



## EN40: Dynamics and Vibrations

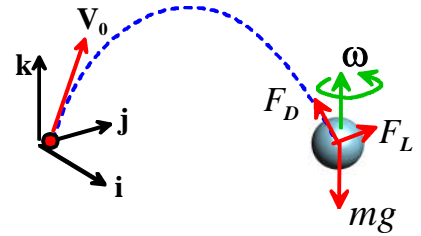
### Solutions to Homework 3: Solving equations of motion for particles MAX SCORE 30 POINTS + 20 POINTS FOR OPTIONAL PROB

Division of Engineering  
Brown University

1. A spherical projectile with diameter  $D$  and mass  $m$  is launched from the origin with initial velocity vector  $\mathbf{V}_0 = V_x \mathbf{i} + V_y \mathbf{j} + V_z \mathbf{k}$  and angular velocity (spin)  $\boldsymbol{\omega} = \omega_x \mathbf{i} + \omega_y \mathbf{j} + \omega_z \mathbf{k}$ . The projectile is subjected to the force of gravity (acting in the negative  $\mathbf{k}$  direction) together with lift and drag forces

$$\mathbf{F}_L = \rho C_L \frac{\pi D^2}{4} \boldsymbol{\omega} \times \mathbf{v} \quad \mathbf{F}_D = -\frac{1}{2} \rho C_D \frac{\pi D^2}{4} V \mathbf{v}$$

where  $V = \sqrt{v_x^2 + v_y^2 + v_z^2}$  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,  $C_L$  and  $C_D$  are the lift and drag coefficients, and  $\rho$  is the air density.



1.1 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. There is no need to use Newton's laws to do this – simply write down the definition of acceleration.

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

[1 POINT]

1.2 Write down the vector equation of motion for the particle ( $\mathbf{F} = m\mathbf{a}$ ).

Note that

$$\begin{aligned} \boldsymbol{\omega} \times \mathbf{v} &= (\omega_x \mathbf{i} + \omega_y \mathbf{j} + \omega_z \mathbf{k}) \times (v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}) \\ &= (\omega_y v_z - \omega_z v_y) \mathbf{i} + (\omega_z v_x - \omega_x v_z) \mathbf{j} + (\omega_x v_y - \omega_y v_x) \mathbf{k} \end{aligned}$$

Therefore, the equation of motion is

$$\begin{aligned} &\rho C_L \frac{\pi D^2}{4} [(\omega_y v_z - \omega_z v_y) \mathbf{i} + (\omega_z v_x - \omega_x v_z) \mathbf{j} + (\omega_x v_y - \omega_y v_x) \mathbf{k}] \\ &- \frac{1}{2} \rho C_D \frac{\pi D^2}{4} V (v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}) - mg \mathbf{k} = m \left( \frac{d^2 x}{dt^2} \mathbf{i} + \frac{d^2 y}{dt^2} \mathbf{j} + \frac{d^2 z}{dt^2} \mathbf{k} \right) \end{aligned}$$

[1 POINT]

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

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -c_1 V v_x + c_2 (\omega_y v_z - \omega_z v_y) \\ -c_1 V v_y + c_2 (\omega_z v_x - \omega_x v_z) \\ -g - c_1 V v_z + c_2 (\omega_x v_y - \omega_y v_x) \end{bmatrix}$$

and give formulas for the constants  $c_1$  and  $c_2$ .

The  $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$  components of the EOM give

$$\begin{aligned} -\frac{1}{2} \rho C_D \frac{\pi D^2}{4} V v_x + \rho C_L \frac{\pi D^2}{4} (\omega_y v_z - \omega_z v_y) &= m \frac{d^2 x}{dt^2} \\ -\frac{1}{2} \rho C_D \frac{\pi D^2}{4} V v_y + \rho C_L \frac{\pi D^2}{4} (\omega_z v_x - \omega_x v_z) &= m \frac{d^2 y}{dt^2} \\ -\frac{1}{2} \rho C_D \frac{\pi D^2}{4} V v_z + \rho C_L \frac{\pi D^2}{4} (\omega_x v_y - \omega_y v_x) - mg &= m \frac{d^2 z}{dt^2} \end{aligned}$$

If we divide through by  $m$  and define

$$c_1 = \frac{\pi \rho C_D D^2}{8 m} \quad c_2 = \frac{\pi \rho C_L D^2}{4 m}$$

these equations become

$$\begin{aligned} -c_1 V v_x + c_2 (\omega_y v_z - \omega_z v_y) &= \frac{d^2 x}{dt^2} \\ -c_1 V v_y + c_2 (\omega_z v_x - \omega_x v_z) &= \frac{d^2 y}{dt^2} \\ -g - c_1 V v_z + c_2 (\omega_x v_y - \omega_y v_x) &= \frac{d^2 z}{dt^2} \end{aligned}$$

