Magnetic dipole earth – modelling the aurora borealis

M. Woxholt[a]

[a] *Institutt for fysikk, Norges Teknisk-Naturvitenskapelige Universitet, N-7491 Trondheim, Norway.*

Abstract

This article aims to discuss the model of the earth as a magnetic dipole. Using numerical methods and computer programming, the goal has been to simulate the movement of the solar wind particles in the earth's inner magnetic field that are the origin of the aurora borealis.

1. Introduction

The earth's magnetic field can be approximated as a magnetic dipole with the magnetic south pole ironically lying on the northern hemisphere. I will in this article present results and visualizations of the magnetic field based on this simple model. To the magnetic field I will apply a simplified model of the aurora borealis, sending single particles into the magnetic field and mapping the trajectory.

2. Theory

To construct the model we need to take into consideration the following aspects of electromagnetic theory.

The magnetic field from a dipole moment $\mathbf{m}$ at a position $\mathbf{r}$ relative to the dipole can be described as

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi}\frac{\mathbf{m}\times\hat{\mathbf{r}}}{r^2}. \tag{1}$$

For a particle moving in a magnetic and electric field, the Lorentz force

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v}\times\mathbf{B}) \tag{2}$$

will apply. However in our simplified model we will not consider the earth's electric field. As we seek to model the trajectory of a solar wind particle we care for the acceleration of said particle, hence we simplify the Lorentz equation

$$\ddot{\mathbf{x}} = \frac{q}{m_p}(\mathbf{v}\times\mathbf{B}). \tag{3}$$

As the dipole model of the earth's magnetic field mainly holds true in the inner magnetosphere; up to $6R_E$[**Pankaj, et al.**], and our model particle is moving orders of magnitude below the speed of light, we neglect the relativistic effects of the particle movement. The kinetic energy of particles trapped in the earth's magnetic field vary between 1eV to 100MeV[**Pankaj et al.**].

3. Numerical method

In geomagnetism it is conventional to model the dipole moment of the earth in spherical coordinates[**Hill**]. However of practical reasons, I have chosen to generalize the dipole moment in cartesian coordinates for my numerical model [**Kievelson**]

$$\mathbf{B} = \begin{pmatrix} (3x^2 - r^2) & 3xy & 3xz \\ 3xy & (3y^2 - r^2) & 3yz \\ 3xz & 3yz & (3z^2 - r^2) \end{pmatrix}\mathbf{M} \tag{4}$$

Here $\mathbf{M}$ is

$$B_0\frac{R_E^3}{r^5}\mathbf{m} \tag{5}$$

where B_0 is the measured magnitude of the magnetic field at the magnetic equator of the earth's surface[**Pankaj, et al.**]. Meanwhile, $\mathbf{m}$ is the magnetic dipole moment tilted θ degrees with respect to the earth's rotational axis

$$\mathbf{m} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{6}$$

For modelling the trajectory of the solar wind particle I apply the Runge-Kutta-4(5) method.

$$\begin{aligned}
&k_1 = hf(t_k, y_k),\\
&k_2 = hf\left(t_k + \frac{1}{4}h, y_k + \frac{1}{4}k_1\right),\\
&k_3 = hf\left(t_k + \frac{3}{8}h, y_k + \frac{3}{32}k_1 + \frac{9}{32}k_2\right),\\
&k_4 = hf\left(t_k + \frac{12}{13}h, y_k + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right),\\
&k_5 = hf\left(t_k + h, y_k + \frac{439}{216}k_1 - \frac{8}{216}k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right),\\
&k_6 = hf\left(t_k + \frac{1}{2}h, y_k - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right),\\
&y_{k+1} = y_k + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4101}k_4 - \frac{1}{5}k_5.
\end{aligned} \tag{7}$$

To implement this method I will use the differential equation solver «integrate.RK45()» from the Scipy-library.

4. Results and discussion

The magnetic dipole moment

We lay the foundations of our model with modelling the earth as a perfect magnetic dipole. From figure 1 we can see that our model behaves as expected in the XZ-plane with magnetic field lines flowing from the magnetic north pole to the south pole.

