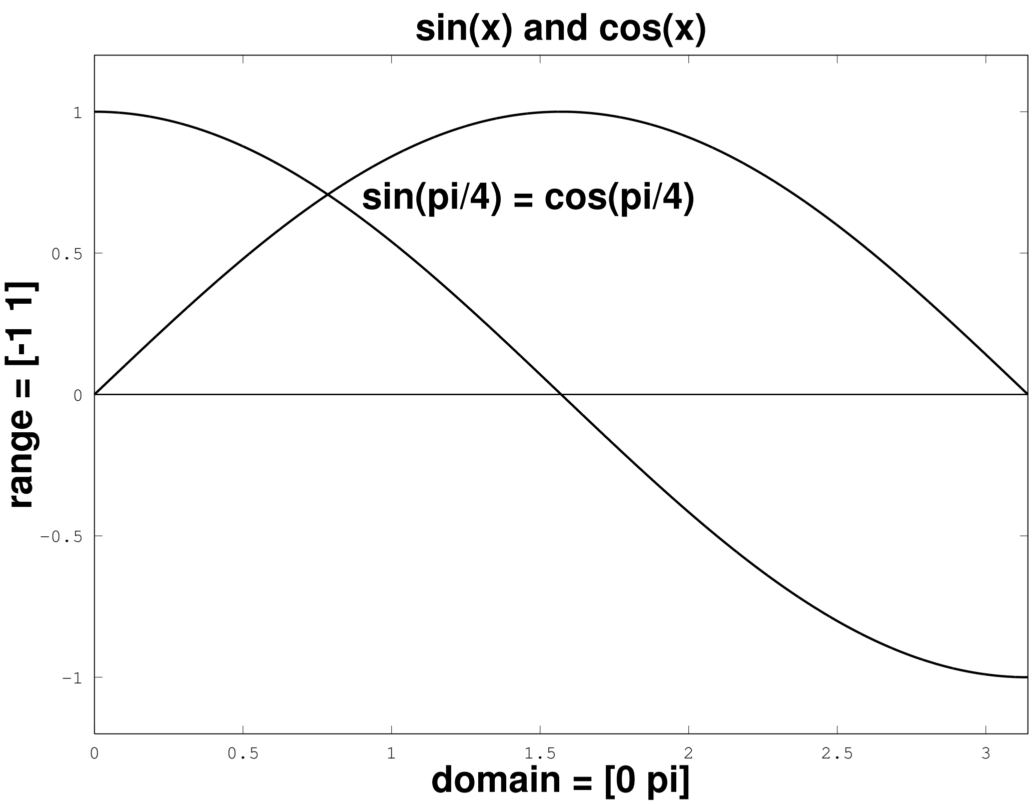
Section 07: Graphing

Octave uses Gnuplot to do the graphing. One could learn how to use Gnuplot directly, but it is better to learn Octave and let it do the work.

The following example, in which we plot y = sin(x) and y = cos(x) on the interval $[0, \pi]$, shows all the basic steps involved in plotting, labelling, and printing. There are many options, e.g. using dashed lines, label fonts and sizes, but one should first learn the basics. You should start with one function, then add a second function, then labels etc. You can play with line widths etc. to suit your taste.




```
%    first clear all preceeding graphs
octave:1> clf % = ''clear function''
%
%    tell Octave the range of values of interest
%    the sine and cosine go from -1 to 1, but the graph
%    will be nicer if we go slightly below and above
%
%    draw the Octave axes, which show the values
octave:2> axis([0 pi -1.2 1.2])
%
%    keep the Octave axes and all succeeding plots
octave:3> hold
%
```

