Modeling Geomagnetic Induced Currents Using Computer-Based Simulations

Abstract
Geomagnetic disturbances (GMDs) are caused when the sun emits a coronal mass ejection (CME), a large expulsion of magnetic fields. When the CME collides with Earth's magnetic field, the collision generates currents that create Geomagnetically Induced Currents (GICs). GICs can disturb electrical systems in the affected areas. The March 1989 Quebec incident displays an example of the effects of GMDs, causing a nine-hour power outage. Boteler, Pirjola, and Marti 2019 works to derive the equations necessary to model GMDs. However, computer-based implementations require specific methods for accurate modeling. This project researches the development of a GMD model on a computer using the Julia programming language.

Keywords: Geomagnetic Disturbances, Geomagnetic Induced Currents, Julia

1 Introduction

Geomagnetic disturbances, also referred to as "geomagnetic storms", are caused by a transfer of energy of charged particles from the sun to the space surrounding Earth. GMDs are especially dangerous due to the variability of their effects. While the presence of GMDs can be predicted, the effect and magnitude cannot be. The currents produced by the GMDs create electric fields at the Earth's surface. The currents can flow into conducting networks such as power grids or telephone networks, damaging equipment and devices in these networks. Based on the scale of the GMD, the effect can be as large as city-wide blackouts. The process to help mitigate the effects of GMDs is made up of three parts. First, data must be collected for the required area. Second, the collected data must be processed by a transfer function model. Lastly, the data from the model can be used to determine the necessary currents to properly power the affected networks. This paper focuses on the implementation of the second stage, the transfer function model. For the model, the 5 layer Quebec model developed in Boteler, Pirjola, and Marti 2019 is used.

2 Implementation

The implementation of the 5 layer Quebec model is done in the Julia programming language, using `Trapz` for trapezoidal integration, `PyPlot` for plots, and `FFTW` for Fourier transforms.

2.1 Synthetic Dataset

First, a synthetic dataset was created for testing. The synthetic field variation was calculated using the sum of 7 sinusoidal functions. The code for the data was implemented using the following code:

```
A =   [200, 90, 30, 17, 8, 3.5, 1]
Phi = [10, 20, 30, 40, 50, 60, 70]

f = [0.00009259, 0.00020833,
0.00047619, 0.00111111, 0.00238095,
0.00555555, 0.025]

function B(t) # Equation (20)
  y = 0
  for i in 1:7
    y += A[i] * sin(
      2 * pi * f[i] * t
        + (pi / 180 * Phi[i])
      )
      end
  return y
end
```

Using the `B` function, the synthetic variation can be found for any second `t`. In Julia, this can be implemented simply using the vectorization operators like so:

```
timestep = 1
X = 1:timestep:86400
Y = B.(X)
```

2.2 Discretization

Because this implementation is done on a computer with a finite amount of memory, processing power, etc., the continuous function cannot be represented on a computer and must be discretized using a Fourier transform. The following code is used to discretize the `B` function using the FFTW library:

```
f = fft(Y)
```

Additionally, the horizontal geoelectric field, E(f) is calculated as the convolution of B and K, defined in Julia like so:

```
function K(f)
    mu = pi * 4e-6
    sigma = 1000
    i = 1im
    return sqrt(
        (i * 2 * pi * f) /
        (mu * sigma)
    )
end
```

and the convolution of the signals is calculated like so:

```
freq = fftfreq(length(Y), timestep)
conv = K.(freq) .* f
```

This code creates and processes the synthetic field variation data. Next, a model needs to be developed to model the input data.

