Lorentz Force Simulator

Chares Moustakas

15/12/2022

1 Introduction

The current software is attempting to simulate the behaviour of a particle near a finite long charged wire. It is important to underline that the aforementioned software is just a fun, non-optimized project and as a consequence the reader can feel free to use, modify, improve and play with it. It is notable to mention that this software was developed in C++ using relatively low level APIs and Libraries such as modern OpenGL, custom fragment-vertex shaders through GLSL, Dear ImGUI and CMake, while all the Objects-meshes were constructed through Blender. I would also like to thank http://www.opengl-tutorial.org/ for their great explanations.

2 Mathematics

The concept of this simulation is to calculate the Lorentz Force that the wire exterts to the spherical particle, using primarily the current that flows through the wire as a datum. To begin with, it is crucial to visualize from the Mathematical perspective the goal, being said that the formula for the precious calculation is:

$$\overrightarrow{\mathrm{F_L}} = q(\overrightarrow{\mathrm{E}} + \overrightarrow{\mathrm{u}} \times \overrightarrow{\mathrm{B}})$$

Where **q** is Coulomb charge of the particle, **E** the electric field, **u** particle's velocity and **B** the magnetic's field flux density.

2.1 B calculation

The calculation of **B** is based on Ampere's formula:

$$\oint_C \overrightarrow{\mathrm{B}}\,\overrightarrow{\mathrm{dl}} = \mu I$$

Due to **B** and **dl** parallelism, the equation above is tranformed as:

$$|\overrightarrow{\mathrm{B}}| \oint_C dl = \mu I$$

$$|\overrightarrow{\mathrm{B}}| 2\pi r = \mu I$$

$$|\overrightarrow{\mathrm{B}}| = \frac{\mu I}{2\pi r}$$

$$\overrightarrow{\mathrm{B}} = \frac{\mu I}{2\pi r}\hat{\phi} \tag{1}$$

Cylindrical Coordinates:

$$\hat{\phi} = -\frac{z}{r}\hat{x} + \frac{x}{r}\hat{z}$$

It is notable that the direction of **B** is tangent to the closed circle-loop C. Consequently **B** calculation is possible if the position of the particle and the current are known magnitudes.

2.2 E calculation

Regarding Electric field calculation, patience is the key... So lets construct the main acceptance, first of all lets assume that the wire has **L** length. An infinitesmal charged part of the wire **dq** that is **y** units away from the end of the cable causes Electric field **dE** for a random point P that it is **r** units away from the wire:

