Project Report – (GROUP10)

Made by-

Yash Tiwari – EP19BTECH11025

Abhinav Poddar – EP19BTECH11023

Raj Nikesh Vardhan - EP19BTECH11009

Title: Simulation of particle in Electric and Magnetic field in Python

In this project we made a python code that can almost simulate trajectory of point particles in generalized Magnetic and Electric Field in Non-Relativistic and Relativistic cases. We have covered every variation of E and B. But you can do most of the simulation in this application

Motivation:

- In Relativistic Mechanics we were finding it difficult to understand the motion of particle in EM field.
- To get a deeper understanding of motion of particle in EM field where both E & B can vary w.r.t 3D spatial coordinates as well as time.
- It is very difficult to solve equations to find the relation between position and time.

  $\frac{dv_x}{dt} = \frac{q}{m}\left(v_y B_z - v_z B_y + E_x\right)$ is case dependent and will be difficult to solve.

Challenges faced:

- How to avoid integration? (Computationally Expensive)
- Making the simulator user friendly
- Testing (Checking Correctness)
- Finding environment suitable for simulation and visualization
- Learning how to work on these environments
- Finding the correct math through research papers and books

Working of the project:

- The Project is made in python with extensive use of the following libraries from it
  - NumPy
  - Tkinter
  - Equations
  - Vpython

Physics behind the Project:

As we know a particle is in E and B it experiences Force which is given by Lorentz formula which is

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

If E and B are the function of x, y, z, t integration becomes quite difficult and also integration is computationally expensive so to avoid that we assume differential of t (dt) to be very small constant (appx. 10^-6), so that we can approx. change in quantities such as B, E and v & (x, y, z) without using integration. trajectory of the particle. By these relations

For Nonrelativistic:

For small dt:

$$(v_x)_t = (v_x)_{t-dt} + q/m\left(v_y B_z - v_z B_y + E_x\right)dt$$

Similarly,

$$x_t = x_{t-dt} + (v_x)_t dt$$

Similarly, for y and z also.

For Relativistic part:

Constant E in x direction:

$$(v_x)_t = (v_x)_{t-dt} + \frac{q}{mc\gamma^3}(E_x)_{t-dt}dt$$

$$x = (v_x)_t dt$$

When in B in z direction and constant:

$$(v_x)_t = (v_x)_{t-dt} + \frac{q}{mc}\left(\frac{B_z v_y}{\gamma^2}\right) dt$$

$$(v_y)_t = (v_y)_{t-dt} - \frac{q}{mc}\left(\frac{B_z v_x}{\gamma^2}\right) dt$$

$$dx = v_x dt$$

$$\gamma = \frac{1}{\sqrt{1 - \frac{mag(v)^2}{c^2}}}$$

Where ,

$$mag(v) = \sqrt{{v_x}^2 + {v_y}^2}$$

The most important factor in this calculation is the dt part as the smaller it becomes the more accurate the simulation becomes.

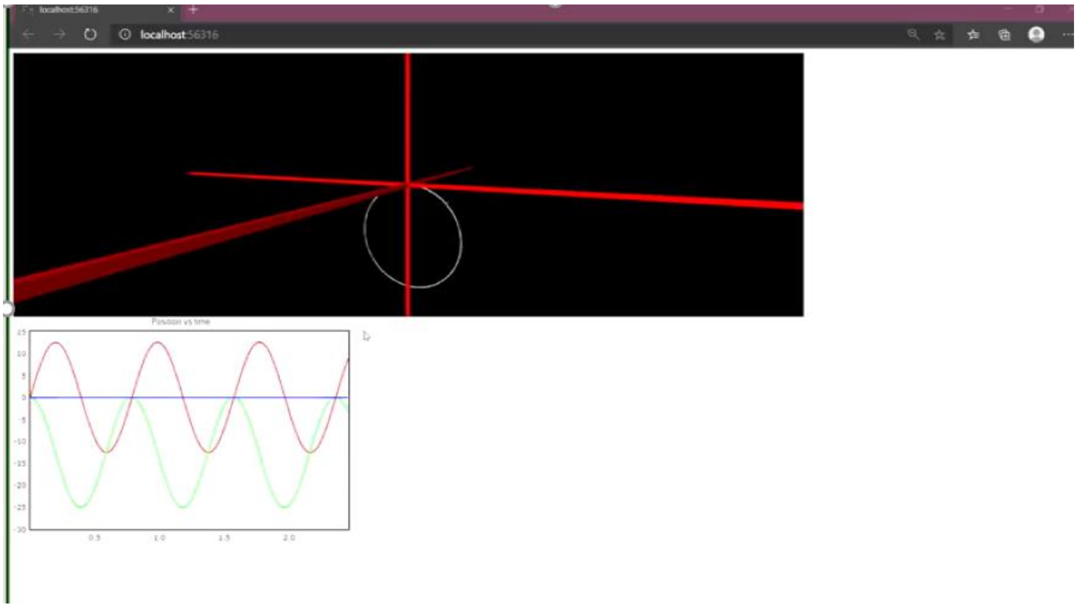
In relativistic part we used the field tensor from references to find the trajectory relation of the particle.

