



School of Engineering
Brown University

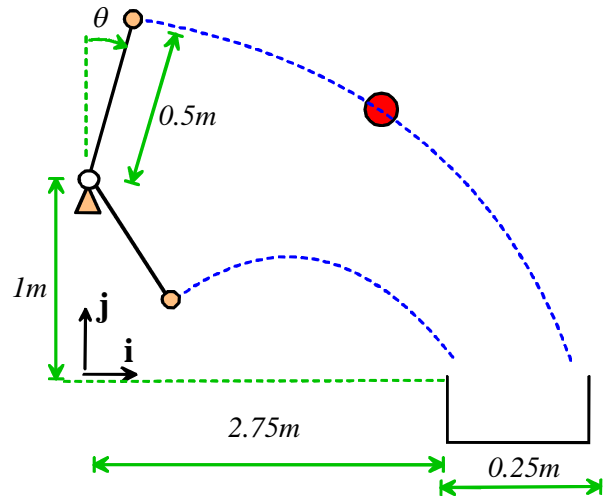
EN40: Dynamics and Vibrations

Homework 3: Solving equations of motion for particles Solutions

MAX SCORE 57+24 EXTRA CREDIT

1. Suppose that you wish to throw a piece of garbage into a garbage can. Is it better to use an underarm or overarm throw? A detailed analysis of this important question has been published by Professor Mahadevan and his group at Harvard, (who is evidently putting his MacArthur fellowship to good use). You will work through a simplified version of their calculations. To keep things simple, we will make the following assumptions:

- You throw by rotating your arm at the elbow, in a seated position. Your elbow has the same height for both under- and over-arm throws
- You throw the projectile with a speed of 5 m/s, for both over- and under-arm throws.
- Relevant dimensions are shown in the figure.



Your garbage will go into the basket only if you release the projectile at the right point. Specifically, there will be some range of release angles $\theta_1 < \theta < \theta_2$ for which the garbage will fall within the can. If the range is large, the shot is easy; if the range is small, the target is hard to hit.

1.1 Start by analyzing an overarm throw. Assume that the projectile is released with your arm at angle θ to the vertical. Find a formula for the position vector of the projectile at time t after release, using the coordinate system shown in the figure.

The trajectory formulas give $\mathbf{r} = 0.5 \sin \theta \mathbf{i} + (1 + 0.5 \cos \theta) \mathbf{j} + (5 \cos \theta \mathbf{i} - 5 \sin \theta \mathbf{j})t - 9.81t^2 / 2 \mathbf{j}$

[2 POINTS]

1.2 Hence, calculate the time that the projectile reaches height $h=0$.

$$\text{When } h=0, \text{ we have } (1 + 0.5 \cos \theta) - 5 \sin \theta t - 9.81t^2 / 2 = 0$$

$$\text{Solve for } t = \frac{-5 \sin \theta + \sqrt{(5 \sin \theta)^2 + 2 \times 9.81(1 + 0.5 \cos \theta)}}{9.81}$$

[1 POINT]

1.3 Derive an equation for the release angle that will just hit the outside of the garbage can (don't try to solve the equation by hand). Using MAPLE, find two possible release angles that satisfy the equation. (if you would like to stop MAPLE giving complex valued solutions, you can type with(RealDomain) at the start of your MAPLE file.

The angle is computed by setting $x=3$ at the instant when $y=h$, giving

$$3 = 0.5 \sin \theta + 5 \cos \theta \frac{-5 \sin \theta + \sqrt{(5 \sin \theta)^2 + 2 \times 9.81(1 + 0.5 \cos \theta)}}{9.81}$$

```
[> with(RealDomain) :
> 3=0.5*sin(q)+5*cos(q)*(-5*sin(q)+sqrt((5*sin(q))^2 +
2*9.81*(1+0.5*cos(q))))/9.81:
> evalf(180*solve(%,q)/Pi); # Angle in degrees
-6.828407800, -49.15139072
```

[3 POINTS]

1.4 Derive an equation for the release angle that will just hit the inside of the garbage can, and once again, use MAPLE to find the two possible release angles satisfying the equation.

The only change necessary is to switch 3 to 2.75 in the preceding equation, so that

```
[> with(RealDomain) :
> 2.75=0.5*sin(q)+5*cos(q)*(-5*sin(q)+sqrt((5*sin(q))^2 +
2*9.81*(1+0.5*cos(q))))/9.81:
> evalf(180*solve(%,q)/Pi); # Angle in degrees
-53.88193924, 0.4190988467
```

There are two possible ranges of angles that would hit the target

[2 POINTS]

1.5 Hence, calculate $\Delta\theta = \theta_2 - \theta_1$ for the overarm throw.

$$-53 < \theta < -49$$

$$-6.8 < \theta < 0.42$$

The second of these is preferable (it is larger, and would be much more comfortable than bending your arm way back). The range is 7.22 degrees.

[1 POINT]

1.6 Repeat 1.1-1.5 for an underarm throw.

The underarm case can be calculated immediately by simply changing the sign of the throwing speed. Here are the predicted angles, for the two cases

```
[> with(RealDomain) :
> 3=0.5*sin(q)-5*cos(q)*(5*sin(q)+sqrt((5*sin(q))^2 +
2*9.81*(1+0.5*cos(q))))/9.81:
> evalf(180*solve(%,q)/Pi); # Angle in degrees
154.7376193, 119.7899411
> with(RealDomain) :
> 2.75=0.5*sin(q)-5*cos(q)*(5*sin(q)+sqrt((5*sin(q))^2 +
2*9.81*(1+0.5*cos(q))))/9.81:
> evalf(180*solve(%,q)/Pi); # Angle in degrees
159.7830894, 115.4407286
```

Now we have

$$154.74 < \theta < 159.78$$

$$115.44 < \theta < 119.790$$

The first of these has the bigger range – just over 5 degrees

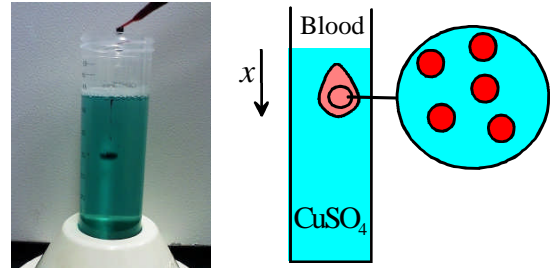
[5 POINTS]

Which throwing style is preferable?

Overarm is better – but not by very much, for the particular choice of parameters used here.

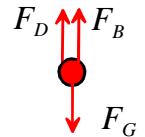
[1 POINT]

2. The figure illustrates a simple screening test for [anemia](#) (known as the CuSO4 test – see, e.g. Rudmann “[Handbook of blood banking and transfusion medicine](#)”). The technician drops a blood specimen into a CuSO4 solution with carefully controlled concentration, and observes the drop for 15sec. If the drop floats, or sinks less than a critical distance, the test indicates that the patient may have anemia. The goal of this problem is to determine the required concentration of the solution. Assume that



- A blood cell can be idealized as a sphere with radius R and mass density ρ
- The CuSO4 solution has mass density ρ_s and viscosity η
- The drag force on a sphere radius R moving through a viscous fluid with speed v is $6\pi\eta Rv$

2.1 Draw a free body diagram showing the forces acting on one of the blood cells within the blood droplet.



The drag, buoyancy and gravity forces are

$$F_D = 6\pi\eta R dx / dt \quad F_B = \rho_s 4\pi R^3 g / 3 \quad F_G = \rho 4\pi R^3 g / 3$$

[2 POINTS]

2.2 Write down Newton’s law of motion relating the depth x of the blood cell to relevant parameters. Show that x satisfies the differential equation

