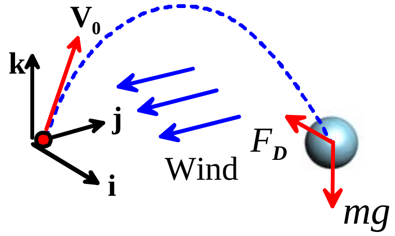
EN40: Dynamics and Vibrations

Homework 3: Solving equations of motion for particles

MAX SCORE 55 POINTS + 3 extra credit

Division of Engineering
Brown University



1. A spherical projectile with diameter D and mass m is launched from the origin with initial velocity vector $\mathbf{V}_0 = V_x\mathbf{i} + V_y\mathbf{j} + V_z\mathbf{k}$. A surface wind blows with air velocity $\mathbf{V}_w = W_x\mathbf{i} + W_y\mathbf{j} + W_z\mathbf{k}$, subjecting the projectile to a drag force

$$\mathbf{F}_D = -\frac{1}{2}\rho C_D \frac{\pi D^2}{4} V\left((v_x - W_x)\mathbf{i} + (v_y - W_y)\mathbf{j} + (v_z - W_z)\mathbf{k}\right)$$

where $V = \sqrt{(v_x - W_x)^2 + (v_y - W_y)^2 + (v_z - W_z)^2}$ is the magnitude of the particle's velocity relative to the air, $\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is the projectile velocity, C_D is the drag coefficient, and ρ is the air density.

a. The motion of the system will be described using the (x,y,z) coordinates of the particle. Write down the acceleration vector in terms of time derivatives of these variables.

$$\mathbf{a} = \frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j} + \frac{d^2z}{dt^2}\mathbf{k}$$

[1 POINT]

b. Write down the vector equation of motion for the particle (**F**=m**a**).

The equation of motion is

$$\mathbf{F}_D = -\frac{1}{2}\rho C_D \frac{\pi D^2}{4} V\left((v_x - W_x)\mathbf{i} + (v_y - W_y)\mathbf{j} + (v_z - W_z)\mathbf{k}\right) - mg\mathbf{k} = m\left(\frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j} + \frac{d^2z}{dt^2}\mathbf{k}\right)$$

[1 POINT]

c. Show that the equation of motion can be expressed in MATLAB form as

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -cV(v_x - W_x) \\ -cV(v_y - W_y) \\ -g - cV(v_z - W_z) \end{bmatrix}$$

and give a formula for the constant c.

The {**i**,**j**,**k**} components of the EOM give

$$-\frac{1}{2}\rho C_D \frac{\pi D^2}{4} V\left(v_x - W_x\right) = m\frac{d^2x}{dt^2}$$

$$-\frac{1}{2}\rho C_D \frac{\pi D^2}{4} V\left(v_y - W_y\right) = m\frac{d^2y}{dt^2}$$

$$-\frac{1}{2}\rho C_D \frac{\pi D^2}{4} V\left(v_z - W_z\right) - mg = m\frac{d^2z}{dt^2}$$

If we divide through by *m* and define

$$c = \frac{\pi}{8}\frac{\rho C_D D^2}{m}$$

these equations become

$$-cV\left(v_x - W_x\right) = \frac{d^2x}{dt^2} \qquad -cV\left(v_y - W_y\right) = \frac{d^2y}{dt^2} \qquad -cV\left(v_z - W_z\right) - g = \frac{d^2z}{dt^2}$$

Finally, we introduce v_x, v_y, v_z as additional unknowns, writing the equations as

$$\frac{dx}{dt} = v_x \qquad \frac{dy}{dt} = v_y \qquad \frac{dz}{dt} = z_y$$

$$\frac{dv_x}{dt} = -cV\left(v_x - W_x\right) \qquad \frac{dv_y}{dt} = -cV\left(v_y - W_y\right) \qquad \frac{dv_z}{dt} = -cV\left(v_z - W_z\right) - g$$

[2 POINTS]

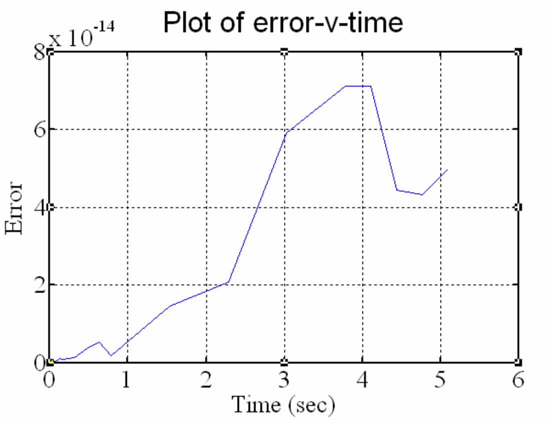
d. Modify the MATLAB code discussed in class to define a function that will calculate and plot the trajectory. Include an ‘event’ function that will stop the calculation when the projectile hits the ground (at *z=0*) (THERE IS NO NEED TO SUBMIT A SOLUTION TO THIS PROBLEM)

```
function trajectory_3d
% Function to plot trajectory of a projectile

 rho = 1.02; % Air density
 D = 0.04114; % Projectile diameter
 Cd = 0.25; % Drag coefficient
 m=0.0459; % projectile mass
 c = rho*Cd*pi*D^2/8/m;
 Wx=0;Wy=6.7;Wz=0; % Wind
 V0 = [67*cos(35*pi/180),0,67*sin(35*pi/180)]; % Inicial velocity
 time=30;

y0 = [0,0,0,V0]; % Initial conditions
options = odeset(‘Events’,@events);
[t,y] = ode45(@odefunc,[0,time],y0,options);

plot(y(:,2),y(:,3))

function dydt = odefunc(t,y)
   vmag = sqrt((y(4)-Wx)^2+(y(5)-Wy)^2+(y(6)-Wz)^2);
   dydt = [y(4);y(5);y(6);
  -c*(y(4)-Wx)*vmag;-c*(y(5)-Wy)*vmag;-9.81-c*(y(6)-Wz)*vmag];
end
function [eventvalue,stopthecalc,eventdirection] = events(t,y)
     % Function to check for a special event in the trajectory
```

```
        eventvalue = [y(3)];
        stopthecalc = [1];
        eventdirection = [-1];

    end
   end
```