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

$$\begin{aligned} \frac{dx}{dt} &= v_x & \frac{dy}{dt} &= v_y & \frac{dz}{dt} &= v_z \\ \frac{dv_x}{dt} &= -c_1 V v_x + c_2 (\omega_y v_z - \omega_z v_y) \\ \frac{dv_y}{dt} &= -c_1 V v_y + c_2 (\omega_z v_x - \omega_x v_z) \\ \frac{dv_z}{dt} &= -g - c_1 V v_z + c_2 (\omega_x v_y - \omega_y v_x) \end{aligned}$$

**[3 POINTS]**

1.4 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 (use the 'plot3' command). (THERE IS NO NEED TO SUBMIT A SOLUTION TO THIS PROBLEM). A basic script is shown below.

```
function trajectory_3d
rho = 1.02; % Air density
D = 0.0748; % Projectile diameter
Cd = 0.3; % Drag coefficient
Cl = 0.01; % Lift coefficient
m=0.142; % projectile mass
c1 = rho*Cd*pi*D^2/8/m;
c2 = rho*Cl*pi*D^2/4/m;
wx = 0;wy=0;wz=0; % Angular velocity components
V0 = [34.4*cos(pi/180),0,34.4*sin(pi/180)]; % Initial velocity
d = 18.4; % Distance traveled
time=30;

y0 = [0,0,0,V0]; % Initial conditions
options = odeset('Events',@events);
[t_vals,w_vals] = ode45(@projectile_equations,[0,time],y0,options);
end_position_1 = w_vals(length(t_vals),1:3)

figure1 = figure;
axes1 = axes('Parent',figure1,'FontSize',12);
view([18.5 22]);
grid('on');
hold('all');
plot3(w_vals(:,1),w_vals(:,2),w_vals(:,3),'LineWidth',2)

wz = 132; % Solution with spin
[t_vals,w_vals] = ode45(@projectile_equations,[0,time],y0,options);
end_position_2 = w_vals(length(t_vals),1:3)
deflection = end_position_2-end_position_1;
deflected_dist = sqrt(dot(deflection,deflection))

plot3(w_vals(:,1),w_vals(:,2),w_vals(:,3),'LineWidth',2,'Color',[1 0 0]);

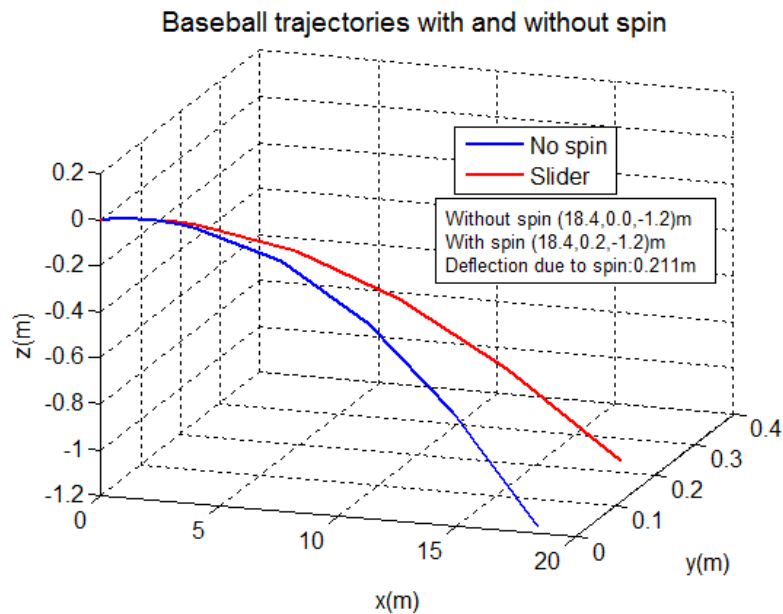
function dwdt = projectile_equations(t,w)
% The vector w contains [x,y,z,vx,vy,vz]
x = w(1); y=w(2); z=w(3); vx = w(4); vy = w(5); vz = w(6);
vmag = sqrt(vx^2+vy^2+vz^2);
dwdt = [vx;vy;vz; ...
        -c1*vx*vmag+c2*(wy*vz-wz*vy); ...
        -c1*vy*vmag+c2*(wz*vx-wx*vz); ...
        -9.81-c1*vz*vmag+c2*(wx*vy-wy*vx)];
end
function [eventvalue,stopthecalc,eventdirection] = events(t,w)
% Function to check for a special event in the trajectory
x = w(1); y=w(2); z=w(3);
hor_dist_traveled = sqrt(x^2+y^2);
eventvalue = [hor_dist_traveled-d];
stopthecalc = [1]; % This makes MATLAB stop the calc
eventdirection = [1]; % We look for a zero crossing from below

end
end
```

