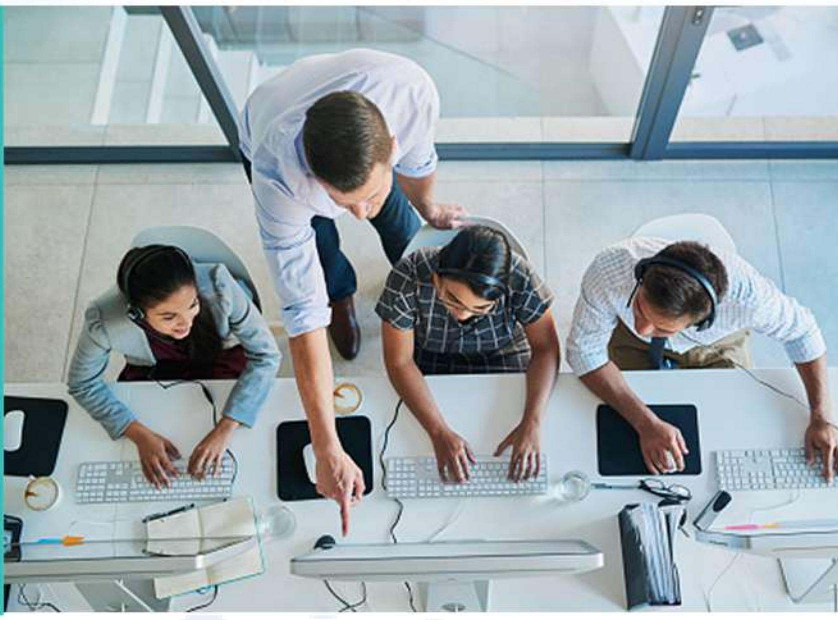
Andhra Pradesh State Skill Development Corporation



SciLab

Plotting in Scilab

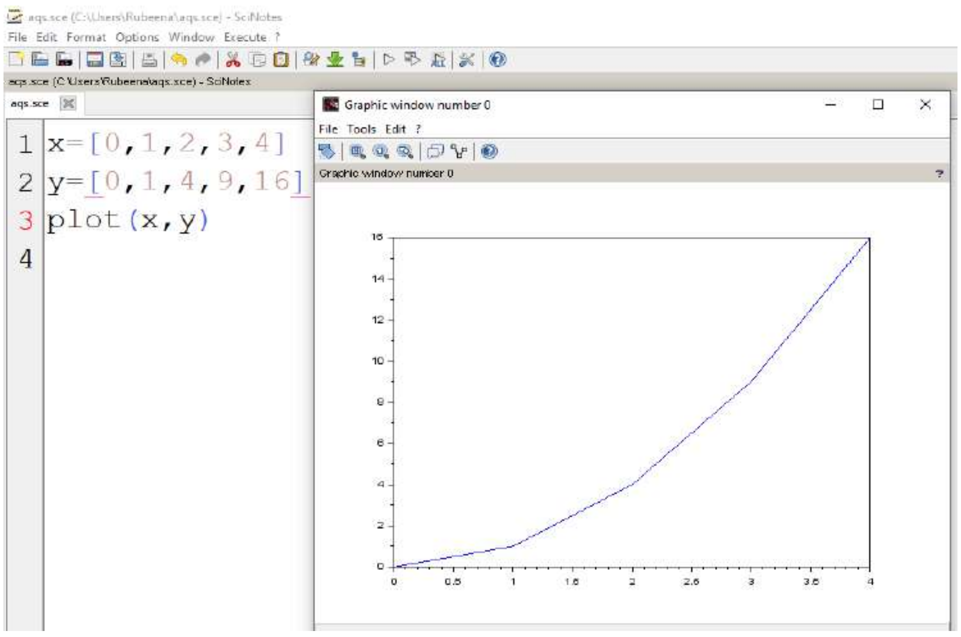
Plotting in Scilab

INTRODUCTION

Scilab has a large number of graphic commands to satisfy various graphic output requirements. We had earlier seen simple instances such as plot(x,y) which plotted the variable x against y, with default graphics settings. In this chapter we explore a selected group of commands for 2D and 3D plotting, in some detail:

Before we proceed, let us once again look closely at a graphic window of Scilab. Let us consider the simple example we are already familiar with:

- x=[0,1,2,3,4]
- y=[0,1,4,9,16]
- plot(x,y)


aqs.sce (C:\Users\Rubeena\aqs.sce) - SciNotes
File Edit Format Options Window Execute ?
aqs.sce
Graphic window number 0
File Tools Edit ?

```
x=[0,1,2,3,4]
y=[0,1,4,9,16]
plot(x,y)
```

Fig. A simple plot

We can note that the window title bar displays "Graphic Window number (0)" Scilab can open up to 20 such windows at the same time.

