Study the motion of a charged particle in Uniform Electric Field, Uniform Magnetic Field, and in combined Electric and Magnetic Field

1 Introduction

1. **Electric Field**: It is an electric property associated with each point in space in the presence of a charge distribution. It is defined as the electric force acting per unit charge. It is a physical field with SI unit N/C.

2. **Magnetic Field**: It is a physical field that arises due to current distribution, i.e. due to changes in relative motion from the observation point. Its SI unit is Tesla (T).

2 Parameters Required for the C++ Code

To simulate the motion of a charged particle in the presence of electric and magnetic fields, we need the following parameters:

Parameters Input by User	**Symbol**
Charge on the particle	q
Mass of the particle	m
x-, y- and z- components of velocity	v_x , v_y and v_z
x-, y- and z- components of electric field	E_x , E_y and E_z
x-, y- and z- components of magnetic field	B_x , B_y and B_z

Table 1

Step size (h) and total number of iterations (n) are pre-defined parameters in this problem.

3 Motion of a charged particle in Uniform Electric Field

When a particle of charge q and mass m is placed in an electric field E, the electric force exerted on the charge is qE. If this is the only force exerted on the particle, it must be the net force and so must cause the particle to accelerate. In this case, Newton's second law applied to the particle and it gives,

$$F = qE = ma \tag{1}$$

The direction of force is parallel to the field if the charge is +ve and opposite to the field if the charge is –ve. The acceleration of the particle is therefore:

$$a = F/m \implies qE/m \tag{2}$$

If E is uniform (i.e., constant in magnitude and direction), then the acceleration is constant. If the particle has a positive charge, then its acceleration is in the direction of the electric field. If the particle has a negative charge, then its acceleration is in the direction opposite the electric field.
Suppose an electron of charge - e is projected horizontally into this field with an initial velocity v_i. Because the electric field E is in the positive y direction, the acceleration of the electron is in the negative y direction. That is,

$$a = -eE/m \tag{3}$$

Because the acceleration is constant, we can apply the equations of kinematics in two dimensions with

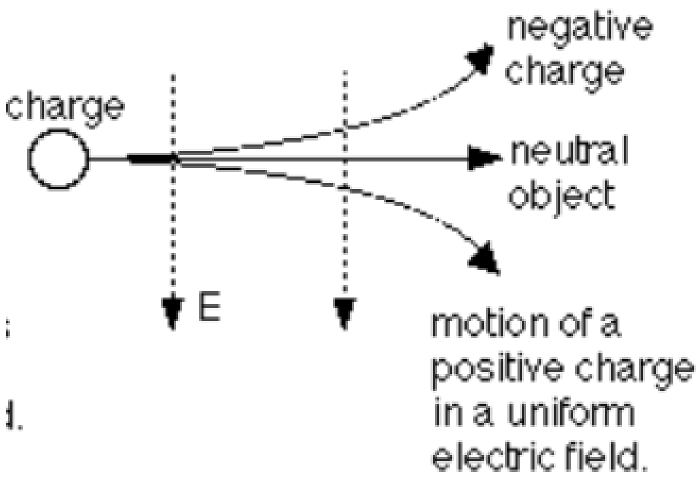
$$v_x i = v_i \tag{4}$$

$$v_y i = 0 \tag{5}$$

After the electron has been in the electric field for a time t, the components of its velocity are