[0 POINTS]

e. Test your code by calculating the trajectory of a projectile that is launched with velocity $\mathbf{V}_0 = 13\mathbf{i} + 25\mathbf{k}$ m/s, with *c=0,* and compare the MATLAB solution with the exact solution. A good way to do this is to plot the distance between the points where the particle should be, and where MATLAB predicts it to be, as a function of time.



The figure shows a plot of the difference between the exact solution and the MATLAB solution, defined as

$$e = \sqrt{\left(x^{exact} - x\right)^2 + \left(y^{exact} - y\right)^2}$$

The plot was produced using the following MATLAB statements

```
   for i =1:length(t)
       exact = exact_pos(t(i));
       error(i) = sqrt((exact(1)-y(i,1))^2 + (exact(2)-y(i,3))^2);
   end
   plot(t,error);

   function y = exact_pos(t)
       y(1) = V0(1)*t;
       y(2) = V0(3)*t-9.81*t^2/2;
   end
```

Note to graders – the graphs may not look like the one displayed above – the difference between MATLAB and the exact solution is just rounding errors in this case. Any graph with a small error is fine.

[3 POINTS]

f. Finally, run simulations with parameters representing a golf-ball (m=0.0459kg, $C_D = 0.25$, D=4.115cm, air density 1.02 kg/m^3) launched at 67 m/s (150 mph) at an angle of 35 degrees to the horizontal. Calculate difference in length of a drive hit into a 6.7m/s (15 mph) head wind to that hit into a 6.7m/s tailwind. Also, calculate the lateral deflection caused by a 6.7 m/s side-wind.

The simulations predict that:

1. With a 6.7m/s headwind, the ball travels 191m (208 yards)
2. With a 6.7m/s tailwind, the ball travels 242m (264 yards)

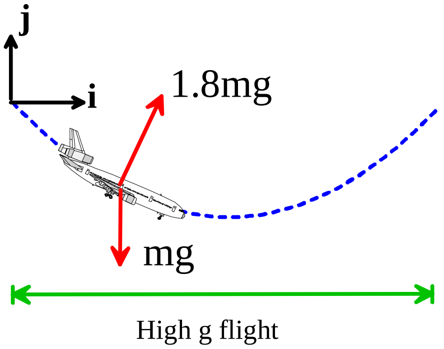
The difference is 51 m (56 yards)
With a 6.7m/s side wind the lateral deflection is 16m (17.5 yards)
(Numbers may not come out exactly equal to these – anything close should get credit.

[3 POINTS]
TOTAL 10 POINTS

2. In this problem we revisit the NASA 'zero-g simulator' aircraft trajectory. In the preceding homework you analyzed the `zero g' portion of the flight-path. The goal here is to calculate the velocity and trajectory of the aircraft as it pulls out of the dive during the 'high-g' portion of the flight.

The figure shows a free body diagram for the aircraft. No engine power is used during this phase of the flight (can you see why?) and drag has been neglected for simplicity. Note that:

- The lift force acts perpendicular to the path of the aircraft.
- The aircraft is flown to keep the magnitude of the lift force constant (this is to avoid structural damage to the airframe). The magnitude is 1.8*mg* where *m* is the aircraft mass and *g* is the gravitational acceleration.



a. The motion of the aircraft will be described by the components of its position vector (*x*,*y*). Write down the acceleration vector in terms of these variables.

The position vector is $\mathbf{r}=x\mathbf{i}+y\mathbf{j}$; the acceleration vector is $\mathbf{a}=\dfrac{d^2x}{dt^2}\mathbf{i}+\dfrac{d^2y}{dt^2}\mathbf{j}$

[1 POINT]

b. Let $\mathbf{v}=v_x\mathbf{i}+v_y\mathbf{j}$ denote the velocity vector of the aircraft. Note that this vector is tangent to the path, and hence find a formula for a unit vector parallel to the lift force, in terms of v_x and v_y. Use the result to write down the resultant force vector acting on the aircraft.

