

DOING PHYSICS WITH MATLAB

GETTING STARTED WITH MATLAB

XY PLOTS

Ian Cooper

School of Physics, University of Sydney

ian.cooper@sydney.edu.au

DOWNLOAD DIRECTORY FOR MATLAB SCRIPTS

Basic XY Plotting

For scientists, engineers, economists and others, XY graphs are an essential tool for analysing data and establishing relationships between two quantities. The ability to visualise data graphically rather than present numbers in tables or through mathematical equations is very important. This visualise of data can be done extremely well using Matlab. In this document you will explore some of the many ways in which XY data can be visualised using Matlab. Although most of the information about graphics is contained within the Matlab help files, the information is in a reference format, while the information presented here aims to teach you how to use the Matlab graphics commands through many illustrative examples.

An XY graph gives a graphical picture of one variable X plotted against another variable Y with linear or log scales. One has to decide the range of values for both X and Y and how the X and Y axes are subdivided for tick marks or for drawing a grid. A plot should have the X and Y axes labelled and a graph often has a title. The XY data can be plotted in many different formats, for example a series of circles or a continuous line. A Matlab plot has a plotting region and an external border region. The various elements of a graph can be shown in different colours and the plots can be annotated. The appearance of a plot can be changed by direct editing of the plot window using the tool bars at the top of the figure window or by using specific Matlab commands contained in m-scripts.

For our first illustrative example, we will consider graphing the function

$$h = A_0 \sin\left(\frac{2\pi t}{T}\right). \quad (1)$$

This function represents the oscillatory motion of an object moving up and down on the end of a spring. The variables to be plotted are h (Y) and are t (X) where h is the position of the object about an equilibrium position and t is the time. A_0 is called the amplitude and is the maximum distance from the equilibrium position and T is the period of oscillation or the time for one complete oscillation.

An m-script is used to calculate a function and to draw the graph in a figure window. Titles can be added to the graph using the commands **title**, **xlabel** and **ylabel**.

Various line types, plot symbols and colors may be obtained with the command **plot(X,Y,S)** where S is a character string made from one element from any or all the three columns in table 1.

Line style	Marker style	Color
- solid	. point	b blue
: dotted	o circle	g green
-. dashdot	x x mark	r red
-- dashed	+ plus	c cyan
	* star	m magenta
	s square	y yellow
	d diamond	k black
	v triangle down	w white
	^ triangle up	
	< triangle left	
	> triangle right	
	p pentagram	
	h hexagram	

Table 1. Line and marker styles: **plot(X,Y,S)**. For example, S = 'c+' plots a cyan dotted line with a plus at each data point and S = 'bd' plots blue diamond at each data point but does not draw any line.

Exercise 1

An m-script ex1.m to plot the sinusoidal function in equation 1 is given at the end of this document. The resulting plot of the sinusoidal function is shown in figure 1. All m-script are written in a structured form with typical sections: *Setup*, *Inputs*, *Initialise*, *Calculations* and *Graphics*. The m-script is like a template, to plot another function, only minimum changes to the code are required.

By analysing the m-script code and comments line by line you will be able to work out what the script is doing.

Change the m-script to plot the data with different line and marker styles and with different colours.