$$v_x = v_i = constant \tag{6}$$

$$v_y = at = -eEt/m \tag{7}$$

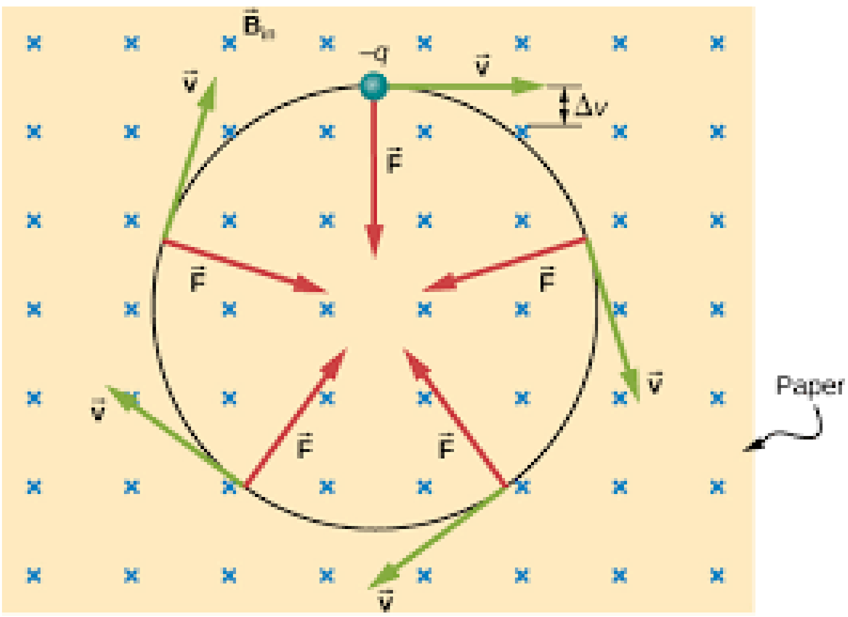


4 Motion of a charged particle in Uniform Magnetic Field

Let us consider a uniform magnetic field of induction B acting along the Z-axis. A particle of charge q and mass m moves in XY plane. At a point P, the velocity of the particle is v. The magnetic Lorentz force on the particle is

$$F = q(\vec{v} \times \vec{B}) \tag{8}$$

Hence, F acts perpendicular to the plane containing v and B. Since the force acts perpendicular to its velocity, the force does not do any work. So, the magnitude of the velocity remains constant and only its direction changes. The force F acting towards the centre acts as the centripetal force and makes the particle to move along a circular path.



If v and B are at right angles to each other

$$q(\vec{v} \times \vec{B}) = Bqvsin(90) \implies Bqv \tag{9}$$

This magnetic Lorentz force provides the necessary centripetal force.

$$Bqv = \frac{mv^2}{r} \tag{10}$$

$$r = mv/Bq \tag{11}$$

It is evident from this equation, that the radius of the circular path is proportional to (i) mass of the particle and (ii) velocity of the particle.

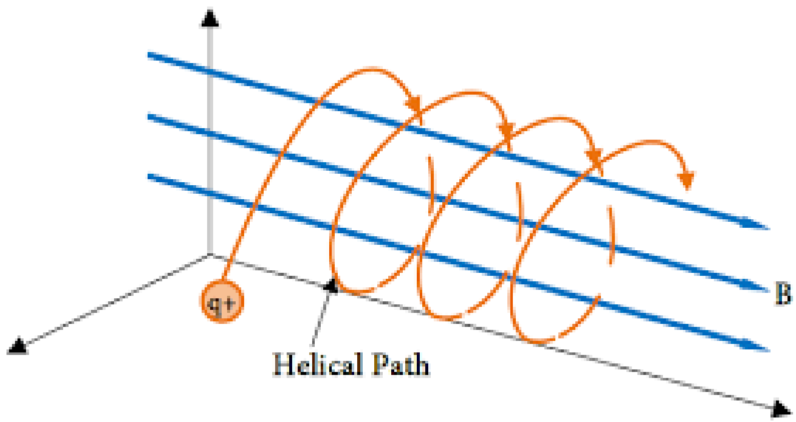
5 Motion of charged particle in electromagnetic field

Consider a particle of mass 'm' and electric charge 'q' moving in a uniform electric and magnetic field, E and B.
Suppose that the fields are crossed(i.e perpendicular to each other) so that E B = 0 .The force acting on the particle is given by the familiar Lorentz law:

$$F = q(E + \vec{v} \times \vec{B}) \tag{12}$$

where v is the particle's instantaneous velocity. Hence from Newton's second law, the particle's equation of motion can be written as

$$m\frac{dv}{dt} = q(E + \vec{v} \times \vec{B})) \tag{13}$$



6 Algorithm

1. **Include all the required header files.**

2. **Define a class 'EM_field'** which has all the above-listed parameters in Table 1 and pre-defined parameters declared as private members.

