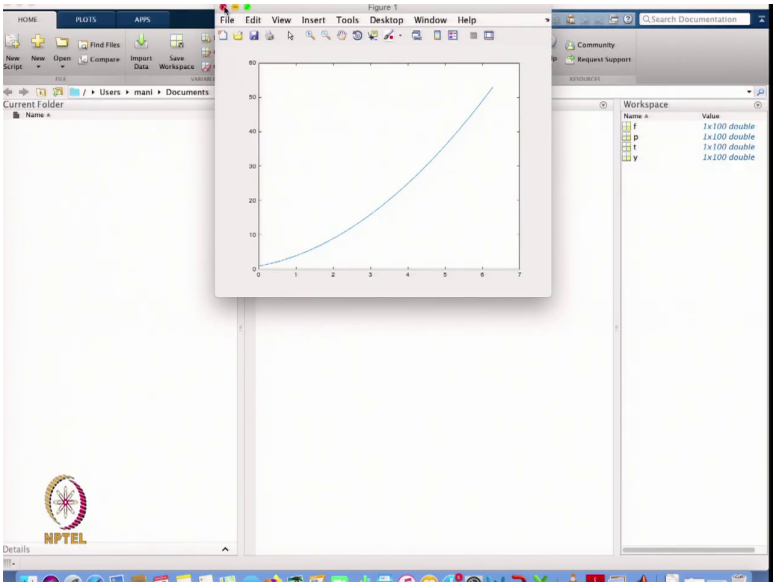
So, that is why related to this magnitude, the cos function we know that it is always lying between -1 to 1. So, it looked like a straight line.

(Refer Slide Time: 14:52)



But that is not. If I just want to plot this function separately, so it is the cos function we can see that, this is the cos function we have.

(Refer Slide Time: 15:07)

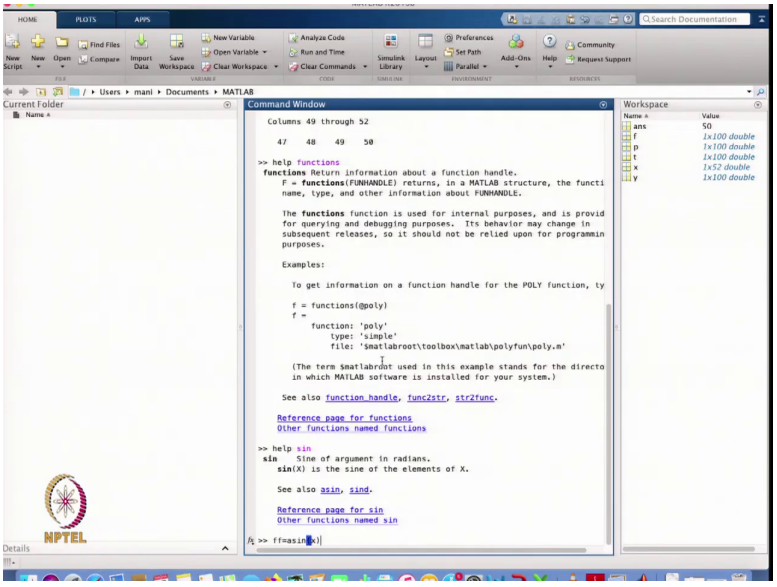


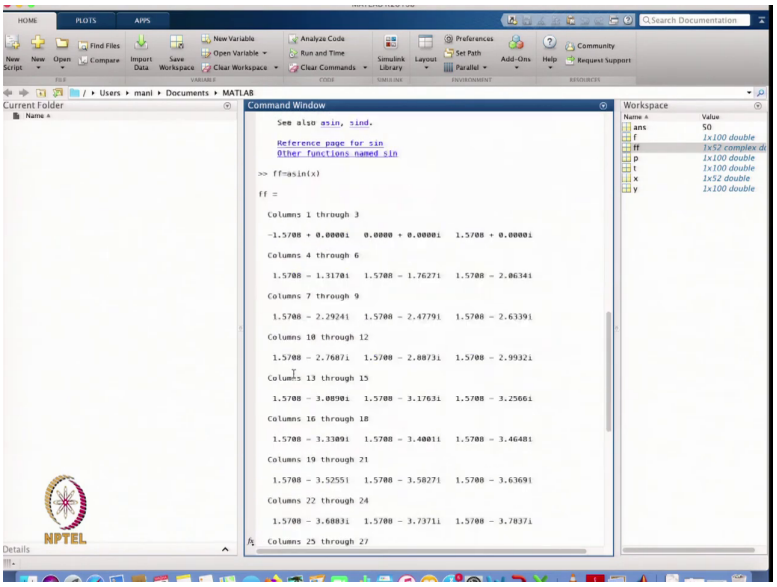
So, similarly, I can define the other function also. Now, suppose I take the polynomial function I can define that, I just defined a polynomial p, and so I just defined t exponential 2 plus t 2 star t plus 1, so this is the polynomial I am defining, and then I plot this one, t with p, so this is the polynomial we are getting.