We can create a unit vector in the correct direction by calculating $\dfrac{\mathbf{k}\times\mathbf{v}}{|\mathbf{k}\times\mathbf{v}|}$. To see this, recall that the cross product of two vectors is perpendicular to both vectors – so the vector is perpendicular to **k** (and so lies in the plane of the figure) and also perpendicular to **v** (and so perpendicular to the path. Evaluating the cross product gives

$$\mathbf{n}=\frac{-v_y\mathbf{i}+v_x\mathbf{j}}{\sqrt{v_x^2+v_y^2}}$$

The resultant force is therefore

$$\mathbf{F}=1.8mg\left(\frac{-v_y\mathbf{i}+v_x\mathbf{j}}{\sqrt{v_x^2+v_y^2}}\right)-mg\mathbf{j}$$

[2 POINTS]

c. Write down the vector equation of motion for the aircraft **F**=m**a**.

$$\mathbf{F}=m\mathbf{a}\Rightarrow 1.8mg\left(\frac{-v_y\mathbf{i}+v_x\mathbf{j}}{\sqrt{v_x^2+v_y^2}}\right)-mg\mathbf{j}=m\left(\frac{d^2x}{dt^2}\mathbf{i}+\frac{d^2y}{dt^2}\mathbf{j}\right)$$

[1 POINT]

d. Hence, show that the equation of motion for the aircraft can be expressed in the MATLAB form

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ -1.8gv_y / V \\ -g + 1.8gv_x / V \end{bmatrix}$$

where $V = \sqrt{v_x^2 + v_y^2}$.

As usual we need to introduce v_x, v_y as additional unknowns. These are related to the position by

$$\frac{dx}{dt} = v_x \qquad \frac{dy}{dt} = v_y$$

In addition, we re-write the two components of the equations of motion as

$$\frac{dv_x}{dt} = -\frac{v_y g}{\sqrt{v_x^2 + v_y^2}} \qquad \frac{dv_y}{dt} = \frac{v_x g}{\sqrt{v_x^2 + v_y^2}} - g$$

Collecting the equations of motion into vector form gives the solution

[2 POINTS]

e. Write a MATLAB script that will solve this equation of motion, given initial values for [x,y, v_x ,v_y]. (THERE IS NO NEED TO SUBMIT A SOLUTION TO THIS PROBLEM)

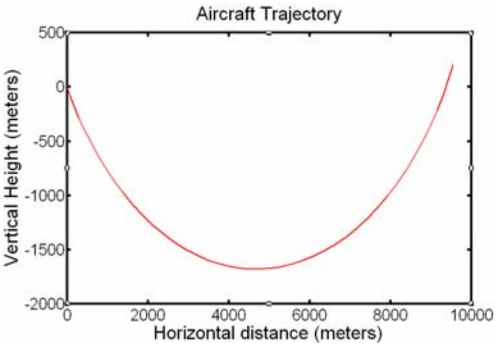
Here's a basic MATLAB script (not required)

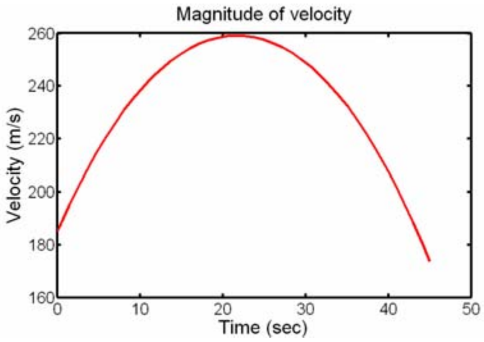
```
function aircraft_trajectory
g = 9.81;
y0 = [0,0,123.5,-137.9];
time=50;
[t,y] = ode45(@eom,[0,time],y0);
plot(y(:,1),y(:,2));
figure
vmag = sqrt(y(:,3).^2 + y(:,4).^2);
plot(t,vmag);

function dydt=eom(t,y)
vmag = sqrt(y(3)^2+y(4)^2);
dydt = [y(3);y(4);-1.8*g*y(4)/vmag;-g+1.8*g*y(3)/vmag];
end
```

[0 POINTS]

f. Assume that the aircraft enters the 'high-g' portion of flight with velocity $\mathbf{v} = 244\mathbf{i} - 264\mathbf{j}$ knots (CONVERT TO m/s IF YOU ARE USING SI UNITS FOR g!!). You can assume that the aircraft is at the origin at this instant. Plot (i) the subsequent trajectory of the aircraft; and (ii) the magnitude of the velocity of the aircraft as a function of time, for a period of 50 seconds. Please label your plots What is the maximum speed of the aircraft during the dive?

The figures are shown below





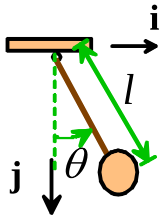
[4 POINTS]

This calculation suggests that the aircraft is being flown very close to its design limits. The max airspeed for a 737 is listed as Mach 0.84, which is 251 m/s at 30000ft. Our calculation predicts that this speed is exceeded. We have neglected drag, so it is likely that the aircraft just hits its max allowable speed at the bottom of the dive. It is this limitation, together with the 1.8mg lift force limitation, that restrict the maximum time for zero g flight. NASA have clearly done their homework…

[TOTAL: 10 POINTS]

3. To understand the motivation for this problem, read through the following publication:

> Crisco Joseph J; Fujita Lindsey; Spenciner David B, 'The dynamic flexion/extension properties of the lumbar spine in vitro using a novel pendulum system.' Journal of biomechanics 2007; 40(12):2767-73

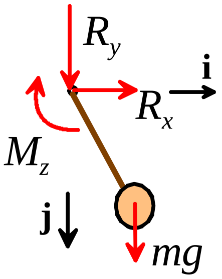


The goal of this problem is to analyze the motion of the pendulum system used in Professor Crisco's experiments. You will calculate the swing angle of the pendulum θ shown in the figure as a function of time.

