

Plan for student Colloquium,

Nikolaj Roager Christensen

February 12, 2022

Contents

1	Introduction, and this document	1
1.1	Topic, and why is this interesting	1
2	Plan	2
3	Theory	2
4	Electric- and magnetic fields and forces	2
4.1	To be done	3
5	The simulation	3
5.1	The differential equation solver	3
5.2	Testing the simulation	4
	References	6

1 Introduction, and this document

In this document, I will go through my plan for this student Colloquium, I will cover, what exactly my topic is, how I plan to structure the time until the trial at March 11. The theoretical background for both my presentation and simulation. Finally I will show the results I have achieved so far.

1.1 Topic, and why is this interesting

I want to talk about physics simulations, in this project I want to talk about how I have simulated (classical) particles subject to electric and magnetic forces, and in particular how this simulation can be used to predict how charged particles move through the Earth's magnetic field.

I largely want to focus on the simulation part of this project, and how the method presented here can be generalized to other systems with particles subject to some forces, for instance charged particles in quasars, or gravitationally bound systems.

2 Plan

At the time of writing, it is Sunday 13th of February, the trial colloquium is on the 11th of March 8:30 to 10:00, this leaves a little less than four weeks.

As of the time of writing, the simulation is largely working, and I can, in principle, plot the results in 2 and 3 dimensions. Though the simulation is still stuck with constant fields. My suggestion for a plan is as follows

1. Week 7: Implement a model for the Earth magnetic field, find sources for missing theory (see below).
2. week 8: Write abstract, write first draft of presentation, finalize simulation results.
3. week 9: Write presentation slides (Using the beamer package), practice.
4. week 10: Practice presentation, trial colloquium Friday morning.
5. week 11: Implement suggestions from trial colloquium, colloquium on Thursday.

I suggest we meet weekly to discuss the progress so far.

3 Theory

4 Electric- and magnetic fields and forces

A particle with charge q moving at velocity \vec{v} through an electric and magnetic field \vec{E} and \vec{B} will experience the Lorentz force (for instance [1] chapter 5.1):

$$\vec{F}_{lor} = q(\vec{E} + \vec{v} \times \vec{B}) \quad (1)$$

If a (classical) particle is subject to no other forces, knowing $\vec{r}(t_0)$, $\vec{v}(t_0)$ and $\vec{E}(\vec{r})$ and $\vec{B}(\vec{r})$. We simply need to solve the 2nd order equation: $m\ddot{\vec{r}} = \vec{F}_{lor}(\vec{r})$.

The \vec{E} and \vec{B} fields are not the only way of characterizing a particular setup, an alternative is using the electric and magnetic vector potentials V, \vec{A} where $\vec{E} = -\vec{\nabla}V$ and $\vec{B} = \vec{\nabla} \times \vec{A}$.

These are arguably easier to work with when calculating the potential from some setup is, but the equation of motion does not become any simpler, and for this reason I will work with the fields rather than potentials in the simulation.

One useful result from the magnetic vector potential is its multipole expansion, as outlined in [1] chapter 5.4, we can write the magnetic potential from a current loop as:

$$\vec{A}(\vec{r}) = \frac{\mu_0 I}{4\pi} \sum_{n=0}^{\infty} \frac{1}{r^{n+1}} \oint (r')^n P_n(\cos(\alpha)) d\vec{l} \quad (2)$$

The exact form of each term is not really important here, the main point is that we have a monopole term r^{-1} , a dipole term r^{-2} , a quadropole term r^{-3} etc. and in the absence of true magnetic monopoles, the monopole term is always zero and at large distances the dipole term dominates, and, for a current loop centered at the origin, we can write the vector potential as:

$$\vec{A}_{dib}(\vec{r}) = \frac{\mu_0}{4\pi} \frac{\vec{m} \times \hat{r}}{r^2} \quad (3)$$

Here \vec{m} is the magnetic dipole term, in principle $\vec{m} = I \int_A d\vec{a}$ for a current I with area A , though in the case of the Earth's magnetic field, we can hardly model it as generated by a simple current loop. However, the Earth magnetic field is still generated by the motion of charges, so a dipole approximation is a passable first approximation.

4.1 To be done

Wikipedia lists a dipole model for Earth magnetic field [2], which I will try implementing, though Wikipedia does not give a lot of sources for this, and it is made clear that the model is wrong at larger distances, due to the sun's own magnetic field a better model is the Tsyganenko model, which incorporates the effects of the sun on the Earth's magnetic field.

I thus far still need to read more into the particles involved in northern light: is it primarily one mass, and what is the distribution of velocities involved.

5 The simulation

The simulation is in essence just two primary classes: a composite (E and B) field class with a function `vec composite_field::get_Efield(vec position)` and `vec composite_field::get_Bfield(vec position)` for getting the field, and a particle class, with a function `void particle::update(const composite_field& Fields, size_t i, double dt)` for updating the velocity and position of at simulation step number i , a finite time time step Δt .

