Projectile Motion with Air Resistance

This worksheet will show how one can use Maple to solve the problem of a projectile moving under the influence of a graivtational force and a resistive force that depends linearly on the velocity.

One should always begin with a restart command and then hit enter after each line consecutively.

```
> restart:
```

Equations of Motion

We first enter the *x* and *y* components from Newton's Second Law. One has to provide the positions (*x* and *y*) as functions of time. We also will need to refer to the equations, so we store the equation using a variable name (xcomp and ycomp).

```
> xcomp:=m*diff(x(t),t$2)=-k*m*diff(x(t),t);
```

$$xcomp := m\left(\frac{d^2}{dt^2}\mathrm{x}(t)\right) = -k\,m\left(\frac{d}{dt}\mathrm{x}(t)\right)$$

```
> ycomp:=m*diff(y(t),t$2)=-k*m*diff(y(t),t)-m*g;
```

$$ycomp := m\left(\frac{d^2}{dt^2}\mathrm{y}(t)\right) = -k\,m\left(\frac{d}{dt}\mathrm{y}(t)\right) - m\,g$$

We can obtain the general solution of each of these second order differential equations using dsolve. Note that Maple inserts its own form for the arbitrary constants.

```
> dsolve(xcomp); dsolve(ycomp);
```

$$\mathrm{x}(t) = \_C1 + \_C2\,\mathbf{e}^{(-k\,t)}$$

$$\mathrm{y}(t) = -\frac{\mathbf{e}^{(-k\,t)}\,\_C1}{k} - \frac{g\,t}{k} + \_C2$$

If you want partiular solutions, then you need to provide initial conditions for each differential equation. We will set the origin as the release point and give the projectile an initial velocity as $(U,V) = (v0\cos(\theta), v0\sin(\theta))$. Note how the initial conditions are inserted with the equation inside braces. We also use the differential operator *D*.

```
> dsolve({xcomp,x(0)=0,D(x)(0)=U});
  dsolve({ycomp,y(0)=0,D(y)(0)=V});
```

$$\mathrm{x}(t) = \frac{U}{k} - \frac{U\,\mathbf{e}^{(-k\,t)}}{k}$$

$$\mathrm{y}(t) = -\frac{\mathbf{e}^{(-k\,t)}\,(V\,k+g)}{k^2} - \frac{g\,t}{k} + \frac{V\,k+g}{k^2}$$

We want to plot these functions for various values of the parameters. Even though these look like useful functions, they cannot be used yet. So, we will store the right hand sides of the expressions in new variables, *X* and *Y*.

```
> X:=rhs(dsolve({xcomp,x(0)=0,D(x)(0)=U}));
  Y:=rhs(dsolve({ycomp,y(0)=0,D(y)(0)=V}));
```

$$X := \frac{U}{k} - \frac{U\,\mathbf{e}^{(-k\,t)}}{k}$$

$$Y := -\frac{\mathbf{e}^{(-k\,t)}\,(V\,k+g)}{k^2} - \frac{g\,t}{k} + \frac{V\,k+g}{k^2}$$

Let's specify some of the constants. Since the angle is in degrees, we make sure we convert to radians.

```
> g:=9.8: v0:=600: theta:=60:
  U:=v0*cos(convert(theta*degrees,radians));
  V:=v0*sin(convert(theta*degrees,radians));
```

$$U := 300$$

$$V := 300\sqrt{3}$$

Let's look at the solutions. They depend upon *t* and *k*.

```
> X; Y;
```

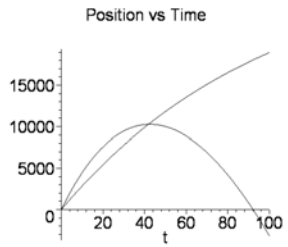
$$\frac{300}{k} - \frac{300\,\mathbf{e}^{(-k\,t)}}{k}$$

$$-\frac{\mathbf{e}^{(-k\,t)}\,(300\sqrt{3}\,k+9.8)}{k^2} - \frac{9.8\,t}{k} + \frac{300\sqrt{3}\,k+9.8}{k^2}$$

We can pick a value for *k* and plot these as functions of time. We could set k:=0.01. But this is a good time to use the command subs. 2D Plotting is done using the plot command. It takes the form plot(expression, range). There are many other things one can control in a plot. Here we plot two functions on the same axes by placing multiple expressions between braces. Also, to allow one to use different *k* values, we will store the value of interest in *k0*.

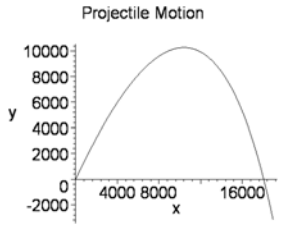
Notice how the plot of *X* is not linear. Also, the rise time appears different than the fall time.

```
> k0:=0.01:
  plot({subs(k=k0,X),subs(k=k0,Y)},t=0..100,title=`Position vs
  Time`,color=black);
```



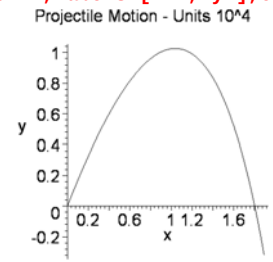
We can also plot *Y* vs *X*. When one has parametric equations, then one uses the form plot([x(t),y(t),t=a..b]). Notice that the result is not a parabola.

```
> plot([subs(k=k0,X),subs(k=k0,Y),t=0..100],title=`Projectile
  Motion`,labels=[`x`,`y`],color=black);
```



The units are large, so we can rescale the axes.

```
> plot([subs(k=k0,X/10^4),subs(k=k0,Y/10^4),t=0..100],title=`Projectile Motion - Units 10^4`,labels=[`x`,`y`],color=black);
```