The figure shows a free body diagram for the pendulum. The shaft of the pendulum is treated as a massless frame. The moment acting at the pivot represents the effects of elastic stiffness and viscous damping in the spinal joint. The magnitude of the moment is related to the rate of swing of the pendulum by

$$M_z = \eta\omega + k\theta$$

where η, k are constants and $\omega = d\theta / dt$



a. Write down the position vector of the mass on the end of the pendulum in terms of θ.

$$\mathbf{r} = l\left(\sin\theta\mathbf{i} + \cos\theta\mathbf{j}\right)$$

[1 POINT]

b. Hence, calculate the acceleration vector of the mass on the end of the pendulum, in terms of θ and its time derivatives.

This can be done with MAPLE:

```
> with(VectorCalculus):
```

```
Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: *,
+, ., Vector, diff, int, limit, series
```

```
> r := <l*sin(theta(t)),l*cos(theta(t))>;
```

$$r := l\sin(\theta(t))\,\mathbf{e}_{\mathbf{x}} + l\cos(\theta(t))\,\mathbf{e}_{\mathbf{y}}$$

```
> diff(%,t$2);
```

$$\left(-l\sin(\theta(t))\left(\frac{d}{dt}\theta(t)\right)^2 + l\cos(\theta(t))\left(\frac{d^2}{dt^2}\theta(t)\right)\right)\mathbf{e}_{\mathbf{x}} + \left(-l\cos(\theta(t))\left(\frac{d}{dt}\theta(t)\right)^2 - l\sin(\theta(t))\left(\frac{d^2}{dt^2}\theta(t)\right)\right)\mathbf{e}_{\mathbf{y}}$$

So the solution is

$$\mathbf{a} = \left[-l\left(\frac{d\theta}{dt}\right)^2\sin\theta + l\frac{d^2\theta}{dt^2}\cos\theta\right]\mathbf{i} - \left[l\left(\frac{d\theta}{dt}\right)^2\cos\theta + l\frac{d^2\theta}{dt^2}\sin\theta\right]\mathbf{j}$$

[3 POINTS]

c. Hence, write down **F**=m**a** for the system.

The force vector is easy to write down – we get

$$\mathbf{F} = R_x\mathbf{i} + \left(R_y + mg\right)\mathbf{j} = m\left\{\left[-l\left(\frac{d\theta}{dt}\right)^2\sin\theta + l\frac{d^2\theta}{dt^2}\cos\theta\right]\mathbf{i} - \left[l\left(\frac{d\theta}{dt}\right)^2\cos\theta + l\frac{d^2\theta}{dt^2}\sin\theta\right]\mathbf{j}\right\}$$

[1 POINT]

d. Write down **M**=**0** for the pendulum and its shaft, taking moments about the center of mass of the system (i.e. the particle)

Using any of the methods in the review notes, we get

$$R_x l\cos\theta + R_y l\sin\theta + M_z = 0$$

[1 POINT]

e. Eliminate the unknown reactions R_x, R_y from the equations of motion in (c) and (d) (you can solve the two components of (c) for the reactions and substitute into (d)) to show that θ satisfies the equation

$$ml^2\frac{d^2\theta}{dt^2} + \eta\frac{d\theta}{dt} + k\theta + mgl\sin\theta = 0$$

The two components of **F**=m**a** give

$$R_x = m\left[-l\left(\frac{d\theta}{dt}\right)^2 \sin\theta + l\frac{d^2\theta}{dt^2}\cos\theta\right]$$

$$R_y = -mg - m\left[l\left(\frac{d\theta}{dt}\right)^2 \cos\theta + l\frac{d^2\theta}{dt^2}\sin\theta\right]$$

Substituting these into $R_x l\cos\theta + R_y l\sin\theta + M_z = 0$ and using $\sin^2\theta + \cos^2\theta = 1$ gives

$$ml^2\frac{d^2\theta}{dt^2} + mgl\sin\theta + M_z = 0$$

Finally, substituting the formula given for M_z gives the required solution.

[2 POINTS]

f. Hence, show that the equations of motion can be expressed in MATLAB form as

$$\frac{d}{dt}\begin{bmatrix}\theta \\ \omega\end{bmatrix} = \begin{bmatrix}\omega \\ \dfrac{-\eta}{ml^2}\omega - \dfrac{k}{ml^2}\theta - \dfrac{g}{l}\sin\theta\end{bmatrix}$$

As always we introduce $\omega = d\theta / dt$, and re-write the second equation of motion as

$$\frac{d\omega}{dt} = -\frac{\eta}{ml^2}\omega - \frac{k}{ml^2}\theta - \frac{g}{l}\sin\theta = 0$$

Writing the two equations as vectors gives the required solution.

