



TOPIC 6

Computer application for drawing 2D Graph

Plotting Elementary Functions

Suppose we wish to plot a graph of $y = \sin 3\pi x$ for $0 \leq x \leq 1$. We do this by sampling the function at a sufficiently large number of points and then joining up the points $(x; y)$ by straight lines. Suppose we take $N + 1$ points equally spaced a distance h apart:

```
>> N = 10; h = 1/N; x = 0:h:1;  
defines the set of points  $x = 0; h; 2h; \dots; 1-h; 1$ .
```

Alternately, we may use the command **linspace**.

The general form of the command is **linspace (a,b,n)** which generates $n + 1$ equispaced points between a and b , inclusive. So, in this case we would use the command

```
>> x = linspace (0,1,11);
```

The corresponding y values are computed by

```
>> y = sin(3*pi*x);  
and finally, we can plot the points with  
>> plot(x,y)
```

The result is shown in Figure 1, where it is clear that the value of N is too small.

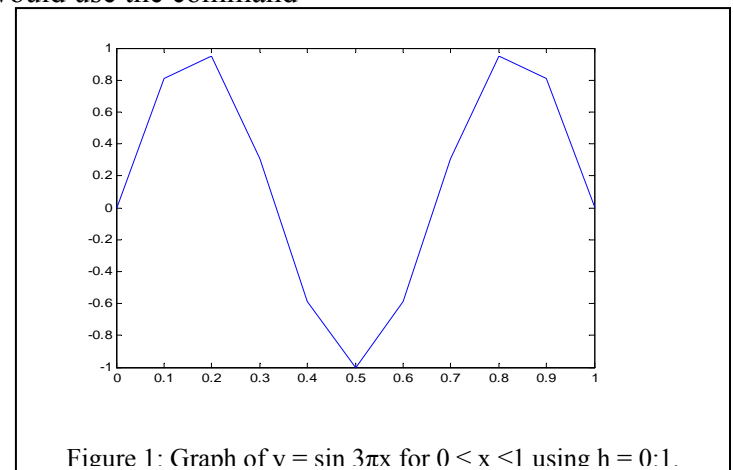


Figure 1: Graph of $y = \sin 3\pi x$ for $0 < x < 1$ using $h = 0:1$.

On changing the value of N to 100:

```
>> N = 100; h = 1/N; x = 0:h:1;  
>> y = sin(3*pi*x); plot(x,y)  
we get the picture shown in Figure 2.
```

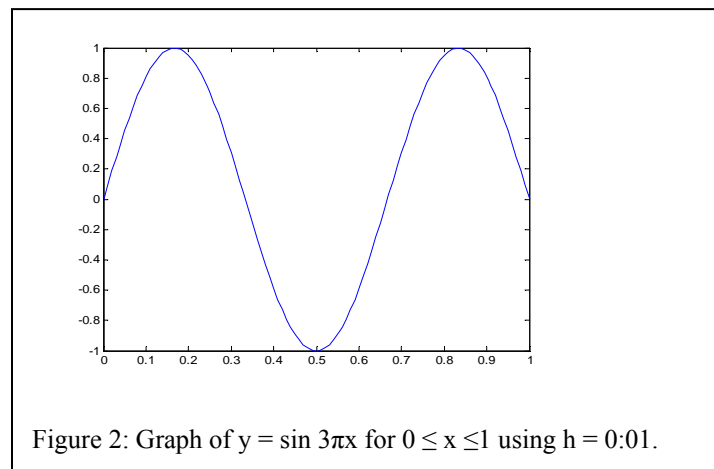


Figure 2: Graph of $y = \sin 3\pi x$ for $0 \leq x \leq 1$ using $h = 0:01$.

Plotting-Titles & Labels

To put a title and label the axes, we use

```
>> title('Graph of y = sin(3pi x)')
>> xlabel('x axis')
>> ylabel('y-axis')
```

The strings enclosed in single quotes, can be anything of our choosing.

Grids

A dotted grid may be added by

```
>> grid
```

This can be removed using either `grid` again, or `grid off`.

Line Styles & Colors

The default is to plot solid lines. A solid white line is produced by

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

The third argument is a string whose first character specifies the color (optional) and the second the line style. The options for colors and styles are:

	Colours		Line Styles
y	yellow	.	point
m	magenta	o	circle
c	cyan	x	x-mark
r	red	+	plus
g	green	-	solid
b	blue	*	star
w	white	:	dotted
k	black	-.	dashdot
		--	dashed

The number of available plot symbols is wider than shown in this table. Use **help plot** to obtain a full list.

Multi-plots

Several graphs may be drawn on the same figure as in

```
>> plot(x,y,'w-',x,cos(3*pi*x),'g--')
```

A descriptive legend may be included with

```
>> legend('Sin curve','Cos curve')
```

which will give a list of line-styles, as they appeared in the plot command, followed by a brief description.

