Simulation of the Earth’s Van Allen Radiation Belt

Tyler Pritchard

Project 1: Phys 527, Fall 2008

Abstract

In this project I attempt to simulate an electron travelling through the Earth’s Van-Allen Radiation Belt. I model the magnetic field causing this dense region of charged particles as a magnetic dipole using classical mechanics and including a relativistic mass correction. Certain expected physical characteristics of the system are evidenced in the simulations – such as the spiraling of electrons along field lines as tan electron heads towards the pole with increasing synchrotron frequency, and the deflection of some electrons in the ecliptic plane.

1. Magnetic Dipole

In this project we choose to model the Earth’s Magnetic Field as a dipole field, as shown in Equation (1). Note that we scale the particles radius by that of the earth such that that quantity is between 1 and 0, and it is this quantity that we use as a variable in this project.

$$B = B_0 \left( \frac{R_e}{r} \right)^3 [3(\hat{p} \times \hat{r})\hat{r} - \hat{p}]$$  (1)

This is a coordinate-independent form, and to calculate the magnetic field at a point, we must both ascribe the dipole moment a direction and choose a coordinate system. The most natural one to do this in is the canonical spherical coordinate system – that is, θ measured from the +z axis, and φ as the angle between the positive x-axis and the projection of the radius onto the xy-plane. This then allows us to write the magnetic field as described in Equation (2).

$$B = B_0 \left( \frac{R_e}{r} \right)^3 [-2\cos(\theta) \hat{r} - \sin(\theta)\hat{\theta} + 0\hat{\phi}]$$  (2)

This field, as a typical dipole field, is symmetric about the dipole moment, in this case aligned with the +z axis, and therefore has no dependence upon φ. Now, while this is a perfectly serviceable expression for the field, when I simulate the I use the Open Source Physics Display3DFrame Class, which is designed to
work in Cartesian coordinates. We can therefore express our Spherical unit vectors in terms of Cartesian ones as in Equations (3-5).

\[
\hat{r} = \sin(\theta) \cos(\varphi) \hat{x} + \sin(\theta) \sin(\varphi) \hat{y} + \cos(\theta) \hat{z} \tag{3}
\]

\[
\hat{\theta} = \cos(\theta) \cos(\varphi) \hat{x} + \cos(\theta) \sin(\varphi) \hat{y} - \sin(\theta) \hat{z} \tag{4}
\]

\[
\hat{\phi} = \sin(\theta) \hat{x} + \cos(\varphi) \hat{y} + 0 \hat{z} \tag{5}
\]

Inserting these into Equation (2), we can then group our unit vectors and take advantage of several trig identities to arrive at Equation (6).

\[
B = -B_0 \left(\frac{R_0}{r}\right)^3 \left[ 1.5 \sin(2\theta) \cos(\varphi) \hat{x} + 1.5 \sin(2\theta) \sin(\varphi) \hat{y} + \left(2\cos^2(\theta) - \sin^2(\theta)\right) \hat{z} \right] \tag{6}
\]

Now that our magnetic field is in a convenient basis, the last thing we need to be able to know it is how to convert \((x,y,z)\) into \((r,\theta,\phi)\) and vice versa. This may be seen in Equations (7-9).

\[
x = r \times \sin(\theta) \cos(\varphi) \quad r = \sqrt{x^2 + y^2} \tag{7}
\]

\[
y = r \times \sin(\theta) \sin(\varphi) \quad \tan(\theta) = \frac{\sqrt{x^2 + y^2}}{z} \tag{8}
\]

\[
z = r \times \cos(\varphi) \quad \tan(\varphi) = \frac{y}{x} \tag{9}
\]

So, throughout the simulation, we will record our positions and velocities in Cartesian coordinates, convert them to polar, calculate the new acceleration values, and then add them to our previous Cartesian values.