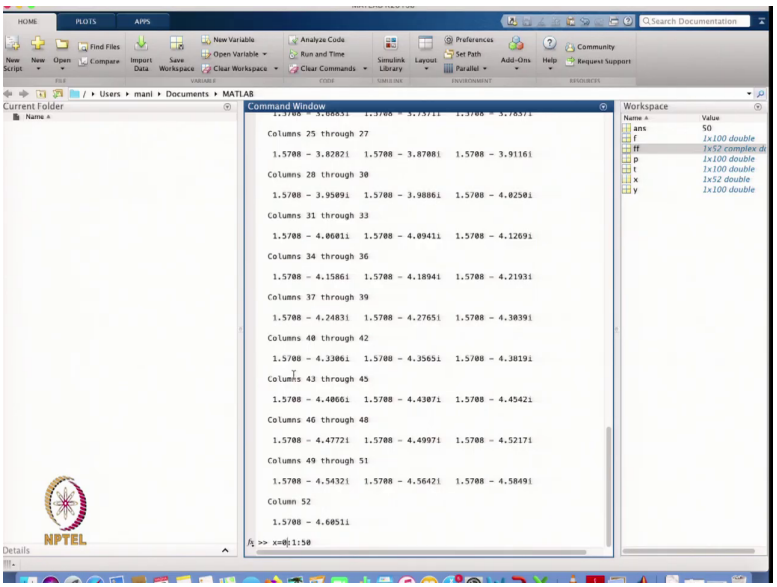
So, you can define lots of functions whatever the function you want to define even I can define the inverse function also. So, suppose I take variable x starting from 1 to, maybe I can

take -1 to 1 with 50 points. So, this is the, this should be semicolon. So, that is the mesh we got, Now, I can define the function ff= cos or sin inverse. So, I want to define how we can define the sin inverse. So, I can take the help functions.

(Refer Slide Time: 16:44)