Matlab fits the legend in a suitable position, so as not to conceal the graphs whenever possible. For further information do *help plot* etc. The result of the commands:

```
>> plot(x,y,'w-',x,cos(3*pi*x),'g--')
>> legend('Sin curve','Cos curve')
>> title('Multi-plot ')
>> xlabel('x axis'), ylabel('y axis')
>> grid
```

is shown in Figure 3. The legend may be moved manually by dragging it with the mouse.

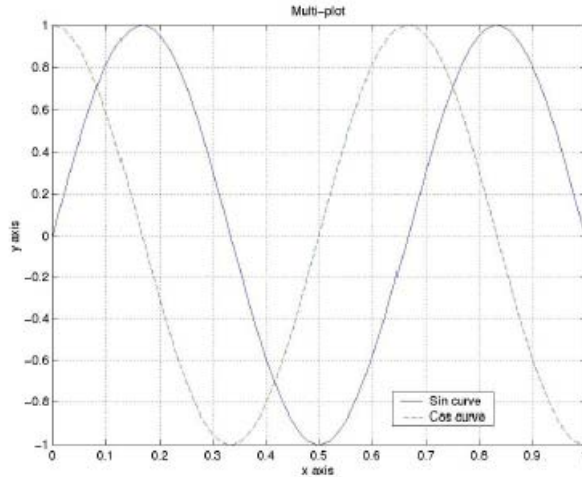


Figure 3: Graph of $y = \sin 3\pi x$ and $y = \cos 3\pi x$ for $0 \leq x \leq 1$ using $h = 0:01$.

Hold

A call to plot clears the graphics window before plotting the current graph. This is not convenient if we wish to add further graphics to the figure at some later stage. To stop the window being cleared:

```
>> plot(x,y,'w-'), hold on
```

```
>> plot(x,y,'gx'), hold off
```

“hold on” holds the current picture; “hold off” releases it (but does not clear the window, which can be done with clf). “hold” on its own toggles the hold state.

Subplot

The graphics window may be split into an $m \times n$ array of smaller windows into which we may plot one or more graphs. The windows are counted 1 to mn row-wise, starting from the top left. Both hold and grid work on the current subplot.

```
>> subplot(221), plot(x,y)
```

```
>> xlabel('x'),ylabel('sin 3 pi x')
```

```
>> subplot(222), plot(x,cos(3*pi*x))
```

```
>> xlabel('x'),ylabel('cos 3 pi x')
```

```
>> subplot(223), plot(x,sin(6*pi*x))
```

```
>> xlabel('x'),ylabel('sin 6 pi x')
```

```
>> subplot(224), plot(x,cos(6*pi*x))
```

```
>> xlabel('x'),ylabel('cos 6 pi x')
```

subplot(221) (or subplot(2,2,1))

specifies that the window should be split into a 2×2 array and we select the first subwindow.

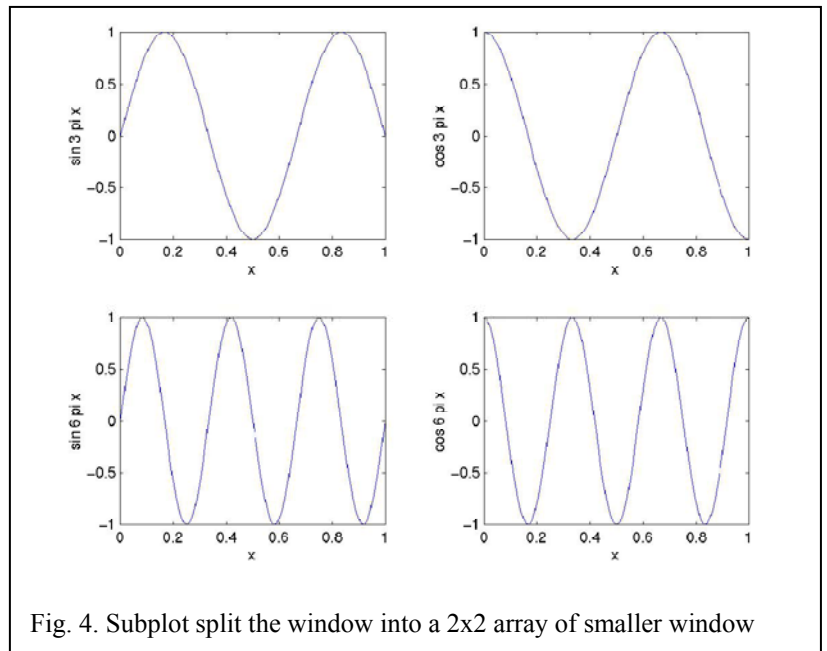


Fig. 4. Subplot split the window into a 2×2 array of smaller window

Formatted text on Plots

It is possible to change to format of text on plots so as to increase or decrease its size and also to typeset simple mathematical expressions. We shall give two illustrations.

