Modelling and control summaries

The University Of Sheffield.

by Anthony Rossiter

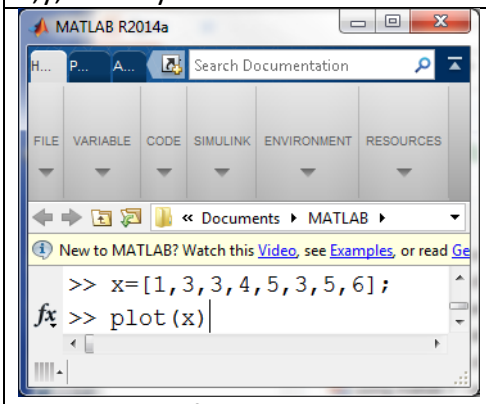
USE OF MATLAB 12 – plotting functions

OVERVIEW: These notes gives a very narrow view of MATLAB and thus demonstrate how to do a limited number of things. In general students are encouraged to become flexible independent learners using the provided MATLAB helps as this is a widely required skill in industry.

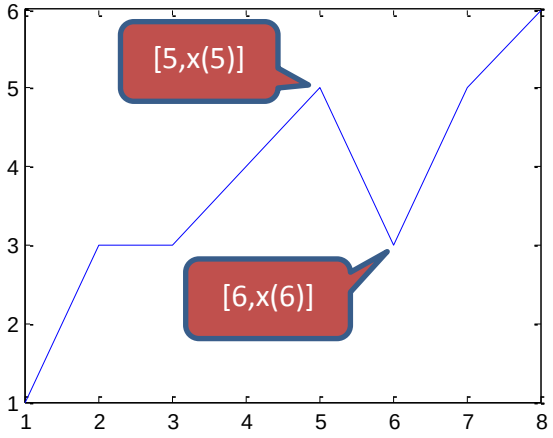
PLOTTING FUNCTIONS: This note will consider only relatively simple functions and MATLAB functionality and thus illustrate the basics.

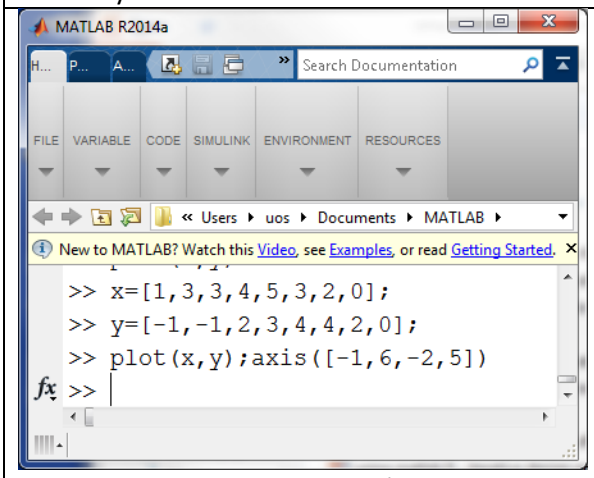
1. I have a set of numbers stored in MATLAB, how do I plot them?
2. I have a function y=f(x), how do I plot this over a specified domain?
3. I have a function which is already in MATLAB as a symbolic expression, how do I plot this?

BASIC PLOTTING: Assume that numbers are stored in the workspace – for convenience we will use x,y, but any variable names will do. The basic MATLAB command is **plot**.

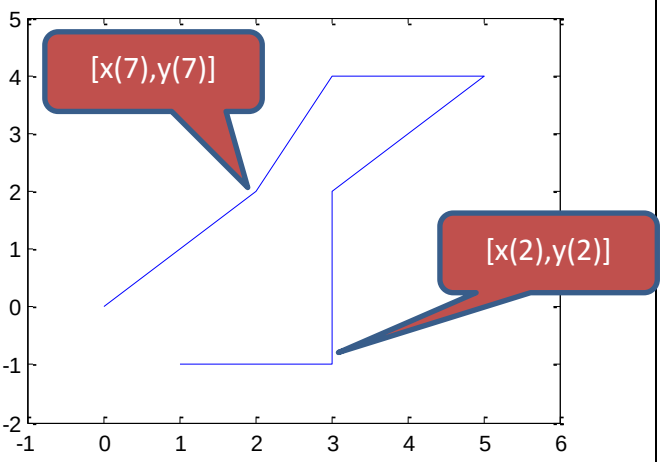


Only one set of numbers is supplied, that is x, so these are on vertical axis against position in the array.





