

## Creating simple plots

### For example in 2D :

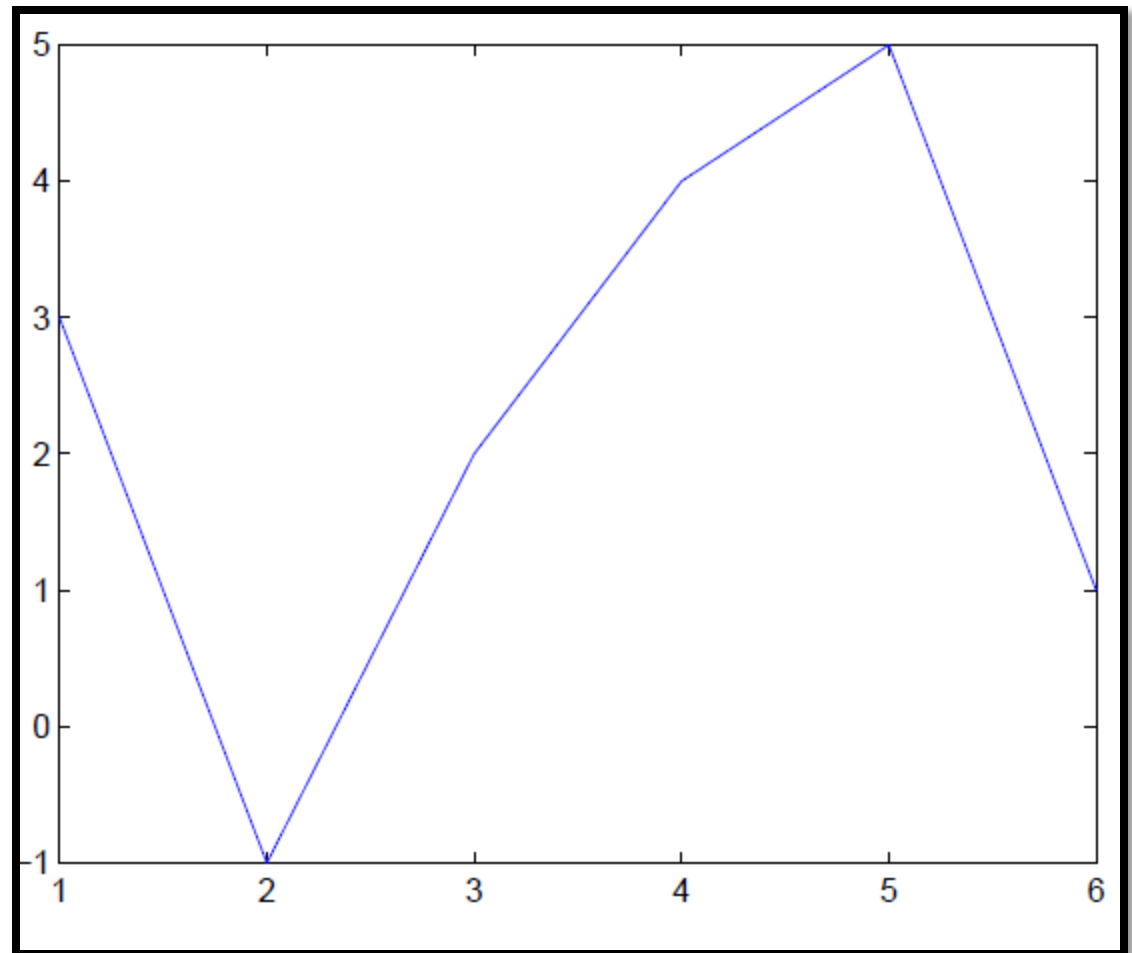
- 1- Is to take a vector of x- coordinates,  $x = (x1; : : : ; xN)$ , and
- 2- A vector of y-coordinates,  $y = (y1; : : : ; yN)$ ,
- 3- Locate the points  $(xi; yi)$ , with  $i = 1; 2; : : : ; n$
- 4- And then join them by straight lines.
- 5- Prepare  $x$  and  $y$  in an identical array form
- 6- namely,  $x$  and  $y$  are both row arrays or column arrays of the same length.
- 7- The MATLAB command to plot a graph is **plot(x,y)**