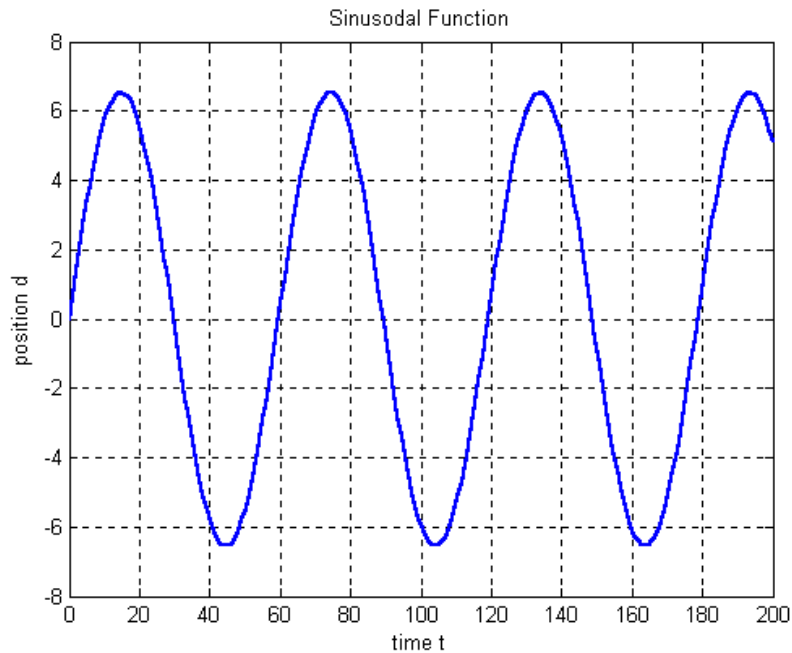


Fig. 1 Plot of the Sinusoidal function (Eq. 1) using the m-script ex1.

A number of plots can be shown in the one figure window using the subplot command **subplot(m,n,p)** where the window is split into $m \times n$ plot areas and p gives the current plot. For the oscillating object at the end of the spring you may want to plot four graphs to show the position d (equation 1), velocity v (equation 2) and acceleration a (equation 3) as functions of time t and a phase plot of v against hd . The use of the subplot command is shown in figures 2 and 3.

$$v = A \left(\frac{2\pi}{T} \right) \cos \left(\frac{2\pi t}{T} \right) \quad (2)$$

$$a = -A \left(\frac{2\pi}{T} \right)^2 h \quad (3)$$

Subplots are shown in figures 2 and 3.

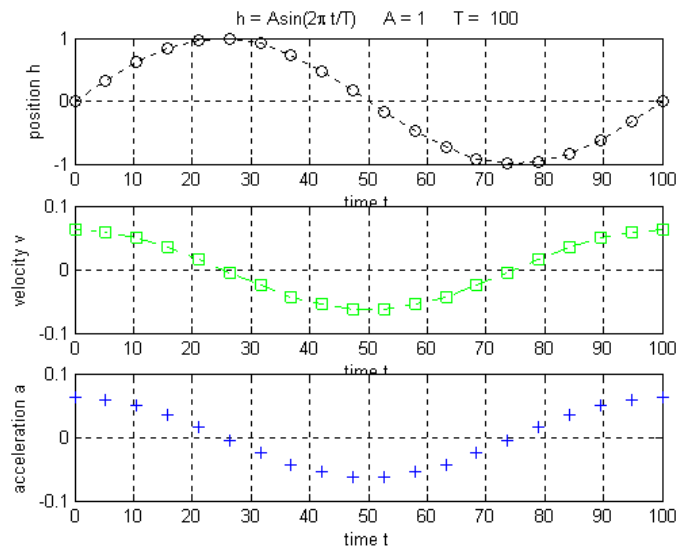


Figure 2. Three plots in a figure window using **subplot(3,1,p)** where $p = 1, 2, 3$.

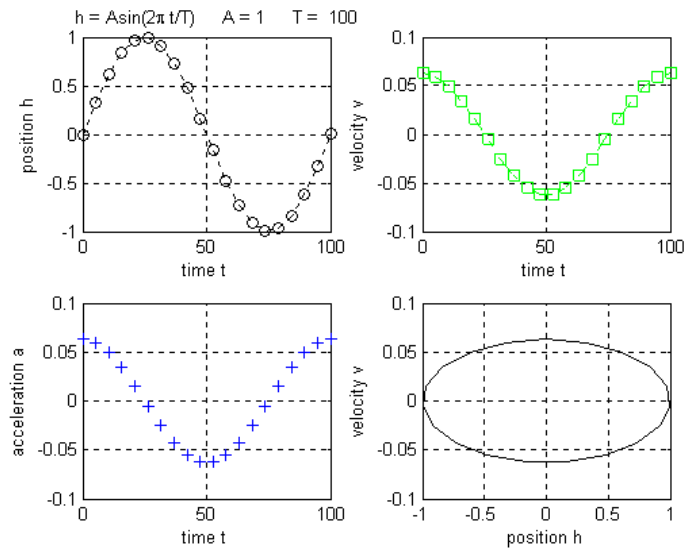


Figure 3. Four plots in a figure window using **subplot(2,2,p)** where $p = 1, 2, 3, 4$.

Two curves can be plotted simultaneously using the **hold on** command. Figure 4 shows two plots showing the oscillatory motion with two different amplitudes and periods.