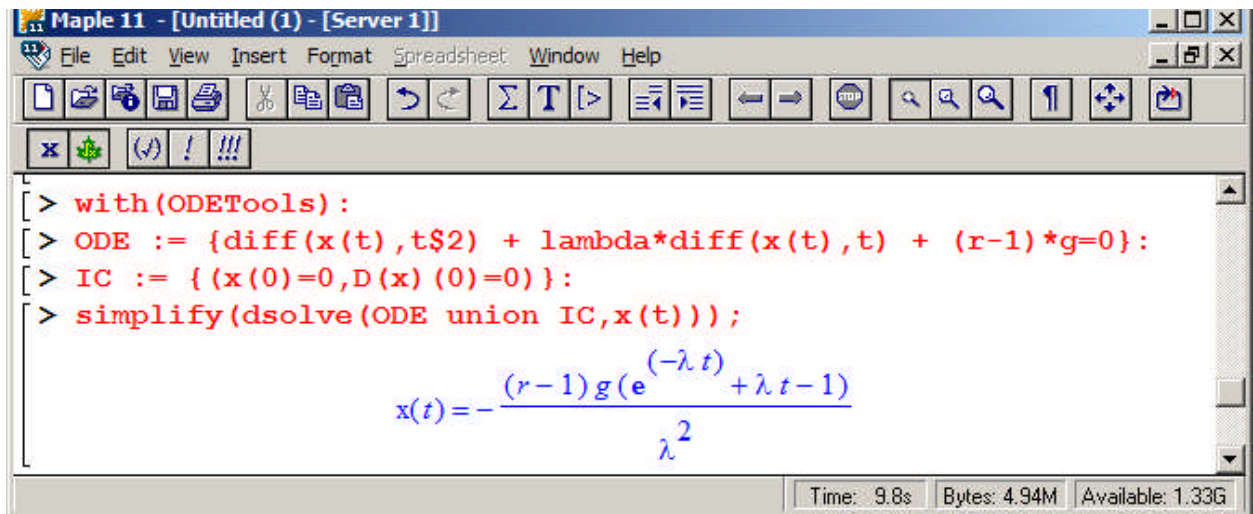
$$\mathbf{F=ma} \text{ in the vertical direction: } \frac{4\pi R^3 \rho}{3} \frac{d^2 x}{dt^2} = -6\pi\eta R \frac{dx}{dt} + \frac{4\pi R^3}{3} (\rho - \rho_s) g$$

Rearranging this equation gives the ODE stated.

[2 POINTS]

2.3 Assume that a representative blood cell starts at $x=0$ and has zero speed at time $t=0$. Hence, calculate expressions for the speed, and depth, of the blood cell in terms of time and other relevant parameters. Use the MAPLE ‘dsolve’ function to solve the differential equation.

Here’s the solution. We have set $\rho_s / \rho = r \quad \lambda = 9\eta / (2\rho R^2)$ to simplify the solution



```

> with(ODETools):
> ODE := {diff(x(t),t$2) + lambda*diff(x(t),t) + (r-1)*g=0}:
> IC := {(x(0)=0,D(x)(0)=0)}:
> simplify(dsolve(ODE union IC,x(t)));

```

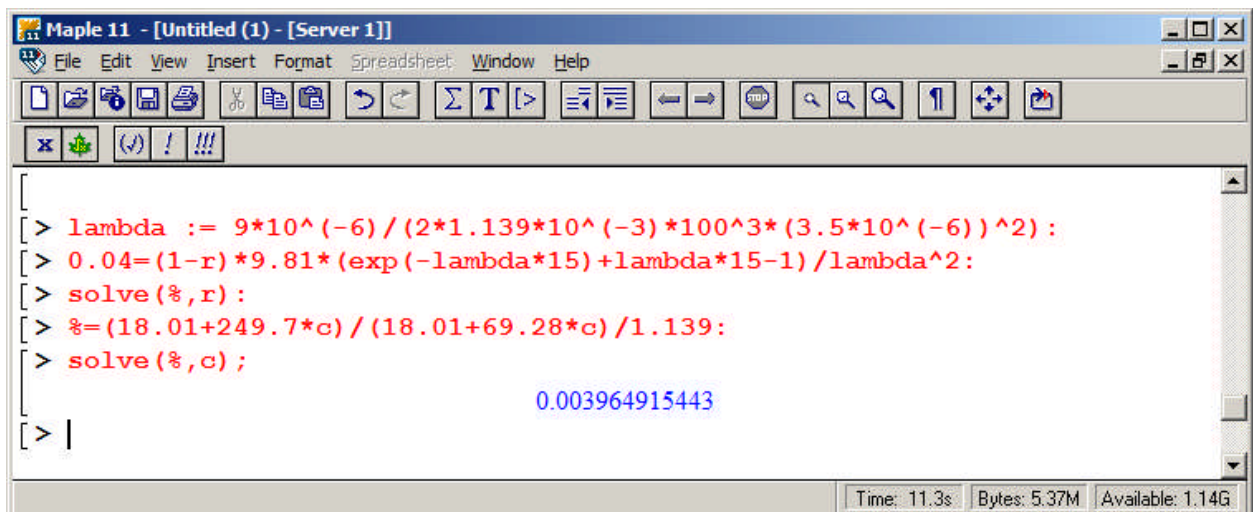
$$x(t) = -\frac{(r-1)g(e^{-\lambda t} + \lambda t - 1)}{\lambda^2}$$

Time: 9.8s Bytes: 4.94M Available: 1.33G

[3 POINTS]

2.4 Use the following values for parameters: mass density of normal blood cells $\rho = 1.139 \text{ gram cm}^{-3}$, mass density of CuSO4 solution $\rho_s = (18.01 + 249.7c) / (18.01 + 69.28c) \text{ gram / cm}^3$, where c is the molar concentration of CuSO4 in the solution (the number of mols of CuSO4 pentahydrate divided by the number of mols of water), viscosity of water $\eta = 10^{-6} \text{ Nsm}^{-2}$, radius of a blood cell $R = 3.5 \mu\text{m}$. Determine the concentration c that will result in blood sinking by 4cm during the 15 second test period.

We just need to substitute numbers into the formula for x and solve for c . Here's a MAPLE solution



```

> lambda := 9*10^(-6) / (2*1.139*10^(-3)*100^3*(3.5*10^(-6))^2):
> 0.04 = (1-r)*9.81*(exp(-lambda*15) + lambda*15 - 1) / lambda^2:
> solve(%, r):
> % = (18.01 + 249.7*c) / (18.01 + 69.28*c) / 1.139:
> solve(%, c):

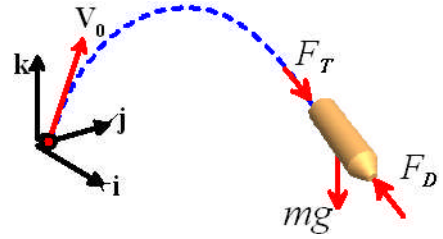
```

0.003964915443

Time: 11.3s Bytes: 5.37M Available: 1.14G

[3 POINTS]

3. The figure shows a rocket, which may be idealized as a particle with mass m , and which travels close to the earth's surface. It is launched from the origin with initial velocity vector $\mathbf{V}_0 = V_x\mathbf{i} + V_y\mathbf{j} + V_z\mathbf{k}$. The projectile is subjected to the force of gravity (acting in the negative \mathbf{k} direction), a drag force



$$\mathbf{F}_D = -\frac{1}{2}\rho C_D A V \mathbf{v}$$

where $V = \sqrt{v_x^2 + v_y^2 + v_z^2}$ is the magnitude of the rocket's velocity, $\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ is the projectile velocity, C_L is the drag coefficient, and ρ is the air density. In addition, a thrust force $\mathbf{F}_T = F_0\mathbf{v}/V$ is exerted on the rocket by its motor (the thrust acts in the direction of motion of the rocket).