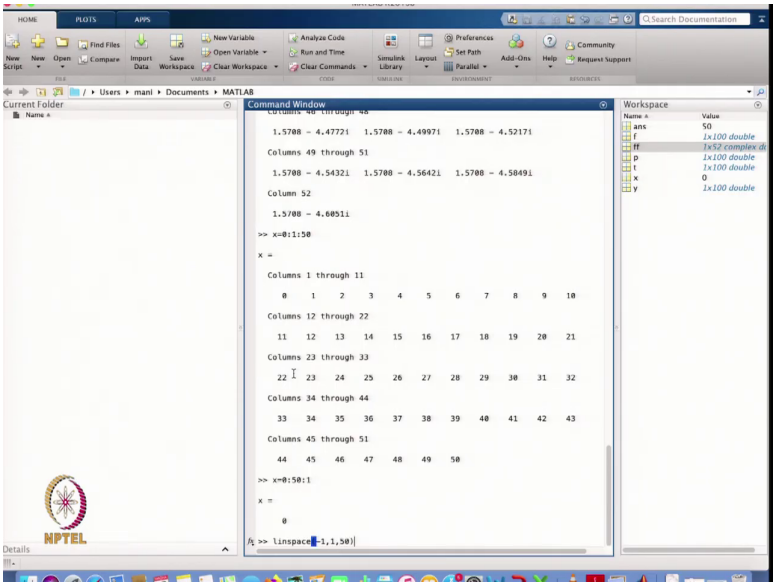


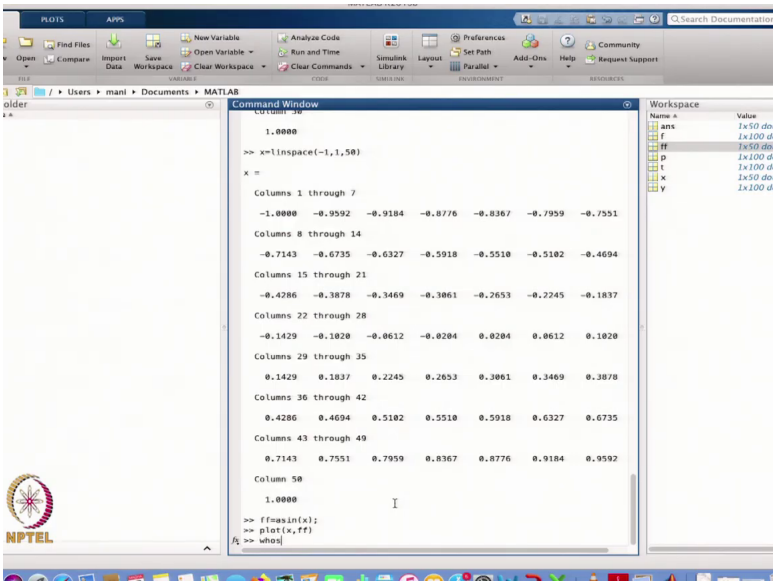




So, when I write the help functions, it will tell you which type of function we can define. So, here we can define the functions, maybe I can define sin function or asin means the inverse of the sin. So, let us define the function ff = asin(x). So, that is the function we are getting for asin, inverse of the sin. And, it is coming to the complex here.

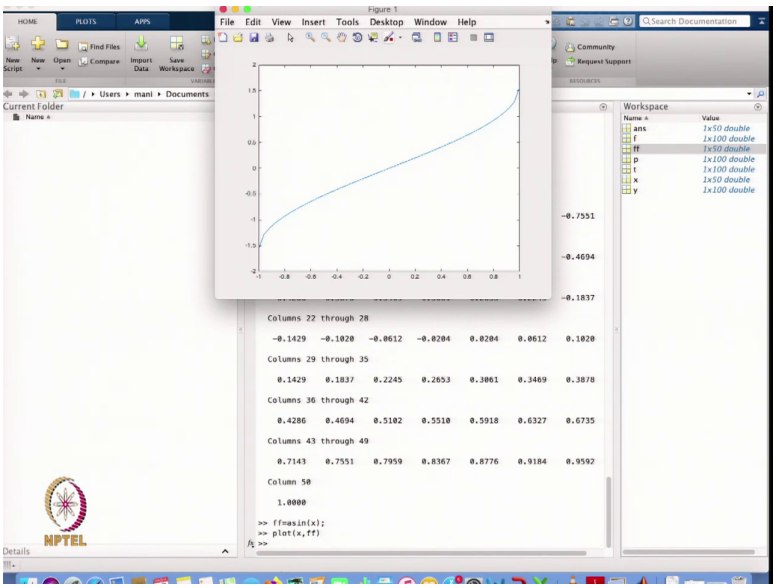
(Refer Slide Time: 17:36)