2.3 5 Layer Quebec Model

The model used in this paper is the 5 layer Quebec model used to model Earth conductivity in Boteler, Pirjola, and Marti 2019. The model is represented by 5 equations, each of which use the previous equation. This is implemented in Julia using recursion with the base case being the topmost layer. Each layer has distinct thicknesses and relativities. The thicknesses and relatives are as followed:

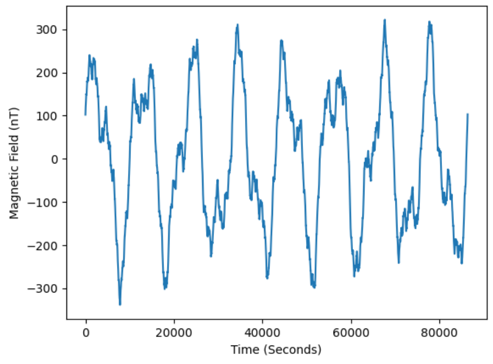
Layer	Thickness (Km)	Resistivity (Ωm)
1	15	1/20000
2	10	1/200
3	125	1/1000
4	200	1/100
5		1/3

Layer 5 is not defined using the table and recursive function. Instead, layer 5 is the base case. The base case and recursive function are defined as so:

```
thickness = [1/20000, 1/200, 1/1000,
    1/100, 1/3] * 1e3
resistivity = [20000, 200, 1000, 100]

N = 5 # number of layers in the model
mu = pi * 4e-6
i = 1im

function K_Base(f)
    return sqrt((i * 2 * pi * f) /
    (mu * thickness[N]))
end

function k(n, f)
    return sqrt(i * 2 * pi * f
    * mu * thickness[n])
end
```

Continued On Next Page

```
function eta(n, f)
    return i * 2 * pi * f
    / k(n, f)
end

function eExpr(n, f)
```



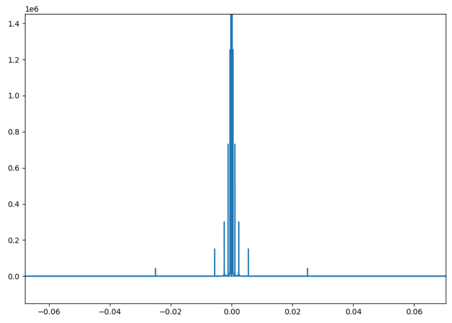


Figure 1: Synthetic Field Variation Graph

Figure 2: Discrete Fourier Transformer of B(t) - A correct implementation should include 7 lines on each side of x=0, representing the 7 sine functions in B(t)

```
    return e ^ (−2 * k(n, f)
    * resistivity[n])
end

function K(n, f) # Equation 19
    if n == N
            return K_Base(f)
    end

    K_NPrev = K(n+1, f)
    e = eExpr(n, f)

    num = K_NPrev * (1 + e)
    + eta(n, f) * (1 − e)

    den = K_NPrev * (1 − e)
    + eta(n, f) * (1 + e)

    return eta(n, f) * num / den
end

function K(f)
    return K(1, f)
end
```

The function K (not to be confused with the K in 2.2) is the complete implementation of the 5-layer Quebec model.

3 Benchmark

To evaluate the accuracy of the computer implementation, it must be compared to the results from the original paper. Table 4 in Boteler, Pirjola, and Marti 2019 shows the values calculated by the transfer function. This can be compared to the results calculated using the computer implementation.

Based on the tables, the computer simulation is very accurate; any inaccuracy is most likely due to the inaccuracy of calculations, constants, or the FFTW library.

4 Real World Application

The 5 layer model implemented above can be used with real data rather than a simulated dataset. In this paper, a Fort Simpson, Canada dataset is used. For each value in the time-series data, the value was used as input to the transfer function `K`. This can be vectorized in Julia like so:

```
Y = K.(X)
```

The lines below show both the input from the Fort Simpson data as well as the transfer function data.

5 Conclusion

This project develops a computer-based implementation of [CITATION]. Some challenges faced in this project were related to the nature of a computer-based implementation. For example, discretization was needed due to computer limitations. Additionally, speed was not a concern during the development. Because of this, the Julia programming language was often a bottleneck due to the Just In Time (JIT) compilation method. This makes Julia great for long running programs but often too slow for quick scripts. If reimplemented, a new implemention might include a compiled programming language like C or C++ for a faster runtime. Additionally, if further researched, a GIC solver would be created to determine the actual voltages and currents of the system that would be required.