5.1 The differential equation solver

The update function can essentially be any solver for a 2nd order differential equation. The simplest approach, and the one used so far is:

```
void particle::update(const composite_field& Fields, size_t i, double dt)
{
    if (i < positions.size() && i > 0)
    {
        //The force acting on the particle is due to the Lorentz force:
        vec dpdt = charge*(Fields.get_Efield(positions[i]) + glm::cross(
            velocity, Fields.get_Bfield(positions[i])));

        //First order time evolution
        velocity += dt * dpdt * inv_mass;
        positions[i] = positions[i-1] + dt * velocity;
    }
}
```

We essentially let $d\vec{R}/dt \approx \Delta\vec{R}/\Delta t$. This method is the Euler method for solving ordinary differential equation, and it is not great. For one thing this implementation uses a fixed Δt , which needs to be small enough to work everywhere: but suppose we have a particle moving in a field which is only significant close to origin, in this case a Δt which is short enough far from the origin would not work near it, and one which works near the origin is wasteful far from the origin.

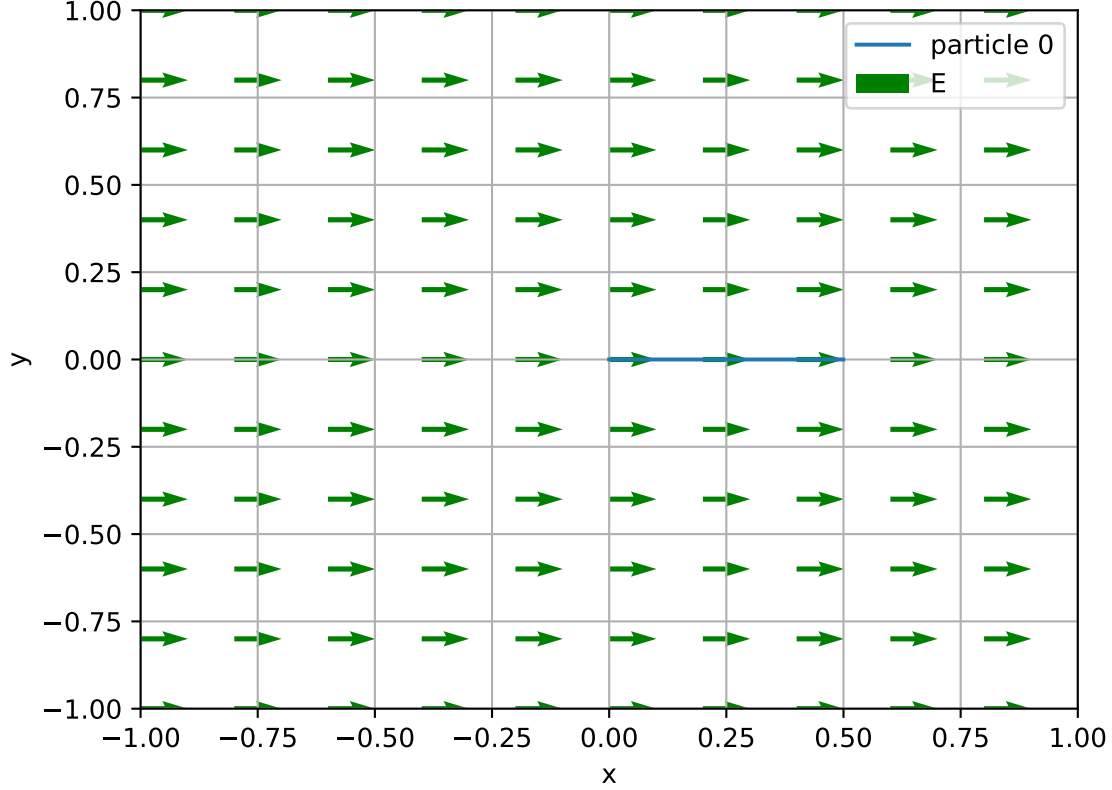


Figure 1: Constant electric field.

A far better option is to use a third party ODE solver library such as `boost::odeint` [3], which implements adaptive step sizes. This also allows us to use higher order solvers, such as the Dorman-Prince algorithm which is used as standard in for instance Matlab's `ode45` function. I have not yet implemented this at the time of writing, but it should simply be a matter of replacing the update function.

5.2 Testing the simulation

Even with the inferior solver, I have implemented three simple test-cases with a particle moving in a constant field:

The main challenge thus far has been finding a way to visualize the simulation results, at this point I simply show the field at select points as a vector plot, and the path of the particle as a line, this is bad, since it is not clear where the particle is when, placing some markers at 1 time unit intervals or something like that would possibly help.

The first case is a particle with mass and charge 1 starting at rest in a constant electric field $\vec{E} = \hat{x}$ and the simulation is run for 1 time units with 1000 time-steps; the units used are whatever is convenient, if we interpret them as SI units (kg, C and V/m), then the particle position is measured in meter and the simulation runs over 1 second. The particle should experience a constant acceleration of 1 length units per time unit squared, and should travel $(1/2)1^2 = 1/2$ length units, The path of the particle is shown on figure 1, and as we see the particle travels exactly as far as expected.