Testing:

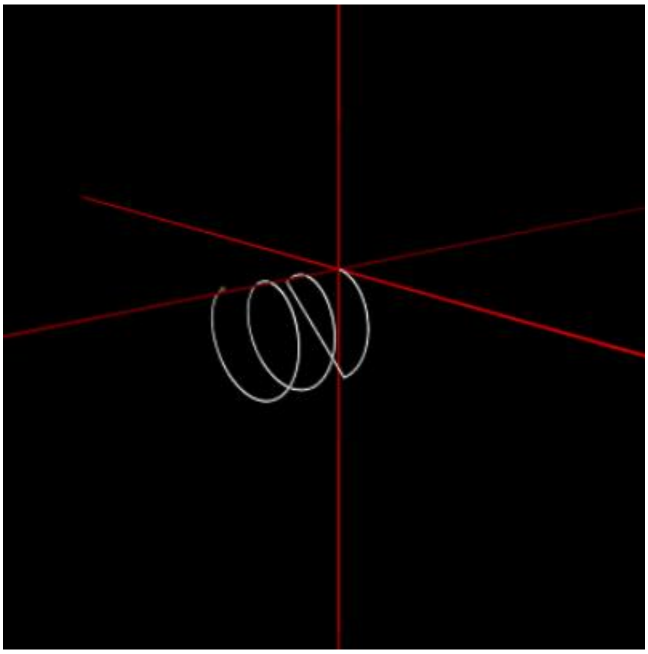
This was the most difficult part as the we cannot say what is trajectory of the particle, but we know few cases which are easy to solve.

Example

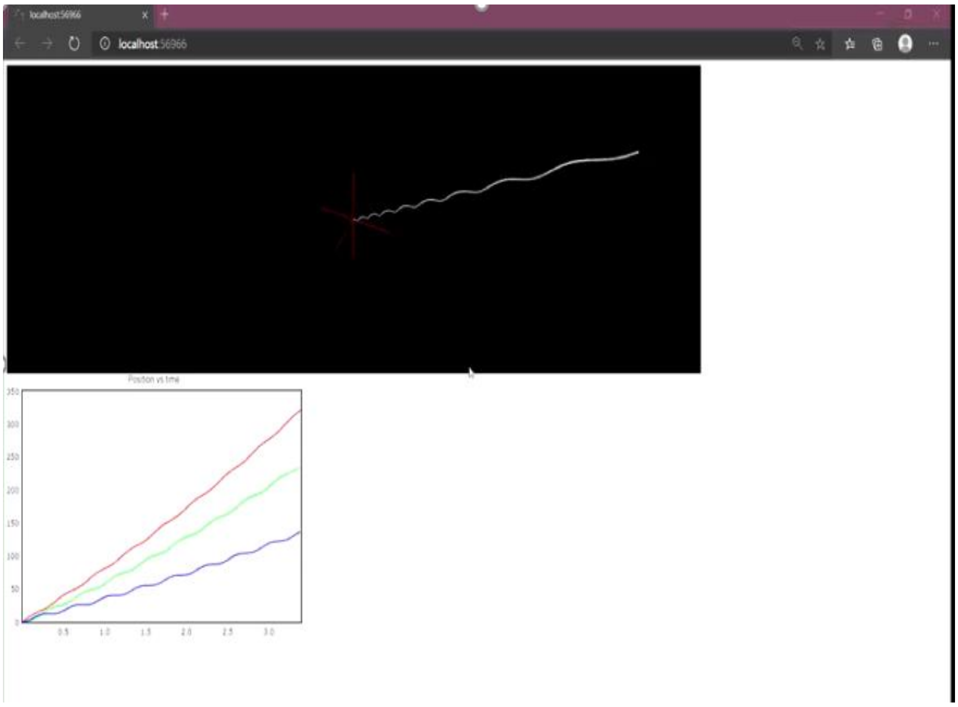
1. Particle moves the circular path when B is applied perpendicular velocity



2. Particle moves in helical path when B is applied.



3. Electric Field and magnetic field in all directions is applied on particle.



Tutorial: How to work with the project?

