CS201: Introduction to computational physics

Assignment 4

Id: 201401114

Name: Omkar Damle

Q1. Charge Particle trajectories under Lorentz force: Write a general MATLAB code to reproduce charge particle motions under Lorentz force in the following cases (as shown in Lecture-15 slide, choice of correct initial conditions is important to reproduce the trajectories). Report on the initial conditions and the rationale behind observed motion. Support your answer with several supporting graphs. (Try different 3D plotting schemes in MATLAB for better visualization, other than "plot3" as discussed in the class). Analyze the motion for t=0 to a reasonable value of t=t_final.

(i) Static and uniform B field. (for +ve and –ve charges) - done

(ii) Static and uniform E and B. (ExB drift) (what happens when v=v0x; B=B0z; and E=E0y; and v0=E0/B0). -

(iii) Static and non-uniform B field (grad B drift)

(iv) Static and uniform B, and under gravitational force (for different mass)

Investigate for at-least three different initial conditions for all the cases (i-iv). Compare the results for different cases

Solution :

$$\vec{F} = q(\vec{E} + \vec{v}\times\vec{B}).$$

$$m\vec{a} = q(\vec{E} + \vec{v}\times\vec{B})$$

$$\begin{vmatrix} i & j & k \\ v_x & v_y & v_z \\ B_x & B_y & B_z \end{vmatrix} = i(v_yB_z - v_zB_y) - j(v_xB_z - v_zB_x) + k(v_xB_y - v_yB_x).$$

$$m\frac{dv_x}{dt} = q(E_x + v_yB_z - v_zB_y).$$

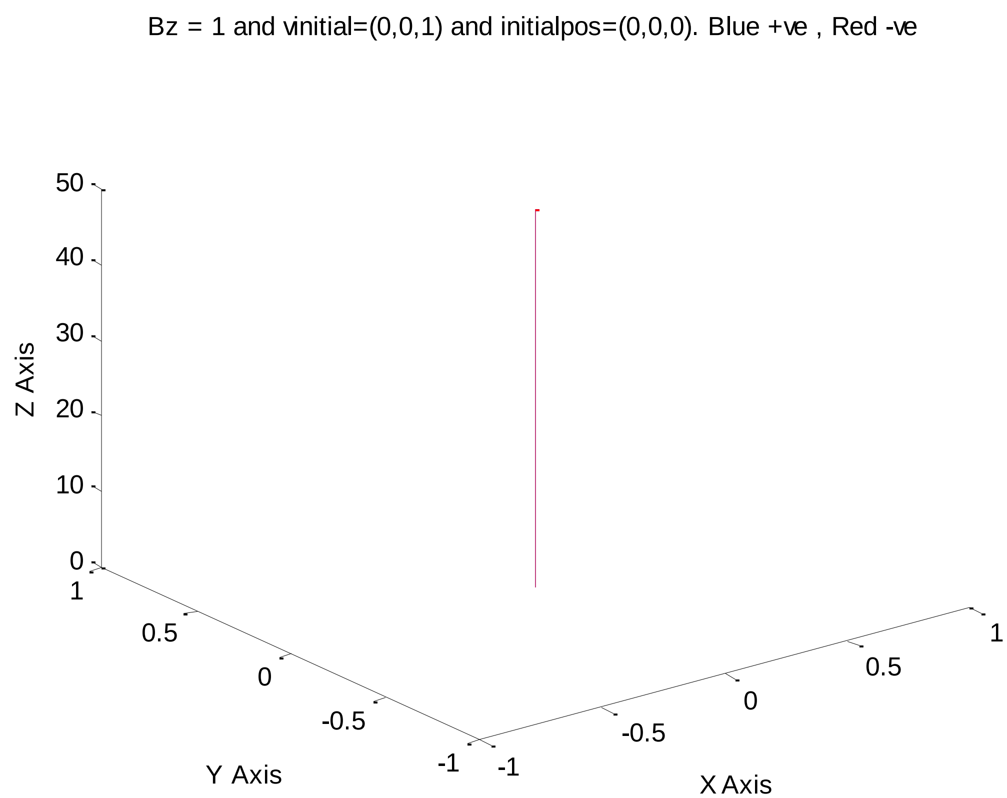
$$m\frac{dv_y}{dt} = q(E_y - v_xB_z + v_zB_x)$$

$$m\frac{dv_z}{dt} = q(E_z + v_xB_y - v_yB_x).$$

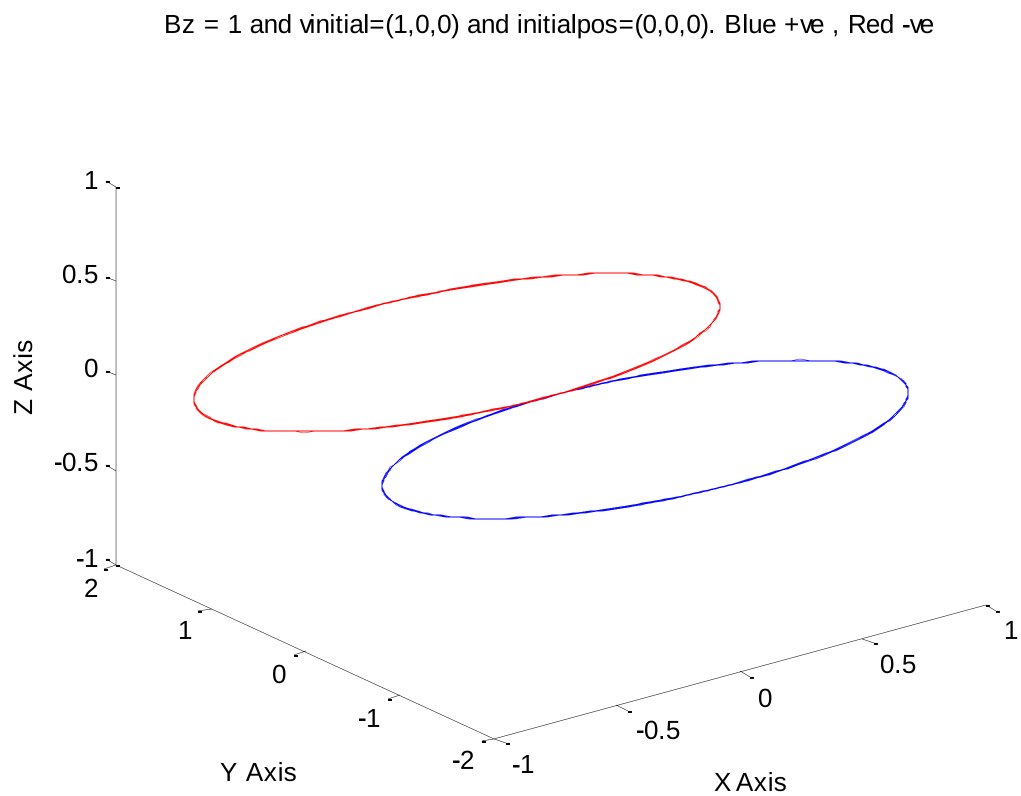
(i) Static and uniform B field

Put E = 0 in the above equations.
Case 1 : If initial velocity is 0, particle remains stationary