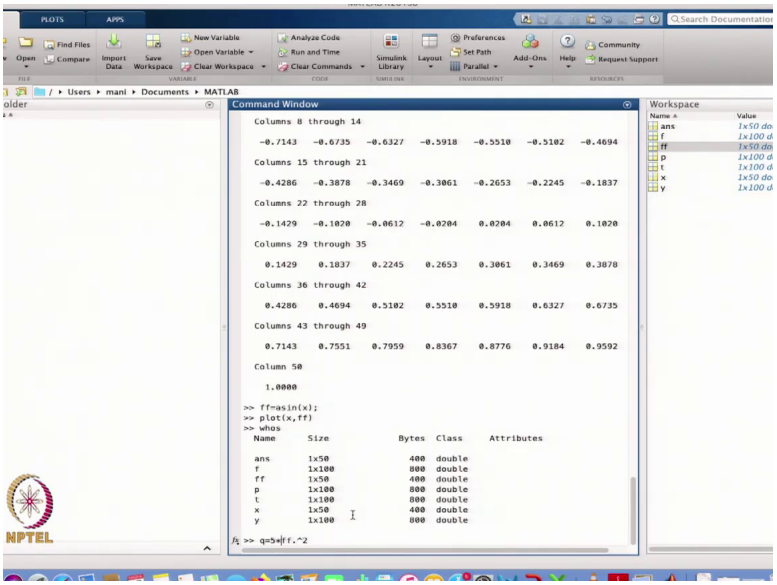
Because so here I should define my mesh from 0 to 1, that is it, this is, it should be, so it should be 0 to 1, 0 to, or maybe I can define this function with the linspace starting from -1 to 1 with 50 points, so that is the value we are getting. So, this I can now define as a x, x is equal to this, now I can define a function ff = asin (x).

(Refer Slide Time: 18:22)



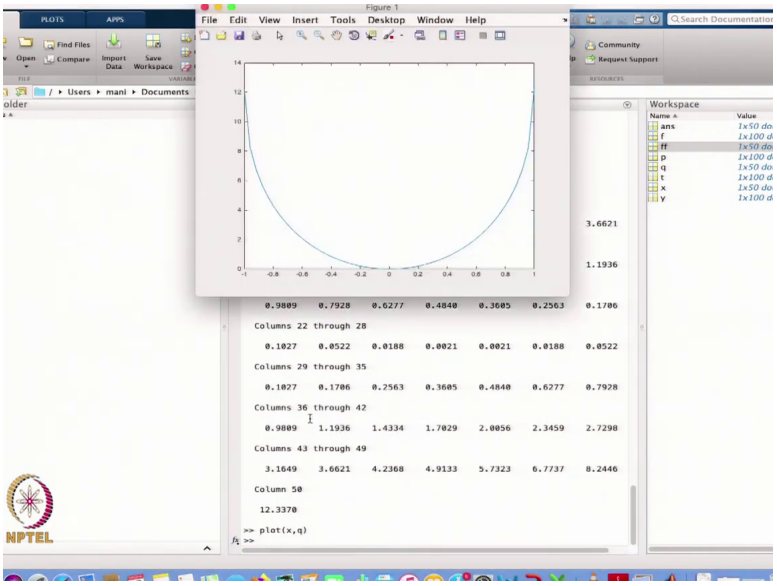
So, now I can plot this function plot this function x with this and ff. So, that is the plot we are getting. So, this is the plot of the sin inverse function, when x is defined from -1 to 1 with the 50 number of mesh points. So, this is the way we can define the function in an inverse. So, similarly we can define suppose I have, now ff, and I want to, so I have two types of functions.

(Refer Slide Time: 19:04)



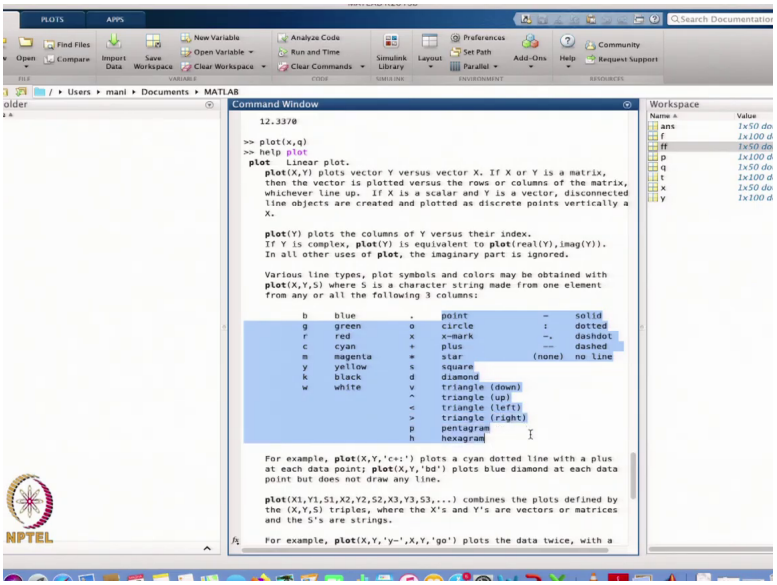
So, this is, now I have two types of functions, one is f, another is ff. So, I know, suppose I want to define the new function, so I take another function as a q and that is defined ff^{*2}. So, this is a function, a new function I have defined multiply by maybe I should define here multiply by 5. So, this is a function, a new function I have defined.