[0 POINTS]

1.5 Finally, run simulations with parameters representing a baseball ( $m=0.142\text{kg}$ ,  $C_D = 0.3$ ,  $C_L = 0.01$ ,  $D=7.48\text{cm}$ , air density  $1.02 \text{ kg/m}^3$ ) launched at  $34.4 \text{ m/s}$  ( $77 \text{ mph}$ ) at an angle of  $1$  degrees to the horizontal. Run simulations with  $\boldsymbol{\omega} = \mathbf{0}$  and  $\boldsymbol{\omega} = 132\mathbf{k} \text{ rad/sec}$  (the second case corresponds to a 'slider' from a right-handed pitcher). In each case, calculate the position vector of the ball when it crosses the plate (at a distance  $d=18.44\text{m}$  from the launch point). Hence, calculate the deflection of the ball due to spin. There is no need to hand in MATLAB code or graphs for this problem – just report your values.

Here's a graph showing the two trajectories. The end positions and deflected distance are shown on the figure. Solutions for the deflection may differ, depending on how the 'distance traveled' is interpreted (using just the  $x$  distance is fine) and on the accuracy of the computation – so any solution near  $0.2\text{m}$  deflection is fine.



[3 POINTS]

2. In this problem, you will write a MATLAB script to predict the orbit of a satellite around the earth. This is the first step in the calculations you will need to do for the lunar impact design project later in the semester. To simplify calculations:

- Neglect the gravitation of the moon and sun
- Assume that the earth is stationary

2.1 The position of the satellite will be described using its  $(x,y,z)$  coordinates (the design project handout describes how the  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  directions are chosen, but this is not important here). Write down the velocity and acceleration of the satellite in terms of these variables

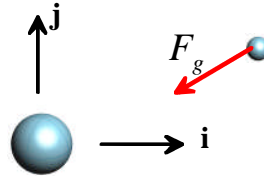
$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

$$\mathbf{v} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k}$$

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

[2 POINTS]

2.2 Draw a free body diagram showing the forces acting on the satellite.



[1 POINT]

2.3 Write down Newton's law of motion  $\mathbf{F} = m\mathbf{a}$  for the satellite.

Following the procedure described in class (or e-notes)

$$-\frac{GMm}{R^2} \frac{(x\mathbf{i} + y\mathbf{j} + z\mathbf{k})}{R} = m \left( \frac{d^2x}{dt^2} \mathbf{i} + \frac{d^2y}{dt^2} \mathbf{j} + \frac{d^2z}{dt^2} \mathbf{k} \right)$$

[2 POINTS]

2.4 Show that the equations of motion can be expressed in MATLAB form as

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ -GMx/R^3 \\ -GM_y/R^3 \\ -GM_z/R^3 \end{bmatrix}$$

Where  $G$  is the gravitational constant,  $M$  is the mass of the earth, and  $R = \sqrt{x^2 + y^2 + z^2}$  is the distance of the satellite from the earth's center.

The three components of the vector equation of motion yield

$$\frac{d^2x}{dt^2} = -\frac{GMx}{R^3} \quad \frac{d^2y}{dt^2} = -\frac{GM_y}{R^3} \quad \frac{d^2z}{dt^2} = -\frac{GM_z}{R^3}$$

Each of these equations must be re-written as two first-order equations, as follows

$$v_x = \frac{dx}{dt} \quad v_y = \frac{dy}{dt} \quad v_z = \frac{dz}{dt}$$
$$\frac{dv_x}{dt} = -\frac{GMx}{R^3} \quad \frac{dv_y}{dt} = -\frac{GM_y}{R^3} \quad \frac{dv_z}{dt} = -\frac{GM_z}{R^3}$$

Collecting these into vector form gives the solution stated.