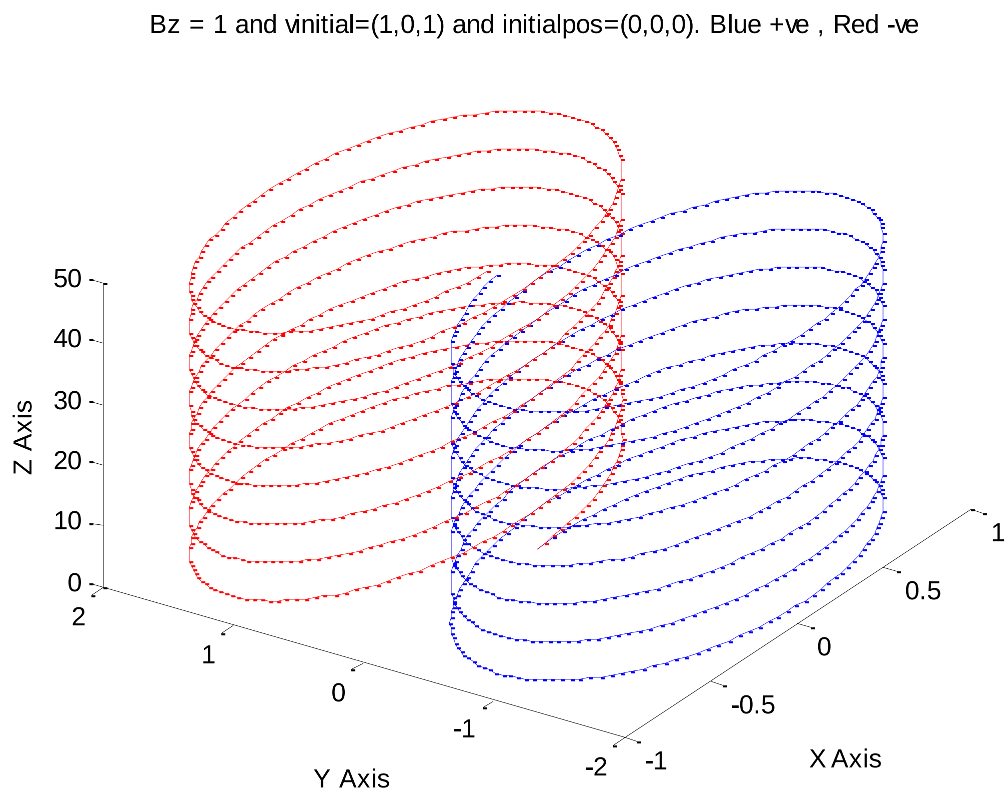
Case 2 : If initial velocity is parallel to the magnetic field, then the force acting on the particle is 0 and hence it continues travelling with the same constant velocity. Note that both +ve as well as -ve charge follows same path.



Case 3: Initial velocity is perpendicular to magnetic field. The particle performs UCM – uniform circular motion. The magnetic field provides the centripetal force.



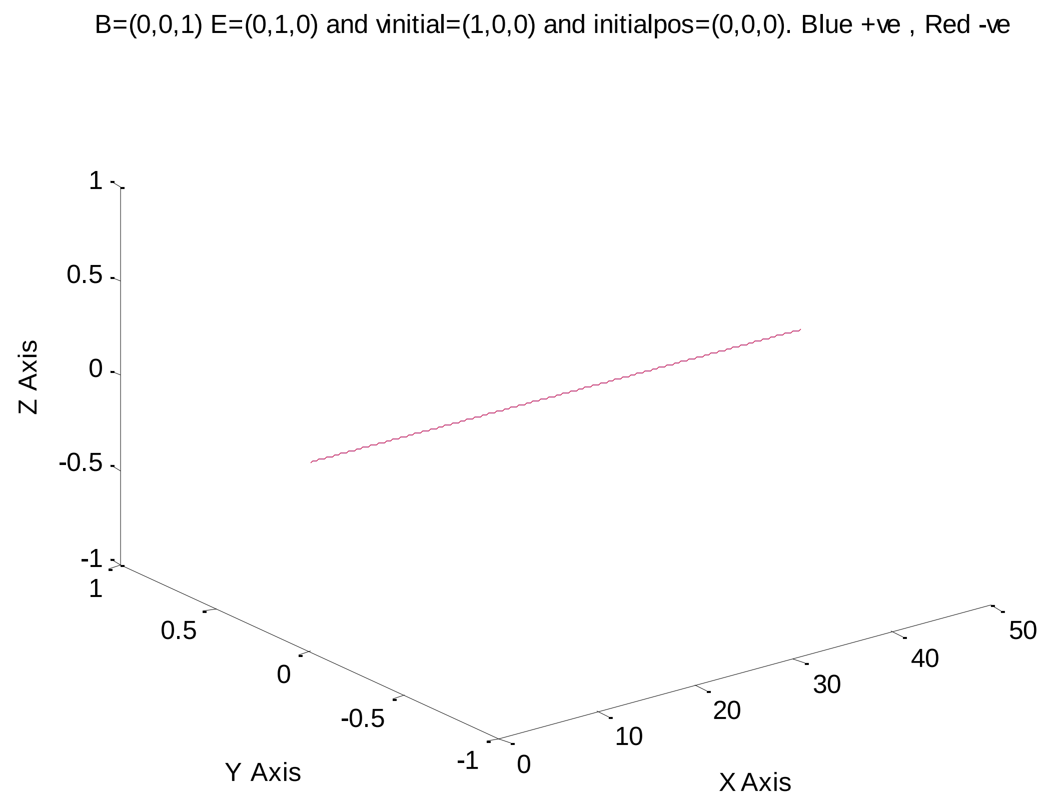
Case 4: The initial velocity is neither perpendicular nor parallel to the magnetic field. Here we can decompose the initial velocity into perpendicular and parallel components. The parallel component remains unaffected. We get a helix .



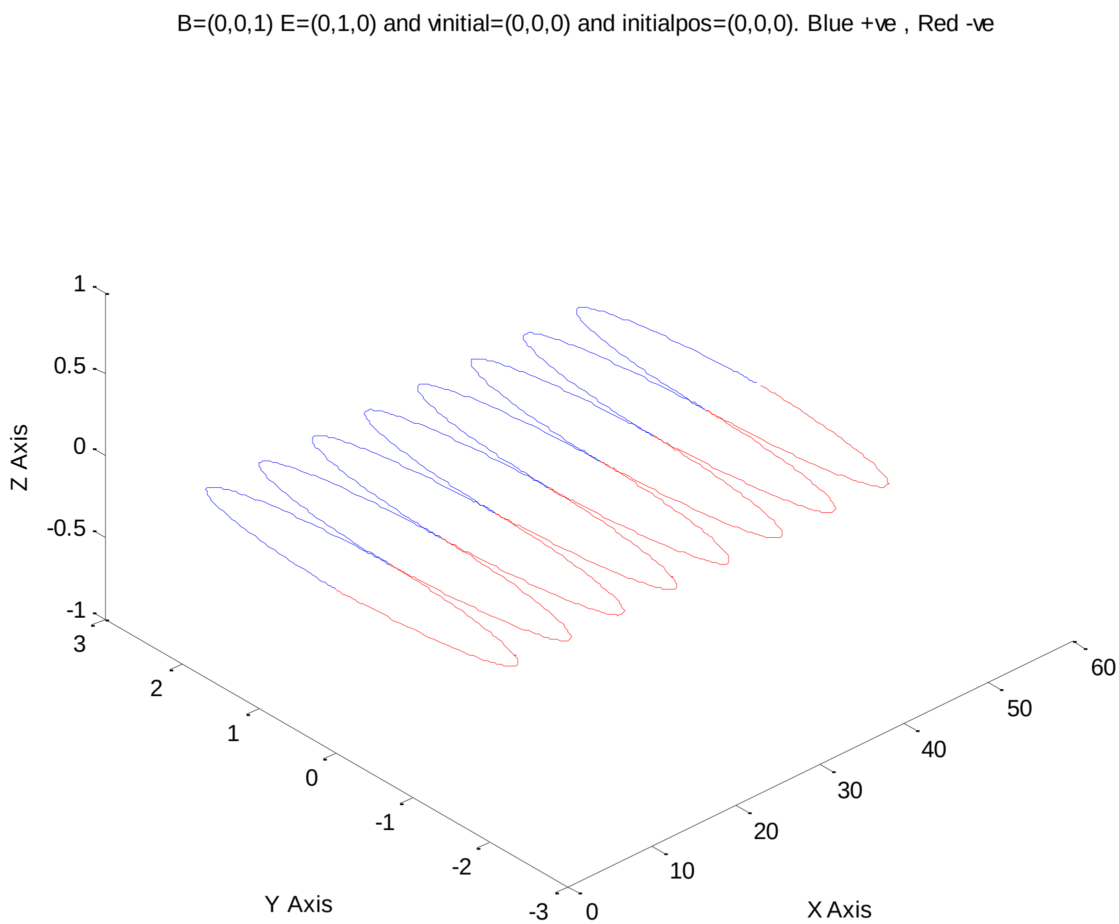
(ii) Static and uniform E and B. (ExB drift)