- First run the program a window will appear as shown in fig.
- Enter the values of q and m.
- Initially values of B, E & V is set to 0.
- Choose non-relativistic or relativistic motion.
- In case of non- relativistic enter the values of B and E.
- Enter velocity in all direction.
- Click on confirm values.
- Then click simulate.
- In case of relativistic choose one of the two environments (due to calculation complexity we could not add more options).
- Choose the values click on confirm and the simulate.
- Please refrain from choosing one option and then going back and choosing other as it may lead unexpected behavior of simulation.

Tutorial: how to input values in the input window?

- Click on the buttons to input the values.
- For e.g.: if you want to input 1+2 first click on “1” then “+” then “2”.
- i.e. press the buttons in a series order to input the same way one will write in paper.
- To put trigonometric inputs please remember to press the “)” button after inserting your data E.g.: to input sin(x) press “Sin” then “x” then “)”.
- Please refrain from directly inputting the data in field using keyboard as it can cause syntax error.
- **<u>IMPORTATNT!!!:</u>** For relativistic case please input big values (> 10^11) for E, B and (comparable to speed of light) for V

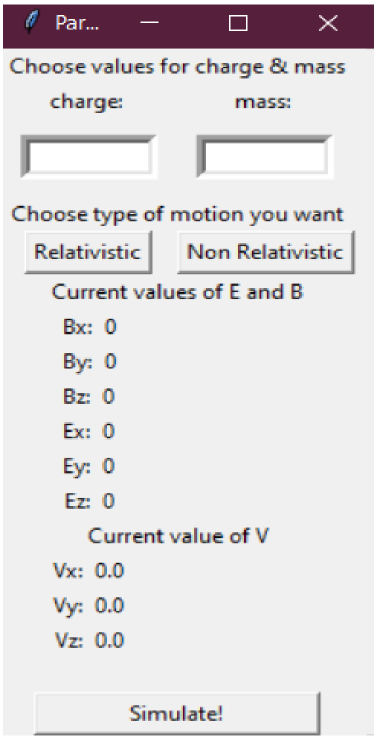


Fig 1 – main window of simulator

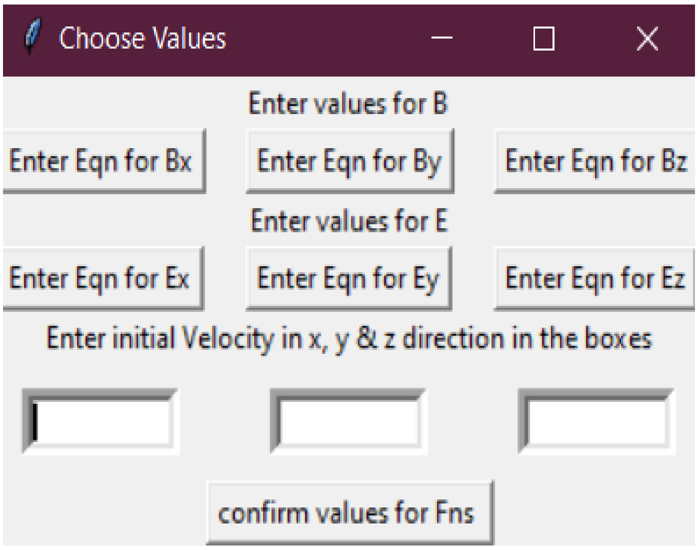


Fig 2 – window to choose values B, E, v for non-relativistic

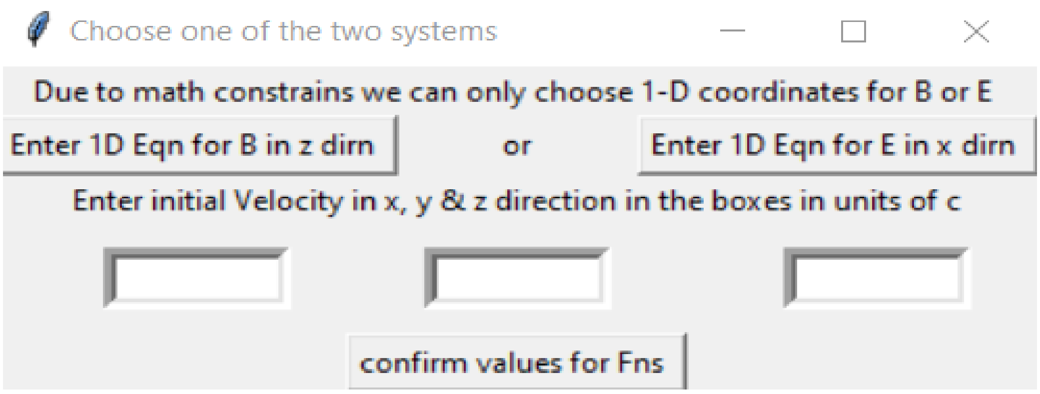