[2 POINTS]

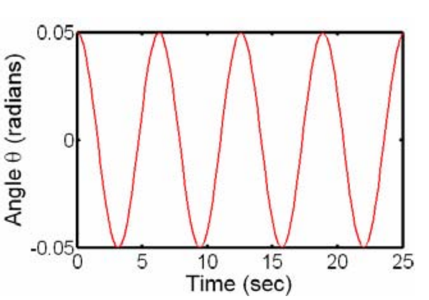
g. Write a MATLAB script to integrate the equations of motion. Check your code by running a simulation with $\eta = k = 0$, l=g and initial conditions $\theta = 0.05$ radians, $\omega = 0$ at time t=0. Plot θ as a function of time and compare the solution with the approximate result $\theta = 0.05\cos(t\sqrt{g/l})$ for small amplitude oscillation of a pendulum.

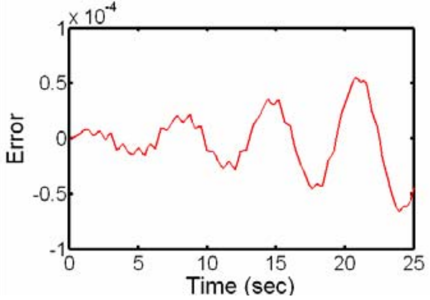
Here's a possible MATLAB script (but with parameters for part (h)). The MATLAB code need not be submitted.

```
function pendulum_hw3
% Solution to pendulum problem

l = 0.46; % Length of pendulum, in m
m = 50; % Mass of pendulum, in kg
k = 200; % Stiffness of pendulum, in Nm/radian
eta = 4; % Viscosity (in Nm/s)
time = 5; % Time of simulation, in sec
g = 9.81; % Gravitational acceleration, in m/s^2
y0 = [0.0277, 0];   % Initial conditions

[t,y] = ode45(@eom,[0,time],y0);

plot(t,y(:,1));

function dydt = eom(t,y)
```

```
dydt = [y(2);
          -eta*y(2)/(m*l^2)-k*y(1)/(m*l^2)-g*sin(y(1))/l];
end

end
```

The results and the difference between the MATLAB solution and the exact solution are plotted below:



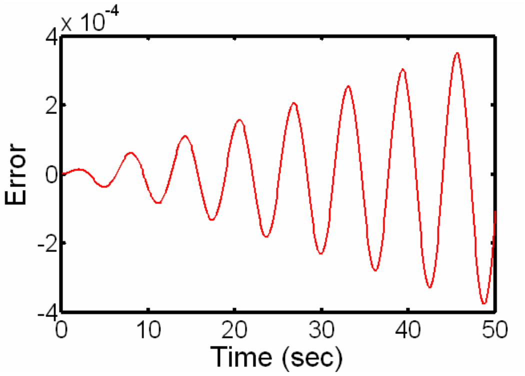


Note to graders – it's not necessary to do the comparison by calculating the error – it is also OK to simply plot the MATLAB solution and the analytical solution on the same plot (they come out on top of each other so you can't see the difference)

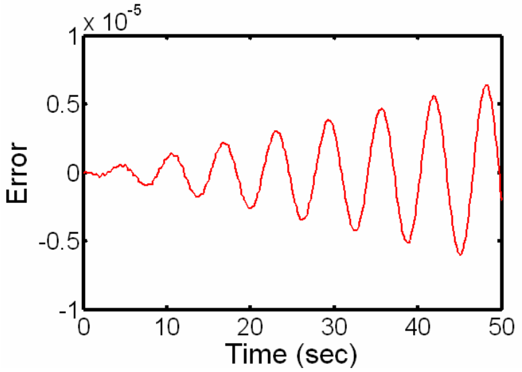
[3 POINTS]

The rest of this discussion was not required – but any comments or discussion should be given up to **[3 POINTS EXTRA CREDIT]**

The error is increasing with time – it would be a good idea to refine the tolerance with the 'options' and see what happens. In this case we see



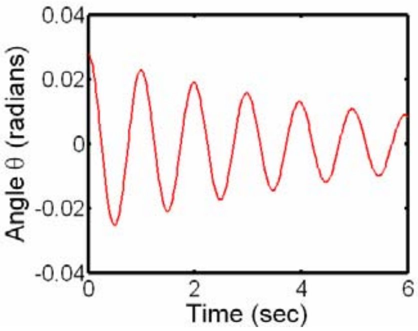
Now the solution is nice and smooth, suggesting that the error is not just computer error, but is something real. In fact, in this case the MATLAB solution is 'correct' and the error is in the approximate solution to the pendulum problem, which gets the period of oscillation slightly wrong. We can check this by reducing the amplitude of the swing – here's the result with $\theta = 0.005$ at *t=0*



Reducing the amplitude by a factor of 10 reduces the error by over a factor of 10, confirming our suspicion.

h. Finally, run a simulation with parameters similar to those used in the experiments: $l = 46cm$, m=50kg, k=200Nm/radian, $\eta = 4Nms / radian$ (the units reported for the viscous term in the paper are ambiguous). Use initial conditions $\theta = 0.0277$ radians and $\omega = 0$. Compare the predicted period of oscillation with the value of 1 second measured experimentally.

The predicted variation of angle with time is shown below



The predicted period is 0.99 sec – in very good agreement with experiment.