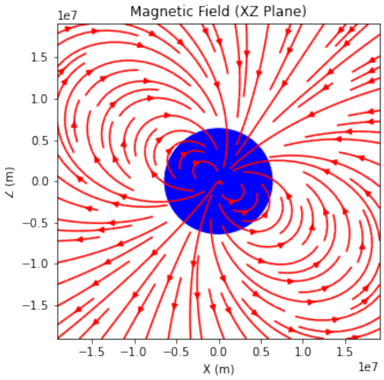

Figure 1: Magnetic field lines seen "from space". The axis into the paper is radial to the ecliptic.

The tilt of the earth and the magnetic field is applied so that the rotational axis of the earth is tilted 23°with regards to the axes of the plots, and the magnetic dipole moment is tilted 11°with regards to the rotational axis.

We can see the same expected behaviour from figure 3 and 2 as they respectively show the magnetic field lines in the XY-plane at the cross section of the equator (the cross section is symmetrical to the ecliptic, so we look upon the tilted magnetic field) and at the geographical north pole.

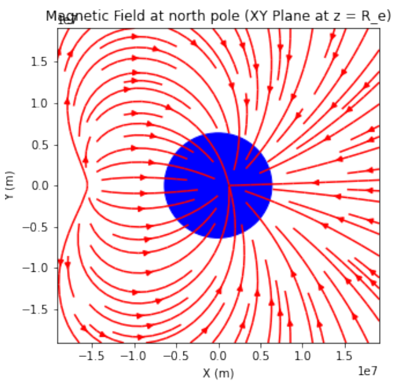

Figure 2: Magnetic field lines at the geographical north pole. The axis entering the plane is perpendicular to the ecliptic. The magnetic field lines are meeting at the magnetic south pole tilted 34°with regard to our z-axis.

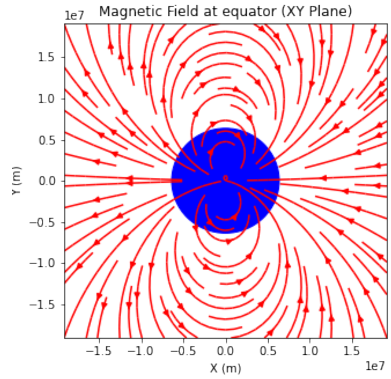

Figure 3: Magnetic field lines seen at the equatorial cross section of the earth.

The particle trajectory

I have modelled the particle trajectory solving the PDE for velocity and position from the Lorentz force (3) using (7). The particle is modelled as a proton with positive charge 1 eV entering the magnetic field of the earth along the x-axis (axis setup described as above) with initial velocity $v_0 = 30 \times 10^6$ m/s, which corresponds to a kinetic particle energy of about 12MeV. As we only consider the magnetic field at the distance $r = 6R_E$ form the earth, all simulations in this article starts at distance $r = 2R_E$ towards the sun.

Considering the small scale particle, compared to the earth and its magnetic field, it is reasonable to expect the particle to behave as if moving in a uniform magnetic field. As a result, we expect gyration around the magnetic field lines due to diamagnetic effects. As expected, from figure 5 and 6 we can clearly see the small oscillations about the magnetic field lines. As the earth and its magnetic field is tilted with regards to the ecliptic, the particle with velocity parallel to the ecliptic plane contains a component parallel to the magnetic field lines. It will therefore follow the field lines in addition to the oscillations and create a helical orbit.

Another known consequence of the Lorentz force is that the resulting velocity when entering the magnetic field must be perpendicular to both the magnetic field and the Lorentz force

$$\mathbf{v}_f = \frac{1}{q}\frac{\mathbf{F} \times \mathbf{B}}{B^2}. \tag{8}$$

As a result of the helical orbit discussed above, this perpendicular velocity will have a westward drift. Figure 4 confirms this.