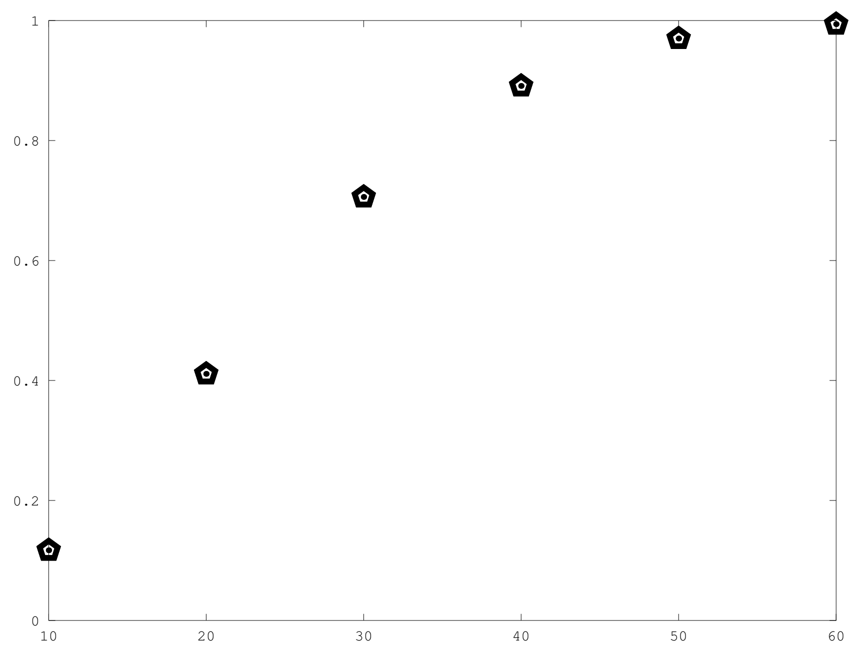
```
%    Draw the real x-axis from (0,0) to (0,pi)
%    a = [0 pi] gives the x coordinates of the x-axis
%    b = [0 0] gives the y coordinates of the x-axis
%
octave:4> a = [0 pi]; % NOT [0:.001:pi]
octave:5> b = [0 0];
%                     FIRST, SECOND
%    ALWAYS: plot( x-values, y-values)
%    increase the line thickness, by writing 'linewidth', 3
octave:6> plot(a, b , 'linewidth', 3)
%
%    now we are ready to plot the two graphs
%    give the x-values for the plot; use increments of .005
octave:6> x = [0 : .005 : pi];    % could use .001
%
%    first graph y = sin(x); use y1 for the name of the vector
octave:7> y1 = sin(x);
%
%    we plot the y-values (y1) against the x-values (x)
%    from now on graphs are thicker using 'linewidth', 5
octave:8> plot(x, y1, 'linewidth', 5)
%
% the same font descriptions were used)
%    second graph y = cos(x); use y2 for the name of the vector
octave:9> y2 = cos(x);
octave:10> plot(x, y2, 'linewidth', 5)
%
%    now add a title
%    the text is enclosed in apostrophes, not quotation marks
%    from now on we use:  'fontweight', "bold", 'fontsize', 20
octave:11> title('sin(x) and cos(x)', 'fontweight', "bold",
      'fontsize', 20)    % if the line is too long, push return
%
%    labels for the x and y axes, note ``xlabel'', not ``x-label''
octave:12> xlabel('domain = [0 , pi]')
octave:13> ylabel('range = [-1 , 1]')
%
%    include a text to indicate the point of intersection
%    the text is placed at (.9 , .7) [obtained by trial and error]
octave:14> text(.9 , .7, 'sin(pi/4) = cos(pi/4)')
%
%    save the graph in three formats:
%    1. png: image, better than jpg
%    2. pdf: for immediate printing
%    3. eps: encapsulated postscript; for importing into a Latex file
```

```
%    specify the full name of the output file
octave:15> print -dpng graph1.png
octave:16> print -dpdf graph1.pdf
octave:17> print -deps graph1.eps
%    check that the graphs are there using ls = ''list''
octave:18> ls *.png
      graph1.png    % yes, its there!
```

Above we plotted a continuous function, but the same holds for discrete data. The one difference is that to indicate that we just want to plot the points using a pentagon ("p") of large size 12 we add **"p", "markersize",12** to the plot command. [Other options are "o" , "+" , "x" , "*", "h" (hexagon), "s" (square),"^" (triangle).]