Case 1 : `B=(0,0,1) E=(0,1,0) and vinitial=(0,0,1)`

This is a very special case where the magnetic and the electrical forces cancel each other perfectly. Hence charged particle continues moving with constant velocity. In general - when v=v0x; B=B0z; and E=E0y; and v0=E0/B0.



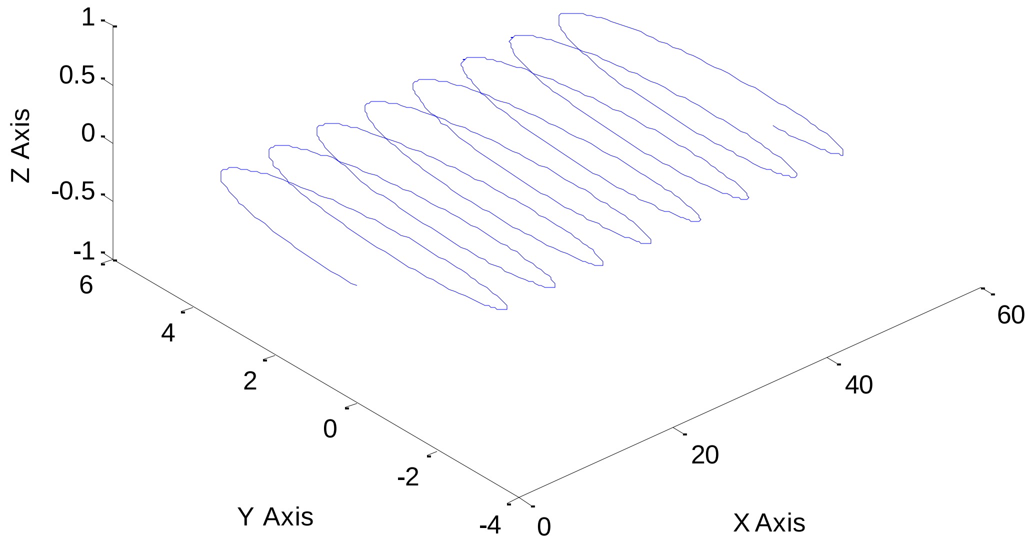
Case 2 : If the electric and the magnetic field is perpendicular to each other and the initial velocity is 0. We get a cycloid.



Case 3: `B=(0,0,1) E=(0,1,0) and vinitial=(0,3,0)`

The particle drifts in the direction of ExB . The drift velocity is 1m/s calculated by formula.

B=(0,0,1) E=(0,1,0) and vinitial=(0,3,0) and initialpos=(0,0,0). Blue +ve , Red -ve

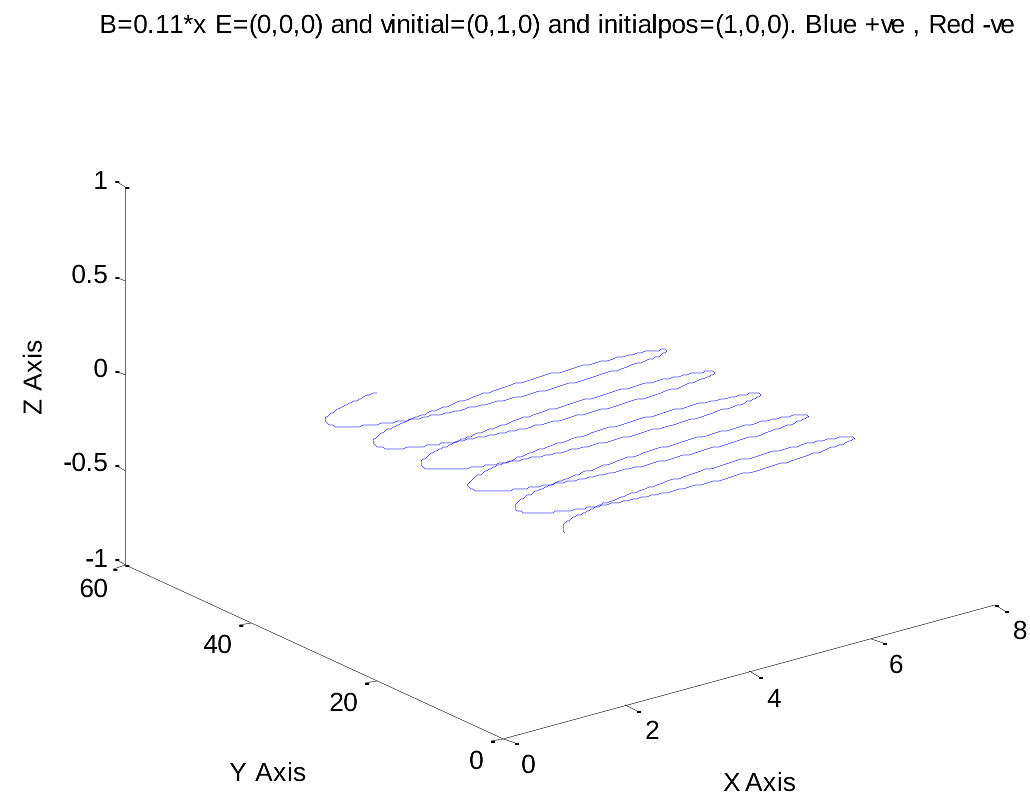


(iii) Static and non-uniform B field (grad B drift)