Let's find the time of flight and the range. Since Y = 0 when the object returns to its launch height, we can solve the equation

```
> subs(k=k0,Y)=0;
```

$$-10000.\, e^{(-0.01\,t)}\,(3.00\sqrt{3}+9.8)-980.\,t+30000.\sqrt{3}+98000.=0$$

However, this is a trascendental equation. In Maple we can use the fsolve command. But, we see that the result returned is $t = 0$. This makes sense as the object is at zero height when it is first launched.

```
> fsolve(subs(k=k0,Y)=0,t);
```

$$0.$$

From the above plot of Y vs t, we see that the root is in the range [80, 100]. We can put this ranger into the fsolve command and get the time of flight.

```
> T:=fsolve(subs(k=k0,Y)=0,t=80..100);
```

$$T := 92.10191668$$

Inserting this time into X, we obtain the range.

```
> subs({k=k0,t=T},X);
```

$$30000. - 30000.\, e^{(-0.9210191668)}$$

This needs to be evaluated using evalf.

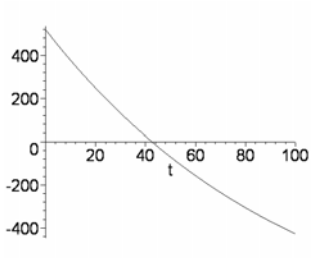
```
> evalf(subs({k=k0,t=T},X));
```

$$18056.60728$$

Let's find the rise time. At the top of the path the vertical velocity is zero. First, we need the vertical velocity.

```
> v:=diff(Y,t); plot(subs(k=k0,v), t=0..100,color=black);
  Trise:=fsolve(subs(k=k0,v));
```

$$v := \frac{e^{(-k\,t)}\,(300\sqrt{3}\,k+9.8)}{k}-\frac{9.8}{k}$$



$$Trise := 42.54112774$$

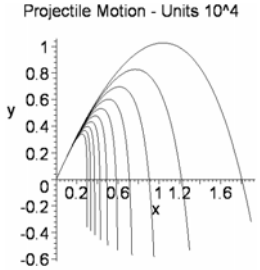
We can now find the time of fall (time of flight less rise time).

```
> T-Trise;
```

$$49.56078894$$

Finally, we can plot multiple solutions on one set of axes for different values of k. We make use of the seq command.

```
> plot({seq([subs(k=n*.01,X/10^4),subs(k=n*.01,Y/10^4),t=0..100],n=1..10)},title=`Projectile Motion - Units 10^4`,labels=[`x`,`y`],color=black);
```



Perturbation Approach to Flight Time

We will seek the time of flight using approximation methods in the case that k is sufficiently small. First, we clear all variables. Then recall the above solution for $y(t)$.

```
> restart: y:=-1/k^2*exp(-k*t)*(V*k+g)-1/k*g*t+(V*k+g)/k^2;
```

$$y := -\frac{e^{(-k\,t)}\,(V\,k+g)}{k^2}-\frac{g\,t}{k}+\frac{V\,k+g}{k^2}$$

Let T be the time of flight. At $t = T$, $y = 0$. So, we have the following:

```
> -1/k^2*exp(-k*T)*(V*k+g)-1/k*g*T+(V*k+g)/k^2=0;
```

$$-\frac{e^{(-kT)}(Vk+g)}{k^2}-\frac{gT}{k}+\frac{Vk+g}{k^2}=0$$

Rewriting, we have

```
> T = (kV+g)/g/k*(1-exp(-k*T));
```

$$T=\frac{(kV+g)(1-e^{(-kT)})}{gk}$$

Recall the Maclaurin series expansion

```
> series(exp(x),x=0,5);
```

$$1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+O(x^5)$$

Thus, for kT small,

```
> T = (kV+g)/g/k*(k*T-1/2*(k*T)^2+1/6*(k*T)^3);
```

$$T=\frac{(kV+g)\left(kT-\frac{1}{2}k^2T^2+\frac{1}{6}k^3T^3\right)}{gk}$$

Collect T terms and then divide out the factor $\frac{T}{2k^2}$:

```
> (g*k/(k*V+g)-k)*T=-1/2*k^2*T^2+1/6*k^3*T^3;
```

$$\left(\frac{gk}{Vk+g}-k\right)T=-\frac{1}{2}k^2T^2+\frac{1}{6}k^3T^3$$

```
> (g*k/(k*V+g)-k)=-1/2*k^2*T+1/6*k^3*T^2; simplify(%*2/k^2);
```

$$\frac{gk}{Vk+g}-k=-\frac{1}{2}k^2T+\frac{1}{6}k^3T^2$$

$$-\frac{2V}{Vk+g}=\frac{T(-3+kT)}{3}$$

For small k we the left side is approximated as

```
> series(-2*V/(V*k+g),k=0,3);
```

$$-\frac{2V}{g}+\frac{2V^2}{g^2}k-\frac{2V^3}{g^3}k^2+O(k^3)$$

Inserting the first terms in for T and then approximating T^2 term using $T=\frac{2V}{g}$, we have

```
> T= 1/3*k*T^2-((-2*V/g)+2*V^2/g^2*k); subs(T^2=(2*V/g)^2,%);
```

$$T=\frac{kT^2}{3}+\frac{2V}{g}-\frac{2V^2k}{g^2}$$

$$T=-\frac{2V^2k}{3g^2}+\frac{2V}{g}$$

Now, we check to see how good the approxmiation is and compare it to the previous result for the time of flight.

```
> g:=9.8: v0:=600: theta:=60:k0:=0.01:
  V:=v0*sin(convert(theta*degrees,radians)):
> evalf(-2/3*V^2/g^2*k0+2*V/g);
```

87.30173625

```
>
```