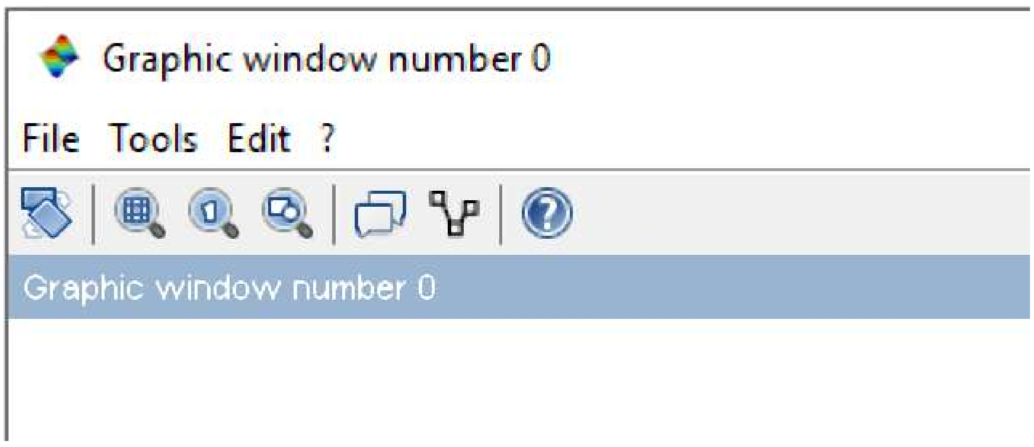


Fig. A Graphic window with menus and submenus

There are 7 buttons on the graphic window there are:

- Rotate
- Zoom area
- Original view
- Reframe to contents
- Toggle datatip mode
- Toggle curve data modification
- Help browser

There are three menus below the title bar, File,Tools and Edit. Let us look at the options in each menu. The File menu which is very frequently used has the following menu options:

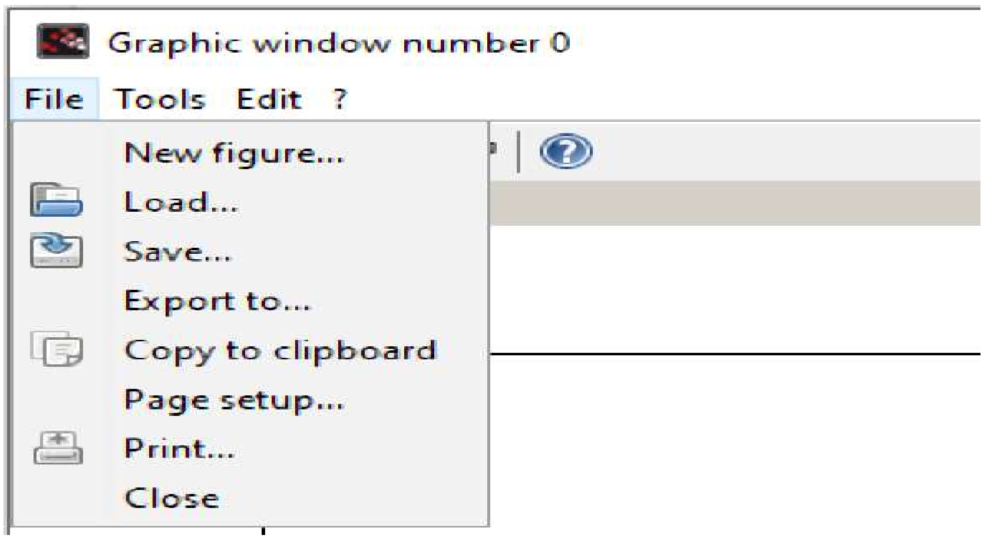


Fig. File Menu options

New: Open a new Scilab window

Load: Load a file

Save: Save the plot on a file with a specified name

Export: Export a file copy to clipboard

Page setup: Enable to page setup

Print: Enable printing

Close: Exit from Scilab graphics window

Options under Tools menu and edit menu are self explanatory and are given in figure.