First we plot the first 100 terms in the sequence $\{x_n\}$ given by $x_n = \left(1 + \frac{1}{n}\right)^n$ and then graph the function $\phi(x) = x^3 \sin^2(3\pi x)$ on the interval $-1 \leq x \leq 1$. The commands:

```
>> set(0,'Defaultaxesfontsize',16);
>> n = 1:100; x = (1+1./n).^n;
>> subplot (211)
>> plot(n,x,'.',[0 max(n)],exp(1)*[1 1],...
'--','markersize',8)
>> title('x_n = (1+1/n)^n','fontsize',12)
>> xlabel('n'), ylabel('x_n')
>> legend('x_n','y = e^1 = 2.71828...',4)
>> %%%%%%%%%%%
>> subplot (212)
>> x = -2:.02:2; y = x.^3.*sin(3*pi*x).^2;
>> plot(x,y,'linewidth',2)
>> legend('y = x^3 sin^2 3\pi x',4)
>> xlabel('x')
```

produce the graph shown in fig. 5.

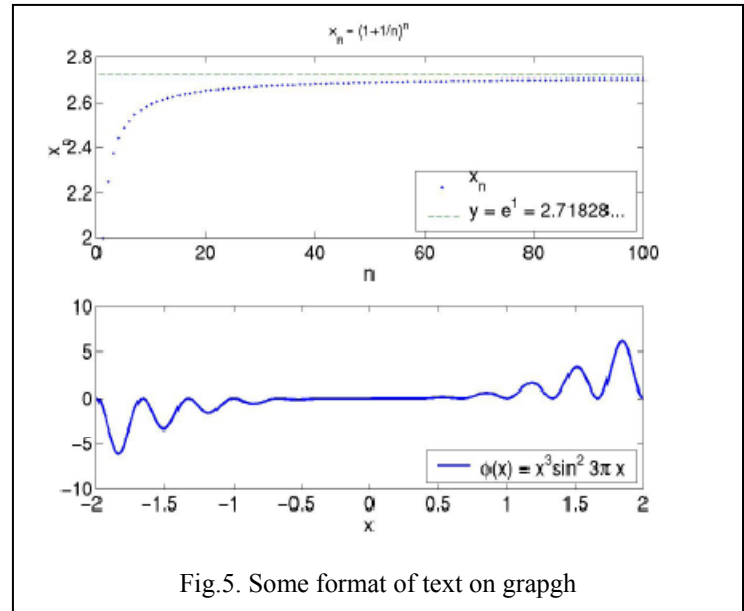


Fig.5. Some format of text on graph

The salient features of these commands are

1. The first line increases the size of the default font size used for the axis labels, legends and titles.
2. The size of the plot symbol “.” is changed from the default (6) to size 8 by the additional string followed by value “ 'markersize',8 ”.
3. The strings x_n are formatted as x_n to give subscripts while x^3 leads to superscripts x^3 . Note also that $\sin^2 3\pi x$ translates into the Matlab command $\sin(3*\pi*x).^2$ -the position of the exponent is different.
4. Greek characters $\alpha, \beta, \dots, \omega, \Omega$ are produced by the strings `'\alpha'`, `'\beta'`, ..., `'\omega'`, `'\Omega'`. The integral symbol: \int is produced by `'\int'`.
5. The thickness of the line used in the lower graph is changed from its default value (0:5) to 2.
6. Use **help legend** to determine the meaning of the last argument in the legend commands.

One can determine the current value of any plot property by first obtaining its “handle number” and then using the **get** command such as

```
>> handle = plot (x,y,')
>> get (handle,'markersize')
ans =
```

6

Controlling Axes

Once a plot has been created in the graphics window you may wish to change the range of x and y values shown on the picture.

```
>> clf, N = 100; h = 1/N; x = 0:h:1;
>> y = sin(3*pi*x); plot(x,y)
>> axis([-0.5 1.5 -1.2 1.2]), grid
```

The axis command has four parameters, the first two are the minimum and maximum values of x to use on the axis and the last two are the minimum and maximum values of y. Note the square brackets. The result of these commands is shown in Figure 4. Look at help axis and experiment with the commands axis equal, axis verb, axis square, axis normal, axis tight in any order.