3.1 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]

3.2 Write down the vector equation of motion for the particle ($\mathbf{F}=\mathbf{ma}$).

$$\mathbf{F} = \mathbf{ma} \Rightarrow \left(F_0 - \frac{1}{2}\rho A V^2 \right) \frac{v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}}{V} - mg\mathbf{k} = \frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j} + \frac{d^2z}{dt^2}\mathbf{k}$$

[2 POINTS]

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

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -c_1 V v_x + c_2 v_x / V \\ -c_1 V v_y + c_2 v_y / V \\ -g - c_1 V v_z + c_2 v_z / V \end{bmatrix}$$

and give formulas for the constants c_1 and c_2 in terms of relevant parameters.

As usual we introduce (v_x, v_y, v_z) as additional unknowns. Using the definitions of the velocity components, and writing $\mathbf{F}=\mathbf{ma}$ in terms of (x,y,z) and these variables, we have

$$v_x = \frac{dx}{dt} \quad v_y = \frac{dy}{dt} \quad v_z = \frac{dz}{dt}$$

$$\frac{dv_x}{dt} = \frac{F_0}{m} \frac{v_x}{V} - \frac{1}{2m}\rho A V v_x \quad \frac{dv_y}{dt} = \frac{F_0}{m} \frac{v_y}{V} - \frac{1}{2m}\rho A V v_y \quad \frac{dv_z}{dt} = -g + \frac{F_0}{m} \frac{v_z}{V} - \frac{1}{2m}\rho A V v_z$$

If we define

$$c_1 = \frac{\rho A}{2m} \quad c_2 = \frac{F_0}{m}$$

and write the differential equations in vector form, we get the result stated.

[2 POINTS]

3.4 Modify the MATLAB code discussed in class (or the online notes, or the MATLAB tutorial, or last year's HW3) to calculate and plot the trajectory. Add an 'Event' function to your code that will stop the calculation when the rocket reaches the ground (THERE IS NO NEED TO SUBMIT A SOLUTION TO THIS PROBLEM).

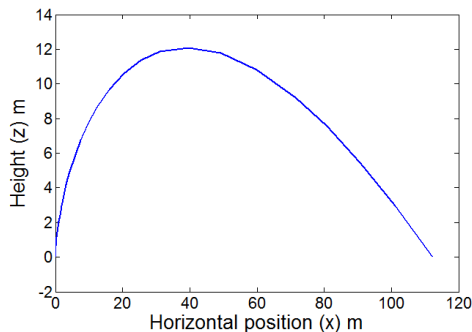
```
function rocket
    A = 0.015;      % Projected area of rocket, m^2
    rho = 1.02;    % Air density, kg/m^3
    CD = 0.1;      % Drag coeft
    g = 9.81;      % Gravitational accel
    m = 1;         % Mass
    F0 = 20;       % Motor thrust, N
    c1 = 0.5*rho*CD*A/m;
    c2 = F0/m;
    theta = 0.001*pi/180; % Angle of launch
    w0 = [0,0,0,0.0001*sin(theta),0.,0.0001*cos(theta)];

    options = odeset('Events',@event);
    [times,sols] = ode45(@eq_of_mot,[0,200],w0,options);
    times(end)
    sols(end,1)
    plot(sols(:,1),sols(:,3));

function dwdt = eq_of_mot(t,w)
    x=w(1);y=w(2);z=w(3);vx=w(4);vy=w(5);vz=w(6);
    V=sqrt(vx^2+vy^2+vz^2);
    dwdt = [vx;vy;vz;...
            -c1*V*vx+c2*vx/V;...
            -c1*V*vy+c2*vy/V;...
            -g-c1*V*vz+c2*vz/V];
end
function [ev,stop,dir] = event(t,w)
    ev = w(3); %w(3) is the height
    stop = 1;
    dir = -1;
end
end
```

[10 POINTS]

3.5 Finally, run simulations with parameters representing a (large) model rocket ($m=1\text{kg}$, $C_D=0.1$, $A=0.015\text{m}^2$, air density 1.02 kg/m^3 , $F_0=20\text{N}$) launched in the (x - z) plane at 0.0001 m/s at an angle of 0.001 degrees to the vertical. Plot the trajectory of the missile in the (x , z) plane, and determine the time of flight, and the total horizontal distance traveled.

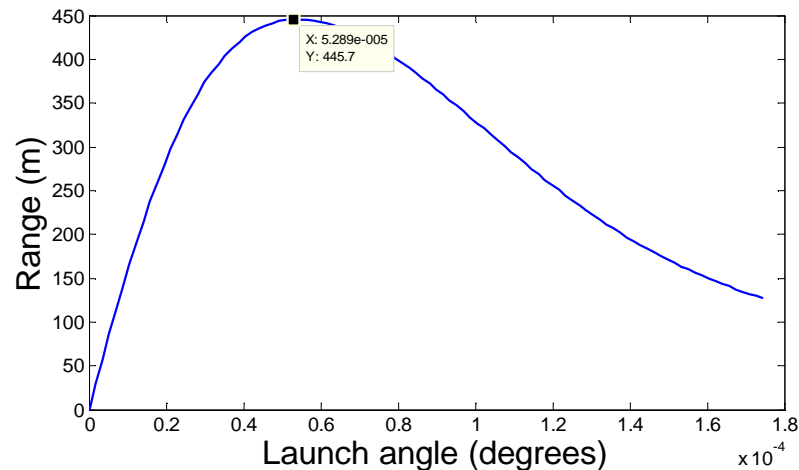


With ode45 and default tolerance the time of flight is 3.98 sec and the distance traveled is 112.16 m – but this is not very accurate so anything close to these values is OK.

[4 POINTS]

3.6 **Optional – for extra credit.** Suppose the rocket motor has a 5sec burn (i.e. the thrust is zero after 5sec). With the same initial velocity, find the initial launch angle that maximizes the horizontal range of the rocket, and find the corresponding range. **NOTE** – ode45 is not accurate enough for this problem – you should use ode113 instead. Also, set the 'Reltol' parameter to 10^{-10} or better.

The simplest way to do this is just to plot a graph of range as a function of angle and read off the maximum.



The max range is 446m and the angle is $5.3 \cdot 10^{-5}$ degrees. A MATLAB code is listed below.

[5 POINTS]

```
function rocket
    A = 0.015;      % Projected area of rocket, m^2
    rho = 1.02;    % Air density, kg/m^3
    CD = 0.1;      % Drag coeft
    g = 9.81;      % Gravitational accel
    m = 1;         % Mass
    F0 = 20;       % Motor thrust, N
    c1 = 0.5*rho*CD*A/m;
    c2 = F0/m;
    for i = 1:100
        theta(i) = 0.01*(i-1)*pi/(180*99.);
        range(i) = compute_range(theta(i));
    end
    plot(theta,range)
    function range = compute_range(angle)
        w0 = [0,0,0,0.0001*sin(angle),0.,0.0001*cos(angle)];
        options = odeset('Events',@event,'RelTol',1.e-10);
        [times,sols] = ode113(@eq_of_mot1,[0,5],w0,options); % First 5 secs
        w0 = sols(end,:); % Solution at end give initial conditions
        range = sols(end,1);
        if (sols(end,3)>0) % continue with no thrust if not yet on the ground
            [times,sols] = ode113(@eq_of_mot2,[5,200],w0,options);
```