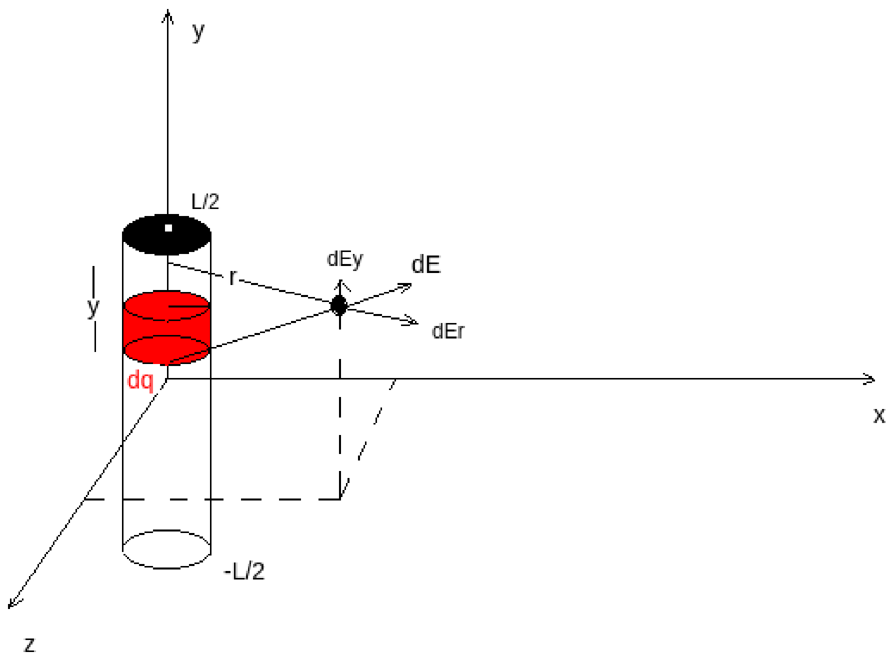

Figure 1: Calculation of Electric field

$$\frac{dq}{dy} = \frac{Q}{L}$$
$$dq = \frac{Q}{L}dy$$

$$\overrightarrow{\mathrm{dE}} = dE_y sin\theta \hat{y} + dE_r cos\theta \hat{r} \tag{2}$$

$$sin\theta = \frac{y}{\sqrt{y^2+r^2}}, cos\theta = \frac{r}{\sqrt{y^2+r^2}} \tag{3}$$

$$dE = k\frac{dq}{y^2+r^2}$$

Based on the above:

$$dE_y = k\frac{Q}{L}\frac{y}{\sqrt{(y^2+r^2)^3}}dy$$

$$\int_0^{E_y} dE_y = k\frac{Q}{L}\int_{-L/2}^{L/2}\frac{y}{\sqrt{(y^2+r^2)^3}}dy = k\frac{Q}{L}(\frac{-1}{\sqrt{(r^2+\frac{L^2}{4})}} + \frac{1}{\sqrt{(r^2+\frac{L^2}{4})}})$$

$$E_y = 0$$

The only non zero coordinate is the coordinate of r unit vector, based on the (1) ! Lets calculate the r coordinate.

$$dE_r = k\frac{Q}{L}\frac{r}{\sqrt{(y^2+r^2)^3}}dy$$

$$\int_0^{E_r} dE_r = k\frac{Q}{L}\int_{-L/2}^{L/2}\frac{r}{\sqrt{(y^2+r^2)^3}}dy = k\frac{Q}{L}\left(\frac{L/2}{r\sqrt{(r^2+\frac{L^2}{4})}}+\frac{L/2}{r\sqrt{(r^2+\frac{L^2}{4})}}\right)$$

$$E_r = k\frac{Q}{r\sqrt{r^2+\frac{L^2}{4}}}$$

Consequently, the expression (1) can be written as:

$$\overrightarrow{\mathrm{E}} = k\frac{Q}{r\sqrt{r^2+\frac{L^2}{4}}}\hat{r} \tag{4}$$

Cylindrical coordinates:

$$\hat{r} = \frac{x}{r}\hat{x} + \frac{z}{r}\hat{z}$$

It is crucial to keep in mind that the Coulomb charge **Q** is not constant in time, therefore on each OpenGL iteration the expression (1) and (4) , concerning the Electric field and Magnetic flux density calculation, is taken into account:

$$I = \frac{dQ}{dt}$$

2.3 Force Calculation

As a result of the above, for known initial velocity it is possible to calculate the Lorentz force for the **q** particle. Consequently, on each OpenGL iteration it is critical to calculate classic motion differential equations for the velocity. So, for the aforementioned known force for a specific timestamp in the timeline $\mathbf{t}=\mathbf{t}_i$:

$$\overrightarrow{\mathrm{F_{Li}}} = m\overrightarrow{\mathrm{a_i}}$$

$$\overrightarrow{\mathrm{F_{Li}}} = m\frac{\overrightarrow{\mathrm{du_i}}}{dt}$$

$$\overrightarrow{\mathrm{F_{Li}}} \approx m\frac{\overrightarrow{\Delta\mathrm{u_i}}}{\Delta t}$$

$$\overrightarrow{\mathrm{u_i}} \approx \frac{\Delta t}{m}\overrightarrow{\mathrm{F_{Li}}} + \overrightarrow{\mathrm{u_{i-1}}}$$

Therefore, in order for the inductive calculation above to be successfull, knowledge of the initial velocity is capable and necessary condition.