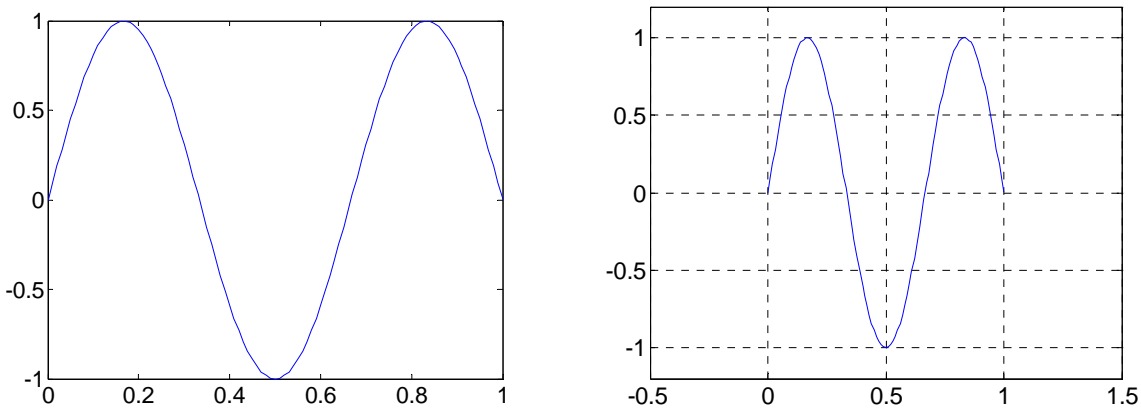


Fig.6. The effect of changing the axes of a plot.

Exercises. Draw graphs of the functions

$$\begin{array}{ll} i) & y = \frac{\sin x}{x} \\ ii) & u = \frac{1}{(x-1)^2} + x \\ iii) & v = \frac{x^2+1}{x^2-4} \\ iv) & w = \frac{(10-x)^{1/3}-2}{(4-x^2)^{1/2}} \end{array}$$

for $0 \leq x \leq 10$.

Stairs

Stairstep graphs are useful for drawing time-history graphs of digitally sampled data.

stairs(Y) draws a stairstep graph of the elements of Y, drawing one line per column for matrices.

When Y is a vector, the x-axis scale ranges from 1 to length(Y). When Y is a matrix, the x-axis scale ranges from 1 to the number of rows in Y.

stairs(X,Y) plots the elements in Y at the locations specified in X. The elements of X must be monotonic. X must be the same size as Y or, if Y is a matrix, X can be a row or a column vector such that length(X) = size(Y,1)

Example:

Create a stairstep plot of a sine wave.

```
>>x = linspace(-2*pi,2*pi,40);
>>stairs(x,sin(x))
```

bar, barh

A bar graph displays the values in a vector or matrix as horizontal or vertical bars.

bar(Y) draws one bar for each element in Y. If Y is a matrix, bar groups the bars produced by the elements in each row. The x-axis scale ranges from 1 to length(Y) when Y is a vector, and 1 to size(Y,1), which is the number of rows, when Y is a matrix.

bar(x,Y) draws a bar for each element in Y at locations specified in x, where x is a monotonically increasing vector defining the x-axis intervals for the vertical bars. If Y is a matrix, bar groups the elements of each row in Y at corresponding locations in x.

bar(...,width) sets the relative bar width and controls the separation of bars within a group. The default width is 0.8, so if you do not specify x, the bars within a group have a slight separation. If width is 1, the bars within a group touch one another.

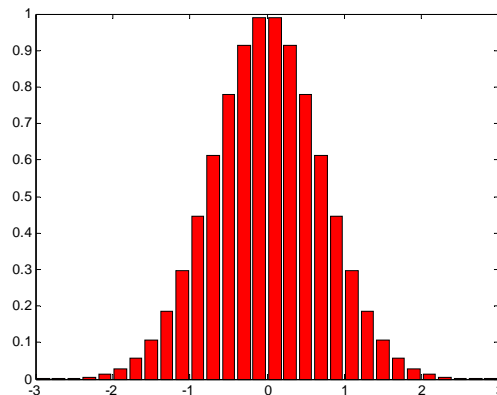
bar(...,'style') specifies the style of the bars. 'style' is 'grouped' or 'stacked'. 'group' is the default mode of display. 'grouped' displays m groups of n vertical bars, where m is the number of rows and n is the number of columns in Y. The group contains one bar per column in Y. 'stacked' displays one bar for each row in Y. The bar height is the sum of the elements in the row. Each bar is multicolored, with colors corresponding to distinct elements and showing the relative contribution each row element makes to the total sum.

bar(...,'bar_color') displays all bars using the color specified by the single-letter abbreviation 'r', 'g', 'b', 'c', 'm', 'y', 'k', or 'w'.

Example:

plots a bell-shaped curve as a bar graph and sets the colors of the bars to red.

```
>>x = -2.9:0.2:2.9;  
>>bar(x,exp(-x.*x),'r')
```



There some MATLAB function to drawing many other 2D graphs, e.g.**hist**, **rose**, **pie**. Use “search” box in help menu of MATLAB to get the information about the other functions.