3. **Declare public member functions** in the class that can be accessed from outside the class using an object or instance of the class defined in the main function (the compiler starts the execution of any C++ program from the main() function)

4. **Definition of Member function 1:** EM_field(){//}
   The constructor initializes the parameters, including mass, charge, the velocity of the charged particle in x, y, and z directions, the electric field in x, y, and z directions, and the magnetic field in x, y, and z directions

   - Return-type: none
     A constructor is a special member function having the same name as the class and it gets invoked automatically as the instance/object of the class is declared.
   - The function doesn't accept any parameters.

5. **Definition of Member Function 2:** double AX (double, double) {//}
This function calculates the acceleration of the charged particle in the x-direction.
From eq.(13), if the cross-product is opened then for x-direction acceleration we get,

$$a_x = q * (E_x + (v_y * B_z - v_z * B_y))/m \tag{14}$$

   - Return type: `double`
     The function returns the calculated value of the particle's acceleration in the x-direction depending on the case if the particle is moving in the presence of an electric, magnetic, or electromagnetic field.
   - The function accepts two parameters
     - v_y: The particle's velocity in the y-direction at the current time-step.
     - v_z: The particle's velocity in the z-direction at the current time-step.

6. **Definition of Member Function 3:** double AY (double, double) {//}
This function calculates the acceleration of the charged particle in the y-direction.
From eq.(13), if the cross-product is opened then for y-direction acceleration we get,

$$a_y = q * (E_y + (v_z * B_x - v_x * B_z))/m \tag{15}$$

   - Return type: `double`
     The function returns the calculated value of the particle's acceleration in the y-direction depending on the case if the particle is moving in the presence of an electric, magnetic, or electromagnetic field.
   - The function accepts two parameters
     - v_z: The particle's velocity in the z-direction at the current time-step.
     - v_x: The particle's velocity in the x-direction at the current time-step.

7. **Definition of Member Function 3:** double AZ (double, double) {//}
This function calculates the acceleration of the charged particle in the z-direction.
From eq.(13), if the cross-product is opened then for y-direction acceleration we get,

$$a_z = q * (E_z + (v_x * B_y - v_y * B_x))/m \tag{16}$$

   - Return type: `double`
     The function returns the calculated value of the particle's acceleration in the z-direction depending on the case if the particle is moving in the presence of an electric, magnetic, or electromagnetic field.
   - The function accepts two parameters
     - v_x: The particle's velocity in the x-direction at the current time-step.
     - v_y: The particle's velocity in the y-direction at the current time-step.

8. **Definition of Member Function 4:** void calc (void) {//}
This function calculates the charged particle's position in x, y, and z-directions for different time steps via the Euler Method.
   - Return data-type: void This function doesn't return any value.
   - The function doesn't accept any parameter
   - The function performs important computations required for the final result:
     - Initializes the value of step size (h), the number of iterations, and, the particle's position in x, y & z-directions.
     - A for-loop is used to implement the Euler method for the calculation of the particle's velocity and position. Also, the function calls to AX (double, double), AY (double, double), and AZ (double, double) are made by this function. The loop gets terminated depending on the value of the total number of iterations.
     - The function finally creates a .dat data file and enters the data computed to this file, which includes time step and displacement.

9. **Define the main function:** `int main()`
   - Creates an instance/object of the class 'EM_field' and makes function calls to all necessary functions.
   - Return 0 to indicate successful execution.
10. The final step in the algorithm involves a crucial step i.e., plotting the data saved in the file by the calc() function using **Gnuplot software**.
    Gnuplot is a graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms. Its source code is freely distributed and is extensively used for data visualization.
    Follow the following steps to plot data on Gnuplot:
    (a) Download Gnuplot.
    (b) Go to → Change directory.
    Select the folder where the 'charge.dat' data file is stored.
    (c) Write the following command in Gnuplot:

```
plot "charge.dat" u 1:2 w l
\* This command asks the Gnuplot to plot Column 2 with respect to Column 1
    with a line-type graph. There are many other variations to this command,
    you can check them out at the official website of Gnuplot or watch some
    free tutorials on YouTube.*/
```