Lab-VII

Devansh Shukla
I18PH021

1 Aim

Write and execute an octave program to simulate/solve motion of a particle in an EM field.

2 Theory

3 Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MotionInEM
%
% Program to solve/simulate the motion of a particle in electromagnetic field
% Author: Devansh Shukla I18PH021
% 9th March, 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


graphics_toolkit gnuplot
pkg load symbolic

% function to compute accelerations
function retval = F(V, E, B, q, m)
   ax = (E(1) + V(2)*B(3) - V(3)*B(2)) * q/m;
   ay = (E(2) + V(3)*B(1) - V(1)*B(3)) * q/m;
   az = (E(3) + V(1)*B(2) - V(2)*B(1)) * q/m;
   retval = [ax, ay, az];
endfunction

% function to compute positions via RK45 method
function retval = Compute(t0, tf, steps, X, V, E, B, q, m)
   increment = (tf - t0) / steps;
   retval = [X];
   for i=t0:increment:tf
      k1 = F(V, E, B, q, m);

      k2 = F(V + increment*k1/2, E, B, q, m);
      k3 = F(V + increment*k2/2, E, B, q, m);
      k4 = F(V + increment*k3, E, B, q, m);

      X = X + increment*V + (increment**2)*(k1 + k2 + k3)/6;
      V = V + (k1 + 2*k2 + 2*k3 + k4)*increment / 6;
      retval = [retval; X];
   endfor
endfunction

printf("Solving equation of motions using RK45 method")

% Initial conditions
B = [0 0 1]
E = [0 0 1]
pos = [1 0 0]
V = [0 1 0.1]
q = 1
m = 1
t0 = 0
tf = 40
steps = 1000;

% calling the compute function to compute positions
X = Compute(t0, tf, steps, pos, V, E, B, q, m);

% plotting the trajectory
figure()
set(gca,'XMinorTick','on','YMinorTick','on');
plot3(X(:, 1), X(:, 2), X(:, 3), "linewidth", 2);
xlabel("X");
ylabel("Y");
zlabel("Z");
title("Trajectory");
set(gcf, 'renderer', 'painters');
print("-dpng", "motion_in_em_traj.png");

% Computing current density
% set the symbolic vars
syms x y z pi;
% set the symbolic funcs
syms B_x(x, y, z) B_y(x, y, z) B_z(x, y, z);

% magnetic filed
B = [B_x; B_y; B_z]
% Magnetic current density
J = curl(B) * 1e7/ 4*pi
```

4 Results

4.1 Terminal output

```
(escape) devansh@ds:~/GitHub/Vault/OctaveLab/Programs/outputs$ octave ../RungaKuttaMotionInEM.m
Solving equation of motions using RK45 methodB =

   0   0   1

E =

   0.10000   0.10000   0.00000

pos =

   1   0   0

V =

   0.00000   1.00000   0.10000

q =  1
m =  1
t0 = 0
tf =  40
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
B = (sym 3×1 matrix)

  ⎡Bₓ(x, y, z) ⎤
  ⎢            ⎥
  ⎢B_y(x, y, z)⎥
  ⎢            ⎥
  ⎣B_z(x, y, z)⎦

J = (sym 3×1 matrix)