(Refer Slide Time: 19:39)



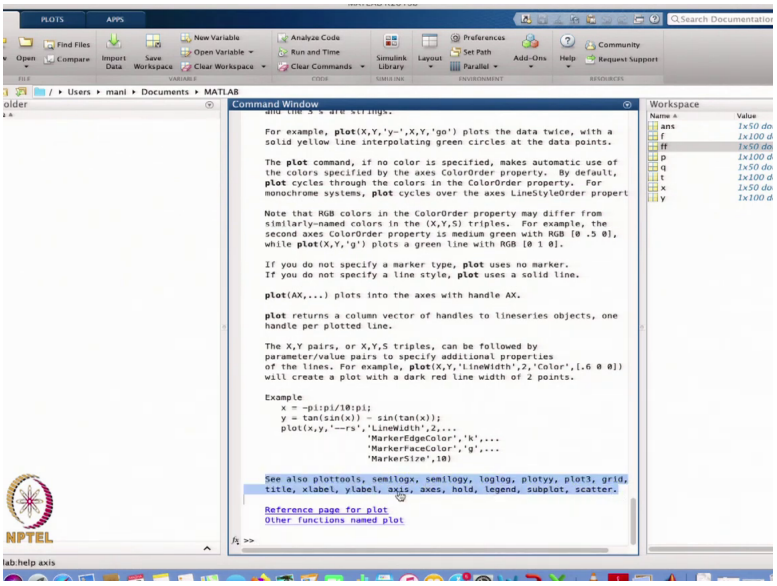
And suppose I want to plot this new function. So, this is a q, and that is a new function whatever we have defined just now, so that is a plot of that function. Now, even I can give the name to this plot. What is the name of this plot. So, this plot I can give the name as I can give the title, so I can write help plot.

(Refer Slide Time: 20:04)



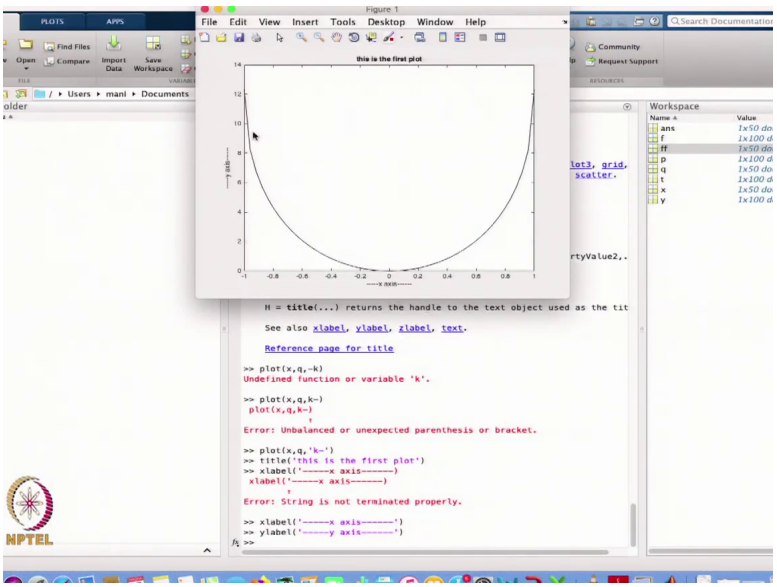
So, help plot give you all this that how we can plot this are the plot with the linear plot, this is the x variable, this is a y variable and then we can give that, what color should be the define for the plot. So, this is the different different color we have and this is the different different styles of the plot that how should be the style, it should be a point, it should be a circle and this is the lines that which line we should follow. It is a solid line, dashed line, so all this we can define.

(Refer Slide Time: 20:39)



And then later on we can also define that like suppose I have, so this is the all the commands used for the plot function. For example, it is giving semi log x, semi log y, plot yy, plot 3 grid, title. So, title is the, that if I suppose I want to be the title to the plot So, this is that how we can define the title.