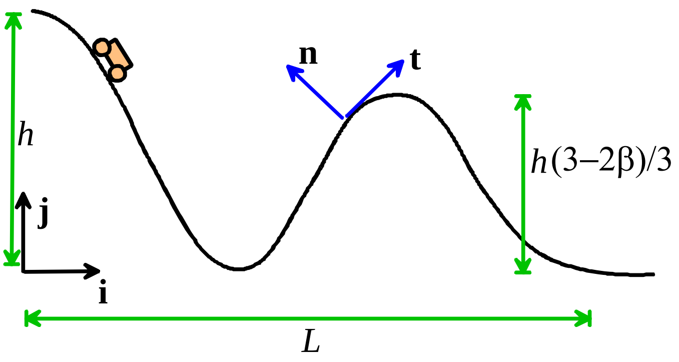
[2POINTS]

TOTAL: 15 POINTS + 3 POINTS EXTRA CREDIT

4. A roller-coaster ride has profile given by

$$y = h(1 - \beta \frac{x}{L})\cos^2 \frac{3\pi x}{2L}$$

as shown in the picture, where $0 < \beta < 1$ is a constant that controls the height of the second peak. The ride must be designed so that cars will travel over the second peak in the ride.



The objective of this problem is to predict the motion of a car as it travels along the path, accounting for the effects of air resistance. The calculation can then be used to estimate a value for β .

a. In solving this problem it will be helpful to introduce the *slope* μ and *bending* κ of the path, defined as

$$\mu = \frac{dy}{dx} \qquad \kappa = \frac{d^2 y}{dx^2}$$

Calculate formulas for μ and κ in terms of x (use MAPLE to do the calculus, and use the command combine(*expression*, trig) to simplify the results).

The expressions are computed below

```
> y := h*(1-beta*x/L)*cos(3*Pi*x/(2*L))^2;
```

$$y := h\left(-\frac{\beta\, x}{L} + 1\right)\cos\left(\frac{3\,\pi\, x}{2\,L}\right)^2$$

```
> diff(%,x):
> combine(%,trig);
```

$$\frac{1}{2}\,\frac{-h\,\beta\cos\left(\frac{3\,\pi\, x}{L}\right)L - h\,\beta\, L + 3\,h\,\pi\,\beta\, x\sin\left(\frac{3\,\pi\, x}{L}\right) - 3\,h\,\pi\, L\sin\left(\frac{3\,\pi\, x}{L}\right)}{L^2}$$

```
> diff(%,x):
> combine(%,trig);
```

$$\frac{1}{2}\,\frac{6\,h\,\beta\sin\left(\frac{3\,\pi\, x}{L}\right)\pi\, L + 9\,h\,\pi^2\,\beta\, x\cos\left(\frac{3\,\pi\, x}{L}\right) - 9\,h\,\pi^2\cos\left(\frac{3\,\pi\, x}{L}\right)L}{L^3}$$

These can be simplified to

$$\mu = -\frac{\beta h}{2L}\left(\cos\left(\frac{3\pi x}{L}\right)+1\right) - \frac{3\pi h}{2}\left(1-\frac{\beta x}{L}\right)\sin\left(\frac{3\pi x}{L}\right)$$

$$\kappa = \frac{3\pi\beta h}{L^2}\sin\left(\frac{3\pi x}{L}\right) - \frac{9h\pi^2}{2L^2}\left(1-\frac{\beta x}{L}\right)\cos\left(\frac{3\pi x}{L}\right)$$

[3 POINTS]

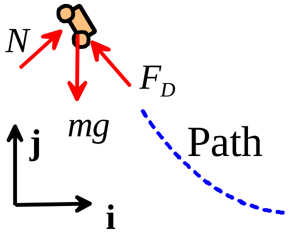
b. The motion of the car will be described using its (x,y) coordinates. Write down the acceleration vector in terms of these coordinates.

The acceleration is

$$\mathbf{a} = \frac{d^2 x}{dt^2}\mathbf{i} + \frac{d^2 y}{dt^2}\mathbf{j}$$

[1 POINT]

c. Draw a free body diagram for the car. Include (i) A reaction force N exerted on the car by the rollercoaster; (ii) An air drag force and (iii) gravity.



[2 POINTS]

d. Note that

$$\mathbf{t} = \frac{(\mathbf{i} + \mu \mathbf{j})}{\sqrt{1+\mu^2}}$$

is a unit vector tangent to the path (can you see this?). Hence, find a unit vector **n** normal to the path. Use these results to write down the resultant force vector acting on the car, in terms of the slope μ and other variables.

We can create a unit vector in the correct direction by calculating $\frac{\mathbf{k} \times \mathbf{t}}{|\mathbf{k} \times \mathbf{t}|}$ (for an explanation, see Problem 2). This gives

$$\mathbf{n} = \frac{(-\mu \mathbf{i} + \mathbf{j})}{\sqrt{1+\mu^2}}$$

[1 POINT]

e. Write down the vector equation of motion for the car **F**=m**a**.