[2 POINTS]

2.5 Write a MATLAB script that will integrate the equations of motion. Be sure to use the 'odeset' command to control the accuracy of the computation, as discussed in class. (THERE IS NO NEED TO SUBMIT A SOLUTION TO THIS PROBLEM]

```
function satellite

GM = 3.986012e05; % Grav constant (km and sec)
x0 = [-19063,18504,760]; % Initial position (km)
v0 = [-1.6128, -1.5314, -3.1703]; % Initial velocity (km/sec)
w0 = [x0,v0];
time = 24*3600; % Time of run (seconds)
close all
options = odeset('RelTol',0.0001,'Events',@events);
[t_vals,w_vals,te,we,ie] = ode45(@eq_of_motion,[0,time],w0,options);

te % This prints the time for all the events in the MATLAB window.

plot3(w_vals(:,1),w_vals(:,2),w_vals(:,3));

function dwdt = eq_of_motion(t,w)
    x = w(1); y=w(2); z=w(3);
    R = sqrt(x^2+y^2+z^2);
    vx = w(4); vy = w(5); vz = w(6);
    dwdt = [vx;vy;vz;-GM*x/R^3;-GM*y/R^3;-GM*z/R^3];
end

function [eventvalue,stopthecal,eventdirection] = events(t,w)
    % Function to check for a special event in the trajectory
    x = w(1); y=w(2); z=w(3);
    eventvalue = [z]; % We look for z=0
    stopthecal = [0]; % We don't stop at the event
    eventdirection = [0]; % We look for a zero crossing all directions
end

end

end
```

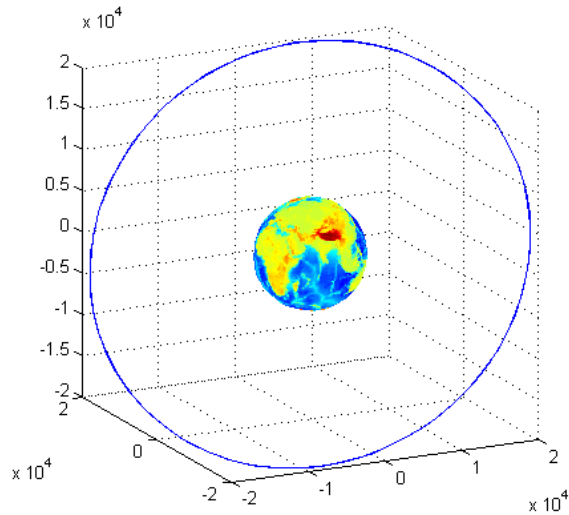
[0 POINTS]

2.6 Test your code by running it with the following parameters:

- $\mu = GM = 3.986012 \times 10^5 \text{ km}^3\text{s}^{-1}$
- Initial position:  $\mathbf{r} = -19063\mathbf{i} + 18504\mathbf{j} + 760\mathbf{k}$  km,
- Initial velocity:  $\mathbf{v} = -1.6128\mathbf{i} - 1.5314\mathbf{j} - 3.1703\mathbf{k}$  km/sec

(this is a typical orbit for a GPS satellite). Hand in a plot showing the satellite orbit.

The orbit is shown below (the earth is shown to scale). Drawing the earth is not required, of course.



[3 POINTS]

2.7 Add an 'Event' function to your code that will determine the times that the satellite crosses the equatorial plane  $z=0$ . Hence, determine the time for one complete orbit.

MATLAB finds zero crossings at:

- 240 sec
- 21794 sec
- 43337 sec
- 64879 sec

Since all zero crossings are found, the time between two events is the time for one half orbit. The time for a complete orbit is 43097 sec (11.97 hours)

[2 POINTS]

3. A 'Penning Trap' is a device that uses specially shaped electrostatic and electromagnetic fields to trap charged particles (or 'ions'). 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 (see, e.g. [Major et al \(2005\)](#) for more information). The ions inside a Penning 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 Penning trap.

2.1 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.

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

$$\mathbf{v} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k}$$

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

[1 POINT]

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

$$\mathbf{F} = Q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$$

where  $\mathbf{r}$  is the ion's (instantaneous) position vector,  $\mathbf{v}$  is its velocity vector,  $Q$  is its charge, and

$$\mathbf{E} = \frac{E_0}{d}(x\mathbf{i} + y\mathbf{j} - 2z\mathbf{k}) \quad \mathbf{B} = B_0\mathbf{k}$$

are the electric and magnetic field vectors. Here,  $E_0, d, B_0$  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

$$\frac{d^2x}{dt^2} - \omega_c v_y - \frac{\omega_z^2}{2}x = 0 \quad \frac{d^2y}{dt^2} + \omega_c v_x - \frac{\omega_z^2}{2}y = 0 \quad \frac{d^2z}{dt^2} + \omega_z^2 z = 0$$

where

$$\omega_c = \frac{QB_0}{m} \quad \omega_z = \sqrt{\frac{2QE_0}{dm}}$$

are two parameters (known as the 'cyclotron frequency' and 'axial frequency' of the system).