(Refer Slide Time: 21:04)

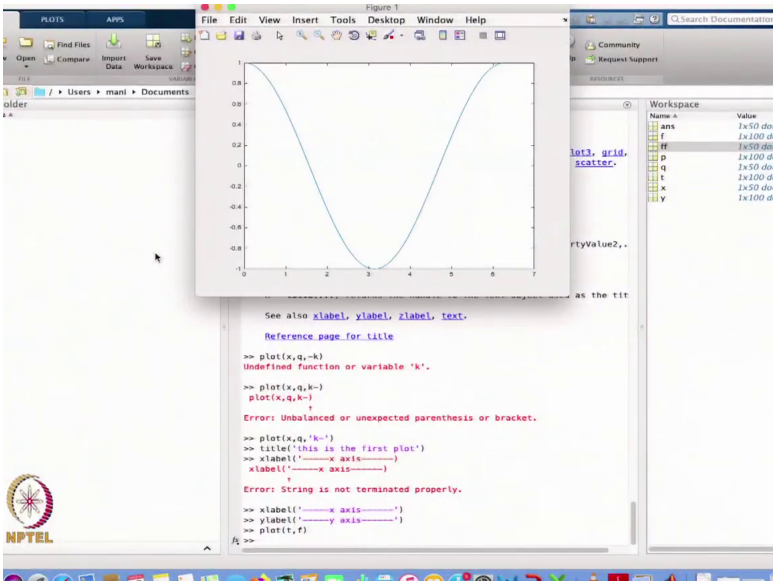


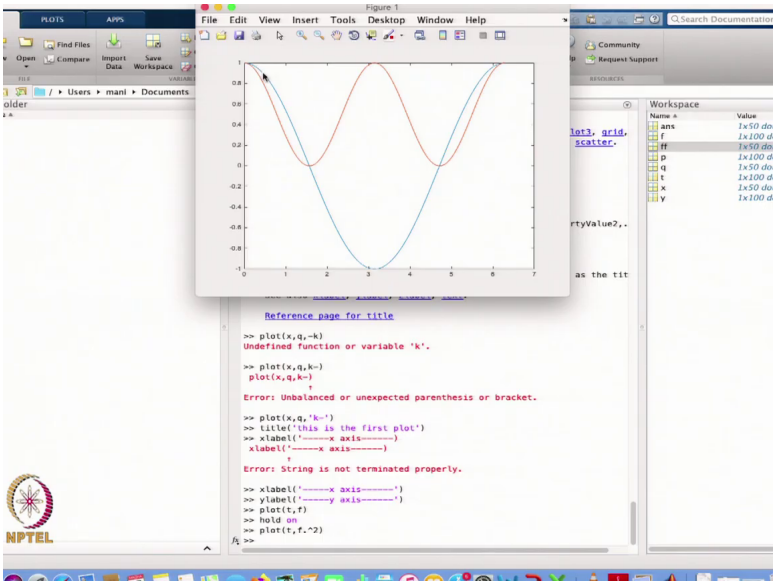
So, let us define this one plot I have done. So, now I want to give that it should be a k. So, that is the plot we have defined. So, this is the k, so we have to define this one in the colons you can see that this we have to define in the colons like this one. So, that is why it is giving the error. So, we have to define this one in colon.

And then this is the plot we got. Now, suppose I want to define the title to this plot. So, I write the title and this is a colon, inverted commas. So, I can write that this is the first plot. Now, you can see that the title is coming that this is the first plot or maybe I can define the x label or y label. So, x label means I can define the x axis. So, I can define the x label, so I write the x axis.

So, you can see that this is the x axis is coming. Similarly, I can define the same thing with the y axis, and that is the, so, now your plot is complete, this is x axis, this is the y axis and that is the title though even. So, all the things we can do with the help of all these commands given on the. Now suppose I want to plot another plot So, plot is suppose I take the same function. So, I have to check if f is defined and t is defined, so I just want to take t and f. So, this one I plot.