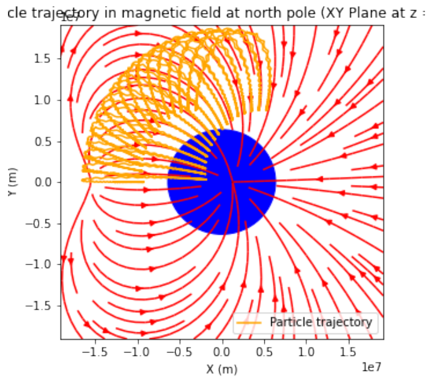

Figure 4: Trajectory of proton seen from the north pole. The axis into the paper is perpendicular to the ecliptic. 30 second simulation.

The last expected motion of the particle is the bounce over mirror points when closing onto the poles[**Pankaj et al.**] and change direction parallel to the field. As the magnetic moment of the particle

$$\mu = \frac{mv_{\perp}^2}{2B} \tag{9}$$

must remain constant we will see a rise of perpendicular velocity when near the poles. For the kinetic energy

$$\mathcal{E} = \frac{1}{2}mv_{\|}^2 + \frac{1}{2}mv_{\perp}^2 \tag{10}$$

of the particle to also remain constant, the parallel component must drop to zero and eventually switch signs so that the particle is repelled from the dense magnetic field [**Wikipedia/Magnetic_mirror_point**]. We see clearly that all trajectory figures contain the motion back and forth between the poles.

Testing accuracy of model

The fact that magnetic fields do no work is a fundamental principle of electromagnetism. I have used this principle to do a simple test of the accuracy of model considering the interaction between the field and the moving particle. We expect:

$$\int q(\mathbf{v} \times \mathbf{B}) \cdot \mathbf{v} dt = 0, \tag{11}$$

and find numerically that the work done on particle by the magnetic field $W_B = 1.9669675099231124e - 23$ which is more than a satisfactory result.

5. Conclusion

From the results and discussion above I will conclude that the dipole moment approximation of the earth's magnetic field provides a simple but quite effective model for

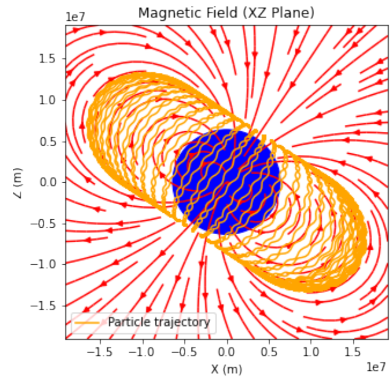

Figure 5: Trajectory of proton seen "from space". The axis into the paper is perpendicular to the ecliptical. 50 second simulation.

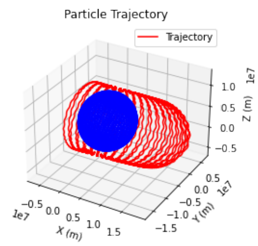

Figure 6: 3D-plot of trajectory. 20 second simulation.

visualizing the effect of the earth's magnetism on solar winds. Returning to the discussion of the aurora borealis, we can clearly see how the particle is the closest to the surface near the poles and that it gyrates in a circular motion around the magnetic poles. This corrwsponds to what we now about the aurora borealis being most prominent in a belt circulating around the magnetic north pole.s