Example : The vectors  $x = (1; 2; 3; 4; 5; 6)$

and  $y = (3; -1; 2; 4; 5; 1)$

**Solution:**

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



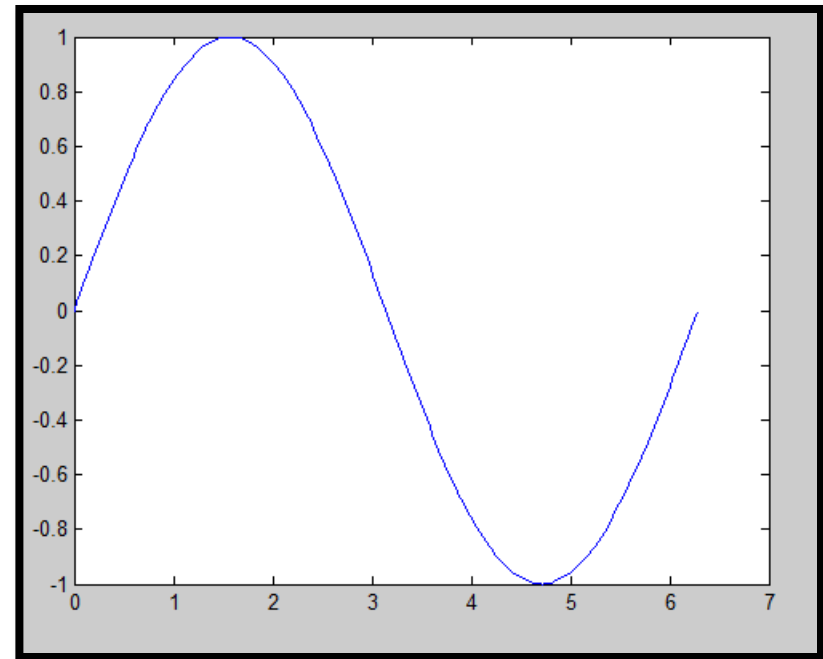
**For example**, to plot the function **sin (x)** on the interval  $[0; 2\frac{1}{4}]$ , we first create a vector of **x** values ranging from  $[0$  to  $2\pi]$ , then compute the sine of these values, and finally plot the result:

**Solution:**

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

Notes:

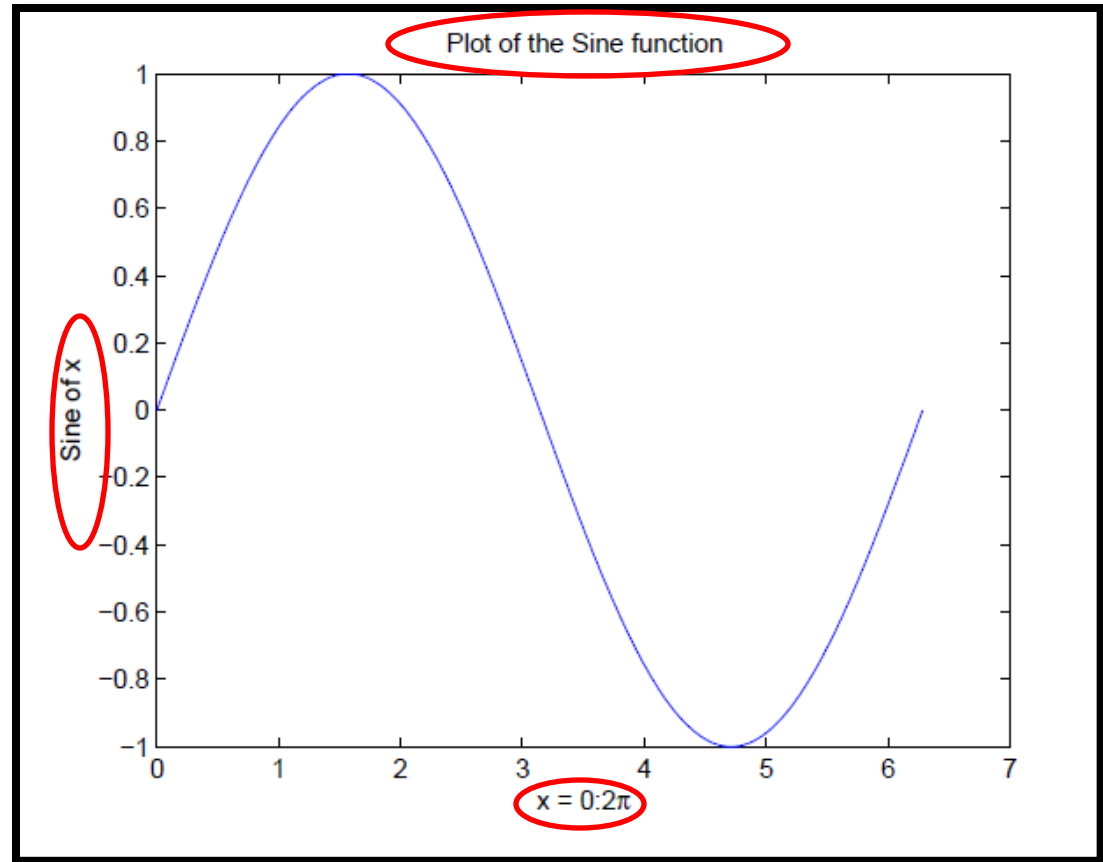
- 1-  **$0:\pi/100:2*\pi$**  yields a vector that
  - starts at **0**,
  - takes steps (or increments) of  **$\pi/100$** ,
  - stops when  **$2\pi$**  is reached.
- 2- If you omit the increment, MATLAB automatically increments by **1**.