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

$$\frac{QE_0}{d}(x\mathbf{i} + y\mathbf{j} - 2z\mathbf{k}) + Q(v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}) \times B_0\mathbf{k} = m \left( \frac{d^2x}{dt^2}\mathbf{i} + \frac{d^2y}{dt^2}\mathbf{j} + \frac{d^2z}{dt^2}\mathbf{k} \right)$$

Evaluating the cross product, and rearranging the  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  components yields three equations

$$\frac{d^2x}{dt^2} - \frac{QB_0}{m}v_y - \frac{QE_0}{dm}x = 0$$

$$\frac{d^2y}{dt^2} + \frac{QB_0}{m}v_x - \frac{QE_0}{dm}y = 0$$

$$\frac{d^2z}{dt^2} + \frac{2QE_0}{dm}z = 0$$

This clearly reduces to the expressions given.

[3 POINTS]

2.3 Re-write the equations of motion in problem 2.2 as 6 first-order differential equations that can be integrated using MATLAB.

As always, we introduce the velocity components as additional unknowns. Here is the solution

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_c v_y + \frac{\omega_z^2}{2}x \\ -\omega_c v_x + \frac{\omega_z^2}{2}y \\ -\omega_z^2 z \end{bmatrix}$$

[2 POINTS]



2.4 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. Make your script plot the trajectory of the ion (use the MATLAB 'plot3' command) You need not submit a solution to this problem.

```

function penning_trap

close all
wz = 1;
wc = 1.413;

w0 = [0.10;0;0;0.0;0.0;0.];

[tvals,wvals] = ode45(@eq_of_motion,[0,300],w0);

plot3(wvals(:,1),wvals(:,2),wvals(:,3));

function dwdt = eq_of_motion(t,w)
    x = w(1); y = w(2); z = w(3);
    vx = w(4); vy = w(5); vz = w(6);

    Fx = wz^2*x/2 + vy*wc;
    Fy = wz^2*y/2 - vx*wc;
    Fz = -wz^2*z;

    dwdt = [vx;vy;vz;Fx;Fy;Fz];

end

end

```

[ 0 POINTS]

2.5 An ion will be trapped if the frequencies  $\omega_c, \omega_z$  satisfy the condition  $\omega_c^2 - 2\omega_z^2 > 0$ . Check this prediction by running two simulations with the following parameters:

(a)  $x=0.1 \quad y=z=0 \quad v_x=v_y=v_z=0 \quad \omega_c=1.415 \quad \omega_z=1$

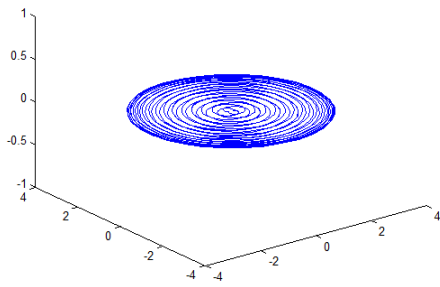
(b)  $x=0.1 \quad y=z=0 \quad v_x=v_y=v_z=0 \quad \omega_c=1.413 \quad \omega_z=1$

Run both tests for a time interval of 300 units. Hand in a graph showing the predicted trajectory for each case.

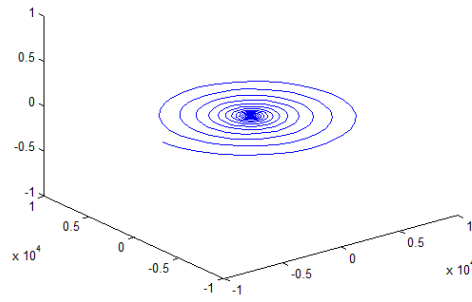
When used as a spectrometer, the ion trajectories are used (indirectly) to determine values for  $\omega_c, \omega_z$ , and hence to find the mass of the particle. To test the influence of  $\omega_c, \omega_z$  on the trajectories, run simulations with

(c)  $x=0.1 \quad y=z=0 \quad v_x=v_y=0 \quad v_z=0.1 \quad \omega_c=1.5 \quad \omega_z=1$

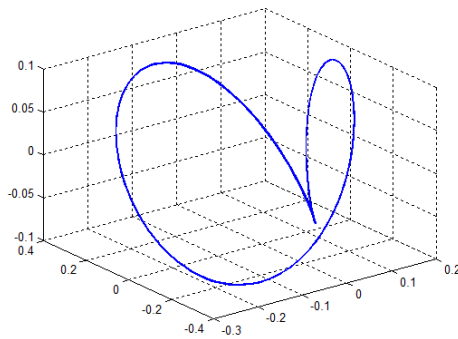
(d)  $x=0.1 \quad y=z=0 \quad v_x=v_y=0 \quad v_z=0.1 \quad \omega_c=2.125 \quad \omega_z=1.5$