6. Appendix A: Python code

```

import matplotlib.pyplot as plt
import numpy as np
from matplotlib.patches import Circle

theta_offset = np.radians(9.6)
M = 3.07*10**(-5) #T*R_e^3
R_e = 6370*10**3 #m
n = 1001

def coord_convert(arr, theta):

    # Rotate dipole moment vector around the x-axis
    x = arr[0] * np.cos(theta) + arr[2] * np.sin(theta)
    y = arr[1]
    z = -arr[0] * np.sin(theta) + arr[2] * np.cos(theta)
```

```python

    return np.array([x, y, z])

def dipole_magfield(x_mesh, y_mesh, z_mesh, m):
    r = np.sqrt(x_mesh**2 + y_mesh**2 + z_mesh**2)
    x, y, z = x_mesh, y_mesh, z_mesh

    B_x = (m[0]*(3*x**2-r**2) + m[1]*3*x*y + m[2]*3*x*z)/r**5
    B_y = (m[0]*3*x*y + m[1]*(3*y**2-r**2) + m[2]*3*y*z)/r**5
    B_z = (m[0]*3*x*z + m[1]*3*y*z + m[2]*(3*z**2-r**2))/r**5

    return B_x, B_y, B_z


m_0 = M*R_e**3*np.array([0,0,-1])
m = coord_convert(m_0, np.radians(34))

x = np.linspace(-3*R_e, 3*R_e, n)
y = np.linspace(-3*R_e, 3*R_e, n)
z = np.linspace(-3*R_e, 3*R_e, n)

xv, yv = np.meshgrid(x, y)
B1, B2, _ = dipole_magfield(xv, yv, 0, m)
B12, B22, _ = dipole_magfield(xv, yv, R_e, m)

xv, zv = np.meshgrid(x, z)
B13, _, B33 = dipole_magfield(xv, 0, zv, m)

# Create subplots
fig, ax = plt.subplots(figsize=(5, 5))

# Plot each magnetic field using streamplot
ax.streamplot(x, y, B1, B2, color='r')
ax.set_title('Magnetic Field at equator (XY Plane)')
ax.set_xlabel('X (m)')
ax.set_ylabel('Y (m)')
ax.add_patch(plt.Circle((0, 0), R_e, color='b', fill=True))
fig.savefig('Magnetic_field_XY_Plane).png')

fig, ax = plt.subplots(figsize=(5, 5))
ax.streamplot(x, y, B12, B22, color='r')
ax.set_title('Magnetic Field at north pole (XY Plane at z = R_e)')
ax.set_xlabel('X (m)')
ax.set_ylabel('Y (m)')
ax.add_patch(plt.Circle((0, 0), R_e, color='b', fill=True))
fig.savefig('Magnetic_field_NP).png')

fig, ax = plt.subplots(figsize=(5, 5))
ax.streamplot(x, z, B13, B33, color='r')
ax.set_title('Magnetic Field (XZ Plane)')
ax.set_xlabel('X (m)')
ax.set_ylabel('Z (m)')
ax.add_patch(plt.Circle((0, 0), R_e, color='b', fill=True))
fig.savefig('Magnetic_field_XZ_Plane).png')

plt.show()

from scipy.integrate import RK45

c = 3*10**8
v_0 = 400*10**5 #m/s
m_p = 1.623*10**(-27) #kg
q = 1.602*10**(-19) #C

def a(v, B_vec):
    return q/m_p*np.cross(v, B_vec)

def func(t, y):
    B = np.array(dipole_magfield(y[0], y[1], y[2], m))
    a_val = a(y[3:], B)
    return np.concatenate((y[3:], a_val))

def run_vel():

    for i in range(0,3,10):
        x = i*R_e
        for j in range(1,4):
            y = j*R_e
            for k in range(1,4):
                z = k*R_e
                y_0 = np.array([x, y, z, -v_0, 0, 0])  # Initial position and velocity
                x,y,z = 0

                t_span = [0, 30]


                integrator = RK45(func, t_span[0], y_0, t_span[1], max_step=0.01)

                # Lists to store the results
                t_vals = [integrator.t]
                y_vals = [integrator.y]

                max_steps = 100000

                # Integrate the differential equation
                step_count=0
                while integrator.status == 'running' and step_count < max_steps:
                    integrator.step()
                    t_vals.append(integrator.t)
                    y_vals.append(integrator.y)
                    step_count+=1

                # Convert the results to NumPy arrays
                t_vals = np.array(t_vals)
                y_vals = np.array(y_vals)

                np.savetxt(f'{i}{j}{k}time.gz', t_vals, fmt='%.18e', delimiter=' ')