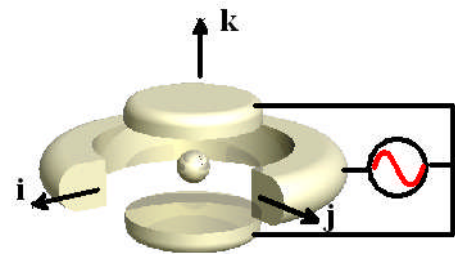
```

        range = sols(end,1);
    end
end
function dwdt = eq_of_mot1(t,w)% EOM with thrust
x=w(1);y=w(2);z=w(3);vx=w(4);vy=w(5);vz=w(6);
V=sqrt(vx^2+vy^2+vz^2);
dwdt = [vx;vy;vz;...
        -c1*V*vx+c2*vx/V;...
        -c1*V*vy+c2*vy/V;...
        -g-c1*V*vz+c2*vz/V];
end
function dwdt = eq_of_mot2(t,w)% EOM with no thrust
x=w(1);y=w(2);z=w(3);vx=w(4);vy=w(5);vz=w(6);
V=sqrt(vx^2+vy^2+vz^2);
dwdt = [vx;vy;vz;...
        -c1*V*vx;...
        -c1*V*vy;...
        -g-c1*V*vz];
end
function [ev,stop,dir] = event(t,w)
ev = w(3); %w(3) is the height
stop = 1;
dir = -1;
end
end
end

```

This is quite a tricky problem, and I am not convinced that the MATLAB solution is correct. The solution appears to converge as Reltol is reduced, but reltol= 10^{-10} is an extremely small value, and may be pushing the ODE solver beyond the limits of floating point precision. Using reltol= 10^{-4} gives a similar max range, but the optimal angle is a factor of 10 smaller. Ode45 gives complete garbage. For design purposes, the solution is really telling you that this approach to controlling the rocket's trajectory is not going to work – there is no way you would be able to design a system that sets the launch angle sufficiently accurately, and tiny fluctuations in air flow etc early on in the launch would give huge variations in trajectory. A guidance system that controls the rocket path in flight would be essential.

4. A 'Paul Trap' is one of several devices that use specially shaped electrostatic and electromagnetic fields to trap charged particles (or 'ions') (see, e.g. [Major et al \(2005\)](#) for more information). It has a number of applications: for example, as a mass spectrometer (used in chemical and biochemical analysis); or as a component in experimental quantum computers. The ions inside a particle trap are continuously moving, and follow very complicated orbits. The goal of this problem is to analyze the motion of a single charged particle inside a Paul trap.



4.1 The motion of the ion will be described by the components of its position vector (x,y,z) . Write down the velocity and acceleration of the ion in terms of these variables.

$$\mathbf{v} = \frac{dx}{dt} \mathbf{i} + \frac{dy}{dt} \mathbf{j} + \frac{dz}{dt} \mathbf{k} \quad \mathbf{a} = \frac{d^2x}{dt^2} \mathbf{i} + \frac{d^2y}{dt^2} \mathbf{j} + \frac{d^2z}{dt^2} \mathbf{k}$$

[1 POINT]

4.2 The ion has mass m , and is subjected to a force

$$\mathbf{F} = Q\mathbf{E}$$

where Q is its charge, and

$$\mathbf{E} = \frac{E_0(1 + \beta \cos \omega t)}{d} (x\mathbf{i} + y\mathbf{j} - 2z\mathbf{k})$$

is a time dependent electric field vector. Here, E_0, d, β are constants that specify the magnitude and geometry of the electric and magnetic fields. Use Newton's law to show that the components of the position vector of the ion satisfies the following equations of motion

$$\frac{d^2x}{dt^2} = \Omega^2(1 + \beta \cos(\omega t))x \quad \frac{d^2y}{dt^2} = \Omega^2(1 + \beta \cos(\omega t))y \quad \frac{d^2z}{dt^2} = -2\Omega^2(1 + \beta \cos(\omega t))z$$

where

$$\Omega = \sqrt{\frac{QE_0}{md}}$$

is a parameter (which has units of frequency).

Newton's law $\mathbf{F} = m\mathbf{a}$ gives

$$\frac{QE_0(1 + \beta \cos \omega t)}{d} (x\mathbf{i} + y\mathbf{j} - 2z\mathbf{k}) = m \left(\frac{d^2x}{dt^2} \mathbf{i} + \frac{d^2y}{dt^2} \mathbf{j} + \frac{d^2z}{dt^2} \mathbf{k} \right)$$

The $\mathbf{i}, \mathbf{j}, \mathbf{k}$ components of this expression reduce to the expression given.

[2 POINTS]

4.3 Re-write the equations of motion as 6 first-order differential equations that can be integrated using MATLAB.

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \Omega^2(1 + \beta \cos(\omega t))x \\ \Omega^2(1 + \beta \cos(\omega t))y \\ -2\Omega^2(1 + \beta \cos(\omega t))z \end{bmatrix}$$

Introducing v_x, v_y, v_z as new unknowns and substituting into $\mathbf{F} = m\mathbf{a}$ in the usual way gives this result

[1 POINT]

4.4 Write a MATLAB script that will solve the equations of motion to determine x, y, z, v_x, v_y, v_z as a function of time. You need not submit a solution to this problem.

A MATLAB script is listed below

```
function paul_trap
close all
Omega = 1;
beta = 0;
omega = 2;
```

```

w0 = [0.1,0,0.1,0,0.100,0]; % Initial condition [x,y,z,vx,vy,vz]
[t,s] = ode45(@eq_of_mot,[0,60],w0);
figure1 = figure;
axes('Parent',figure1,'FontSize',14);
plot3(s(:,1),s(:,2),s(:,3));
grid('on');
xlabel({'x'},'FontSize',14);
ylabel({'y'},'FontSize',14);
zlabel({'z'},'FontSize',14);

beta = 30;
omega = 12;
[t,s] = ode45(@eq_of_mot,[0,60],w0);
figure2 = figure;
axes('Parent',figure2,'FontSize',14);
plot3(s(:,1),s(:,2),s(:,3));
grid('on');
xlabel({'x'},'FontSize',14);
ylabel({'y'},'FontSize',14);
zlabel({'z'},'FontSize',14);

beta = 30;
omega = 11;
[t,s] = ode45(@eq_of_mot,[0,60],w0);
figure3 = figure;
axes('Parent',figure3,'FontSize',14);
plot3(s(:,1),s(:,2),s(:,3));
grid('on');
xlabel({'x'},'FontSize',14);
ylabel({'y'},'FontSize',14);
zlabel({'z'},'FontSize',14);

beta = 30;
omega = 21;
[t,s] = ode45(@eq_of_mot,[0,60],w0);
figure4 = figure;
axes('Parent',figure4,'FontSize',14);
plot3(s(:,1),s(:,2),s(:,3));
grid('on');
xlabel({'x'},'FontSize',14);
ylabel({'y'},'FontSize',14);
zlabel({'z'},'FontSize',14);

function dwdt = eq_of_mot(t,w)

    x=w(1);y=w(2);z=w(3);vx=w(4);vy=w(5);vz=w(6);

    c = Omega^2*(1+beta*cos(omega*t));
    dwdt = [vx;vy;vz;c*x;c*y;-2*c*z];

end
end

```

[0 POINTS]

4.5 An ion will be trapped if the frequency of the electric field ω is related to Ω correctly, and if the amplitude of the oscillation β lies in the correct range. Demonstrate this behavior by running computations with the following parameters (the units are arbitrary – in actual designs the frequencies are very high; typically radio frequencies).