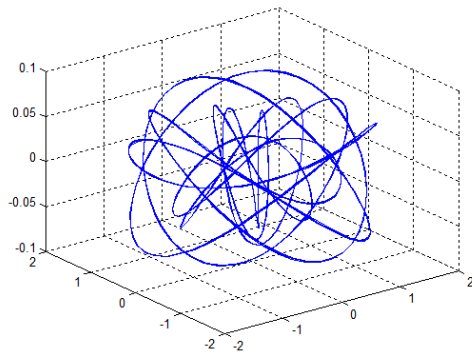
(a)



(b)



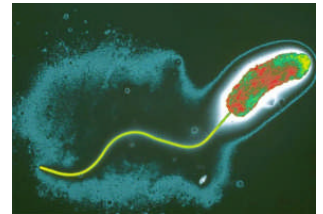
(c)



(d)

[4 POINTS]

**4. OPTIONAL PROBLEM FOR EXTRA CREDIT** There is currently great interest in analyzing the propulsion mechanisms and motion of bacteria. This is partly because bacteria can be genetically engineered to perform useful tasks (for example, bacteria have been used as microfluidic pumps), and partly because there are too many physicists with not enough to do.



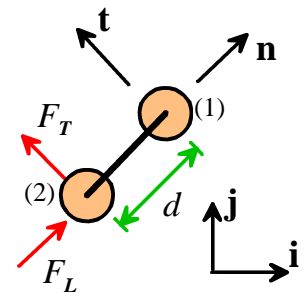
Many bacteria propel themselves by means of a rotating tail (a flagellum) as shown in the figure (from Nature 406, 469-470 3 August 2000) doi:10.1038/35020178

In this problem, you will set up and analyze a simple model that is used to describe the motion of a bacterium swimming through a viscous fluid.

The body of the bacterium is represented by two beads with mass  $m$ , connected by a rigid link with length  $d$ , as shown in the figure. Each bead is subjected to a viscous drag force

$$\mathbf{F}_D = -\eta \mathbf{v}$$

where  $\mathbf{v}$  is the velocity vector of the bead, and  $\eta$  is the viscosity of the fluid. The bacterium is propelled by its rotating flagellum, which is attached to the rearmost bead. The flagellum exerts a constant force  $F_L$  parallel to the body of the bacterium, together with a small fluctuating transverse force



$F_T(t) = F_T \sin \omega t$ . These forces can be expressed as vectors

$$\mathbf{F}_L = F_L \mathbf{n} \quad \mathbf{F}_T = F_T \sin \omega t \mathbf{t}$$

where  $\mathbf{n}$  and  $\mathbf{t}$  are unit vectors parallel and perpendicular to the rigid link, as shown in the figure. For simplicity, we will analyze motion only in two dimensions.

4.1 The motion of the bacterium will be described using the position vectors of the two beads

$\mathbf{r}_1 = x_1 \mathbf{i} + y_1 \mathbf{j}$   $\mathbf{r}_2 = x_2 \mathbf{i} + y_2 \mathbf{j}$ . Write down the velocity and acceleration vector of each bead in terms of  $(x_1, y_1), (x_2, y_2)$  and their time derivatives.

$$\mathbf{v}_1 = \frac{dx_1}{dt} \mathbf{i} + \frac{dy_1}{dt} \mathbf{j} \quad \mathbf{a}_1 = \frac{d^2 x_1}{dt^2} \mathbf{i} + \frac{d^2 y_1}{dt^2} \mathbf{j}$$

$$\mathbf{v}_2 = \frac{dx_2}{dt} \mathbf{i} + \frac{dy_2}{dt} \mathbf{j} \quad \mathbf{a}_2 = \frac{d^2 x_2}{dt^2} \mathbf{i} + \frac{d^2 y_2}{dt^2} \mathbf{j}$$

[2 POINTS]

4.2 Find formulas for the components of the unit vectors  $\mathbf{n}$  and  $\mathbf{t}$  in terms of  $x_1, y_1, x_2, y_2$  and  $d$

It is easy to write down  $\mathbf{n}$ :  $\mathbf{n} = [(x_1 - x_2) \mathbf{i} + (y_1 - y_2) \mathbf{j}] / d$ . The vector  $\mathbf{t}$  can be written down directly, or else can be calculated by noting that  $\mathbf{t}$  is perpendicular to both  $\mathbf{n}$  and  $\mathbf{k}$ , so  $\mathbf{t} = \mathbf{k} \times \mathbf{n}$ , which gives  $\mathbf{t} = [-(y_1 - y_2) \mathbf{i} + (x_1 - x_2) \mathbf{j}] / d$

[2 POINTS]