2. Differential Equation

The force acting on a particle in a magnetic field is \(\mathbf{F} = q(\mathbf{V} \times \mathbf{B})\), where \(q\) is the charge of the particle, \(\mathbf{V}\) is the velocity and \(\mathbf{B}\) is the magnetic field. Using classical mechanics, we may write our differential equation of motion as seen in Equation (10).

\[
\frac{d\mathbf{v}}{dt} = \frac{q}{m} (\mathbf{V} \times \mathbf{B}) \tag{10}
\]

While we are using classical mechanics for this simulation, we wish to add a relativistic mass correction, using Equation (11).
\[ m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (11) \]

Combining all of this, and computing the cross product, we arrive at Equation (12).

\[
\frac{dv}{dt} = q \frac{\sqrt{1 - \frac{v^2}{c^2}}}{m_0} \left[ (v_y B_z - v_z B_y) \hat{\mathbf{r}} + (v_z B_x - v_x B_z) \hat{\mathbf{y}} + (v_x B_y - v_y B_x) \hat{\mathbf{z}} \right] \quad (12)
\]

From here, we may insert our magnetic field vector from Equation (6) and have our equation of motion. I have already stated that the simulation will be using its radial component as a scaled earth radius, and it will be convenient to scale our velocity by \( c \) in a similar manner. Taking this scaling into account, our constant becomes that in Equation (13).

\[
\delta = -\frac{B_0 \cdot q \cdot c}{m_e} = -2.049 \frac{m}{s^2} \quad (13)
\]

Our final equation of motion, then, is Equation (14) below, where \([\ldots]\) is the unit less cross product of our scaled velocity and magnetic field components.

\[
\frac{dv}{dt} = \delta \cdot \sqrt{1 - \frac{v^2}{c^2}} \cdot \left( \frac{R_e}{r} \right)^3 \cdot [\ldots] \quad (14)
\]

Since we have our numerical values for the velocity components and have solved for the magnetic field components in a Cartesian basis vector in Equation (6), it is straightforward to compute the acceleration.

### 3. Implementation

Ideally, we would like to keep the numbers we use to solve our differential equation of motion around unity to minimize computational error, and to that end the actual radii and velocities used in the simulation are scaled quantities \( \left( \frac{v}{c} \right) \) and \( \left( \frac{R}{r} \right) \). To compute our physical quantities, I use a Euler-Cromwell algorithm. While this is not the most precise algorithm, I was unable to successfully implement an RK4 ODE solver into the 3-D ball class for this second order differential equation. The difficulties here arose with the expansion of the state and rate arrays to the third dimension. While this should be tractable, the E-C algorithm is sufficient over most regimes in this problem, and only appears to break down significantly at large velocities, large time steps, or inside of the earth. This is significant because we can control the time step to lessen the second, the physical model becomes inappropriate at large velocities (that is, our relativistic correction is no longer sufficient), and we can control the time step. So all told, the E-C algorithm is sufficient, though not preferable.

Managing the time step is important however, for to large of a time step, particle paths which should spiral in towards the pole instead become chaotic and are ejected in a random direction, due to imprecise calculations. Below, we have 2 figures – both of these particles have the same initial conditions, but the second has a time step a factor of 10 times larger. We see here a situation where
our simulation begins to break down – an improved differential equation algorithm, or variable time step would be two methods of ameliorating this problem.

Figure (1) – An electron with time step $dt=0.01$

Figure (2) – An electron with identical initial conditions as that in Figure (1) and time step $dt=0.1$
Through experimentation with initial conditions, it appears that electrons with initial positions and
velocities close to the pole tend to spiral inwards, while most others are deflected – which is
qualitatively similar to what we expect given what we know of the radiation belt. For a more
quantitative analysis, more information would be required on typical synchrotron periods, as well as the
relative strengths and positions of particles in it. Pictures of some trial runs may be seen below.

Figure (3) – Another example of an electron spiraling in along the dipole field
Figure (4) – The deflection of a distant, slow moving electron. Here $r_0 = 8R_e$, and $v = 0.3c$. 

(time = 66.07)

(time = 89.37)
Figure (5) – The deflection of an electron. An artifact of the simulation, here a high velocity electron’s path is completely reversed.

Figure (6) – An electron that “jumped” field lines close to the earth – likely due to numerical errors in the simulation. Note that its spiraling outwards similarly to it’s entry vector.

4. Conclusion

The Van Allen Belt Simulation models an electron interacting with the magnetic field of the Earth. The Earth’s magnetic field is modeled as a dipole field, and we use a Euler-Cromwell algorithm to compute our electron’s motion. The simulation provides stable, physical paths for some initial conditions and time-steps. There are however a variety of values of these for which our system breaks down due to difficulties with our differential equation solver, physical model, and time step.