When two arrays are supplied (**must be same length**), the first is used for the horizontal axis.



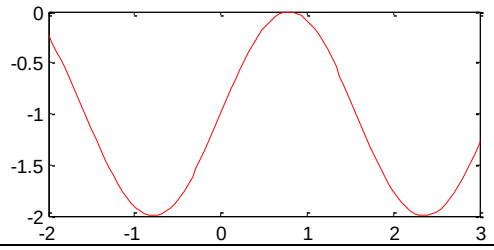
PLOTTING MATHEMATICAL FUNCTIONS: The basic technique is to create the values of the input (domain) as an array and then the corresponding values of range (or output). A plot is then done as illustrated above.

```
x=linspace(-2,3,100);
y=sin(2*x)-1;
plot(x,y,'r');
```

Input values

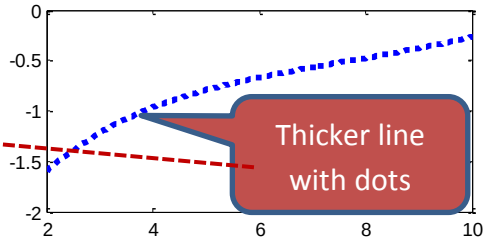
Output values

Plot in red




```
zz=linspace(2,10,200);
w=exp(-0.2*zz).*zz-3*cos(zz/10);
plot(zz,w,'b:','linewidth',3)
```

NOTE: The use of '.' above tells MATLAB to multiply the arrays index by index. As zz is an array zz*zz would be mathematical incorrect (meaningless) otherwise.



PLOTTING SYMBOLIC FUNCTIONS: Symbolic variables in MATLAB are use to support algebra and therefore do not have numeric values. However, one can determine associated numerical values by evaluating the function (variable) at defined inputs – this is done using the function **subs** which is short for substitute in. Once the variable has been evaluated, or translated to numbers, it can be plotted in the normal way.

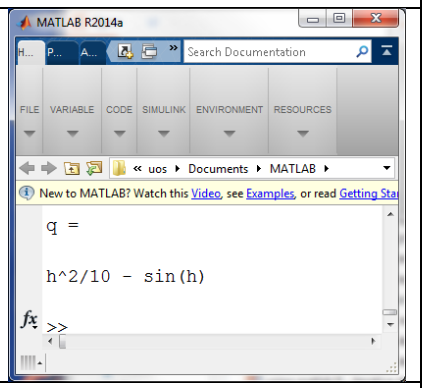
```
syms h
q=(h^2/10)-sin(h);
hval=linspace(0,4,250);
qval=subs(q,hval);
plot(hval,qval);
```

Create symbolic variable

Create symbolic function

Use subs to evaluate symbolic function **q** at given values **hval**.

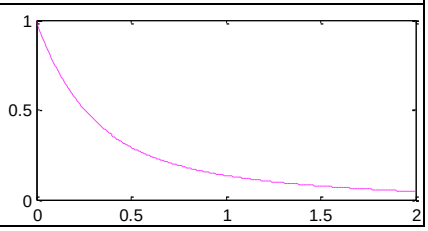
Plot in normal way



The use of subs is a useful tool for plotting the solution to ODEs obtained via **dsolve** as illustrated with the code here.

```
yt=dsolve('Dy+4*y=exp(-t)','y(0)=1');
tt=linspace(0,2,300);
plot(tt,subs(yt,tt),'m--')
```

Use subs to evaluate symbolic function **yt** at given values **tt**.



REMARK: Type **help plot** or **help subs** in the command window to find out more.