# Modelling Motion in a Restive Medium

In labs, we have considered the problem of an object rising (or falling) in the Earth' s gravitational field in the presence of friction. Here, let' s see how we can construct a simple model of this motion. We will assume a linear dependence of friction, in other words, air friction will be proportional the first power of the speed.

If we start with an object being fired upward at some initial velocity, say 10 m/s, our equation of motion is :

$$m\,\ddot{y} = -m\,g - c\,v \Rightarrow \ddot{y} = -g - (c/m)\,v$$

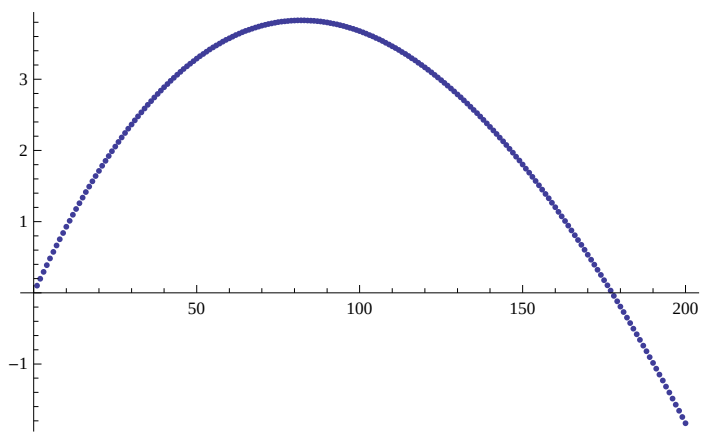
where m and g are mass and acceleration due to gravity, v is the instantaneous speed, and c is some constant which we will arbitrarily vary. The notation $\ddot{y}$ means the second derivative of distance (or acceleration)

We can write a simple Mathematica program :

In[23]:=
```
Clear[m, y, g, c, v, h, nterms]
m = 1.0; y[0] = 0; v[0] = 10; g = 9.81; h = 0.01; c = 0.5;
(* We set the mass to 1 kg,
the coefficient c to 0.5 (which is OK for small objects in the atmosphere),
we start from the ground with an initial upward velocity of 10m/s *)

(* Set up the recursion relations for v and y: *)
v[n_] := v[n] = v[n - 1] + h (-g - (c / m) v[n - 1])
y[n_] := y[n] = y[n - 1] + h (v[n] + v[n - 1]) / 2
(* Let's see what a plot looks like *)
ListPlot[Table[y[n], {n, 200}]]
```

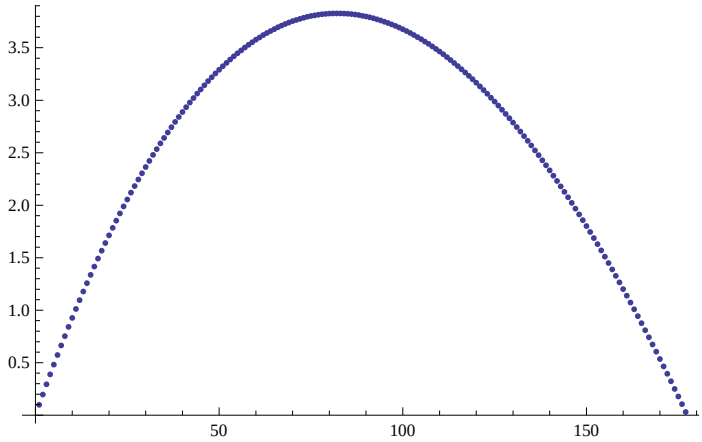
Out[27]=




```
(*And this looks about right; the object rises,
then falls.  Of course we have the situation that by choosing 200 points to plot,
we are producing negative heights, which we know is not physically meaningful.  So,
our first tweak to the program above will be to have the program figure
 out how many points I need to plot, and automatically end the plotting
 when the object hits the ground.  Review Catch and Throw in the doc center
 for more details of the technique below. I will set the variable nterms,
to be the value of n when the object hits (or is close to hitting the ground): *)
```

In[29]:=
```
nterms = Catch[Do[If[y[n] < 0, Throw[n - 1]], {n, 10000}]];
ListPlot[Table[y[n], {n, nterms}]]
```

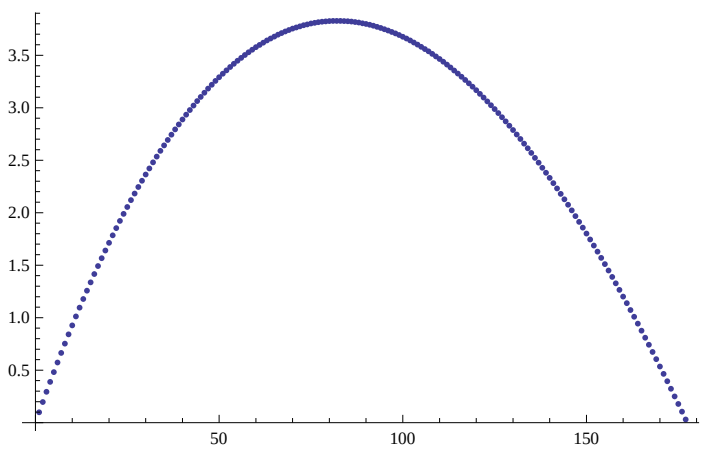
Out[30]=




```
(* And this looks much better! Now,
let's see if we can have the program compute time of flight and maximum height *)
```

In[39]:=
```
maxht = Catch[Do[If[y[n] < y[n - 1], Throw [y[n - 1]], {n, 10000}], {n, 2000}]];
ListPlot[Table[y[n], {n, nterms}]]
Print["The time of flight is ", nterms h,
 "  secs. The time of flight without friction would be ", 2 v[0] / g, " secs"]
Print["The maximum height achieved is ", maxht,
 " meters.  The max ht achieved without friction is ", 2 v[0]^2 / g, " meters"]
```

Out[40]=




```
The time of flight is 1.77
   secs. The time of flight without friction would be 2.03874 secs

The maximum height achieved is 3.82746
  meters.  The max ht achieved without friction is 20.3874 meters
```

With this working model in hand, you can simulate the effects of varying mass, coefficient of drag, etc. See if you can write the equations for a 2 dimensional trajectory with linear friction. (Anticipate being asked this again sometime soon...)