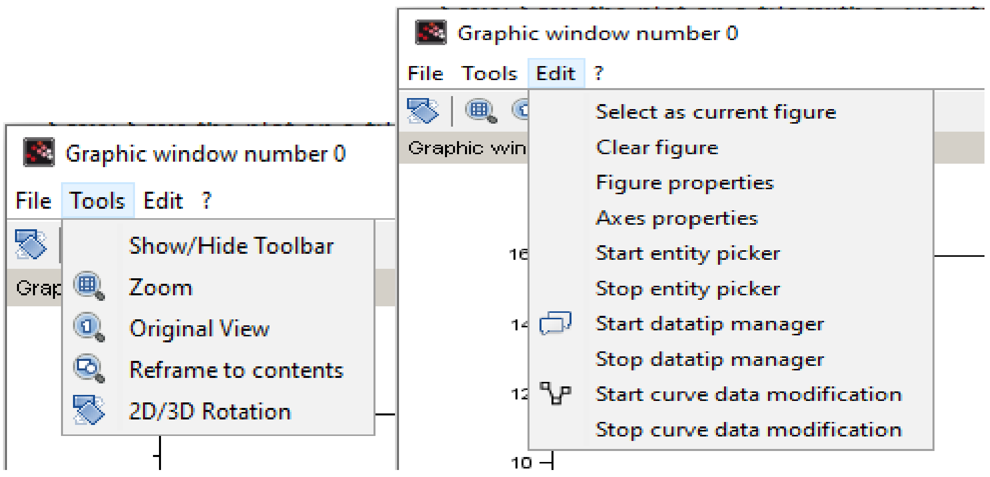


Fig. Tools and edit menu options

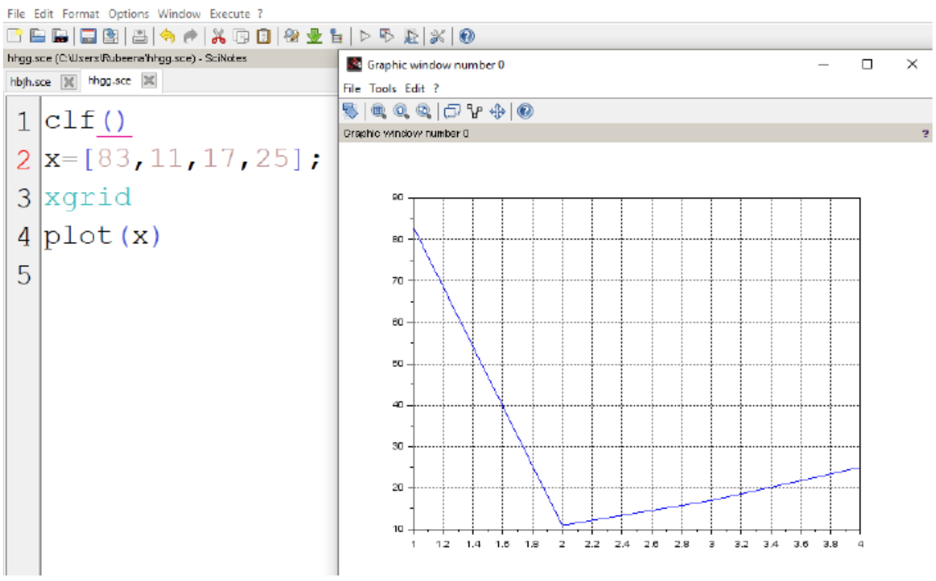
PLOT COMMAND

Plot command in Scilab attempts to simulate the corresponding Matlab command. Hence the syntax of this command is different from the other plot commands of Scilab which we will be addressing soon. There are two changes in the following example.

(i) Only one variable is indicated, and the other is implied (it is taken as the index of the array itself. I.e, y is used as [1, 2, 3, 4].

(ii) The ‘xgrid’ command displays windows using the clf( ) function, before attempting new graphic output.

- clf( )
- x=[83,11,17,25]
- xgrid
- plot(x)


File Edit Format Options Window Execute ?
hhgg.sce (C:\Users\Rubeena\hhgg.sce) - SciNotes
hbjh.sce
hhgg.sce
Graphic window number 0
File Tools Edit ?
Graphic window number 0

```
clf()
x=[83,11,17,25];
xgrid
plot(x)
```

Fig. A plot using x grid command

Colors & Styles:

We have an optional argument in plot command by which we can change the color, line style and marker style of the plot. This is used in the following format:

plot(x, ‘color linestyle markerstyle’)

Where x represents the data vector for plotting. The specifiers for the line style and marker style and color are shown in the given below.