We have

$$N\frac{(-\mu \mathbf{i} + \mathbf{j})}{\sqrt{1+\mu^2}} - F_D \frac{(\mathbf{i} + \mu \mathbf{j})}{\sqrt{1+\mu^2}} - mg\mathbf{j} = m\left(\frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j}\right)$$

$$\Rightarrow -\frac{\mu N + F_D}{\sqrt{1+\mu^2}}\mathbf{i} + \left(\frac{N - \mu F_D}{\sqrt{1+\mu^2}} - mg\right)\mathbf{j} = m\left(\frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j}\right)$$

The drag force is given by the usual expression $F_D = \frac{1}{2}\rho C_D A V^2$

[2 POINTS]

f. A constraint equation is required to ensure that the car follows the path of the rollercoaster. Note that *y* is related to *x* by the roller-coaster profile. Hence, show that the accelerations in the horizontal and vertical direction must be related by

$$\frac{d^2y}{dt^2} = \mu\frac{d^2x}{dt^2} + \kappa\left(\frac{dx}{dt}\right)^2$$

(This is tricky – start by showing $dy/dt = \mu dx/dt$ using the chain rule, and then differentiate again).

The chain rule gives

$$\frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt} = \mu\frac{dx}{dt}$$

Differentiating again, we have that

$$\frac{d^2y}{dt^2} = \frac{d\mu}{dx}\frac{dx}{dt}\frac{dx}{dt} + \mu\frac{d^2x}{dt^2} = \mu\frac{d^2x}{dt^2} + \kappa\left(\frac{dx}{dt}\right)^2$$

[2 POINTS]

g. Show that the equations of motion in (e) and the constraint equation in (f) can be expressed in the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \frac{\mu}{m\sqrt{1+\mu^2}} \\ 0 & 0 & 0 & 1 & \frac{-1}{m\sqrt{1+\mu^2}} \\ 0 & 0 & \mu & -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dv_x}{dt} \\ \frac{dv_y}{dt} \\ N \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ \frac{-cV^2}{\sqrt{1+\mu^2}} \\ \frac{-\mu cV^2}{\sqrt{1+\mu^2}} - g \\ -\kappa v_x^2 \end{bmatrix}$$

where *V* is the magnitude of the car's velocity, and give a formula for the constant c in terms of air density ρ, the drag coefficient of the car C_D; its frontal area *A* and its mass *m*.

As always we introduce v_x, v_y as additional unknowns. These are related to the position by

$$\frac{dx}{dt} = v_x \qquad \frac{dy}{dt} = v_y$$

These equations, together with the **i,j** components of the equation of motion, and the constraint equation, can be expressed as

$$
\begin{array}{cccccc}
1\times\dfrac{dx}{dt} & 0\times\dfrac{dy}{dt} & 0\times\dfrac{dv_x}{dt} & 0\times\dfrac{dv_y}{dt} & 0\times N & = v_x \\
0\times\dfrac{dx}{dt} & 1\times\dfrac{dy}{dt} & 0\times\dfrac{dv_x}{dt} & 0\times\dfrac{dv_y}{dt} & 0\times N & = v_y \\
0\times\dfrac{dx}{dt} & 0\times\dfrac{dy}{dt} & 1\times\dfrac{dv_x}{dt} & 0\times\dfrac{dv_y}{dt} & \dfrac{\mu}{m\sqrt{1+\mu^2}}\times N & = \dfrac{-cV^2}{\sqrt{1+\mu^2}} \\
0\times\dfrac{dx}{dt} & 0\times\dfrac{dy}{dt} & 0\times\dfrac{dv_x}{dt} & 1\times\dfrac{dv_y}{dt} & \dfrac{-1}{m\sqrt{1+\mu^2}}\times N & = \dfrac{-\mu cV^2}{\sqrt{1+\mu^2}} - g \\
0\times\dfrac{dx}{dt} & 0\times\dfrac{dy}{dt} & \mu\times\dfrac{dv_x}{dt} & -1\times\dfrac{dv_y}{dt} & 0\times N & = -\kappa v_x^2
\end{array}
$$

where $c = \dfrac{\rho C_D A}{2m}$

giving the required matrix equation.

[2 POINTS]

h. Write a MATLAB function to integrate the equations of motion. To test your code, run the simulation with $\beta = c = 0$, and set *h=L=m=1*. Start the car running with initial position *x=0*, *y=h*, and initial velocity (0.01,0), and run the simulation for time period [0,10]. Plot the trajectory of the car (use plot(y(:,1),y(:,2)). Note that the exact trajectory should be $y = \cos^2 \dfrac{3\pi x}{2}$. You should find that the trajectory is close to, but not exactly equal to this solution – if it is not close, you messed up your code (if this happens check your formulas for μ and κ in the code carefully).