The second case is the same particle, starting at position $(0,1,0)$ with velocity $\vec{v} = \hat{x}$

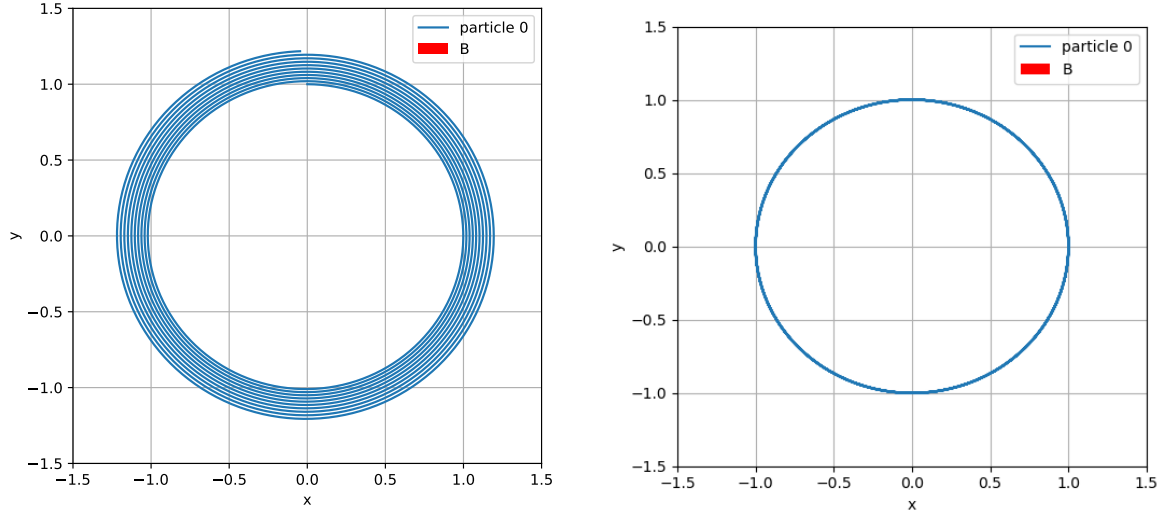


Figure 2: Constant magnetic field in Z direction, around 10 full cycles with too low precision (left) and higher precision (right) the high resolution example has been saved as a png, since the vector based pdf version had too large file size.

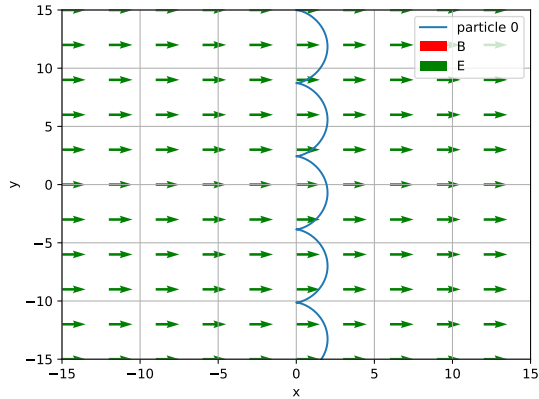


Figure 3: Cycloid motion resulting from a combined constant electric and magnetic field.

distance units per time subject to a magnetic field $\vec{B} = \hat{z}$, in this case we know ([1] chapter 5.1.2) the particle will perform a cyclical motion relation $qv_oB = mv_o^2/r$ between the orbital velocity and the radius so if I let the simulation run for $6.28 \approx 2\pi$ time units we expect to see a full circle in the XY plane with radius $r = mv/Bq = 1$, and as see non figure 2 this is almost what we see, here I have run the simulation 62.8 time units for roughly 10 full circles with a total of 10000 time steps, and as is evident the radius has slowly increased, increasing to 1000000 steps seems to solve the problem, but this highlights how unreliable this method is.

The final example is the same particle, starting at rest at (0,6,0) in a electric field $\vec{E} = \hat{x}$ and $\vec{B} = \hat{z}$. The result on figure 3 is Cycloid motion, this was discussed for instance in our electrodynamics course, and this shows that the simulation can handle both fields at once.

References

- [1] D. J. Griffiths. *Introduction to Electrodynamics*. 4th ed. Cambridge University Press, (2017).
- [2] community. *wikidia on the Dipole model*. 2022. URL: https://en.wikipedia.org/wiki/Dipole_model_of_the_Earth%27s_magnetic_fiel://en.wikipedia.org/wiki/Dipole_model_of_the_Earth%27s_magnetic_field. Website online February 2022.
- [3] Karsten Ahnert and Mario Mulansky. *The odeint library*. 2009. URL: <http://headmyshoulder.github.io/odeint-v2/index.html>. Website online February 2022.