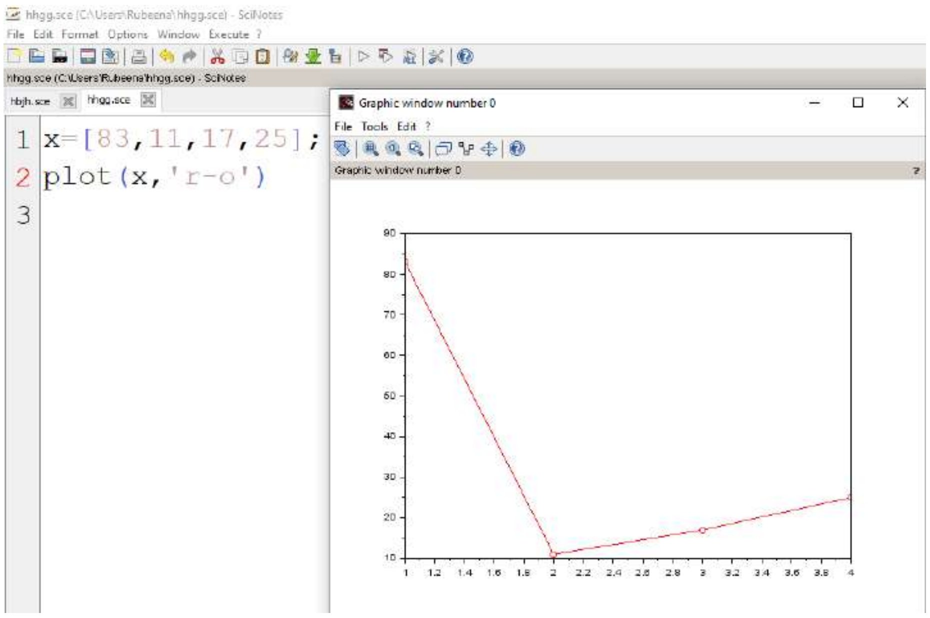
SPECIFIER-1	FOREGROUND COLOR
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black

SPECIFIER-2	LINE STYLE
-	Solid Line
_	Dashed Line
:	Dotted Line
.	Dash Dotted Line
SPECIFIER-2	**MARKER STYLE**
+	Plus sign
O	Circle
*	Asterk
.	Point
X	Cross
‘square’ or ‘s’	Square
‘diamond’ or ‘d’	Diamond
^	Upward pointing triangle
v	Downward pointing triangle
>	Right pointing triangle
<	Left pointing triangle
‘pentagram’	Five pointed star

Fig. Use of plot command along with xgrid command

We can try out different color, line style and marker style using the above table. Here is our first example

- x=[83,11,17,25]
- plot(x, ‘r-o’)


hhgg.sce (C:\Users\Rubeena\hhgg.sce) - SciNotes
File Edit Format Options Window Execute ?
Graphic window number 0
File Tools Edit ?

```
x=[83,11,17,25];
plot(x,'r-o')
```

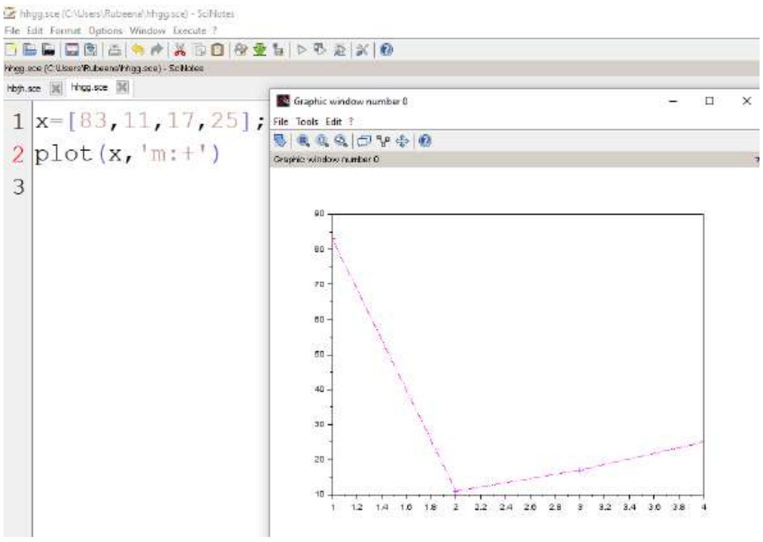
Fig. plot(x, ‘r-o’)

In the above command, the specifier ‘r’ indicates red color, the line style ‘-’ indicates solid line and marker style ‘d’ stands for diamond. The effects are seen on the plot.

Here are some other examples of specifiers.