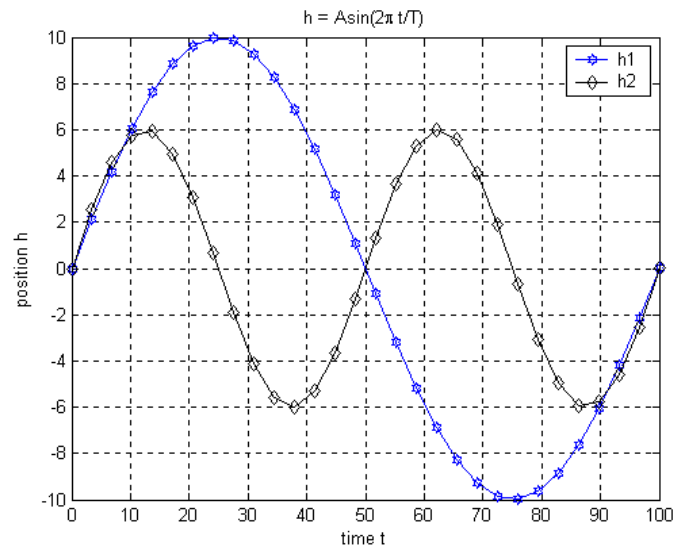


Figure 4. Two curves plotted together using the **hold on** command.

Another type of XY plot is called a stem plot. **stem(X,Y)** plots the data sequence Y as stems from the X-axis terminated with circles for the data value. Figure 5 is a stem plot for graph of equation 1.

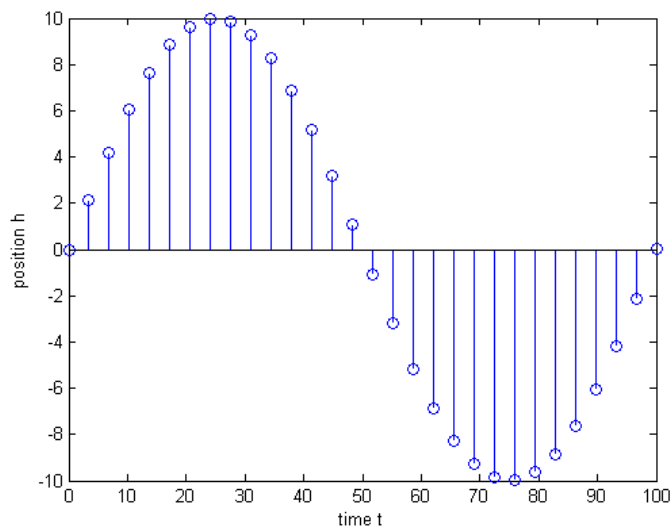


Figure 5. A stem(X,Y) plot for equation 1.

Log plots

A variety of log graphs can be plotted with the commands **semilogx**, **semilogy** and **loglog**. Consider the three function given in equations 4 to 6.

$$y = 10^x \quad (4)$$

$$N = N_0 e^{kt} \quad (5)$$

$$P = AT^4 \quad (6)$$

In equation 4, the variable y increases very rapidly with x and it is appropriate that a log scale is used for the Y-axis. Equation 5 represents an exponential growth in a population N with time t and using a log scale for the X-axis is suitable. Equation 6 could represent the power P radiated from a hot object at temperature T and the graph is best shown as a log-log graph. The m-script ch1003.m is used to create the three different types of log plots as shown in figure 6.