  ⎡   ⎛          ∂                          ∂                 ⎞⎤
  ⎢π⋅⎜- 2500000⋅──(B_y(x, y, z)) + 2500000⋅──(B_z(x, y, z))⎟⎥
  ⎢   ⎝          ∂z                         ∂y                ⎠⎥
  ⎢                                                             ⎥
  ⎢  ⎛         ∂                         ∂                ⎞   ⎥
  ⎢ π⋅⎜2500000⋅──(Bₓ(x, y, z)) - 2500000⋅──(B_z(x, y, z))⎟   ⎥
  ⎢  ⎝         ∂z                        ∂x               ⎠   ⎥
  ⎢                                                             ⎥
  ⎢   ⎛          ∂                         ∂                ⎞ ⎥
  ⎢π⋅⎜- 2500000⋅──(Bₓ(x, y, z)) + 2500000⋅──(B_y(x, y, z))⎟ ⎥
  ⎣   ⎝          ∂y                        ∂x               ⎠ ⎦
```

4.2 Plots

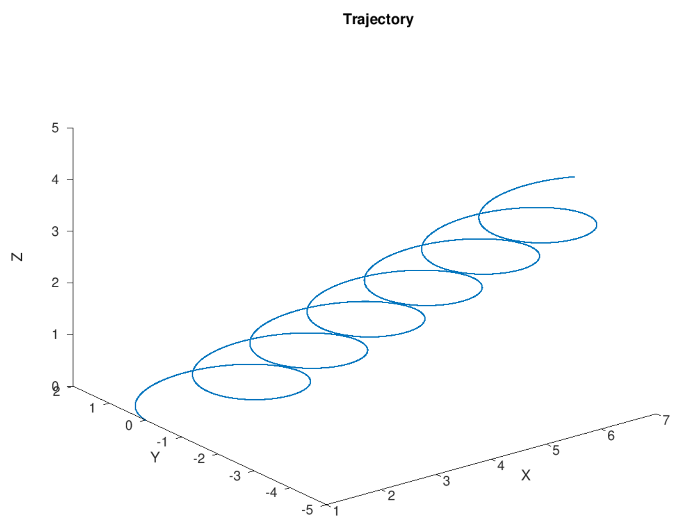

Figure 1: $B = \hat{k}$; $E = 0.1\hat{i}\ +\ 0.1\hat{j}$; $v = \hat{j}\ +\ 0.1\hat{k}$; $r = \hat{i}$; $q = 1C$; $m = 1kg$

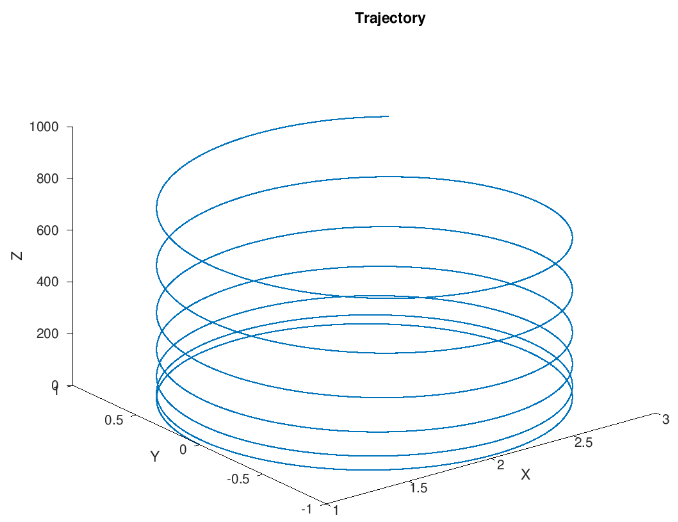

Figure 2: $B = \hat{k}$; $E = 1\hat{k}$; $v = \hat{j} + 0.1\hat{k}$; $r = \hat{i}$; $q = 1C$; $m = 1kg$

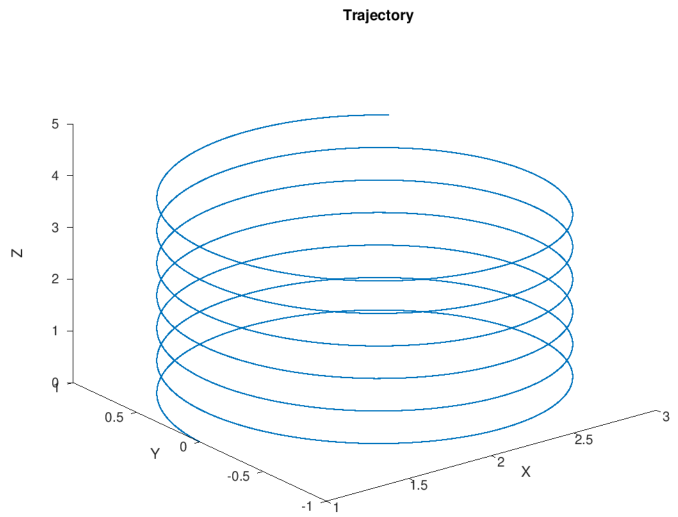

Figure 3: $B = \hat{k}$; $E = 0$; $v = \hat{j} + 0.1\hat{k}$; $r = \hat{i}$; $q = 1C$; $m = 1kg$

5 Remarks

The programs can be used to trace and simulate the motion of any particle in an electromagnetic field by defining the required parameters.