- x=[83,11,17,25]
- plot(x, ‘m:+’)

In the above command, the color ‘m’ for pink color, line style is ‘**:**’ (colon) representing dotted line and marker style is ‘+’ representing the plus sign


hhgg.sce (C:\Users\Rubeena\hhgg.sce) - SciNotes
File Edit Format Options Window Execute ?
Graphic window number 0
File Tools Edit ?

```
x=[83,11,17,25];
plot(x,'m:+')
```

Fig. A plot drawn using pink color, dotted line style and ‘+’ marker style

scf and subplot functions:

scf ( ): set the current graphic figure (window)

The current figure is the destination of the graphic drawing. The scf function allows to change this current figure or to create it if it does not already exist.

scf(num) set the figure with figure_id==num as the current figure. If it does not already exist it is created.

scf(h) set the figure pointed to by the handle h as the current figure. If it does not already exist it is created.

scf( ) is equivalent to scf(max(winsid())+1). It may be used to create a new graphic window. For example of scf function

- x=[0:0.2:2*%pi]
- scf(0)
- y=sin(x)
- plot(x,y,'c')
- scf(8)
- z=cos(x)
- plot(x,z,'g')
- scf(4)
- d=rand(x)
- plot(x,d,'m')

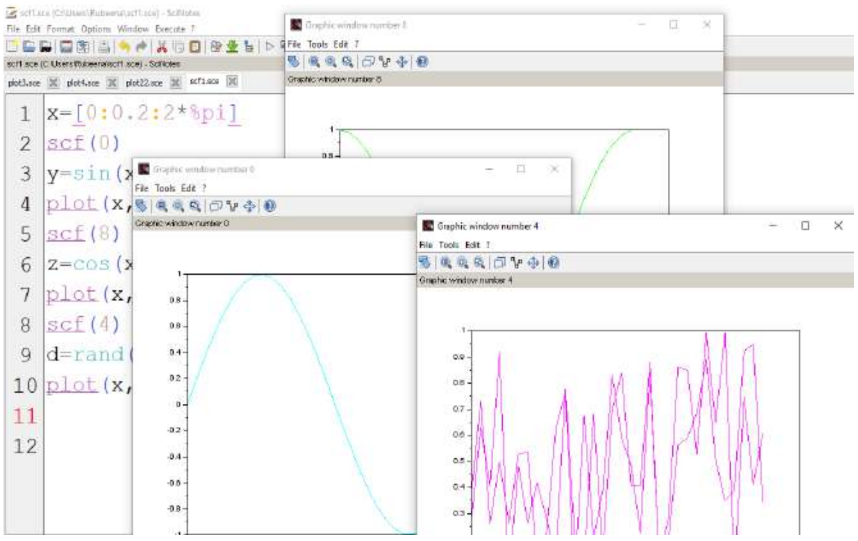

```
x=[0:0.2:2*%pi]
scf(0)
y=sin(x
plot(x,
scf(8)
z=cos(x
plot(x,
scf(4)
d=rand(
plot(x,
```

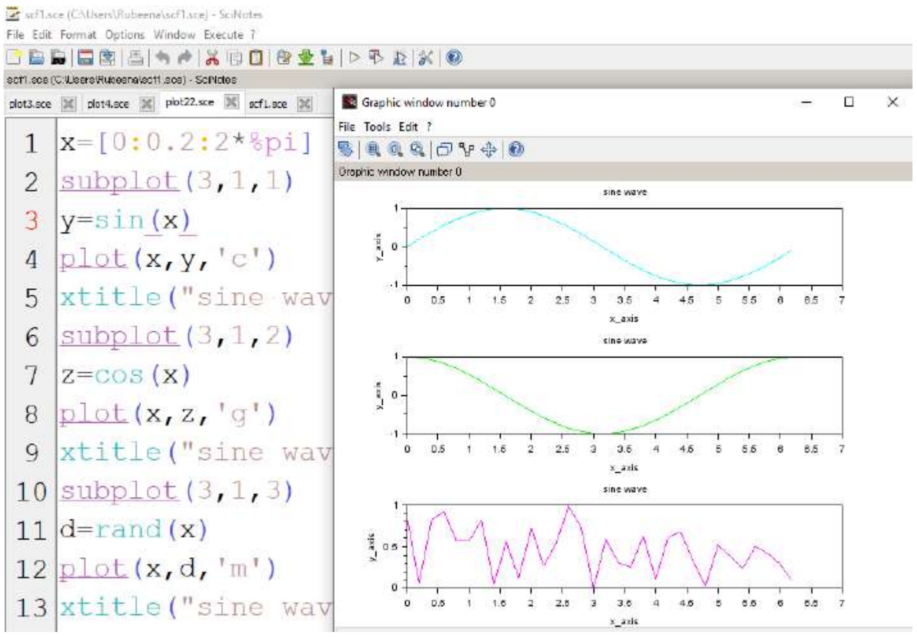
Fig. Separate Graphical windows using scf function

Subplot ( ): Divide a graphics window into a matrix of sub-windows

subplot(m,n,p) or subplot(mnp) breaks the graphics window into an m-by-n matrix of sub-windows and selects the p-th sub-window for drawing the current plot. The number of a sub-window into the matrices is counted row by row ie the sub-window corresponding to element (i,j) of the matrix has number (i-1)*n + j.