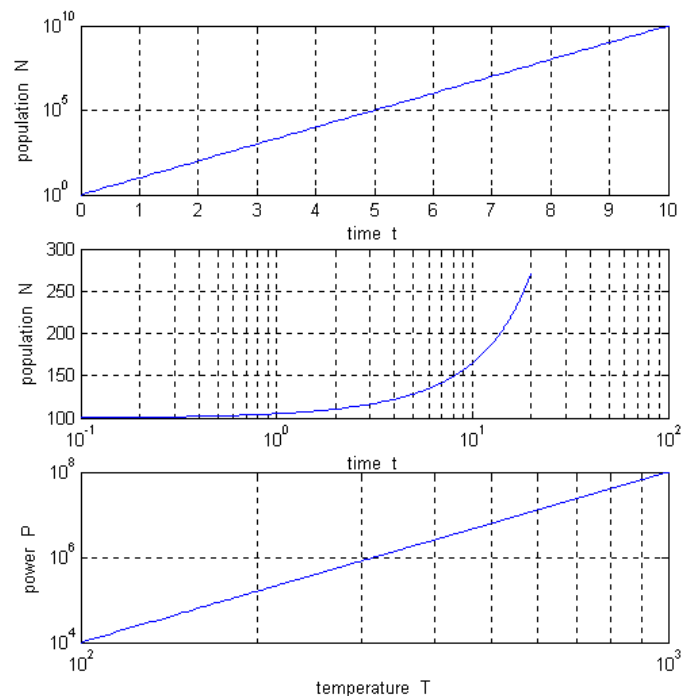


Figure 6. Semilogy, semilogx and loglog plots of equations 4 to 6 respectively.

Two quantities with different scales can be plotted on the one graph with the **plotyy** command in which one quantity is scaled to the left Y axis and the other quantity to the right Y axis as shown in figure 7. The script for figure 7 is in the m-script ch1002.m.

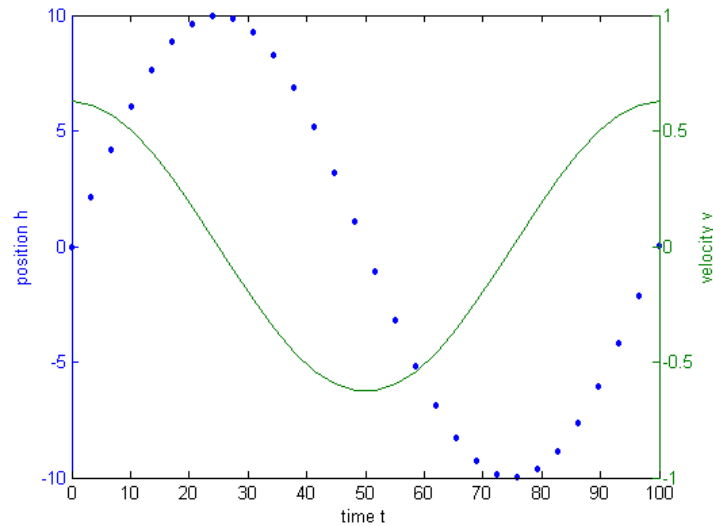


Figure 7. Two plots position h and velocity v in the one figure window using the **plotyy** command.

m-scripts

ex1.m

```
% ch10001.m
% m-script to plot sinusoidal function given by equation (1)
%
% setup -----
clear                % clears all variables
close all            % closes all figure windows
tic                  % starts clock

% inputs -----
n = 200;              % number of data points to be plotted
tmin = 0;            % first point to be plotted
tmax = 100;          % last point to be plotted

A = 1;               % amplitude
T = 100;             % period

% initialise -----
t = zeros(n,1);      % setup array for time t
h = zeros(n,1);      % setup array for position h

t = linspace(tmin,tmax,n); % n equally spaced points for t from tmin to tmax

% calculations
% -----
h = A.*sin((2*pi/T).*t); % calculation of array for position h

% graphics -----
figure(1)            % open figure window for plotting
plot(t,h)            % plot XY graph

% end -----
toc                  % gives execution time in seconds in command window
```