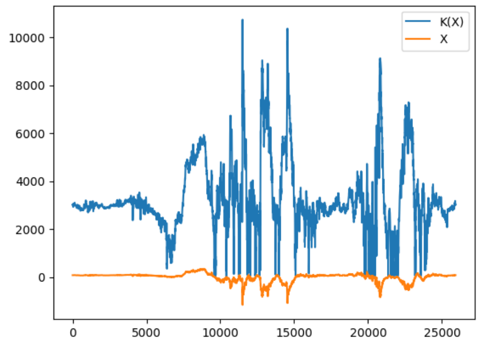


Figure 3: Sensor data and transfer function data from Fort Simpson dataset

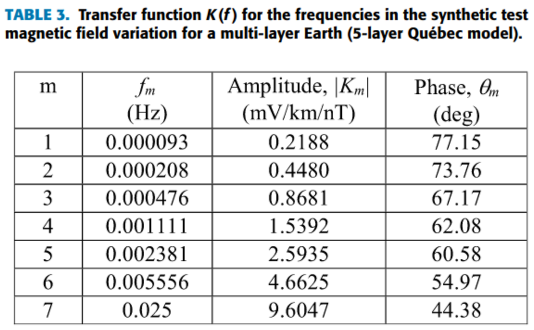
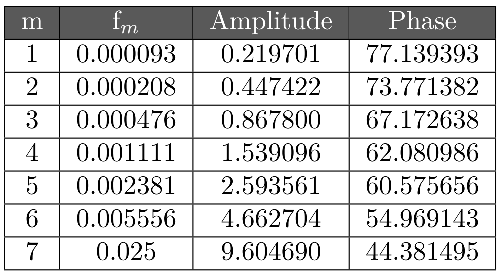


TABLE 3. **Transfer function $K(f)$ for the frequencies in the synthetic test magnetic field variation for a multi-layer Earth (5-layer Québec model).**

m	f_m (Hz)	Amplitude, $\|K_m\|$ (mV/km/nT)	Phase, θ_m (deg)
1	0.000093	0.2188	77.15
2	0.000208	0.4480	73.76
3	0.000476	0.8681	67.17
4	0.001111	1.5392	62.08
5	0.002381	2.5935	60.58
6	0.005556	4.6625	54.97
7	0.025	9.6047	44.38



m	f_m	Amplitude	Phase
1	0.000093	0.219701	77.139393
2	0.000208	0.447422	73.771382
3	0.000476	0.867800	67.172638
4	0.001111	1.539096	62.080986
5	0.002381	2.593561	60.575656
6	0.005556	4.662704	54.969143
7	0.025	9.604690	44.381495

Figure 4: Transfer function results from Boteler, Pirjola, and Marti 2019

Figure 5: Results from function implemented above

References

(N.d.[b]).

(N.d.[c]).

(N.d.[d]). URL: https://onepetro.org/NACECORR/proceedings/CORR01/All-CORR01/NACE-01317/112880.

(N.d.[e]).

Boteler, David H., Risto J. Pirjola, and Luis Marti (2019). “Analytic Calculation of Geoelectric Fields Due to Geomagnetic Disturbances: A Test Case”. In: *IEEE Access* 7, pp. 147029–147037. DOI: 10.1109/ACCESS.2019.2945530.

Garner, Rob (2015). *Solar storm and space weather - frequently asked questions*. URL: https://www.nasa.gov/mission_pages/sunearth/spaceweather/index.html.

Geomagnetically induced currents (gics) (n.d.). URL: https://www.jpl.nasa.gov/infographics/geomagnetically-induced-currents-gics.

Scientist studies whether solar storms cause animal beachings (2017). URL: https://www.sciencedaily.com/releases/2017/02/170206083827.htm.