Here is an Example.

- x=[0:0.2:2*%pi]
- subplot(3,1,1)
- y=sin(x)
- plot(x,y,'c')
- xtitle("sine wave","x_axis","y_axis")
- subplot(3,1,2)
- z=cos(x)
- plot(x,z,'g')
- xtitle("sine wave","x_axis","y_axis")
- subplot(3,1,3)
- d=rand(x)
- plot(x,d,'m')
- xtitle("sinewave","x_axis","y_axis")



```
x=[0:0.2:2*%pi]
subplot(3,1,1)
y=sin(x)
plot(x,y,'c')
xtitle("sine wav
subplot(3,1,2)
z=cos(x)
plot(x,z,'g')
xtitle("sine wav
subplot(3,1,3)
d=rand(x)
plot(x,d,'m')
xtitle("sine wav
```

Fig. A plot drawn 1 columns and 3 separate rows using subplot function

Strings

The following are the set of string commands

xstring	: draw strings
xstringl	: compute a box which surrounds strings
xstringb	: draw strings into a box
title	: add titles on graphics window
xlabel	: add x-axis name on graphics window
ylabel	: add y-axis name on graphics window
xtitle	: add title, x-axis, y-axis names on graphics window

2D PLOTTING

We have seen that the plot command in Scilab follows Matlab. The general plot command is plot2d(string,x,y)

The staring has three parts “abc”. Each of these are explained below:

Part-a:

- c if y is empty.
- o “one”, if many ‘y’ are plotted against one x
- g “general” if x and y are both matrices of the same size and pair of rows from x and y.

Part-b:

- “n” normal (non logarithmic) scales
- “l” logarithmic scale in x-axis

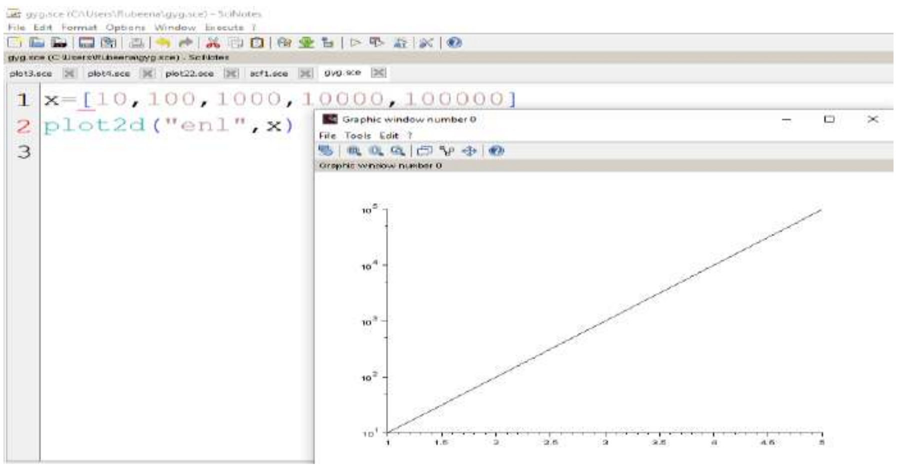
Part-c:

- “n” normal (non logarithmic) scales
- “l” logarithmic scale in y-axis

Let us see some illustrative examples

```
x=[10,100,1000,10000,100000]
plot2d(“enl”,x)
```

Here ‘e’ denotes that the variable y is empty (implied). “N” denotes the normal scale on the x-axis. “I” denotes a logarithmic scale on the y axis. Try changing ‘I’ to ‘n’ and compare the logarithmic and normal plots.