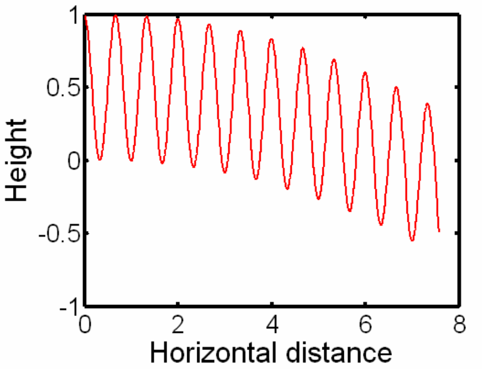
Here's a MATLAB script (including solution to (i), and with parameters in (k)). The MATLAB code need not be submitted with homework.

```
function rollercoaster

c = 0.05*1.02*4/500; % Specific drag coefficient
beta = 0.24; % The geometric parameter
L = 500.0; % Length of ride, in m
h = 100.; % Initial height of ride
m= 500.; % Mass of car
g=9.81; % Gravitational acceleration
y0 = [0,h,0.01,0];
reltol = 0.00000001;
options = odeset('RelTol',0.00000001, 'Events',@events);
[t,y] = ode45(@eom,[0,10],y0,options);
plot(y(:,1),y(:,2));

function dydt = eom(t,y)
      x = y(1);
      mu = -h*beta/L*cos(3*pi*x/(2*L))^2
                 -h*(1-beta*x/L)*(3*pi/(2*L))*sin(3*pi*x/L);
      k = (3*pi*h*beta/L^2)*sin(3*pi*x/L)
```

```
              - 4.5*h*pi^2/L^2*(1-beta*x/L)*cos(3*pi*x/L);
      V2 = y(3)^2+y(4)^2;

      M = eye(5,5);
      M(3,5) = mu/(m*sqrt(1+mu*mu));
      M(4,5) = -1/(m*sqrt(1+mu*mu));
      M(5,3) = mu;
      M(5,4) = -1.;
      M(5,5) = 0;
      f(1) = y(3);
      f(2) = y(4);
      f(3) = -c*V2/sqrt(1+mu^2);
      f(4) = -c*V2*mu/sqrt(1+mu^2)-g;
      f(5) = -k*y(3)^2;
      f = transpose(f);
      sol = M\f;
      dydt = sol(1:4);
end
function [eventvalue,stopthecalc,eventdirection] = events(t,y)
     % Function to stop simulation when x=L
     eventvalue = [L-y(1)];
     stopthecalc = [1];
     eventdirection = [-1];

 end
end
```

Here's the solution with the default MATLAB tolerance (i.e. without the 'options')

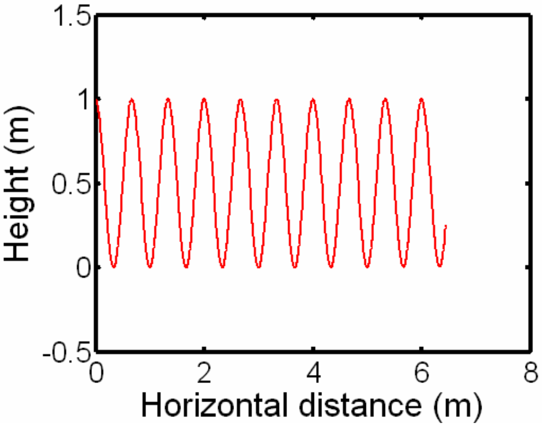
The solution is clearly wrong, since $y = \cos^2 \frac{3\pi x}{2}$ should never be negative.



[3 POINTS]

i. Fix the error by increasing the accuracy of the MATLAB computation.

Here's what happens with higher accuracy.



[1 POINT]

j. The ride is supposed to end when $x=L$ – add an 'event' function to your MATLAB code that will stop the computation when the car reaches $x=L$. (There is no need to submit a solution to this problem)

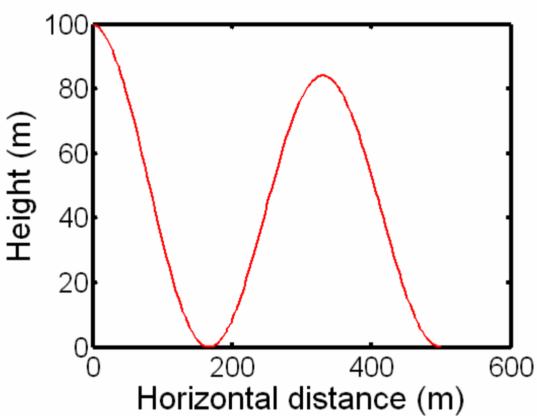
[0 POINTS]

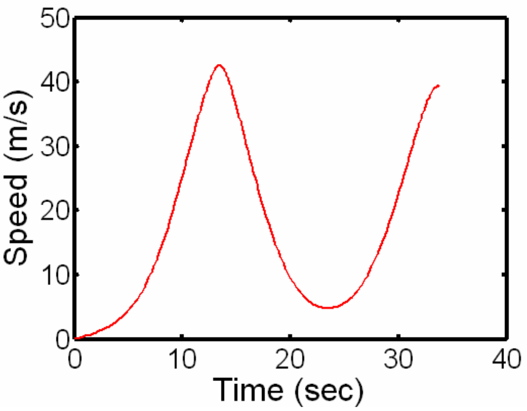
k. Finally, run your simulation with the following parameters:

- Height h = 100m; length L=500m
- Car mass 500kg
- Car frontal area $4m^2$
- Drag coefficient 0.1
- Air density $1.02\,kg\;m^{-3}$

(you will also need to increase the simulation time) and use the simulations to estimate the value of β required for the car to get over the peak at $x=3L/4$. (you can do this by trial and error)

A value of $\beta = 0.24$ seems to work well. Here are the plots of the ride and the speed as a function of time for this value.





The ride lasts for 33.8 sec (you can make it last longer by changing the value of β, as the Actress said to the Bishop…)

(Any reasonable choice of β should get credit)

[3 POINTS]

TOTAL 20 POINTS