(Refer Slide Time: 23:20)

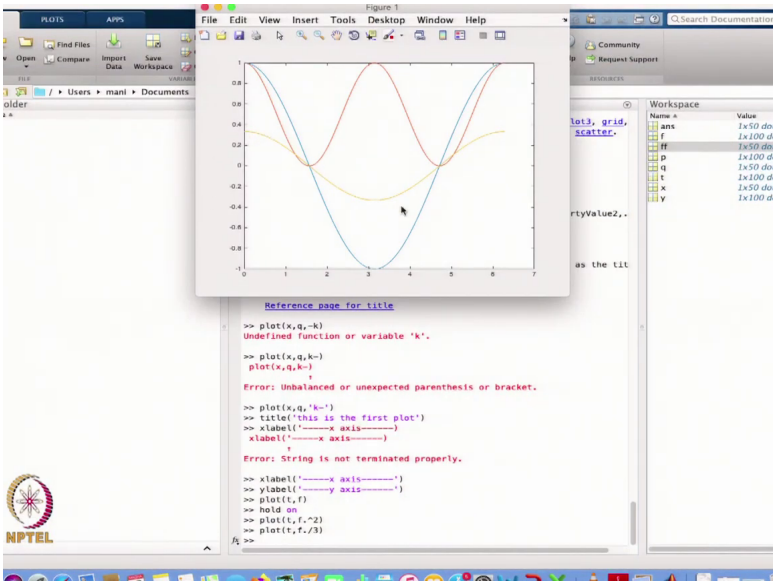




Now, you see that the previous plot has gone automatically, the new plot is coming, but the previous plot because a new plot is coming on the previous plot. So, the previous plot is now deleted and the new plot has come up, but suppose I want to plot the two plots on the same one, then we can define that I have done the plot 1 then I will put hold on. So, this is the hold on command this will hold on the previous plot.

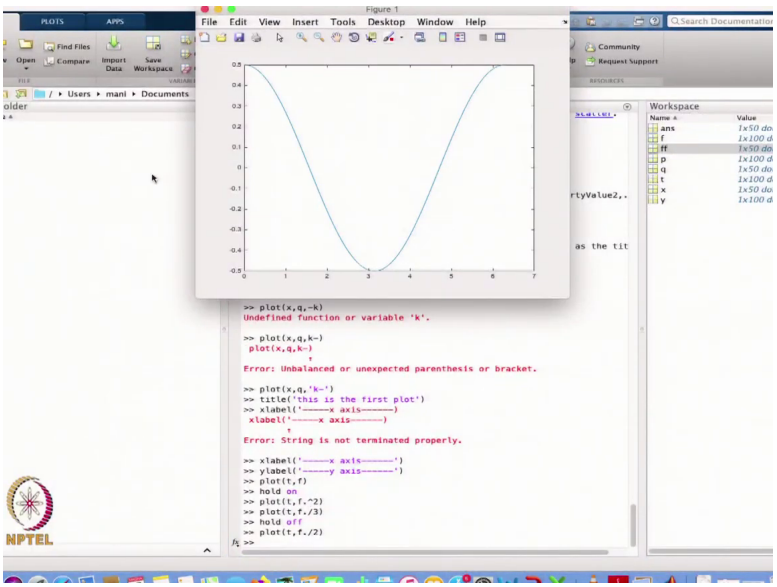
Now, suppose I take the another plot t with respect to f^{*2} , so instead of f, I am plotting now square in the, on the same plot, so it gives you another plot. So, you can see that it was the previous plot and this is the new plot we have plotted on the same graph by the command that is called the hold on.

(Refer Slide Time: 24:18)



Similarly, now I want to again plot the same command by f divided by 3. So, this is another plot we are getting. So, this is the third plot because sometimes in the scientific computing we need to plot different different graphs on the same plotting or different different plots on the same graph. So, this can be done with the help of the, the command, the hold on if I do not want to dismiss, I can write the hold off. So, this will hold off. Now, suppose I want to plot any function, new function by 2.

(Refer Slide Time: 24:59)



So, this plot is gone. Now, you get the new plot. So, these are the basic commands we have done today in this lecture about how we can deal with the graphics, or we can define the

function with the mesh in the MATLAB and we can, how we can plot this function. So, thanks very much for this. In the next one, we will go for more basics of the MATLAB course. Thanks very much.