Fig 3 – window to choose values B(z) / E(x), v for relativistic

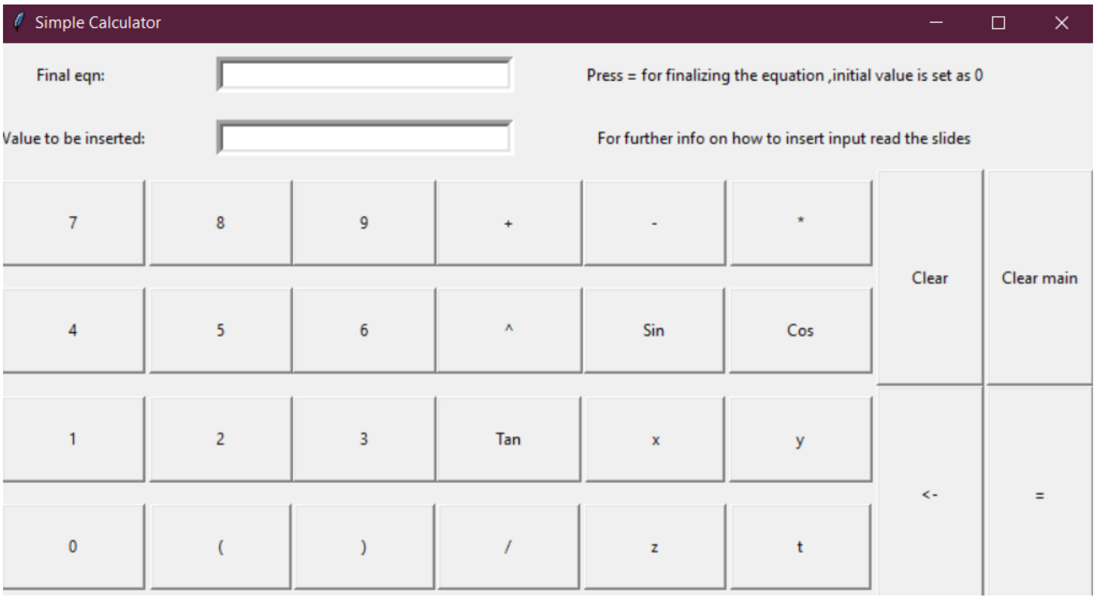


Fig 4 – window to input value of force components

Shortcomings and Further scope:

- Using large values for B and E will lead less to accurate simulation (in case of non- relativistic) due to computer limitations.
- Increasing accuracy (decreasing value of dt) will slow the simulation and will take hours to complete.
- Not all functions are available as input (log, factorial etc.).
- Testing is still in progress.
- 2-D & 3-D relativistic motion is not provided (will require quite significant research to make this possible).
- This simulator can be easily expanded to simulate various cases like movement of gas particles, behavior of light etc.

Fig 5 – Significant Simulation error observed when dt is 10 ^ (-3)

References:

- **For relativistic:**
  - https://www.researchgate.net/publication/262474836_Relativistic_charged_particle_in_a_uniform_electromagnetic_field/fulltext/03957afe0cf2005ef799b9f8/Relativistic-charged-particle-in-a-uniform-electromagnetic-field.pdf
  - http://www.physics.usu.edu/Wheeler/EM/Notes/EMNotes26MotionOfParticles.pdf
- **For libraries:**
  - https://www.vpython.org , https://www.youtube.com/playlist?list=PLdCdV2GBGyXOnMaPS1BgO7IOU_00ApuMo **(for learning Vpython).**
  - https://docs.python.org/3/library/tkinter.html , https://youtu.be/YXPyB4XeYLA **(learning tkinter)**
  - https://pypi.org/project/Equation/ **(for learning Equation library)**