(a) $x = z = 0.1$ $y = 0$ $v_x = v_z = 0$ $v_y = 0.1$ $\Omega = 1$ $\beta = 0$ $\omega = 0$

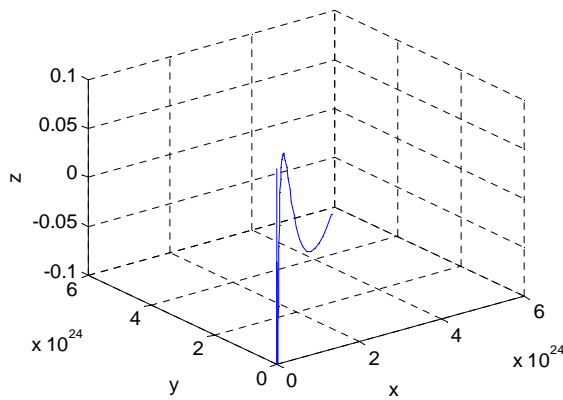
(b) $x = z = 0.1$ $y = 0$ $v_x = v_z = 0$ $v_y = 0.1$ $\Omega = 1$ $\beta = 30$ $\omega = 12$

(c) $x = z = 0.1$ $y = 0$ $v_x = v_z = 0$ $v_y = 0.1$ $\Omega = 1$ $\beta = 30$ $\omega = 11$

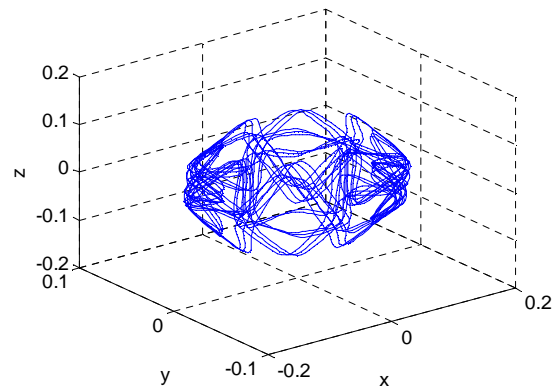
(d) $x = z = 0.1$ $y = 0$ $v_x = v_z = 0$ $v_y = 0.1$ $\Omega = 1$ $\beta = 30$ $\omega = 22$

Run the tests for a time interval of 60 units. Hand in a graph showing the predicted trajectory for each case.

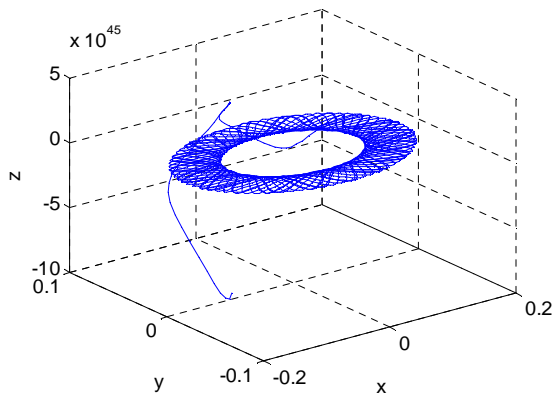
Graphs are shown below. Case (b) traps the particle – the particle escapes for all the other conditions.



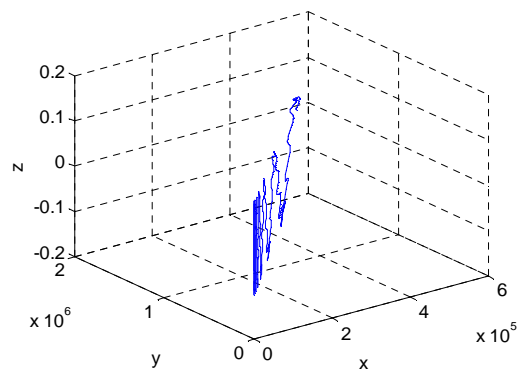
(a)



(b)



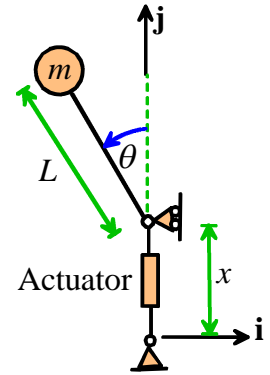
(c)



(d)

[5 POINTS]

5. The figure shows a schematic of the parametrically excited inverse pendulum discussed on the first day of class. The pivot point moves vertically with a displacement $x = X_0 \sin \Omega t$. The goal of this problem is to show that at an appropriate frequency Ω the inverted pendulum is stable.



5.1 Write down the position vector of the mass m . Hence, calculate an expression for the acceleration of the mass, in terms of Ω, X_0, L as well as θ and its time derivatives.

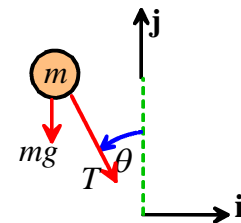
$$\mathbf{r} = -L \sin \theta \mathbf{i} + (X_0 \sin \Omega t + L \cos \theta) \mathbf{j}$$

$$\mathbf{v} = -L \cos \theta \frac{d\theta}{dt} \mathbf{i} + (X_0 \Omega \cos \Omega t - L \sin \theta \frac{d\theta}{dt}) \mathbf{j}$$

$$\mathbf{a} = \left[L \sin \theta \left(\frac{d\theta}{dt} \right)^2 - L \cos \theta \frac{d^2\theta}{dt^2} \right] \mathbf{i} + \left[-X_0 \Omega^2 \sin \Omega t - L \cos \theta \left(\frac{d\theta}{dt} \right)^2 - L \sin \theta \frac{d^2\theta}{dt^2} \right] \mathbf{j}$$

[3 POINTS]

5.2 Draw a free body diagram showing the forces acting on the mass m



[2 POINTS]

5.3 Write down Newton's law of motion of the mass, and hence show that the angle θ satisfies the equation

$$\frac{d^2\theta}{dt^2} - \frac{g}{L} \left(1 - \frac{\Omega^2 X_0}{g} \sin \Omega t \right) \sin \theta = 0$$

$\mathbf{F} = m\mathbf{a}$ gives

$$T \sin \theta \mathbf{i} - T \cos \theta \mathbf{j} - mg \mathbf{j}$$

$$= m \left[L \sin \theta \left(\frac{d\theta}{dt} \right)^2 - L \cos \theta \frac{d^2\theta}{dt^2} \right] \mathbf{i} + m \left[-X_0 \Omega^2 \sin \Omega t - L \cos \theta \left(\frac{d\theta}{dt} \right)^2 - L \sin \theta \frac{d^2\theta}{dt^2} \right] \mathbf{j}$$

The \mathbf{i} and \mathbf{j} components of this equation yield two equations. We can eliminate T by multiplying the \mathbf{i} component by $\sin \theta$ and the \mathbf{j} component by $\cos \theta$ and adding the two equations. This gives

$$-mg \sin \theta = m \left[-L \cos^2 \theta \frac{d^2\theta}{dt^2} \right] + m \left[-X_0 \Omega^2 \sin \Omega t \sin \theta - L \sin^2 \theta \frac{d^2\theta}{dt^2} \right]$$

This equation can be rearranged into the form required.