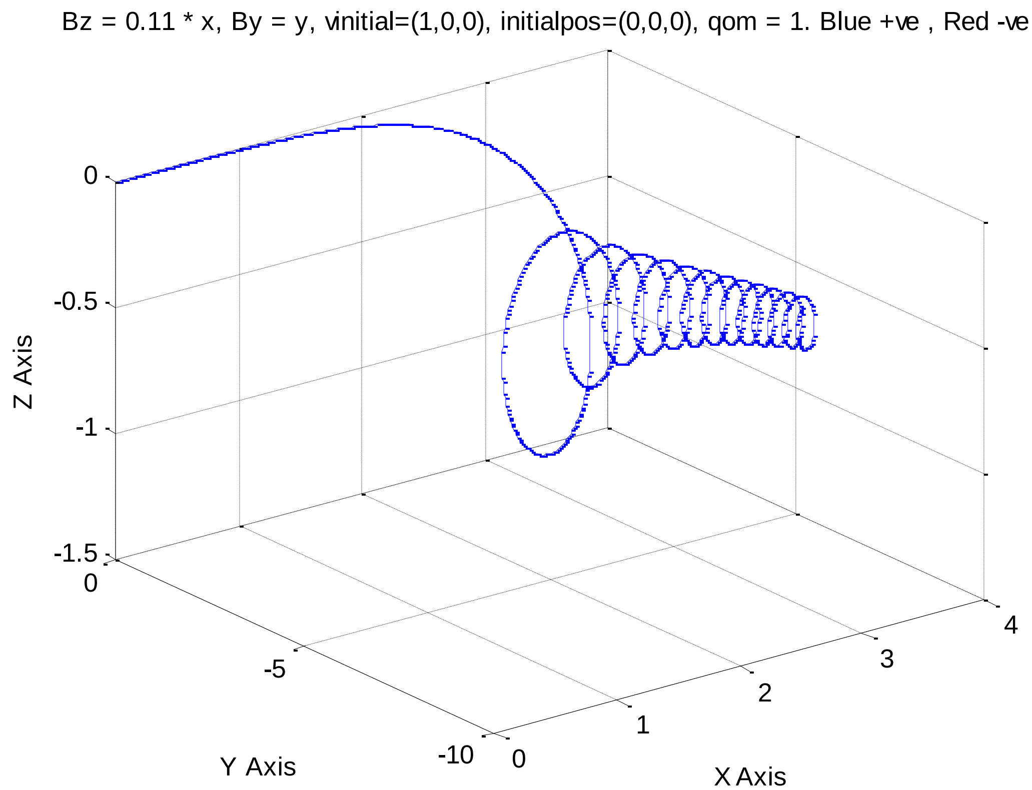
```
Case 1: B=0.11*x E=(0,0,0) and vinitial=(0,1,0) and initialpos=(1,0,0)
```

Here the magnetic field is not uniform. If the magnetic field changes the radius of gyration changes.

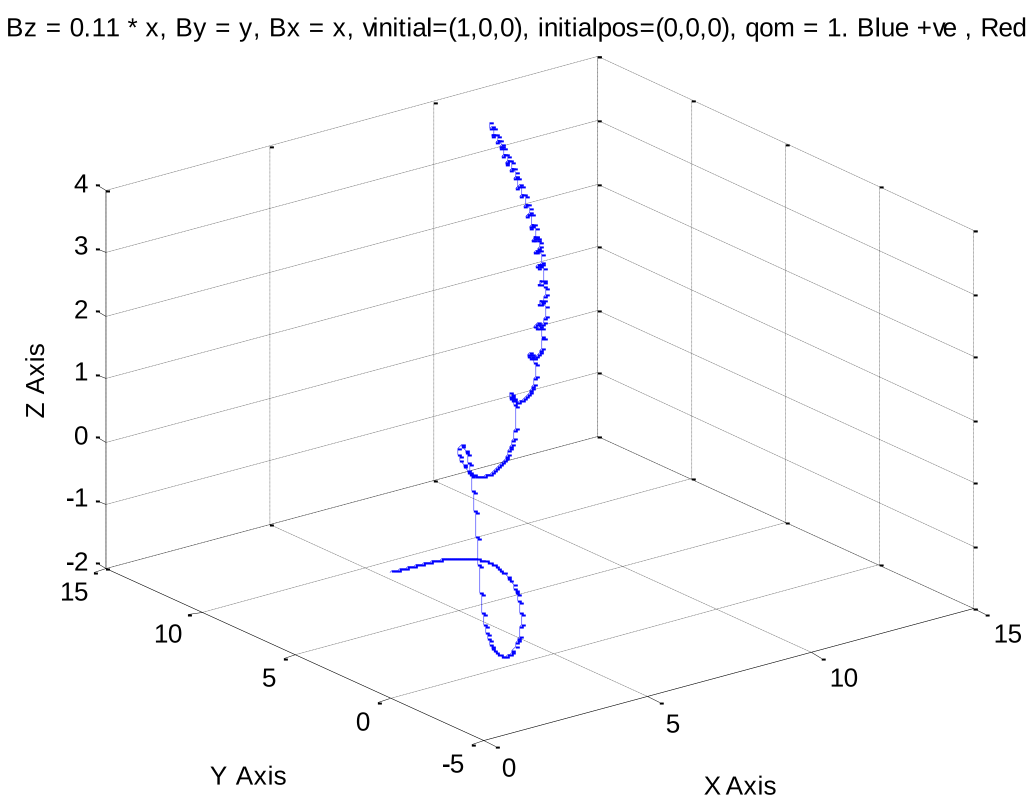
Radius is inversely proportional to the strength of B. This results in the following motion. This is known as grad B drift.



Case 2: Bz = 0.11 * x, By = y, vinitial=(1,0,0), initialpos=(0,0,0), qom = 1.



Case 3 : `Bz = 0.11 * x, By = y, Bx = x, vinitial=(1,0,0), initialpos=(0,0,0),`



(iv) Static and uniform B, and under gravitational force (for different mass)

This case is very similar to the case of static electric field and static magnetic field. Here the only difference is that the gravitational force is independent of the charge of the particle. Also the mass plays an important role in this case.

$$F_x = ma_x.$$

$$\vec{F} = mg\hat{j} + q(\vec{v} \times \vec{B}).$$

$$\rightarrow \cancel{0} = \cancel{m\frac{dv_x}{dt}}.$$

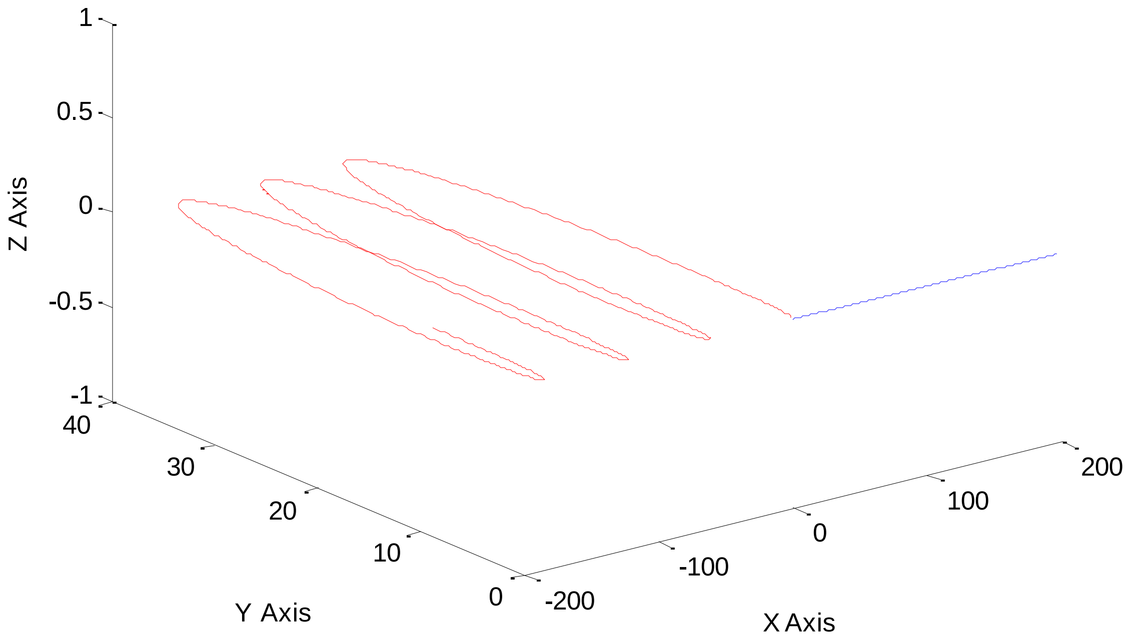
$$m\frac{dv_x}{dt} = q(v_y B_z - v_z B_y).$$

