=Plotting 2D graphs=

Welcome to the spoken tutorial on Plotting 2D graphs with scilab.

Assuming that Scilab is installed on your computer, we will discuss plots in Scilab.

Scilab offers many ways to create and customize various types of 2D and 3D plots.

The several common charts that Scilab can create are: x-y plots, contour plots, 3D plots, histograms, bar charts, etc...

Now open your Scilab console window. I will use Plotting.sce file to cut and paste the commands.

In order to Plot, we need a set of points. Let us create a sequence of equally spaced points. This can be done by the linspace command which creates a linearly equally spaced vector.

For Example

```
--> x=linspace(1,10,5)
```

x is a row vector with 5 points with linearly equally spaced between points 1 and 10

Similarly y is a row vector with linearly equally spaced 5 points between points 1 and 20

```
--> y=linspace(1,20,5);
```

More information on linspace command can be obtained from the Help documentation.

We will now we will plot a graph with the arguments x and y using the Plot function. This is similar to the one used in matlab.

```
--> plot(x,y)
```

creates a graph of x verses y as you see.

Notice that the graphics window is labeled as '0'

We will open another graphic window using the

```
xset('window',1)
```

function. I will close this cut the xset function paste in scilab hit enter.

You will see a graphic window number 1. Note that two arguments are passed to this function namely the window and 1.

The next graph will be plotted on this window.

For scilab plot2d is the native function used to plot 2d graphs.

```
--> plot2d(x,y,style=3)
```

plot2d command plot a graph of x verses y as you see.

Notice that there is a third argument called style. Style argument is optional.It is used to customize the appearance of the plot.

For positive values of style the curve is a plain line with different colours like green for 3 in our case. The default value of style is 1.Try plotting graphs for negative values and see the difference in appearance for yourself.

Also we can set the start points and end points for x and y axis by passing the fourth argument.It is called rect.

```
--> plot2d(x,y,style=3,rect=[1,1,10,20])
```

We have x axis starting from 1 to 10 and y axis from 1 to 20.

The order of argument in the rect command is xmin,ymin,xmax and ymax.

Let us now learn about Title,Axis and Legends

To configure labels to the axis and title of the plot we can use the commands

```
--> title("My title")

--> xlabel("X"); and

--> ylabel("Y");
```

I will cut this set of commands and paste in the console.

You will see that the graph has been labeled x to the x axis, y to the y axis and title of the graph is my title.

You may want to configure the title and axis of the plot in a single command instead of 3 for this purpose we use the xtitle

```
--> xtitle ( " My title " , " X axis " , " Y axis " );
```

command for all the 3 arguments. I will cut this command paste in scilab and hit enter.Now you see that the x axis label is X axis , Yaxis and the title is My title.

The clf() function that i am typing now will clear the graphic window as you see.

It is useful while plotting different graph on the same graphic window.

I will close this window.

Sometimes we need to compare two sets of data in the same 2D plot, that is, one set of x data and two sets of y data.

Let us see an example for this

I will scroll down

We will define the x axis points in a row vector x using the linspace command

Let us define a function

y1 = x^2

plot x verses y1

define another function

y2 = 2*x^2

plot x verses y2

we will also give label and title to our graph

Notice that we have additionally passed the ”o-” and ”+-” commands to the plot function,to change the appearance of the curve

```
--> x = linspace ( 1 , 10 , 50 );

--> y1 = x^2;

--> plot (x ,y1 , "o-")

--> y2 = 2*x ^2;

--> plot (x, y2, "+-")

--> xtitle ("My title" , "X axis" , "Y axis" );
```

These arguments are not a part of the plot2d function.They can be used only with the plot function

I will copy these set of commands and paste in the scilab console

You see the graph

Wouldn't it be of great help to know which curve is associated with which function?

This can be achieved using the legend command as you see

```
--> legend ( " x ^ 2 " , " 2* x ^ 2 " );
```

"o-" curve represents function y1=x^2 function and "+-" curve represents function y2=2*x^2

I will close this graphic window

[[File:title.jpeg]]

We will now discuss about plot2d demos and subplot function

Scilab provides demos for all its major functions

Demos of plot2d can be viewed through demonstration tab click on Graphics,click plot2d_3d plots and select a demo out the various demos provided. I will click on plot2d.

You will see the demo graph

The code for this graph can also be seen by clicking the view code button.

This link does not open in Mac OS but works in windows and linux

Nevertheless in Mac the code can be viewed through the directory.

Let us go to the terminal.

currently i am in demos directory of scilab 5.2

The full path to this directory is shown here.

we will type

```
ls
```

to see the list of demos available as you see here.

then we will select the 2d_3d_plots directory and hit enter

type ls again to see various demo code available in the sce files

```
ls
```

we will view the code for the demo which we have seen earlier type

```
more plot2d.dem.sce
```

and hit enter.

Here you will see the code for the demo graph of plot2d function

I will close the terminal

I will close the demo graph and the demos window

Similarly you can go through the other demos and explore scilab

Let us now discuss about Subplot function

The subplot() function divides a graphics window into a matrix of sub-windows.

To explain this function we will use demos for plotting 2D graphs in scilab.

for eg. type

```
-->plot2d()
```

on tour console and see the demo plot for this function

[[File:plot2d.jpg]]

I will close this window.

The subplot command breaks the graphics window into a 2 by 2 matrix of sub-windows represented by the first two arguments in the subplot command the third argument denotes the current window in which the plot will be plotted

I will execute this whole set of commands by copying in the scilab console

```
-->subplot(221)

-->plot2d()

-->subplot(222)

-->plot2d2()

-->subplot(2,2,3)

-->plot2d3()

-->subplot(2,2,4)

-->plot2d4()
```

[[File:subplot2.GIF]]

You can see 4 plots on a single plot window

The plot obtained can be saved as a image on your computer. Click on the graphic window, go to File menu select export to option.

Give a suitable title to your plot, select a destination folder to save your file select the file format in which you want your image to appear.

I will select the JPEG format and Click Save.

Browse through the directory to open the image and verify yourself whether it has been saved or not.

This brings us to the end of this spoken tutorial on Plotting 2D graphs using Scilab. There are many other functions in Scilab which will be covered in other spoken tutorials. Keep watching the Scilab links.

Spoken Tutorials are part of the Talk to a Teacher project, supported by the National Mission on Education through ICT. More information on the same is available at http://spoken-tutorial.org/NMEICT-Intro.

Thanks for joining. Goodbye.