## 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*.

```
>> x = 0:pi/100:2*pi;  
>> y = sin(x);  
>> plot(x,y)  
>> xlabel('x = 0:2\pi')  
>> ylabel('Sine of x')  
>> title('Plot .....')
```

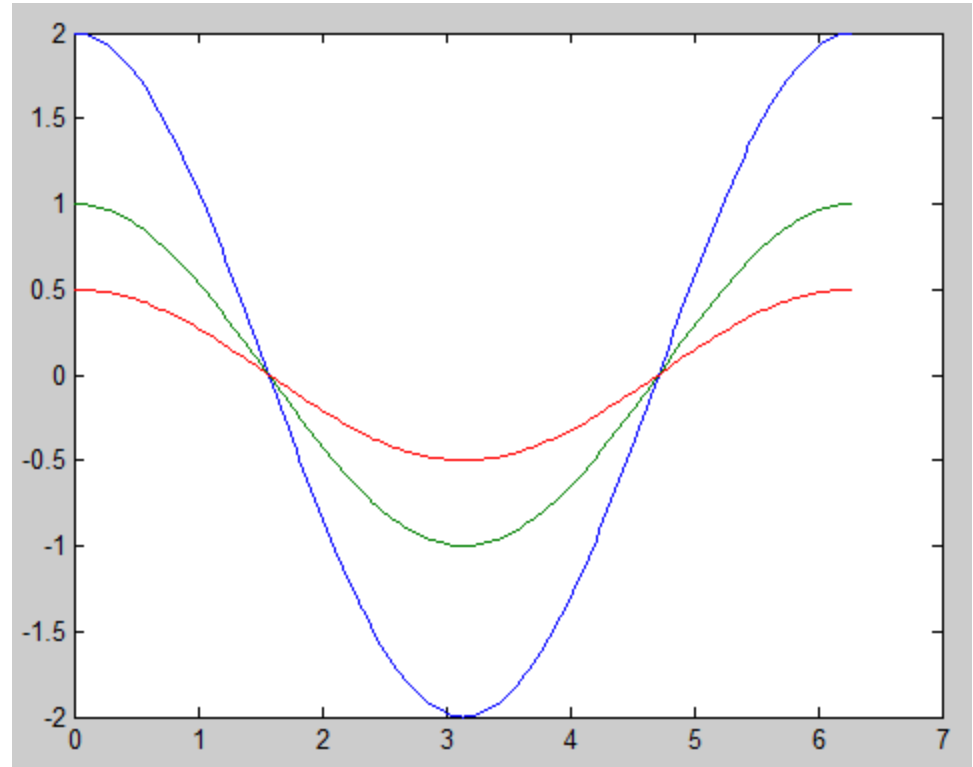


The desired is indicated by a third argument. For example, red is selected by `plot(x,y,'r')`. Note the single quotes, ' ', around r.