[2 POINTS]

5.4 Arrange the equation of motion for θ into a vector form that MATLAB can solve.

As always we introduce $\omega = \frac{d\theta}{dt}$ as an additional unknown. Then, in vector form we can write

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \omega \\ \frac{g}{L} \left(1 - \frac{\Omega^2 X_0}{g} \sin \Omega t\right) \sin \theta \end{bmatrix}$$

[1 POINT]

5.5 Write a MATLAB program to calculate and plot the angle θ as a function of time. Show the angle in degrees. Use a relative tolerance of 0.00001 in the ODE solver (You don't need to submit a solution to this problem)

```
function inverted_pendulum

    close all
    L = 0.25;
    g=9.81;
    X0=0.015;
    Omega = pi*60;
    w0 = [5*pi/180,0];
    options = odeset('RelTol',0.00001);
    [times,sols] = ode45(@eom,[0,30],w0,options);
    plot(times,sols(:,1)*180/pi)

    Omega = pi*50;
    [times,sols] = ode45(@eom,[0,30],w0,options);
    figure
    plot(times,sols(:,1)*180/pi)

    L = 0.5;
    Omega = pi*60;
    [times,sols] = ode45(@eom,[0,30],w0,options);
    figure
    plot(times,sols(:,1)*180/pi)

    L = 0.5;
    Omega = pi*73;
    [times,sols] = ode45(@eom,[0,30],w0,options);
    figure
    plot(times,sols(:,1)*180/pi)

    function dwdt = eom(t,w)
        theta=w(1);omega=w(2);
        dwdt = [omega;g*(1-Omega^2*X0*sin(Omega*t)/g)*sin(theta)/L];
    end
end
```

[0 POINTS]

5.6 Plot graphs of θ as a function of time for the following parameters. Run each simulation for 30 sec, and initial conditions $\theta = 5 \text{ deg}$ $\omega = 0$

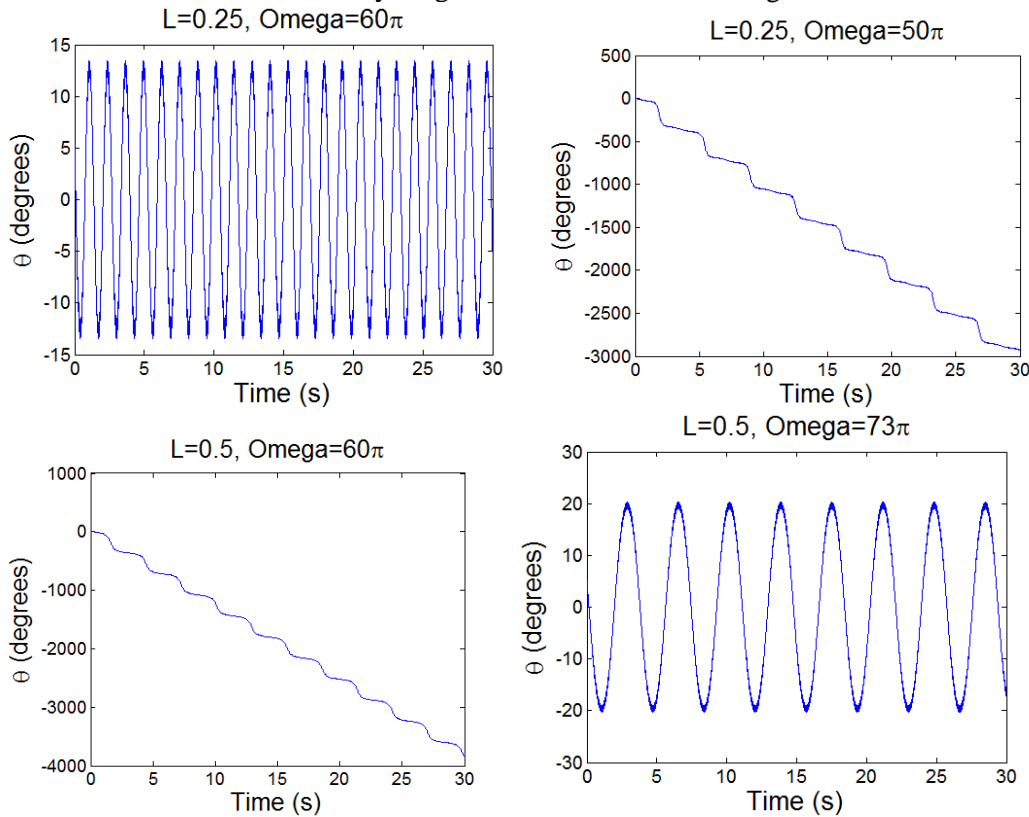
(a) $\Omega = 60\pi$ (30cycles / sec) $L = 0.25m$ $X_0 = 0.015m$

(b) $\Omega = 50\pi$ (25cycles / sec) $L = 0.25m$ $X_0 = 0.015m$

(c) $\Omega = 60\pi$ (30cycles / sec) $L = 0.5m$ $X_0 = 0.015m$

(d) $\Omega = 73\pi$ (36.5cycles / sec) $L = 0.5m$ $X_0 = 0.015m$

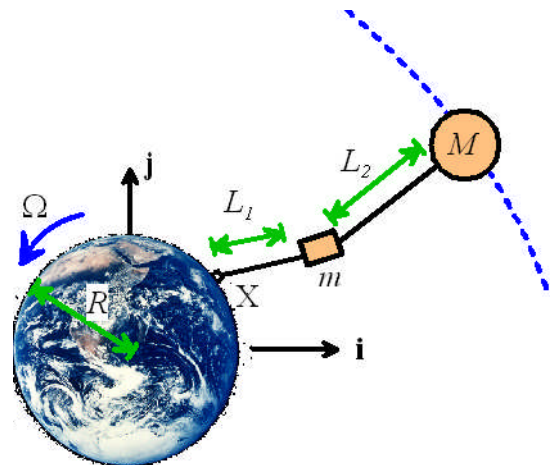
(a) and (c) turn out to be stable... See the figs. The graphs might look slightly different if people run with different tolerances, but anything that looks sensible should get credit.



[5 POINTS]

6. OPTIONAL – For extra credit: The figure shows a conceptual design for a ‘[space elevator](#)’ which (if materials scientists and electrical engineers are able to achieve several technological miracles) offers a very low-cost approach to launching payloads into orbit. It consists of a satellite with mass M , which lies in the equatorial plane and is in geostationary orbit. The satellite is tethered to the earth by a cable. A ‘crawler’ with mass m rides up and down this cable transporting freight and passengers from the earth’s surface to orbit. The goal of this problem is to analyze the motion of this system. For simplicity, we will

- Assume that the system remains in the equatorial plane, so that the position of the satellite and



crawler can be described by their coordinates in a fixed Cartesian basis $\mathbf{r}_1 = x_1\mathbf{i} + y_1\mathbf{j}$,
 $\mathbf{r}_2 = x_2\mathbf{i} + y_2\mathbf{j}$

- Neglect the mass of the cables
- Idealize the cables below and above the crawler as a combination of a spring and damper, which exert forces

$$T_1 = k_2(L_1 - a_1) + \lambda_1 \left(\frac{dL_1}{dt} - \frac{da_1}{dt} \right) \quad T_2 = k_2(L_2 - a_2) + \lambda_2 \left(\frac{dL_2}{dt} - \frac{da_2}{dt} \right)$$

to the objects attached to their ends, where L_1, L_2 are the stretched lengths of the cables, a_1, a_2 are the un-stretched cable lengths, and k_1, k_2 and λ_1, λ_2 are cable stiffnesses and damping coefficients.

6.1 The point X where the cable is attached to the surface of the earth moves as the earth rotates. Assume that at time $t=0$ this point has position vector $\mathbf{r}_X = R\mathbf{i}$, where R is the radius of the earth. Write down (a) the position vector of X as a function of time, and (b) the velocity vector \mathbf{v}_X of point X, in terms of R and the angular speed of the earth Ω

$$\mathbf{r}_X = R \cos \Omega t \mathbf{i} + R \sin \Omega t \mathbf{j} \quad \mathbf{v}_X = -R \sin \Omega t \mathbf{i} + R \cos \Omega t \mathbf{j}$$

[1 POINT]

6.2 Write down formulas for (a) the cable lengths; (b) unit vectors parallel to each cable; (c) unit vectors parallel to the gravitational forces acting on the crawler and satellite in terms of (x_1, y_1) , (x_2, y_2) . Also, find expressions for dL_1/dt and dL_2/dt in terms of (x_1, y_1) , (x_2, y_2) and their time derivatives.