The m-script for figures 2 to 5.

```
% m-script to plot sinusoidal function given by equation (1)
%
% setup -----
clear                % clears all variables
close all            % closes all figure windows
tic                  % starts clock

% inputs -----
n = 30;              % number of data points to be plotted
tmin = 0;            % first point to be plotted
tmax = 100;          % last point to be plotted

A1 = 10;             % amplitude
A2 = 6
T1 = 100;            % period
T2 = 50;

% initialise -----
t = zeros(n,1);      % setup array for time t
h1 = zeros(n,1);     % setup array for position h
h2 = zeros(n,1);
v = zeros(n,1);
a = zeros(n,1);

t = linspace(tmin,tmax,n); % n equally spaced points for t from tmin to tmax

% calculations
% -----
w1 = 2*pi/T1;
w2 = 2*pi/T2;
h1 = A1.*sin(w1.*t); % calculation of array for position h
h2 = A2.*sin(w2.*t);
v = (A1*w1).*cos(w1.*t); % velocity
a = -(A1*w1^2).*h1; % acceleration

% graphics -----
%Plot titles
tm1 = ' ';
tm2 = 'h = A sin(2\pi t/T)';
tm3 = ' A = '
tm4 = num2str(A1);
tm5 = ' T = '
tm6 = num2str(T1);
tm = [tm1 tm2 tm3 tm4 tm5 tm6] % string for main title
tx1 = 'time t'; % string for X axis title
ty1 = 'position h' % string for Y axis title
ty2 = 'velocity v'
ty3 = 'acceleration a'

% Line and marker styles
figure(1) % open figure window for plotting
S = 'b-h'; % string for plotting line style
plot(t,h1,S) % plot XY graph with line and marker style S

title(tm2) % main plot title
xlabel(tx1) % title for X axis
ylabel(ty1) % title for Y axis
grid on % toggle grid on or off: grid on or grid off
```



```

hold on
S = 'k-d';
plot(t,h2,S)

figure(2)
subplot(3,1,1)
S = 'k:o';
plot(t,h1,S)           % plot XY graph with line and marker style S
title(tm)             % main plot title
xlabel(tx1)           % title for X axis
ylabel(ty1)           % title for Y axis
grid on

subplot(3,1,2)
S = 'g-.s';
plot(t,v,S)           % plot XY graph with line and marker style S
xlabel(tx1)           % title for X axis
ylabel(ty2)           % title for Y axis
grid on

subplot(3,1,3)
S = 'b+';
plot(t,v,S)           % plot XY graph with line and marker style S
xlabel(tx1)           % title for X axis
ylabel(ty3)           % title for Y axis
grid on

figure(3)
subplot(2,2,1)
S = 'k:o';
plot(t,h1,S)           % plot XY graph with line and marker style S
title(tm)             % main plot title
xlabel(tx1)           % title for X axis
ylabel(ty1)           % title for Y axis
grid on

subplot(2,2,2)
S = 'g-.s';
plot(t,v,S)           % plot XY graph with line and marker style S
xlabel(tx1)           % title for X axis
ylabel(ty2)           % title for Y axis
grid on

subplot(2,2,3)
S = 'b+';
plot(t,v,S)           % plot XY graph with line and marker style S
xlabel(tx1)           % title for X axis
ylabel(ty3)           % title for Y axis
grid on

subplot(2,2,4)
S = 'k-';
plot(h1,v,S)           % plot XY graph with line and marker style S
xlabel(ty1)           % title for X axis
ylabel(ty2)           % title for Y axis
grid on

figure(4)
stem(t,h1)
xlabel(tx1)           % title for X axis

```

```

ylabel(ty1)                % title for Y axis

figure(5)
[haxis,handle1,handle2] = plotyy(t,h1,t,v,'plot');
xlabel(tx1)
axes(haxis(1))
ylabel(ty1)
axes(haxis(2))
ylabel(ty2)

% end -----
toc                % gives execution time in seconds in command window

```

The m-script for figures 6 and 7 are given in ch1003.m.

```

%ch1003.m
clear
close all
tic
% semilogy plot of  $y = 10^x$  -----

nx = 100;
xmin = 0;
xmax = 10;
x = linspace(xmin,xmax,nx);

y = 10.^x;

% semilogy plot of  $N = N_0 \exp(kt)$ ;
nt = 200;
tmin = 0;
tmax = 20;
k = 0.05;
No = 100;
t = linspace(tmin,tmax,nt);

N = No.*exp(k.*t);

% loglog plot of  $P = AT^4$  -----
nT = 400;
Tmin = 100;
Tmax = 1000;
A = 0.0001;
T = linspace(Tmin,Tmax,nT);

P = A.*T.^4;

% - graphics -----
figure(1)
tx1 = 'x';
ty1 = 'y';
semilogy(x,y)
xlabel(tx1)
ylabel(ty1)
grid on
toc

```

```
figure(2)
tx2 = 'time t';
ty2 = 'population N';
semilogx(t,N)
xlabel(tx2)
ylabel(ty2)
grid on
```

```
figure(3)
tx3 = 'temperature T';
ty3 = 'power P';
loglog(T,P)
xlabel(tx3)
ylabel(ty3)
grid on
```

```
figure(4);
subplot(3,1,1)
semilogy(x,y)
xlabel(tx2)
ylabel(ty2)
grid on
subplot(3,1,2)
semilogx(t,N)
xlabel(tx2)
ylabel(ty2)
grid on
```

```
subplot(3,1,3)
loglog(T,P)
xlabel(tx3)
ylabel(ty3)
grid on
toc
```