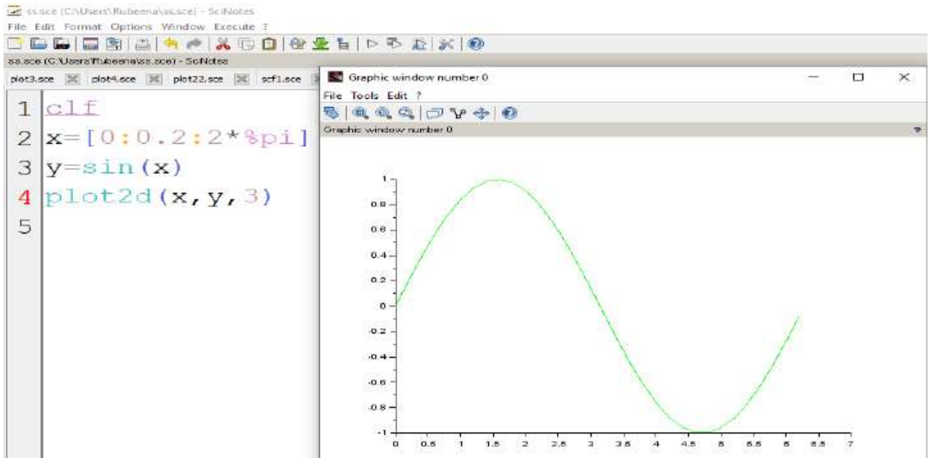

gyg.sce (C:\Users\Rubeena\gyg.sce) - SciNotes
File Edit Format Options Window Execute ?
plot3.sce
plot4.sce
plot22.sce
scf1.sce
gyg.sce
Graphic window number 0
File Tools Edit ?

```
x=[10,100,1000,10000,100000]
plot2d("enl",x)
```

Fig. A logarithmic plot generated with plot2D

Here is another example:

- clf
- x=[0:0.2:2*%pi]
- y=sin(x)
- plot2d(x,y,3)


ss.sce (C:\Users\Rubeena\ss.sce) - SciNotes
File Edit Format Options Window Execute ?
plot3.sce
plot4.sce
plot22.sce
scf1.sce
Graphic window number 0
File Tools Edit ?

```
clf
x=[0:0.2:2*%pi]
y=sin(x)
plot2d(x,y,3)
```

Fig. A simple plot 2D graph

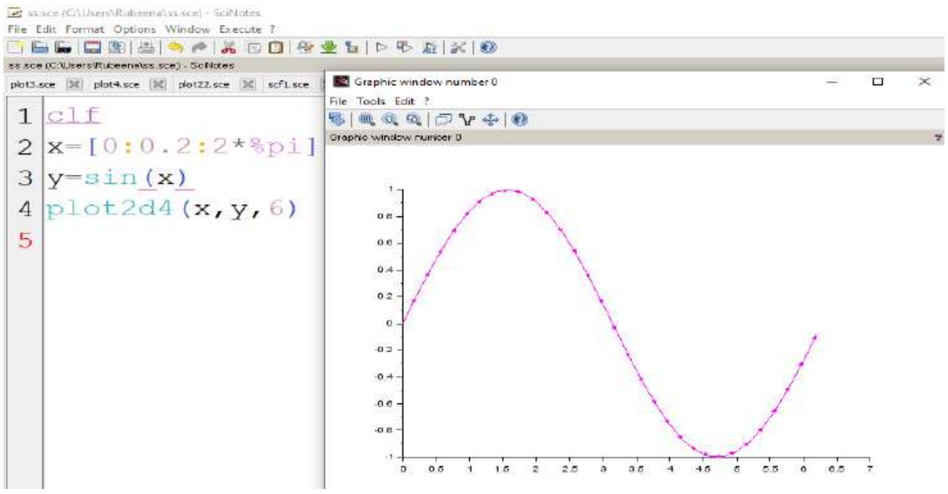
There are three types of 2D plotting

- Plot 2D2
- Plot 2D3
- Plot 2D4

Plot2d4:

plot2d4 is the same as plot2d but curves are plotted using arrows style. Another point to note is that plot2d accepts standard functions as arguments. In the following examples x is a vector with values from 0 to 2, in steps of 0.2.

- clf
- x=[0:0.2:2*%pi]
- y=sin(x)
- plot2d4(x,y,6)


ss.sce (C:\Users\Rubeena\ss.sce) - SciNotes
File Edit Format Options Window Execute ?
plot3.sce
plot4.sce
plot22.sce
scf1.sce
Graphic window number 0
File Tools Edit ?