$$m\frac{dv_y}{dt} = mg + q(-v_x B_z + v_z B_x).$$

$$m\frac{dv_z}{dt} = q(v_x B_y - v_y B_x).$$

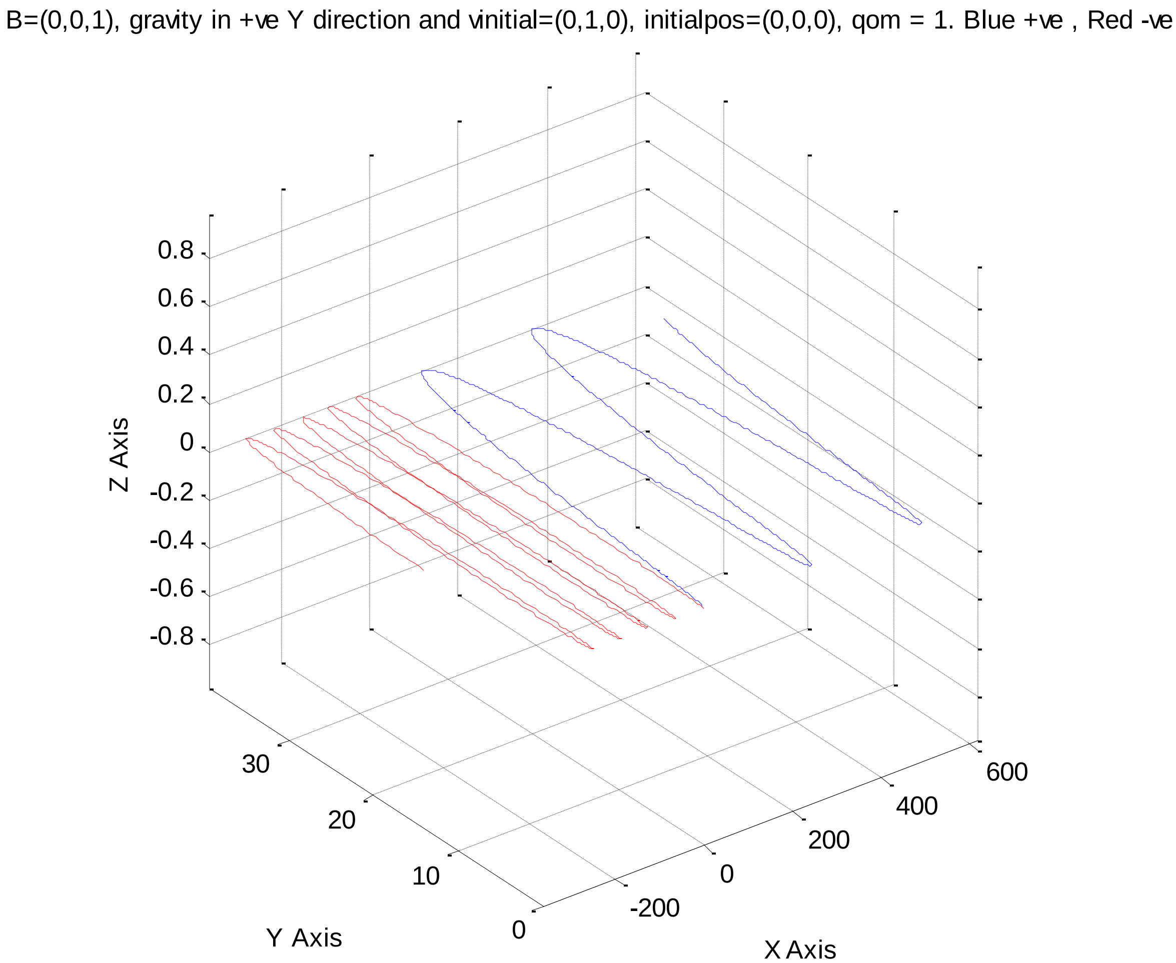
Case 1 : `B=(0,0,1), gravity in +ve Y direction and vinitial=(0,1,0), initialpos=(0,0,0), qom = 1`

B=(0,0,1), gravity in +ve Y direction and vinitial=(0,1,0), initialpos=(0,0,0), qom = 1. Blue +ve , Red -ve

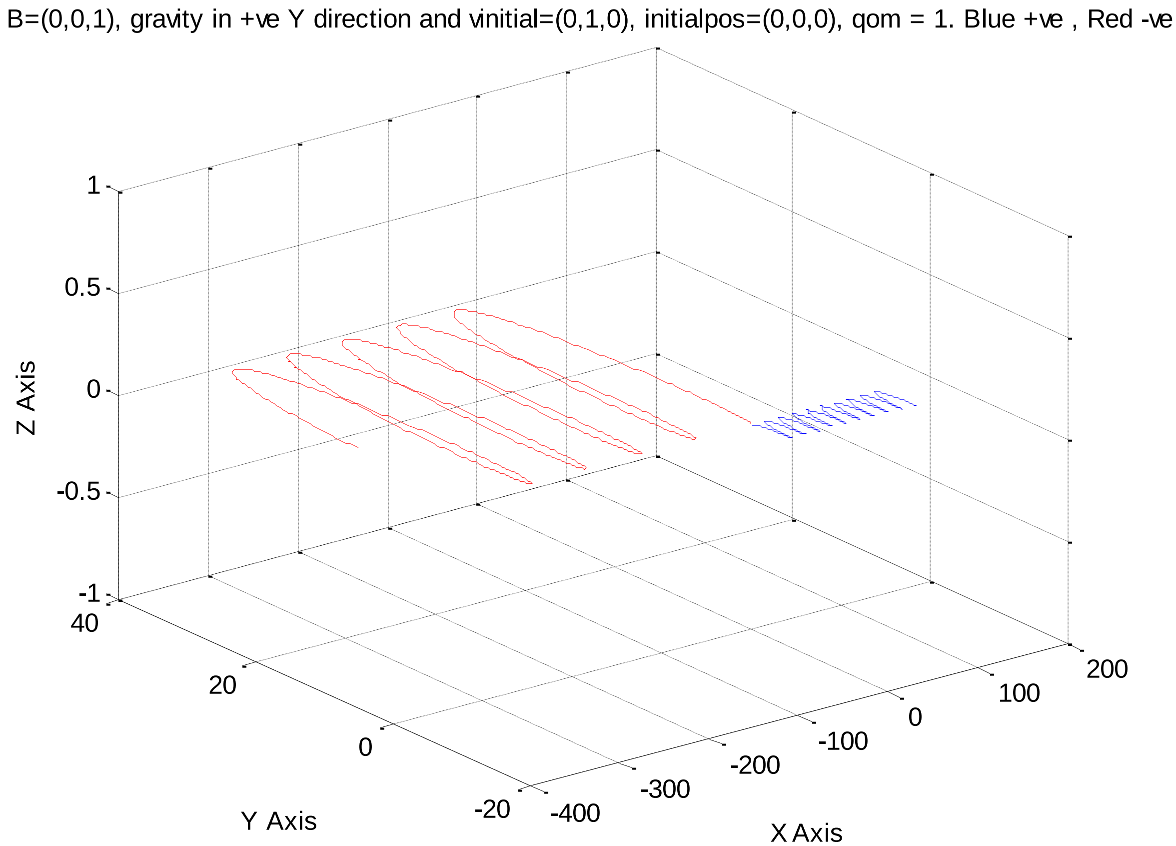


For the +ve particle with q/m = 1, the path is a straight line as the gravitational and the magnetic fields cancel out each other. This is not the case for the negative particle as the direction of the magnetic field is reversed.