4.3 The length of the rigid link remains constant. This requires that  $(\mathbf{r}_1 - \mathbf{r}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) = d^2$ . By differentiating this expression with respect to time, show that the accelerations of the two beads must be related by the following constraint

$$(\mathbf{a}_1 - \mathbf{a}_2) \cdot (\mathbf{r}_2 - \mathbf{r}_1) = (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2)$$

Differentiating using the chain rule gives

$$\frac{d}{dt} (\mathbf{r}_1 - \mathbf{r}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) = 0 \Rightarrow (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) + (\mathbf{r}_1 - \mathbf{r}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2) = 0$$

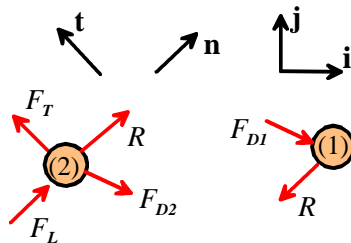
$$\Rightarrow (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) = 0$$

$$\Rightarrow \frac{d}{dt} (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) \Rightarrow (\mathbf{a}_1 - \mathbf{a}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2) + (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2) = 0$$

$$\Rightarrow (\mathbf{a}_1 - \mathbf{a}_2) \cdot (\mathbf{r}_2 - \mathbf{r}_1) = (\mathbf{v}_1 - \mathbf{v}_2) \cdot (\mathbf{v}_1 - \mathbf{v}_2)$$

[2 POINTS]

4.4 Draw two free body diagrams showing the forces acting on each bead. Note that the rigid link exerts equal and opposite forces on the two particles. The forces have unknown magnitude  $R$ , but parallel to the link.



[3 POINTS]

4.5 Write down Newton's laws of motion  $\mathbf{F} = m\mathbf{a}$  for each bead.

$$m\mathbf{a}_1 = -\eta\mathbf{v}_1 - R\mathbf{n}$$

$$m\mathbf{a}_2 = -\eta\mathbf{v}_2 + F_L\mathbf{n} + F_T \sin \omega t \mathbf{t} + R\mathbf{n}$$

or in  $(\mathbf{i}, \mathbf{j})$  components:

$$m \left( \frac{d^2 x_1}{dt^2} \mathbf{i} + \frac{d^2 y_1}{dt^2} \mathbf{j} \right) = -\eta \left( \frac{dx_1}{dt} \mathbf{i} + \frac{dy_1}{dt} \mathbf{j} \right) - R [(x_1 - x_2)\mathbf{i} + (y_1 - y_2)\mathbf{j}] / d$$

$$m \left( \frac{d^2 x_2}{dt^2} \mathbf{i} + \frac{d^2 y_2}{dt^2} \mathbf{j} \right) = -\eta \left( \frac{dx_2}{dt} \mathbf{i} + \frac{dy_2}{dt} \mathbf{j} \right) + (F_L + R) [(x_1 - x_2)\mathbf{i} + (y_1 - y_2)\mathbf{j}] / d + F_T \sin \omega t [-(y_1 - y_2)\mathbf{i} + (x_1 - x_2)\mathbf{j}] / d$$

[3 POINTS]

4.6 Hence, show that the equations of motion for the two beads, together with the constraint equation in problem (3), can be expressed in MATLAB form as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m & 0 & 0 & 0 & (x_2 - x_1) / d & \dot{v}_{x1} \\ 0 & 0 & 0 & 0 & 0 & m & 0 & 0 & (y_2 - y_1) / d & \dot{v}_{y1} \\ 0 & 0 & 0 & 0 & 0 & 0 & m & 0 & (x_1 - x_2) / d & \dot{v}_{x2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & m & (y_1 - y_2) / d & \dot{v}_{y2} \\ 0 & 0 & 0 & 0 & (x_2 - x_1) & (y_2 - y_1) & (x_1 - x_2) & (y_1 - y_2) & 0 & R \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \dot{v}_{x1} \\ \dot{v}_{y1} \\ \dot{v}_{x2} \\ \dot{v}_{y2} \\ R \end{bmatrix} = \begin{bmatrix} v_{x1} \\ v_{y1} \\ v_{x2} \\ v_{y2} \\ -\eta v_{x1} \\ -\eta v_{y1} \\ F_L n_x + F_T t_x - \eta v_{x2} \\ F_L n_y + F_T t_y - \eta v_{y2} \\ (v_{x1} - v_{x2})^2 + (v_{y1} - v_{y2})^2 \end{bmatrix}$$

where  $R$  is the unknown reaction force in the link, the dots over the variables represent time derivatives, and  $n_x = (x_1 - x_2) / d$ ,  $n_y = (y_1 - y_2) / d$ ,  $t_x = -(y_1 - y_2) / d$ ,  $t_y = (x_1 - x_2) / d$

As usual, we introduce the velocity components  $v_{x1}, v_{y1}$  and  $v_{x2}, v_{y2}$  as additional unknowns.

