1. Introduction

MATLAB has an excellent set of graphic tools. Plotting a given data set or the results of computation is possible with very few commands. You are highly encouraged to plot mathematical functions and results of analysis as often as possible. Trying to understand mathematical equations with graphics is an enjoyable and very efficient way of learning mathematics.

2. Creating simple plots

The basic MATLAB graphing procedure, for example in *2D,* is to take a vector of x' coordinates, $x = (x_1, \ldots, x_n)$, and a vector of y-coordinates, $y = (y_1, \ldots, y_n)$, locate the points (x_i, y_i), with $i = 1, 2, \ldots, n$ and then join them by straight lines. You need to prepare *x* and *y* in an identical array form; namely, *x* and *y* are both row arrays or column arrays of the same length.

The MATLAB command to plot a graph is *plot(x,y).* The vectors x = (1, 2, 3, 4, 5, 6) and y = (3, −1, 2, 4, 5, 1) produce the picture shown in Figure.1.

```
>> x = [1 2 3 4 5 6];
>> y = [3 -1 2 4 5 1];
>> plot(x,y)
```

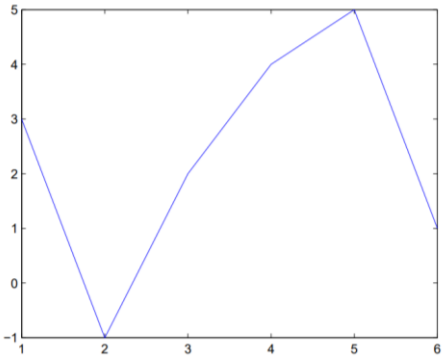

Figure.1: Plot for the vectors x and y

Note: The plot functions has different forms depending on the input arguments. If *y* is a vector *plot(y)* produces a piecewise linear graph of the elements of *y* versus the index of the elements of *y*. If we specify two vectors, as mentioned above, *plot(x,y)* produces a graph of *y* versus *x*.

For example, to plot the function *sin (x)* on the interval [0, 2π], we first create a vector of *x* values ranging from *0 to 2π,* then compute the *sine* of these values, and finally *plot* the result:

```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y)
```

Notes:

• *0 : pi/100 : 2*pi* : yields a vector that

– starts at 0,

– takes steps (or increments) of π/100,

– stops when *2π* is reached.

• If you omit the increment, MATLAB automatically increments by *1*.

3. Adding titles, axis labels, and annotations

MATLAB enables you to add axis labels and titles. For example, using the graph from the previous example, add an *x- and y-axis labels.*

Now *label* the axes and add a *title*. An example of *2D* plot is shown in Figure.2.

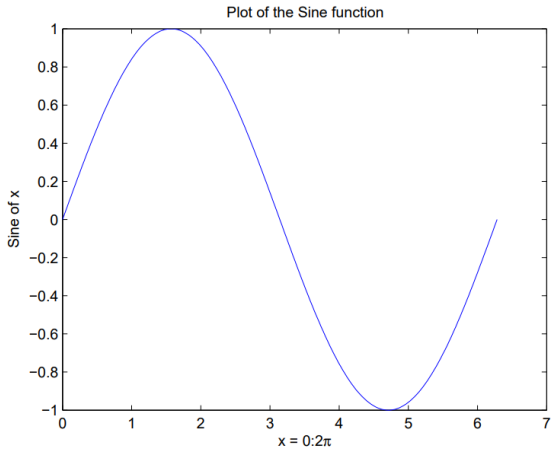

Figure.2: Plot of the Sine function

>> xlabel('x = 0:2\pi')

>> ylabel('Sine of x')

>> title('Plot of the Sine function')

The color of a single curve is, by default, blue, but other colors are possible. The desired color is indicated by a third argument. For example, red is selected by *plot (x,y, 'r').* Note the single quotes, *' '*, around r.

4. Multiple data sets in one plot

Multiple (x, y) pairs arguments create multiple graphs with a single call to plot. For example, these statements plot three related functions of x: y1 = 2 cos(x), y2 = cos(x), and y3 = 0.5 ∗ cos(x), in the interval $0 \leq x \leq 2\pi$.

```
>> x = 0:pi/100:2*pi;
>> y1 = 2*cos(x);
>> y2 = cos(x);
>> y3 = 0.5*cos(x);
>> plot(x,y1,'--',x,y2,'-',x,y3,':')
>> xlabel('0 \leq x \leq 2\pi')
>> ylabel('Cosine functions')
>> legend('2*cos(x)','cos(x)','0.5*cos(x)')
>> title('Typical example of multiple plots')
>> axis([0 2*pi -3 3])
```

The result of multiple data sets in one graph plot is shown in Figure 2.

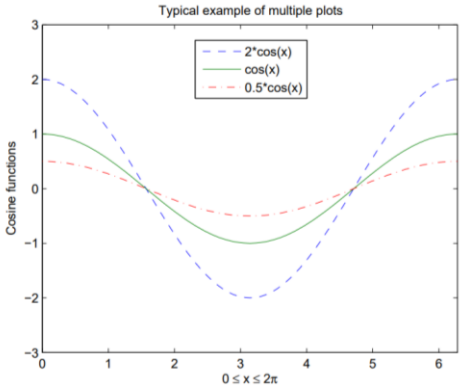

Figure .2: Typical example of multiple plots

By default, MATLAB uses line style and color to distinguish the data sets plotted in the graph. However, we can change the appearance of these graphic components or add annotations to the graph to help explain your data for presentation.

5. Specifying line styles and colors

It is possible to specify line styles, colors, and markers (e.g., circles, plus signs, . . .) using the plot command:

plot(x,y,'style_color_marker')

where *style_color_marker* is a triplet of values from Table 2

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	−	Solid	+	Plus sign
r	Red	−−	Dashed	o	Circle
b	Blue	:	Dotted	*	Asterisk
g	Green	−.	Dash-dot	.	Point
c	Cyan	none	No line	×	Cross
m	Magenta			*s*	Square
y	Yellow			*d*	Diamond

Table 2: Attributes for plot