Case 2: Now let us decrease the q/m ratio to 0.5. This can happen due to increase in mass. In this case the gravitational force will dominate. Hence for +ve particle it moves first in the Y-direction.



Case 3: Now we increase the q/m ratio to 2. This can happen if the mass decreases by half. Here the magnetic force dominates and hence the +ve particle moves in the -ve Y direction at the start.



Code:

Q1.m file ::

```
clear all;
close all;

global q m qom g;
global Bx By Bz;
global Ex Ey Ez;

%q = 1.6e-19;        %why does this not work ??
%m = 9.1e-31;

g=9.8;
qom = 2 ;

Bx = 0;
By = 0;
Bz = 1;

Ex = 0;
Ey = 0;
Ez = 0;
```

```
starttime = 0;
totaltime = 30;
dt = (totaltime-starttime) / 1000;

u0 = zeros (6 , 1);

u0(1) = 0 ;          %x
u0(2) = 0 ;          %y
u0(3) = 0 ;          %z

u0(4) = 9.8 ;          %vx
u0(5) = 0 ;          %vy
u0(6) = 0 ;          %vz

[t1,u1] = ode45(@LorentzForceFunction , [starttime : dt : totaltime] , u0) ;
qom = -1;
[t2,u2] = ode45(@LorentzForceFunction , [starttime : dt : totaltime] , u0) ;

plot3( u1(:,1) , u1(:,2) , u1(:,3) , 'b' , u2(:,1) , u2(:,2) , u2(:,3) , 'r'
)
grid on
xlabel('X Axis');
ylabel('Y Axis');
zlabel('Z Axis');
title('B=(0,0,1), gravity in +ve Y direction and vinitial=(0,1,0),
initialpos=(0,0,0), qom = 1. Blue +ve , Red -ve');
```

LorentzForceFunction.m file ::

```
function F = LorentzForceFunction( t , u )
%LORENTZFORCEFUNCTION Summary of this function goes here

% u
% x y z
% vx vy vz

% F
% vx vy vz
% dvx/dt dvy/dt dvz/dt

global q m qom g;
global Bx By Bz;
global Ex Ey Ez;

% for question 1-part c
% Bz = 0.11 * u(1);

F = zeros ( length(u) , 1);
```

```
F(1) = u (4) ;
F(2) = u (5) ;
F(3) = u (6) ;

%{
F(4) = (qom) * ( Ex + u(5)*Bz - u(6)*By) ;
F(5) = (qom) * ( Ey - u(4)*Bz + u(6)*Bx) ;
F(6) = (qom) * ( Ez + u(4)*By - u(5)*Bx) ;
%}


% for Q1 part4 gravitation y direction
F(4) = ((qom) * (u(5)*Bz - u(6)*By));
F(5) = ((qom) * (-u(4)*Bz + u(6)*Bx)) + g;
F(6) = ((qom) * (u(4)*By - u(5)*Bx));

end
```

Q2) Compute with your matlab code, the cyclotron frequency and the cyclotron radius for – an electron in the Earth's ionosphere at 300 km altitude, where the magnetic flux density B~.00005 Tesla, considering that the electron moves at the thermal velocity (kT/m), with T=1000 K, where "k" is Boltzmann's constant. Plot a graph to show the motion/results and compare your results with analytical calculations.

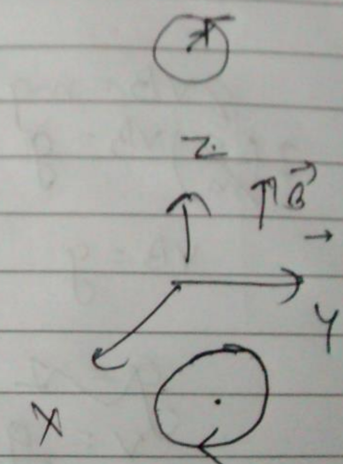
The magnetic flux density at a particular point equals the magnetic field at that point.
Now given,
Magnetic field (B) = 0.00005 Tesla,
Temperature (T) = 1000 K
Mass of electron (m) = 9.1e-31 Kg.
Boltzmann constant (k) = 1.38e-23 J/K

We can find the rms velocity of the electron by the formula

V = sqrt(kT/m);

It comes out to be 123.145e3 m/s.
We can easily calculate the cyclotron radius and the cyclotron frequency by the formula as stated below:

Cyclotron freq = gyro freq.



$$qvB = \frac{mv^2}{r}$$

$$qB = \frac{mv}{r}$$

$$\boxed{r = \frac{mv}{qB}}$$

$$\vec{v} = \vec{\omega} \times \vec{r}$$

$$v = r\omega \Rightarrow \omega = \frac{v}{r} = \frac{qB}{m}.$$

$$\boxed{f = \frac{qB}{2\pi m}}$$

Analytically,

$$r = \frac{mv}{qB} = \frac{9.1\times10^{-31} \times 123.145\times10^{3}}{1.6\times10^{-19} \times 5\times10^{-5}}$$

$$= 14\times10^{-3}\,m$$

$$= \underline{14\,mm}.$$

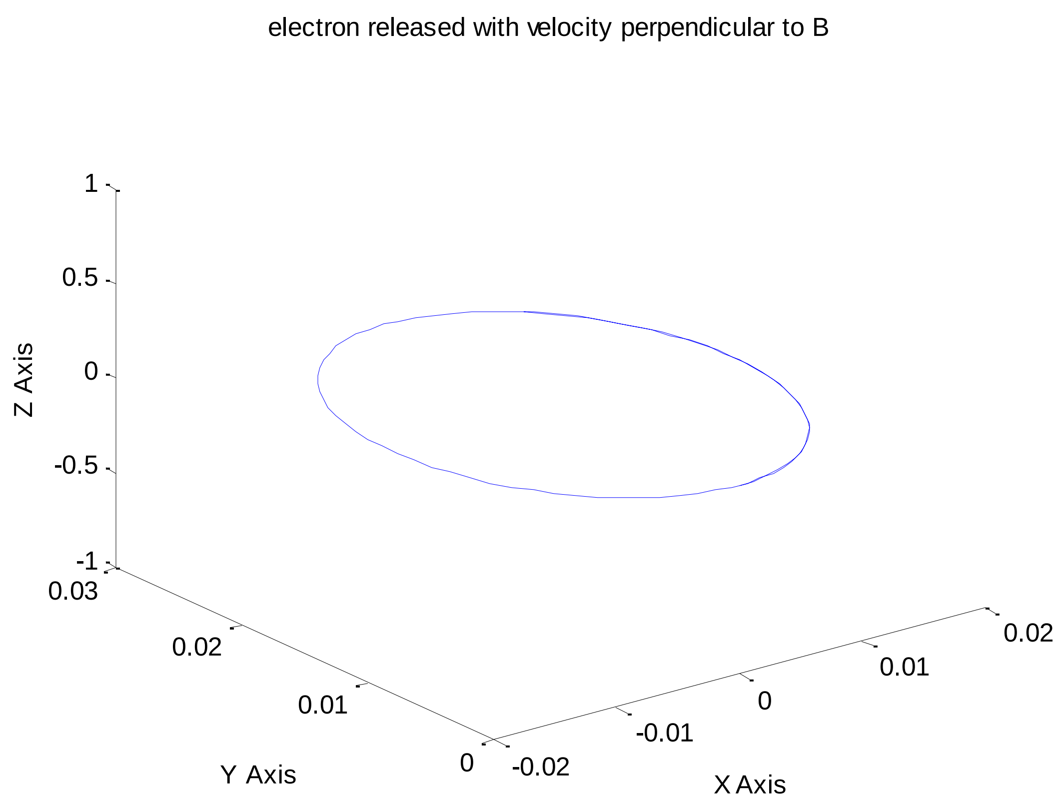
$$f = \frac{qB}{2\pi m} = \frac{1.6\times10^{-19} \times 5\times10^{-5}}{2\pi \times 9.1\times10^{-31}} \Rightarrow T = 7.147\times10^{-7}$$