$$L_1 = \sqrt{(x_1 - R \cos \Omega t)^2 + (y_1 - R \sin \Omega t)^2} \quad L_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\frac{dL_1}{dt} = \frac{(x_1 - R \cos \Omega t)(v_{x1} + R\Omega \sin \Omega t) + (y_1 - R \sin \Omega t)(v_{y1} - R\Omega \cos \Omega t)}{L_1}$$

$$\frac{dL_2}{dt} = \frac{(x_2 - x_1)(v_{x2} - v_{x1}) + (y_2 - y_1)(v_{y2} - v_{y1})}{L_2}$$

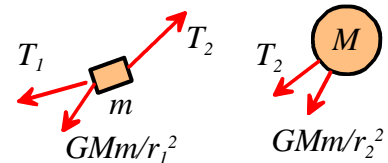
[3 POINTS]

6.3 Write down the acceleration vectors for the crawler and the satellite, in terms of derivatives of (x_1, y_1) , (x_2, y_2)

$$\mathbf{a}_1 = \frac{d^2 x_1}{dt^2} \mathbf{i} + \frac{d^2 y_1}{dt^2} \mathbf{j} \quad \mathbf{a}_2 = \frac{d^2 x_2}{dt^2} \mathbf{i} + \frac{d^2 y_2}{dt^2} \mathbf{j}$$

[1 POINT]

6.3 Draw two free body diagrams showing (a) the forces acting on the satellite, and (b) the forces acting on the crawler.



(the exact directions of the arrows aren't critical, of course)

[3 POINTS]

6.4 Show that the equations of motion for the crawler and satellite can be expressed as

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ v_{x1} \\ v_{y1} \\ v_{x2} \\ v_{y2} \end{bmatrix} = \begin{bmatrix} v_{x1} \\ v_{y1} \\ v_{x2} \\ v_{y2} \\ -\mu x_1 / r_1^3 + T_1(R \cos \Omega t - x_1) / mL_1 + T_2(x_2 - x_1) / mL_2 \\ -\mu y_1 / r_1^3 + T_1(R \sin \Omega t - y_1) / mL_1 + T_2(y_2 - y_1) / mL_2 \\ -\mu x_2 / r_2^3 + T_2(x_1 - x_2) / ML_2 \\ -\mu y_2 / r_2^3 + T_2(y_1 - y_2) / ML_2 \end{bmatrix}$$

where μ is the product of the earth's mass and the gravitational constant and $r_1 = \sqrt{x_1^2 + y_1^2}$ $r_2 = \sqrt{x_2^2 + y_2^2}$

Let $v_{x1}\mathbf{i} + v_{y1}\mathbf{j}$, $v_{x2}\mathbf{i} + v_{y2}\mathbf{j}$ denote the velocity of the car and the satellite, with respect to fixed axes. The position-velocity relation gives

$$v_{x1}\mathbf{i} + v_{y1}\mathbf{j} = \frac{d}{dt}(x_1\mathbf{i} + y_1\mathbf{j}) \quad v_{x2}\mathbf{i} + v_{y2}\mathbf{j} = \frac{d}{dt}(x_2\mathbf{i} + y_2\mathbf{j})$$

Newton's law $\mathbf{F} = m\mathbf{a}$ gives

$$m \frac{d}{dt}(v_{x1}\mathbf{i} + v_{y1}\mathbf{j}) = -\frac{\mu m}{r_1^2} \frac{\mathbf{r}_1}{r_1} + T_1 \frac{(\mathbf{r}_X - \mathbf{r}_1)}{L_1} + T_2 \frac{(\mathbf{r}_2 - \mathbf{r}_1)}{L_2}$$

$$M \frac{d}{dt}(v_{x2}\mathbf{i} + v_{y2}\mathbf{j}) = -\frac{\mu M}{r_2^2} \frac{\mathbf{r}_2}{r_2} + T_2 \frac{(\mathbf{r}_1 - \mathbf{r}_2)}{L_2}$$

The eight equations in the vector ODE are the \mathbf{i} and \mathbf{j} components of these two equations.

[2 POINTS]

6.5 Write a MATLAB script that will calculate the position and velocity of the satellite and crawler as a function of time. Use the following values for parameters in the problem

- The gravitational parameter $\mu = 3.9812 \times 10^5 \text{ km}^3 \text{ s}^{-1}$
- The earth's radius R is 6472km
- The earth's angular velocity is $\Omega = \pi / 43200 \text{ rad/s}$
- The cables have constant stiffness $k_1 = k_2 = 1 \text{ kN} / \text{km}$ and damping $\eta_1 = \eta_2 = 1 \text{ kNs} / \text{km}$
- The satellite has initial position and velocity $\mathbf{r}_0 = 42241.12\mathbf{i} \text{ km}$ $\mathbf{v}_0 = 3.0719\mathbf{j} \text{ km/s}$
- The crawler has initial position and velocity $\mathbf{r}_0 = 7015.1\mathbf{i} \text{ km}$ $\mathbf{v}_0 = 0.5102\mathbf{j} \text{ km/s}$
- As the crawler climbs the cable, the un-stretched lengths of the two cables vary with time. Take the un-stretched lengths (in km) to be

$$a_1 = \begin{cases} 637.8 + 5362.6(2\pi t / T - \sin 2\pi t / T) & t < T \\ 34332 & t > T \end{cases}$$

$$a_2 = 35863 - a_1$$

where T is the time of ascent. There is no need to submit a solution to this problem. Note that this is an intensive computation, – MATLAB will take a few minutes to complete the calculation.

