In this video ,MATLAB will be used to plot a very simple function.

If you open MATLAB for the first time you should see an interface similar to the one in the video. On the top of the window is your toolbar, which is in essence equal to your toolbar in Microsoft Office applications. On the left you can see your current folder, which is a small version of your file explorer. On the right hand side you can see your workspace, this will be covered later in the video. In the middle you will see your command window.

So let’s start of with a very simple matlab program. To create a MATLAB program you have to choose whether you will create a script or a function. The main difference is that a function has input and output variables, whereas the scripts are the simplest type of programs, since they only store commands exactly as you would type them at the command line.

For this demo we will create a script. We are gonna create a new script and save the script as something familiars. For now I will place the script on my desktop and call it demo\_1. What we currently have is an empty script, that should not perform any arithmetics. So if we press run nothing should happen.

Now we get a message that our script is not on our MATLAB path. The path is a subsection of all your folders, which tells MALTAB where to look for files. Since our current script is not found, we should tell MATLAB to add the location, in this example the desktop, to the path, to allow matlab to use it. If we would now run the script we would not get any messages anymore.

Now lets have a look at the exercise. Plot a cosine wave with amplitude 10 frequency 5 hertz and a phase of pi/4 rad on the time domain [0, 0.5] with 1001 data points.

Comments in Matlab are expressed with percentage signs and these lines of codes will therefore be ignored by MATLAB. MATLAB does not use continuous variables because this would lengthen computation significantly and therefore can only create discrete variables. So lets start of with creating a variable t spanning from 0 to 0.5. This can be done using colons. These are used to separate 3 values. The first value is the starting point of the variable and the third value is the end point of the variable. The second value will indicate the size of the steps between different values of the variable. Let us think of this value. The total domain is 0.5 long with roughly 1000 values. So this would mean that the steps would be 0.5/1000 long.

Lets us run the script and see what happens. First of all you will probably see massive amounts of data in your command window. This can be prevented by placing a semicolon behind your line to suppress its output. Secondly you will notice that the variable t appeared in your workspace with size 1x1001. This is kinda strange because we divided by 1000 previously. However, this is completely correct and this will be demonstrated by a simple example. If we count from 1 to 10 we will have in total 10 different values. If we were to include 0 in the counting we would however have 11 different values. This is also the case in our script. Since we include 0 we will always get 1 value extra on top of the amount we divide by.

Now we will create a second variable x which represents the cosine wave. MATLAB requires a lot of multiplication and division signs, which you would normally not write, since MATLAB would otherwise confuse them with variables. The variable x will therefore be equal to 10\*cos(2\*pi\*5+pi/4) and lets add a semicolon to suppress the output. MATLAB already knows the value of pi so you should not enter 3.14 to approximate this number.

If we run this script again we can see that the variable x has appeared in the workspace, with the same size as t, since for every value of t we should have a value of x.

Now we will try to plot the function. To plot a function we will use the plot function of MATLAB which normally accepts 2 inputs. The first input is the data corresponding to the x-axis and the second input is the data corresponding to the y-axis. In our case we would write plot(t,x) and if we would now run the script a figure should pop up corresponding to our cosine wave.