$$= 0.7147\times10^{-6}\,s.$$

$$= 1.399\times10^{6}\,Hz.$$

Firstly we will neglect the effect of gravity because the gravitational force is very small as compared to the magnetic force mainly due to the small order of magnitude of the mass of the electron. This will become clear in Q3 where we calculate the drift velocity.

In order to calculate the frequency computationally, we assume that the electron is released with an initial velocity perpendicular to the magnetic field. In this case it performs uniform circular motion. So we calculate the time it takes to perform one revolution(by using distance formula) and that gives us the time period. Then we can calculate the frequency. Now calculating the radius is easy. We can find the displacement(not the distance) of the electron in half of the time period. That gives us the diameter and eventually the radius.



The results are as follows:

frequency_computattional =

1.3889e+006

frequency_analytical =

1.3992e+006

radius_computattional =

0.0140

radius_analytical =

0.0140

We can observe that the computational and analytical results match.

Matlab code:

1. Q2.m

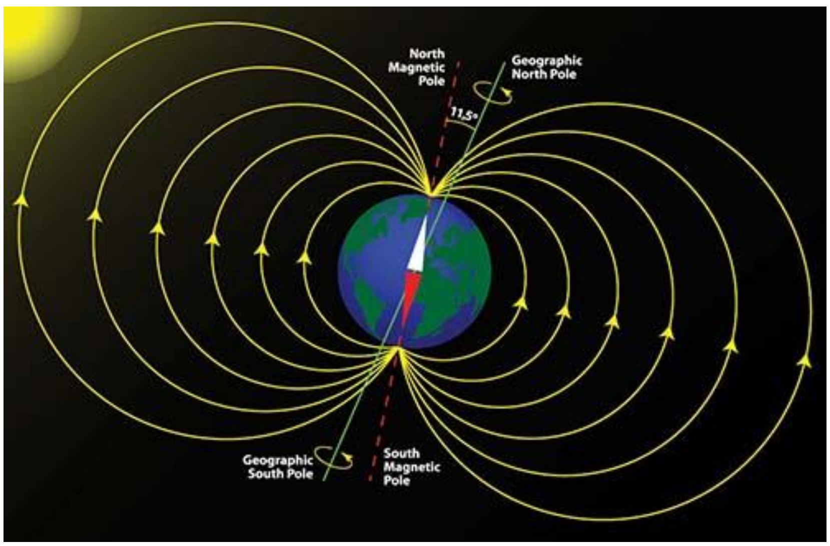
```
clear all;
close all;

global q m qom g;
global Bx By Bz;

q = -1.6e-19;
m = 9.1e-31;

g=9.8;
qom = (q/m) ;

Bx = 0;
By = 0;
Bz = 5e-5;

temp = 1000 ; %in kelvin
k = 1.38e-23 ;

initial_vel = sqrt(k*temp/m);

starttime = 0;
totaltime = 1e-6;
dt = (totaltime-starttime) / 100;

u0 = zeros (6 , 1);

u0(1) = 0 ;           %x
u0(2) = 0 ;           %y
u0(3) = 0 ;           %z

u0(4) = initial_vel ;          %vx
```

```
u0(5) = 0 ;          %vy
u0(6) = 0 ;          %vz

options=odeset('RelTol',1e-5);
[t,u] = ode45(@Cyclotron , [starttime : dt : totaltime] , u0, options) ;

% u contains the position vectors and the velocity vectors at each point.

count = 0;  % counts the number of times (0,0,0) is reached. we will break at
2

index = 1;

%this is for calculating the time period and frequency
while count<2
    if sqrt((u0(1)-u(index,1))*(u0(1)-u(index,1)) + (u0(2)-
u(index,2))*(u0(2)-u(index,2)) + (u0(3)-u(index,3))*(u0(3)-u(index,3))) <=
0.0006
        count = count + 1;
    end
    index = index + 1;
end


% here index applies to time array also

frequency_computattional = (1 / t(index))
frequency_analytical = (q * Bz)/( 2 * pi * m)
% now we have the time period . so let us run for half the time period and
calculate the distance from y axis . that will be the diameter

index_for_half_time = round(index / 2);

diameter = sqrt( power( u0(1) - u(index_for_half_time,1) , 2 ) + power( u0(2)
- u(index_for_half_time,2) , 2 ) + power( u0(3) - u(index_for_half_time,3) ,
2 ) );

radius_computattional = diameter /2

%analytical  solution

radius_analytical  = (m * initial_vel)/(q * Bz)


plot3( u(:,1) , u(:,2) , u(:,3) )
xlabel('X Axis');
ylabel('Y Axis');
zlabel('Z Axis');
title('electron released with velocity perpendicular to B')
```

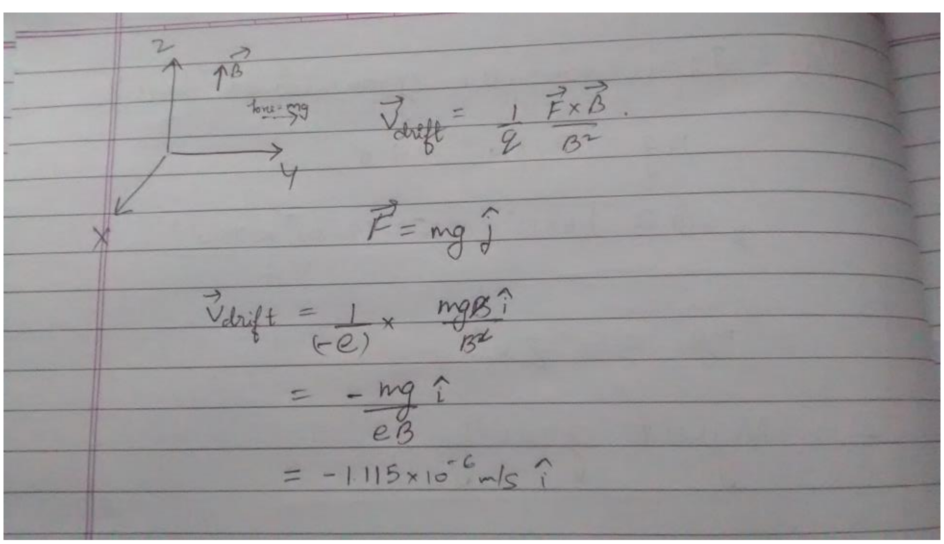
2. Cyclotron.m

```
function F  = Cyclotron( t , u )

%LORENTZFORCEFUNCTION Summary of this function goes here
%   Detailed explanation goes here

% u
% x y z
% vx vy vz

% F
% vx vy vz
% dvx/dt dvy/dt dvz/dt

global q m qom g;
global Bx By Bz;



F = zeros ( length(u) , 1);

F(1) = u (4) ;
F(2) = u (5) ;
F(3) = u (6) ;

%{
F(4) = (qom) * ( u(5)*Bz - u(6)*By) ;
F(5) = (qom) * ( - u(4)*Bz + u(6)*Bx);
F(6) = (qom) * ( u(4)*By - u(5)*Bx) ;
%}

% for Q1 part4 gravitation y direction

F(4) = ((qom) * (u(5)*Bz - u(6)*By));
F(5) = ((qom) * (-u(4)*Bz + u(6)*Bx)) ;%+ g ;
F(6) = ((qom) * (u(4)*By - u(5)*Bx));

end
```