## 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)
```



By default, MATLAB uses *line style and color* to distinguish the data sets plotted in the graph.

However, you can change the appearance of these graphic components or add annotations to the graph to help explain your data for presentation.

```
>> 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')
```

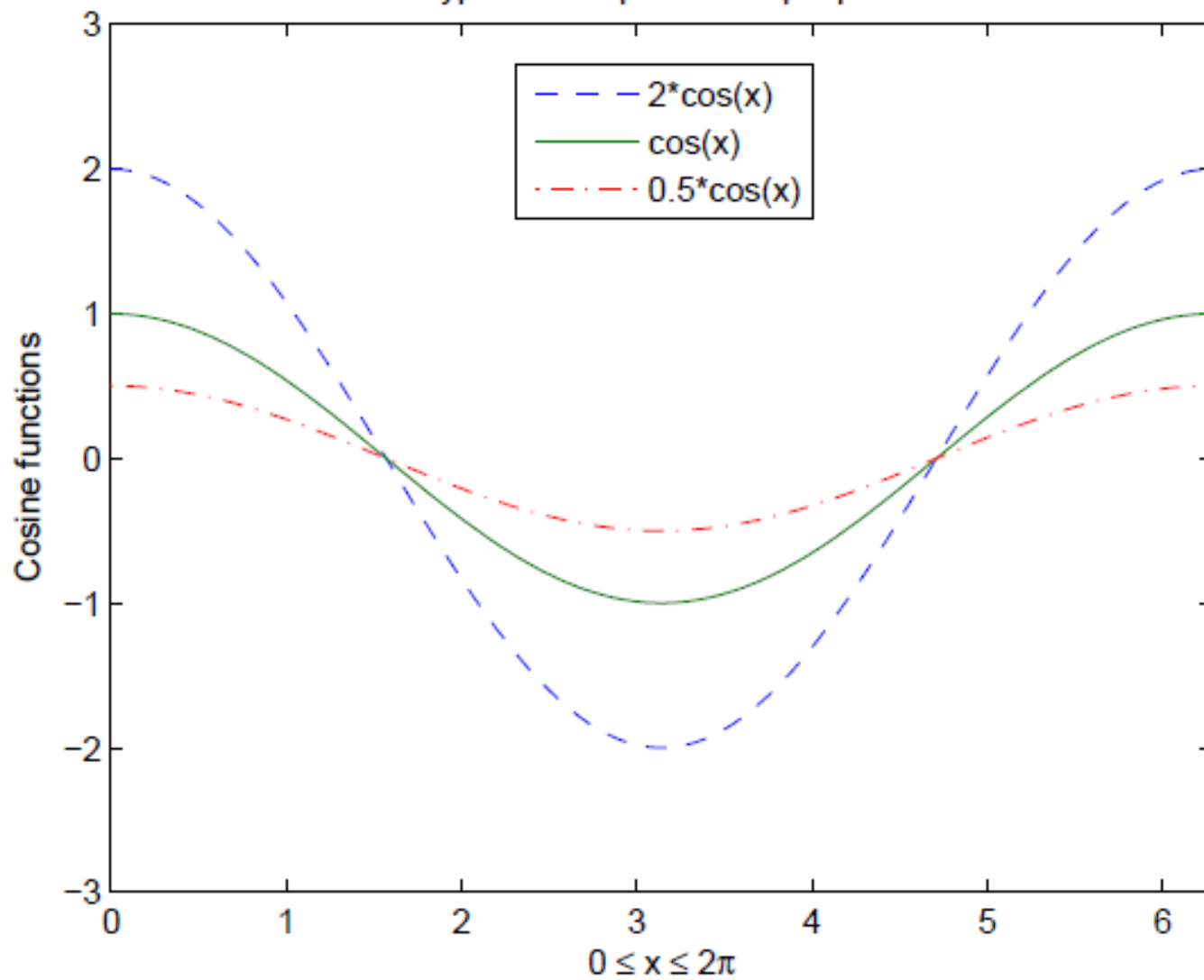
```
      x1 x2   y1 y2
```

```
>> axis([0 2*pi -3 3])
```

To specify line style

Controlling axes

Typical example of multiple plots



**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 2.3*.

Table 2.3: Attributes for plot

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