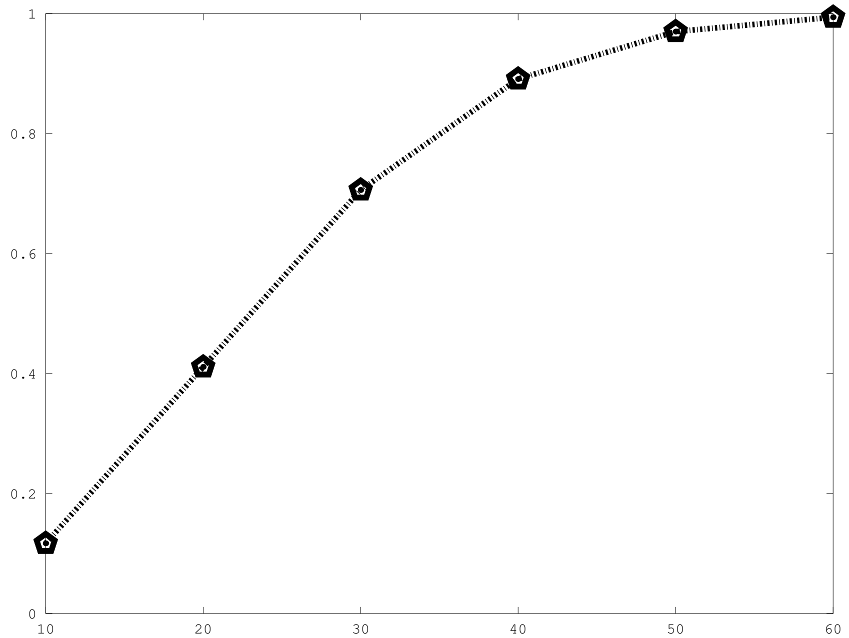
To illustrate this we use the birthday probabilities from section 06A:

```
octave:19> indices = [10 : 10 : 60];
octave:20> birthday_prob = [0.117 0.411 0.706 0.891 0.970 0.994];
octave:21> plot(indices, birthday_prob,"p","markersize",12)
```



If in addition to placing the pentagons at the data points we wanted to connect them by a thin dashed line, all we would have to do is **hold** the graph and then do a new plot which does not have the **"p"** and **,"markersize",12** commands. We can make the line alternate between a dash and a dot by using the command **'linestyle, '-.'**. [Other options are '--' , ':' , '-' (solid).] Lines and curves can also be coloured.

```
octave:22> hold
octave:23> plot(indices , birthday_prob , "linestyle" , '-.')
```



Try these

1. Plot the graphs of $y = x^2$, $y = \sqrt{x}$, $y = x$ on the same axes. Let x vary between -0.5 and 2.5.

   Print the line $y = x$ with dash marks using the command:
   Plot(x,y, **'linestyle', '--'**) [note the apostrophes and the comma between the last two commands.]

   Print the other two curves with a thicker line using the command:
   plot(x, y, **'linewidth', 8**) [note the apostrophes and the comma between the last two commands.]

   Put the labels "x-axis", "y-axis". The two functions are inverse functions of one another and are symmetric about the line $y = x$. Give the title "Inverse functions" to the graph. Label the points of intersection of the two graphs.

2. Consider an isosceles triangle with angles A, B, A (in degrees) and sides a, b, a.

i. Create a function that determines the ratio $\frac{b}{a}$ as a function of B. [suggestion: use the law of sines; you have to convert from degrees to radians.]

ii. Plot the ratio $\frac{b}{a}$ as a function of B, as B varies from 1^o to 179^o. Check the answers at 1^o and 179^o by making a sketch. Why were 0^o and 180^o not included?

iii. Above we used regular pentagons to mark the points. Use (i) to find the ratio of the diagonal of a regular pentagon to its side. [answer: see `G` in section 08.]