Function space_elevator

```
close all
format long
mu = 3.986012e05; % Gravitational parameter km^3/s
spinearth = 2*pi/(24*3600); % Angular velocity of earth (rad/s)
radearth = 6378.1; % Earths radius (km)
msat = 5000000; % Mass of satellite (kg)
mcar = 500; % Mass of car (kg)
k1 = 1; % Stiffness of cable 1 (kN/km)
k2 = 1; % Stiffness of cable 2 (kN/km)
eta1 = 1; % Damping of cable 1 (kN s/km)
eta2 = 1; % Damping of cable 1 (kN s/km)

T = 5*24*3600; % Ascent time (sec)
% Initial conditions [xcar,ycar,vxcar, icar,xsat,ysat,vxsat,vysat]
w0 = [7015.1,0,42241.12,0,0,0.5102,0,3.0719];
options = odeset('RelTol',0.001);
[tsol,wsol] = ode45(@eom,[0,8*24*3600],w0,options);
plot(wsol(:,1),wsol(:,2),'Color',[0 0 1]);
hold on
plot(wsol(:,3),wsol(:,4),'Color',[1 0 0]);

% calculation of tensions
for i = 1:length(tsol)
    t = tsol(i);
    x1 = wsol(i,1); y1 = wsol(i,2); x2 = wsol(i,3); y2 = wsol(i,4);
    xt = radearth*cos(spinearth*t);
    yt = radearth*sin(spinearth*t);
    L1 = sqrt((x1-xt)^2 + (y1-yt)^2);
    L2 = sqrt((x1-x2)^2 + (y1-y2)^2);
    dL1dt = ((x1-xt)*(vx1-vxt) + (y1-yt)*(vy1-vyt))/L1;
    dL2dt = ((x2-x1)*(vx2-vx1) + (y2-y1)*(vy2-vy1))/L2;
    a1 = 637.8 + 5362.6*(2*pi*t/T-sin(2*pi*t/T));
    da1dt = 5362.6*(2*pi/T-2*pi*cos(2*pi*t/T)/T);
    if (t>T) a1= 34332; da1dt=0; end
    a2 = 35863-a1;
    da2dt = -da1dt;
    T1 = k1*(L1-a1)+eta1*(dL1dt-da1dt);
    T2 = k2*(L2-a2)+eta2*(dL2dt-da2dt);
    tt1(i) = T1;
    tt2(i) = T2;
end
figure
plot(tsol,tt1)
hold on
plot(tsol,tt2,'Color',[1 0 0])

function dwdt = eom(t,w)

    x1 = w(1); y1 = w(2); x2 = w(3); y2 = w(4);
    vx1 = w(5); vy1 = w(6); vx2 = w(7); vy2 = w(8);
    r1 = sqrt(x1^2 + y1^2);
    r2 = sqrt(x2^2 + y2^2);
```

```

xt = radearth*cos(spinearth*t);
yt = radearth*sin(spinearth*t);
vxt = -radearth*spinearth*sin(spinearth*t);
vyt = radearth*spinearth*cos(spinearth*t);
L1 = sqrt((x1-xt)^2 + (y1-yt)^2);
L2 = sqrt((x1-x2)^2 + (y1-y2)^2);
dL1dt = ((x1-xt)*(vx1-vxt) + (y1-yt)*(vy1-vyt))/L1;
dL2dt = ((x2-x1)*(vx2-vx1) + (y2-y1)*(vy2-vy1))/L2;
a1 = 637.8 + 5362.6*(2*pi*t/T-sin(2*pi*t/T));
dalddt = 5362.6*(2*pi/T-2*pi*cos(2*pi*t/T)/T);
if (t>T) a1= 34332; dalddt=0; end
a2 = 35863-a1;
da2dt = -dalddt;
T1 = k1*(L1-a1)+eta1*(dL1dt-dalddt);
T2 = k2*(L2-a2)+eta2*(dL2dt-da2dt);
dwdt = [vx1;vy1;vx2;vy2;...
-mu*x1/r1^3+ (T2*(x2-x1)/L2 + T1*(xt-x1)/L1)/mcar ;...
-mu*y1/r1^3+ (T2*(y2-y1)/L2 + T1*(yt-y1)/L1)/mcar ;...
-mu*x2/r2^3 + T2*(x1-x2)/L2/msat;...
-mu*y2/r2^3 + T2*(y1-y2)/L2/msat];

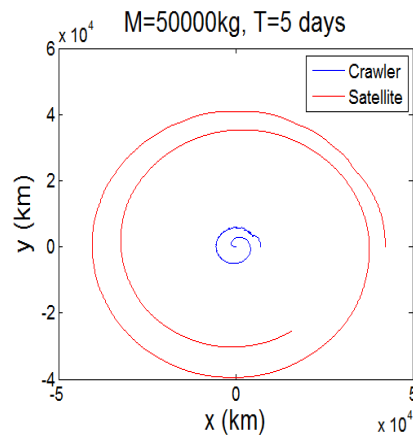
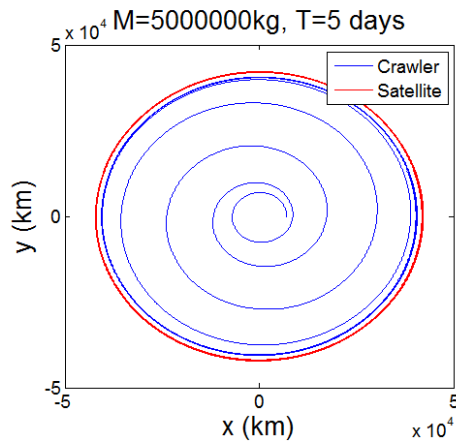
```

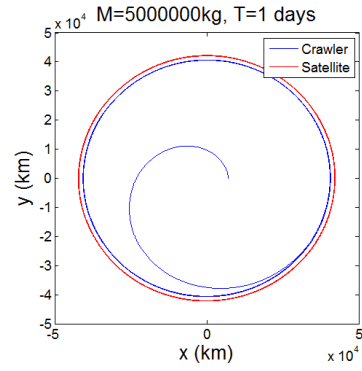
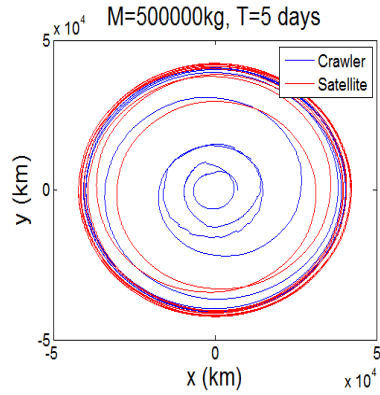
end
end

[0 POINTS]

6.6 Plot a graph showing the trajectory (x-v-y) of the crawler and satellite as the crawler ascends the cable. Plot both trajectories on the same graph, but show them in different colors. Try the following parameters

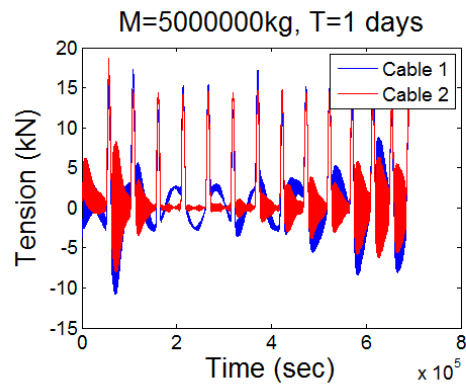
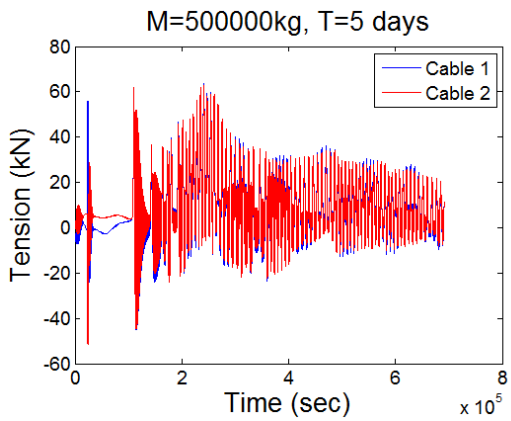
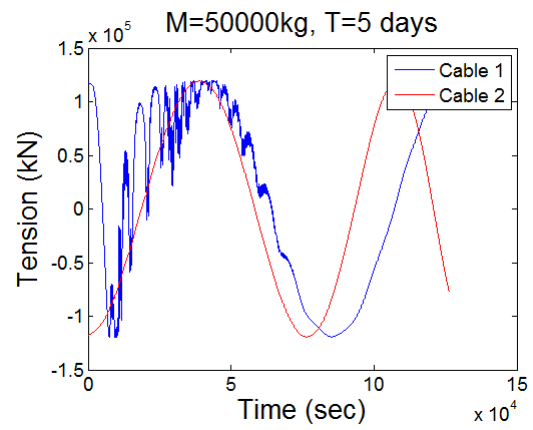
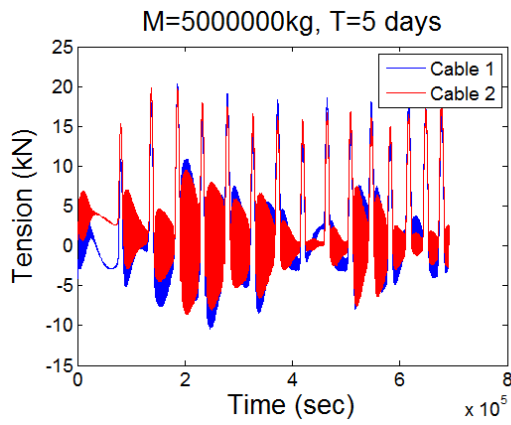
- $M = 5000000kg$ $m = 500kg$ $T = 5$ days
- $M = 50000kg$ $m = 500kg$ $T = 5$ days (this was a typo – it should have said 500000. The system crashes with a small satellite mass)
- $M = 5000000kg$ $m = 500kg$ $T = 1$ day





[5 POINTS]

Plot a graphs showing the tension in each of the cables as a function of time.



[5 POINTS]