The first four equations in the system follow as

$$\dot{x}_1 = v_{x1} \quad \dot{y}_1 = v_{y1} \quad \dot{x}_2 = v_{x2} \quad \dot{y}_2 = v_{y2}$$

The  $\mathbf{i}, \mathbf{j}$  components of the two equations of motion yield the following equations

$$m \frac{dv_{x1}}{dt} + R(x_1 - x_2) / d = -\eta v_{x1} \quad m \frac{dv_{y1}}{dt} + R(y_1 - y_2) / d = -\eta v_{y1}$$

$$m \frac{dv_{x2}}{dt} + R(x_2 - x_1) / d = -\eta v_{x2} + F_L(x_1 - x_2) / d - F_T \sin \omega t (y_1 - y_2) / d$$

$$m \frac{dv_{y2}}{dt} + R(y_2 - y_1) / d = -\eta v_{y2} + F_L(y_1 - y_2) / d + F_T \sin \omega t (x_1 - x_2) / d$$

Writing these equations in matrix form gives the solution.

[3 POINTS]

4.7 Write a MATLAB script that will calculate and plot the trajectory of the bacterium (you only need to plot the trajectory of one of the two beads). There is no need to submit a solution to this problem.

```
function bacteria

eta = 10.1;
m = 0.1;
FL = 0.1;
FT = 0.01;
d = 0.001;
omega = 0.1;

w0 = [0;0;-d;0;0;0.0;0;0];

options=odeset('RelTol',0.0001);
[tvals,wvals] = ode45(@eq_of_motion,[0,300],w0,options);

plot(wvals(:,1),wvals(:,2));
axis square

function dwdt = eq_of_motion(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);

    K = eye(9,9); % This sets up a matrix with 1s on the diags
    K(5,5) = m;
    K(6,6) = m;
    K(7,7) = m;
    K(8,8) = m;
    K(5,9) = (x2-x1)/d;
    K(6,9) = (y2-y1)/d;
    K(7,9) = (x1-x2)/d;
    K(8,9) = (y1-y2)/d;
    K(9,5) = (x2-x1);
    K(9,6) = (y2-y1);
    K(9,7) = (x1-x2);
    K(9,8) = (y1-y2);
    K(9,9) = 0;
    rhs = [vx1;vy1;vx2;vy2;-eta*vx1;...
          -eta*vy1;...
          (FL*(x1-x2)-FT*sin(omega*t)*(y1-y2))/d-eta*vx2;...
          (FL*(y1-y2)+FT*sin(omega*t)*(x1-x2))/d-eta*vy2;...
          (vx1-vx2)^2+(vy1-vy2)^2];
    sol = K\rhs;
    dwdt=sol(1:8);
end

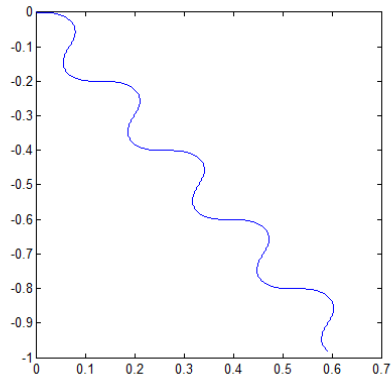
end
```

[0 POINTS]

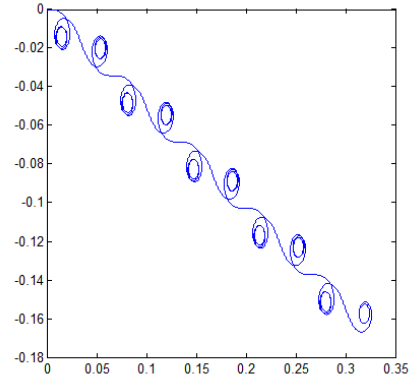
- 5 Test your code by running it with the following parameters:
- $\eta = 10$ ,  $\omega = 0.1$ ,  $F_T = 0.01$ ,  $F_L = 1$ ,  $d = 0.01$ , initial position  $x_1 = y_1 = 0$   $x_2 = -d$ ,  $y_2 = 0$  and zero initial velocity
  - $\eta = 10$ ,  $\omega = 0.1$ ,  $F_T = 0.01$ ,  $F_L = 1$ ,  $d = 0.001$ , initial position  $x_1 = y_1 = 0$   $x_2 = -d$ ,  $y_2 = 0$  and zero initial velocity.

Hand in plots showing the predicted trajectory for each case.

The two cases are shown in figs (a) and (b) below, respectively.



(a)



(b)

**[5 POINTS]**