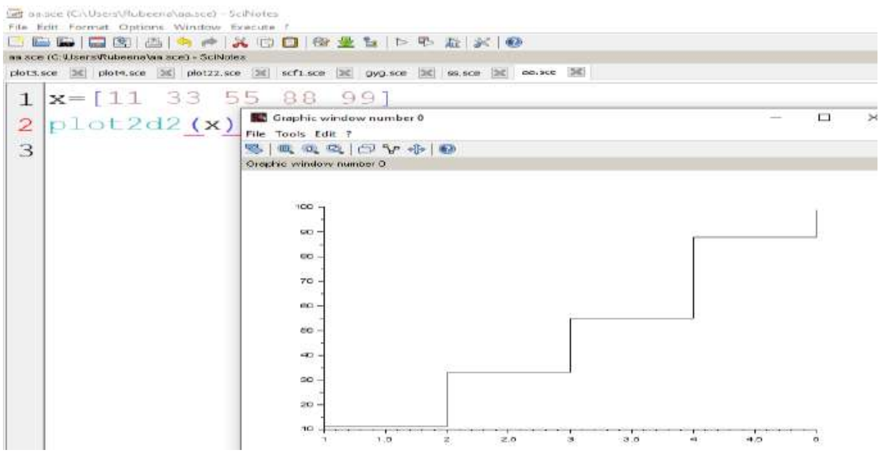
```
clf
x=[0:0.2:2*%pi]
y=sin(x)
plot2d4(x,y,6)
```

Fig. Arrows style generated using plot 2D4

Plot2d2

This curve is used to draw the 2D plot as a step function

- x=[11 33 55 88 99]
- plot2d2(x)

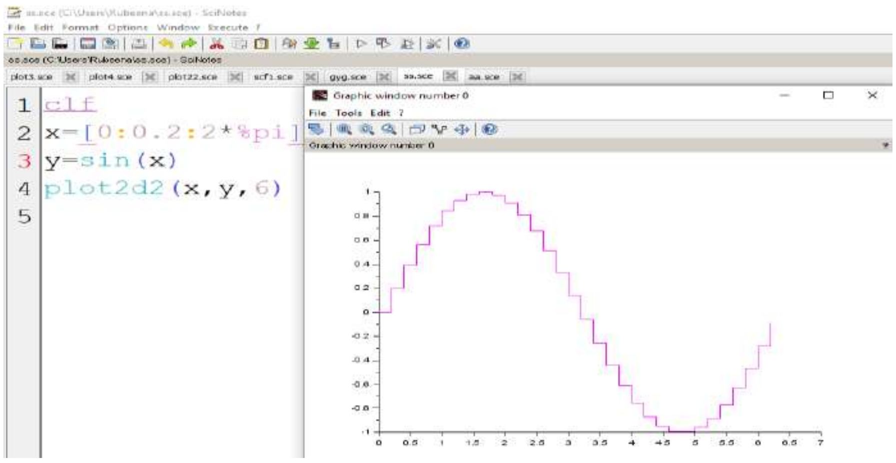


```
x=[11 33 55 88 99]
plot2d2(x)
```

Fig. Using xgrid along with plot2D

In the above example only one variable x was represented. In this example here using trigonometric function (sin) and color. We can now see the step function using trigonometric functions.

- clf( )
- x=[0:0.2:2*%pi]
- y=sin(x)
- plot2d2(x,y,6)



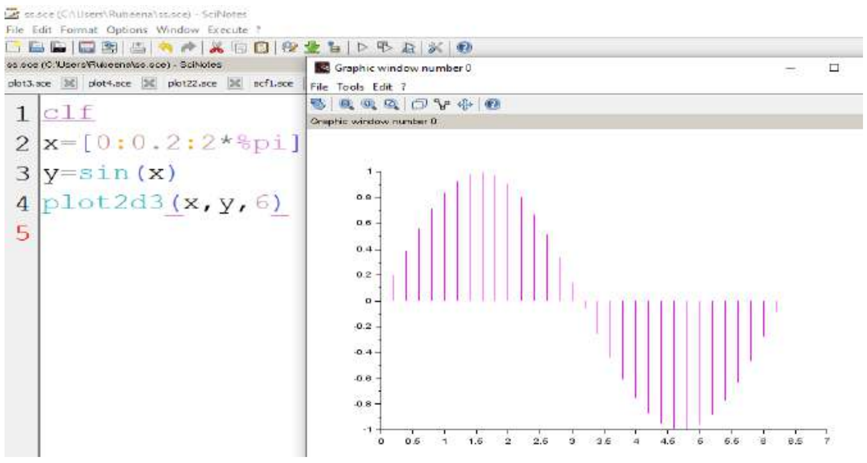
```
clf
x=[0:0.2:2*%pi]
y=sin(x)
plot2d2(x,y,6)
```

Fig. step function generated using plot 2D2

Plot2d3

This command uses vertical lines to effect plotting

- clf
- x=[0:0.2:2*%pi]
- y=sin(x)
- plot2d3(x,y,6)



```
clf
x=[0:0.2:2*%pi]
y=sin(x)
plot2d3(x,y,6)
```

Fig. vertical lines generated using plot 2D3