General Procedure for plotting functions:

- Plan an x range to plot
- Create an x vector
- Calculate the y value for each x
- Plot x vs. y

Revise x range and vector if necessary

Example: Projectile Function

1. Problem Statement:

We wish to create a function that will plot the height vs. time of a projectile thrown straight up with an initial velocity of v_0. The graph should cover from initial launch through when the projectile returns to the ground.

2. Inputs: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Input Source*
v0	the initial velocity	m/sec.	command line

3. Output: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Output type*
NA	A plot of height vs. time	m/s & s	display figure

4. Solution Steps/test calculation:

Calculate end time (solve equation for height = 0)

$$0 = v_0 t - \frac{1}{2} g t^2 \quad \Rightarrow \quad 0 = \left(v_0 - \frac{1}{2} g t\right) t$$

this will be zero for t = 0 and for

$$0 = v_0 - \frac{1}{2} g t \quad \Rightarrow \quad t_{end} = \frac{2 v_0}{g}$$

we then need to create several points from 0 to t_{end}.
then we can plug those times into the height formula & a vector of heights
finally we plot height vs. time and label the graph.

$$h = v_0 t \; - \; \frac{1}{2} g t^2$$

To test a graph we need a couple of example points on the graph – so for v_0 = 50 m/sec. let's calculate the max time and the height at half the max time.

$$t_{end} = \frac{2(50m/s)}{(9.8m/s^2)} = 10.2\sec. \qquad h = (50m/s)(5.1s) - \frac{1}{2}(9.8m/s^2)(5.1s)^2 = 255m - 127.4m$$

h (5.1 sec.) = 127.6m

Develop and validate a MATLAB function to complete this problem

5.
Coding:

```
function projectile(v0)
% This function calculates the height of a projectile versus time when
% launched upward with an initial speed of v0. It plots ten points on the
% graph of height vs. time
%
% function projectile(v0)
% input:          v0 = initial speed (m/sec)
% output:         a graph of height vs. time
% S. Scott Moor        February 2006

% other variables
% g = acceleration of gravity (m/sec2)
% h = vector of heights of the projectile (m)
% t = vector of times corresponding to above heights (m)
% tend = time until projectile hits the ground again (s)

% Calculate final time
g=9.8;
tend = 2*v0/g;
% Set up the time vector
t=0:0.02*tend:tend;
% Calculate the corresponding heights
h=v0*t - 0.5*g*t.^2;
% Plotting the results
plot(t,h)
xlabel 'time (s)', ylabel 'height (m)'
V = num2str(v0);
b =['Projectile height, v_0 = ', V, ' m/sec'];
title(b)
```

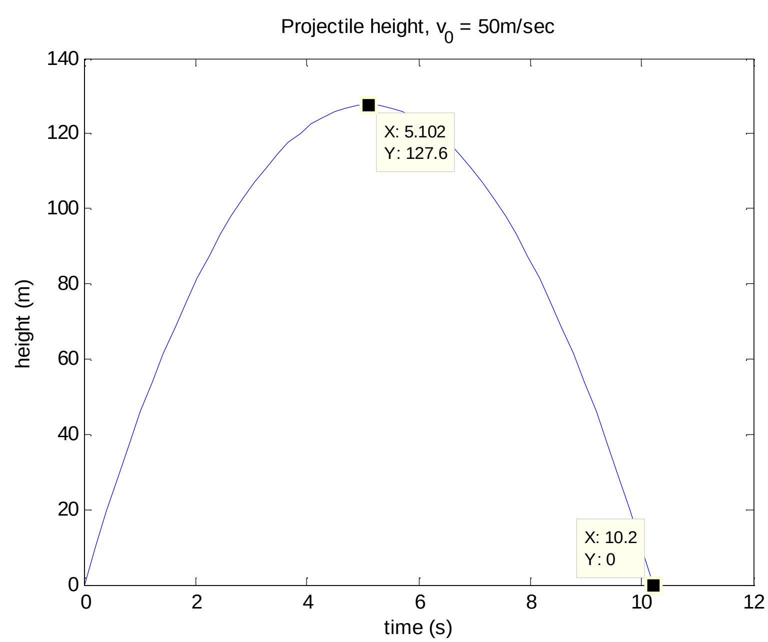
Validation:

6. Run test case (from step 4)

Command issued

```
>> projectile(50)
```

Resulting graph with data cursers used to identify the two test points calculated in step 4 is shown to the right. The two test points match that calculation exactly.



7. Reality check.

For the first time we do need to complete this step because the calculation in 4 does not do the complete calculations the program does. Therefore we need to look at the entire result – in this case the shape of the curve and ask does it look realistic? In this case it does – we expected and got a simple parabola.

8. Test on wider cases:

Now we need to check that it will work in a range of cases. I used the sequence of commands shown in the box at the right to produce the figure below.

```
>> subplot(2,2,1)
>> projectile(0.5)
>> subplot(2,2,2)
>> projectile(15)
>> subplot(2,2,3)
>> projectile(100)
>> subplot(2,2,4)
>> projectile(1000)
>>
```

The program is working well for a wide range of input values producing logical results and properly formatted graphs.