3. What will be the gravitational drift velocity “vg” in the above case? Compare your computational result with theoretical result.



The drift velocity is given by the following formula:



$$\vec{V}_{drift} = \frac{1}{q} \frac{\vec{F} \times \vec{B}}{B^2}.$$

$$\vec{F} = mg\,\hat{j}$$

$$\vec{V}_{drift} = \frac{1}{(-e)} \times \frac{mgB\,\hat{i}}{B^2}$$

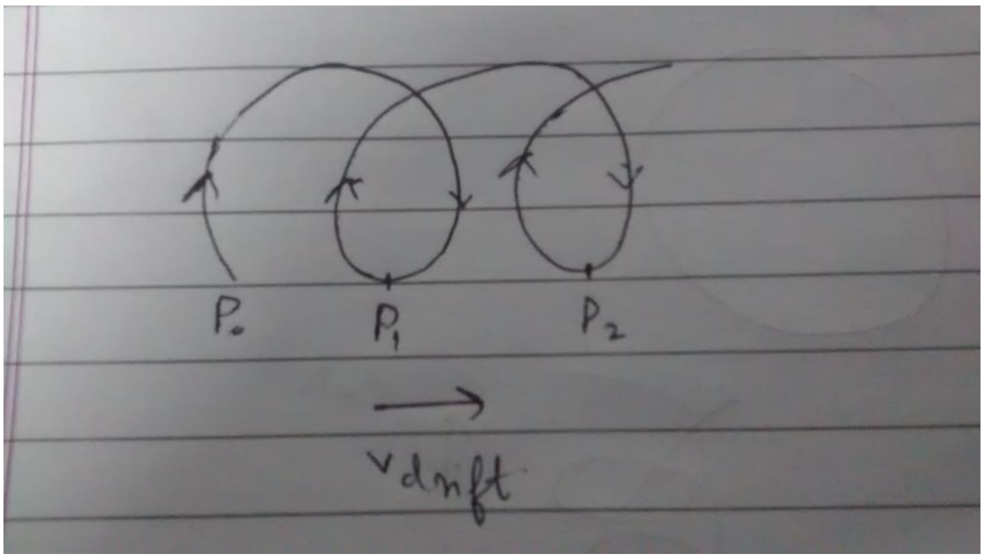
$$= \frac{-mg}{eB}\,\hat{i}$$

$$= -1115 \times 10^{-6}\ m/s\ \hat{i}$$

We can see that the drift velocity has a direction which is the same as cross product of F and B. We can assume that the experiment is being conducted at the earth's equator so that the gravitational force and the magnetic field are perpendicular to each other.

Analytically we get the drift velocity to be around **1.115e-6 m/s**.

Computationally we can find the drift velocity by finding the time taken by some point (say P) to travel and reach positions P1, P2 ,etc. It will be the same as the drift velocity.



Computationally its very difficult to get the drift velocity because it has an order of magnitude of 10^-6 m/s and the time period for 1 circular motion is also 10^-6 s. So the center travels only 10^-6 m in 1 sec. If we run the code for less time (<1e-4) than the distance travelled by the center is so negligible (<1e-10) that it can't be measured. If instead we run the code for more number of iterations, then the time taken comes in hours which is practically impossible.

I tried optimizing the program but the drift velocity came out to be around 0.020m/s and it won't go less than it.

Matlab Code :

1. Q3.m

```
clear all;
close all;

global q m qom g;
global Bx By Bz;
global base_point_x_coordinate ;
global measurement_point_time ;

base_point_x_coordinate = 0;
measurement_point_time = 1;
q = -1.6e-19;
m = 9.1e-31;
```

```
g=9.8;
qom = (q/m) ;

Bx = 0;
By = 0;
Bz = 5e-5;

temp = 1000 ; %in kelvin
k = 1.38e-23 ;

initial_vel = sqrt(k*temp/m);

starttime = 0;
totaltime = 1e-4;
dt = (totaltime-starttime) / 10000000;
npoints = totaltime/dt

u0 = zeros (6 , 1);

u0(1) = 0 ;          %x
u0(2) = 0 ;          %y
u0(3) = 0 ;          %z

u0(4) = initial_vel ;          %vx
u0(5) = 0 ;          %vy
u0(6) = 0 ;          %vz

tic
options=odeset('RelTol',1e-11);
[t,u] = ode45(@Q3Function , [starttime : dt : totaltime] , u0, options) ;

% u contains the position vectors and the velocity vectors at each point.

count = 0;  % counts the number of times (0,0,0) is reached. we will break at
2

index = 1;

% now let us find the drift velocity.- its in the -ve x-direction

%{
while index<=npoints
   if(u(index,2) < 1e-11)
        base_point_x_coordinate = u(index , 1);
        measurement_point_index = index ;
   end
   index = index + 1;
end
%}

toc

base_point_x_coordinate
```

```
drift_velocity = base_point_x_coordinate / measurement_point_time

plot3( u(:,1) , u(:,2) , u(:,3) )
xlabel('X Axis');
ylabel('Y Axis');
zlabel('Z Axis');
title('electron released with velocity perpendicular to B')
```

2. Q3Function.m

```
function F  = Q3Function( t , u )

% u
% x y z
% vx vy vz

% F
% vx vy vz
% dvx/dt dvy/dt dvz/dt

global q m qom g;
global Bx By Bz;
global base_point_x_coordinate ;
global measurement_point_time ;



F = zeros ( length(u) , 1);

F(1) = u (4) ;
F(2) = u (5) ;
F(3) = u (6) ;

%{
F(4) = (qom) * ( u(5)*Bz - u(6)*By) ;
F(5) = (qom) * ( - u(4)*Bz + u(6)*Bx);
F(6) = (qom) * ( u(4)*By - u(5)*Bx) ;
%}

% for Q1 part4 gravitation y direction

F(4) = ((qom) * (u(5)*Bz - u(6)*By));
F(5) = ((qom) * (-u(4)*Bz + u(6)*Bx)) + g ;
F(6) = ((qom) * (u(4)*By - u(5)*Bx));

if(u(2) < 1e-11)
        base_point_x_coordinate = u(1);
        measurement_point_time = t ;
end
```

```
end
```