                np.savetxt(f'{i}{j}{k}pos_vel.gz', y_vals, fmt='%.18e', delimiter=' ')

    return 0


y_0 = np.array([-2*R_e, 0, 0, v_0, 0, 0])  # Initial position and velocity
t_span = [0, 30]


integrator = RK45(func, t_span[0], y_0, t_span[1], max_step=0.01)

# Lists to store the results
t_vals = [integrator.t]
y_vals = [integrator.y]

max_steps = 100000
```

```python

# Integrate the differential equation
step_count=0
while integrator.status == 'running' and step_count < max_steps:
    integrator.step()
    t_vals.append(integrator.t)
    y_vals.append(integrator.y)
    step_count+=1

# Convert the results to NumPy arrays
t_vals = np.array(t_vals)
y_vals = np.array(y_vals)

def acc_test(v_x, v_y, v_z, B):

    dW = np.zeros(v_x.size)
    for i in range(v_x.size-1):
        v = np.array([v_x[i], v_y[i], v_z[i]])
        B_temp = np.array([B[0,i], B[1,i], B[2,i]])
        dW[i] = q*np.dot((np.cross(v, B_temp)),v)
        B_temp = 0
    return abs(np.sum(dW))

# Extract x, y, and z positions from y_vals
x_positions = y_vals[:, 0]
y_positions = y_vals[:, 1]
z_positions = y_vals[:, 2]

v_x = y_vals[:, 3]
v_y = y_vals[:, 4]
v_z = y_vals[:, 5]

B = np.array(dipole_magfield(x_positions, y_positions, z_positions, m))

sim_acc = acc_test(v_x, v_y, v_z, B)
print(f'Work done on particle by magnetic field = {sim_acc}')

theta, phi = np.linspace(0, 2 * np.pi, n), np.linspace(0, np.pi, n)
theta, phi = np.meshgrid(theta, phi)
r = R_e
x1 = r * np.sin(phi) * np.cos(theta)
y1 = r * np.sin(phi) * np.sin(theta)
z1 = r * np.cos(phi)
coord_arr = np.array([x1,y1,z1])
tilt_coord = coord_convert(coord_arr, np.radians(23))

# Plot the trajectory
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(y_positions, x_positions, z_positions, label='Trajectory', color='r')
ax.plot_wireframe(tilt_coord[0], tilt_coord[1], tilt_coord[2], color='b')
ax.set_xlabel('X (m)')
ax.set_ylabel('Y (m)')
ax.set_zlabel('Z (m)')
ax.set_title('Particle Trajectory')
ax.legend()
fig.savefig('Trajectory_3d).png')
plt.show()


fig, ax = plt.subplots(figsize=(5,5))
ax.streamplot(x, y, B12, B22, color='r', zorder=0)
ax.set_title('Particle trajectory in magnetic field at north pole (XY Plane at z = R_e)')
ax.set_xlabel('X (m)')
ax.set_ylabel('Y (m)')
ax.plot(x_positions, y_positions, color='orange', label='Particle trajectory', zorder=1)
ax.add_patch(plt.Circle((0, 0), R_e, color='b', fill=True, zorder =2))
ax.legend()
fig.savefig('Trajectory_XY_plane.png')

fig, ax = plt.subplots(figsize=(5,5))
ax.streamplot(x, z, B13, B33, color='r')
ax.set_title('Magnetic Field (XZ Plane)')
ax.set_xlabel('X (m)')
ax.set_ylabel('Z (m)')
ax.add_patch(plt.Circle((0, 0), R_e, color='b', fill=True))
ax.plot(x_positions, z_positions, color='orange', label='Particle trajectory')
ax.legend()
fig.savefig('Trajectory_XZ_plane.png')

plt.show()
```

References

[1] Pankaj K. Soni, Bharati Kakad, Amar Kakad. Indian Institute of Geomagnetism, New Panvel (West), Navi Mumbai, India. Received 18 July 2020, Revised 11 October 2020, Accepted 15 October 2020, Available online 30 October 2020, Version of Record 23 December 2020.

[2] Hill C. IAEA's Atomic and Molecular Data Unit. "Visualizing the Earth's dipolar magnetic field" Web page: https://scipython.com/blog/visualizing-the-earths-magnetic-field/ (20.3.2024)

[3] Kivelson M.G., Russell C.T. Chap 6.2 Planetary Magnetic fields pp. 165-167. Web page:http://www.ss.ncu.edu.tw/ yhyang/104-1/Ref-KivelsonRussell-p165-167.pdf (20.3.2024)

[4] https://en.wikipedia.org/wiki/Magnetic_mirror_point(20.3.2024)