POYNTING FLUX AND KINETIC ENERGY FLUX DERIVED FROM THE FAST SATELLITE

A Major Qualifying Project submitted to the Faculty of
Worcester Polytechnic Institute

In partial fulfillment of the requirements for the
Degree of Bachelor of Science

by

Nicole Cahill Sebastian Musielak

5th March 2010

Project sponsored by SRI International

Advisors:
Dr. Russell Cosgrove, SRI International
Dr. Hasan Bahcivan, SRI International
Provost John A. Orr, Worcester Polytechnic Institute

CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. ABSTRACT

This project, completed in collaboration with SRI International, will be the first of a four year project to determine the amount of energy that is entering the atmosphere due to auroral phenomena caused by solar wind. MATLAB routines were developed to process data from the FAST satellite for Poynting flux and kinetic energy flux. These routines will be utilized in future years of the project.

2. EXECUTIVE SUMMARY

This report outlines the 2010 Silicon Valley Electrical and Computer Engineering (ECE) Major Qualifying Project (MQP). The 2010 MQP is part 1 (year-1) of a larger, four-year, project proposed by Doctor Russell Cosgrove and Doctor Hasan Bahcivan of SRI International and approved by the National Science Foundation (NSF). The main goal of the NSF project is to get a better understanding of energy contributions to the Earth's atmosphere. Specifically, the NSF project lays out a plan to the study and quantify the interaction between solar wind, made primarily of electromagnetic waves and charged particles called space plasma, and the Earth's own geomagnetic field. By the end of the NSF project, Dr. Cosgrove wishes to create a number of models that describe this complex interaction which will, in turn, be used as inputs in larger-scale models. These larger-scale models, typically referred to as Global (General) Circulation Models (GCM's), are used to describe and predict properties of the atmosphere like density, temperature, and circulations patterns.

The primary tool of use for the NSF project is the Fast Auroral SnapshoT (FAST) Explorer Satellite. The FAST satellite was launched on August 21, 1996 by NASA. NASA's plan was to study the Earth's oval-shaped auroral regions located at the poles (high latitudes). Although scientific theories had already been established on some of the causes and effects of the auroras before FAST, based on new data, some of those theories have been revised. In the simplest terms, the Earth's aurora is caused by the transfer of kinetic energy (collision) between solar wind particles with particles that make up the lower-altitude portion of the Earth's atmosphere. One of the questions that NASA hopes that FAST will answer is how electrons and ions are accelerated in space to create the aurora. The effects of the auroras have been poorly quantified, but the main effect that the NSF project and thus this MQP is concerned with is the heating of the atmosphere through the ionosphere. The ionosphere refers to a region of altitude in the Earth's atmosphere between about 300 km and 1000km and the outer edge of the atmosphere. The ionosphere is named such because the many billions and trillions of collisions between molecules in the ionosphere and other particles cause electrons to be knocked off causing the molecules to become ionized. The ionosphere's interaction with solar wind, ionospheric heating, and all subsequent effects are the main topics explored in the NSF project.

As mentioned previously, Dr. Russell Cosgrove and Dr. Hasan Bahcivan of SRI international were the proposers of the NSF project, thus Cosgrove became the Principal Investigator (PI), and Bahcivan served as CoPI. The project was to be performed and is to be completed at SRI international in Menlo Park, California. This makes SRI International the corporate sponsor of the 2010 MQP. Because Cosgrove slated the first three parts (three years) of the project to be performed by WPI students as part of their MQP's, SRI International will also be a corporate sponsor for the 2011 and 2012 MQP's. SRI International is a contract-based research company with innovations in a multitude of fields including engineering, education, political policy, national defense, robotics, space physics and many others. The students of the 2010-2012 MQP's will be working at the Center for Geospace Studies (CGS) in the Engineering and Systems Division (ESD) of SRI.

The final deliverables for year-1 of the NSF project, and thus the final deliverables for the 2010 MQP, included the development and documentation of methods for the computation of Poynting flux and kinetic energy flux from data measured by the FAST satellite. The Poynting flux is the amount of energy carried by an electromagnetic wave going through an area. Poynting flux is calculated by finding the cross product of the electric field and the magnetic field intensity, which results in the Poynting vector, and multiplying by the desired area. Units of Poynting flux are measured in watts per square meter (W/m^2) or joules per square meter-seconds ($J/m^2 *s$). The kinetic energy flux is the amount of energy carried by particles through an area. Kinetic energy flux is calculated by summing the kinetic energy of all incoming particles through a certain area. Units of kinetic energy flux are also measured in watts per square meter (W/m^2) or joules per square meter-seconds (J/m^2 *s).

Once the students were able to develop the methods for calculating the Poynting flux and kinetic energy flux, the students were asked to use their methods to process a limited number of FAST orbits that contained interesting and representative samples. As a starting point, the students needed to understand why the research was being done. They found that similar research projects studying the energy inputs due to auroral processes had already been conducted, but because electric field measurements in previous projects were calculated using electric field models instead of being taken directly. Because deriving electric field data from models has a probabilistic nature, there is an error or variance included in the electric field data, which created an unknown bias to calculations of the Poynting flux. The NSF project seeks to rectify the problem of electric field variability by using electric field measurement values taken directly instead of deriving values from models.

After the students learned the basic theory behind the project, they were able to begin creating methods for calculating Poynting flux and kinetic energy flux. The first step was to view and obtain data from the FAST satellite. To view FAST data, the students had two options: they could view plots of measurements using the Satellite Data Tool (SDT) which connects directly to FAST data banks or they could navigate to the University of California Berkeley FAST website and view summary plots online. The summary plots contain partially processed data taken from FAST. The processing of data includes de-spinning (i.e. removing the effects of the satellite's spin) electric and magnetic field measurements, subtracting the geomagnetic field from the magnetic field measurements in order to find the perturbations, transforming data into another coordinate system, and a number of other techniques. The students used both methods to view and become familiar with FAST data. The next step was to obtain the actual data. The students also had two options for this: they could download data they view on SDT or they could download summary files, which correspond to summary plots, from the Berkeley website. Because the summary files available on the Berkeley website were already partially processed, the students chose to obtain data that way. If the students wanted to obtain data the other way by using SDT, they would have to process the raw data from scratch using a combination of SDT and Interactive Data Language (IDL) routines written by Berkeley researchers. This would have been the ideal situation because the students would have been able to control exactly how data was processed instead of relying on already partially processed data. In practice however, the students were not able to make use of the IDL routines because of software debugging problems. At this point in the project, the students had an understanding of what the data looks like and how to obtain partially processed data online. The next step was to process the data for Poynting and kinetic energy flux.

The summary data was provided by Berkeley researchers and posted online in Common Data Format (CDF). Because there were so many instruments on FAST, there were actually a couple of versions of the CDF files, but the students only needed 4 of the 6 available versions: the DC fields data, the electron energy spectrum data, the ion energy spectrum data, and the orbit-ephemeris (positional) data. Once all the data was obtained for a particular orbit, they needed to align the fields and particle data with the positional data in time. The aligning of the fields, particle, and positional data is done using a simple MATLAB interpolation function. Once all of the data was aligned in time, the students needed to transform the positional data from a non-earth-fixed coordinate system (Geocentric Equatorial Inertial, GEI) to an earth-fixed coordinate system (Geocentric Geographic, GEO). This allowed the students and mentors to describe the position of events with respect to the Earth. Once the positional data was transformed from GEI to GEO, the students found values for the geomagnetic field using the IGRF-1995 model. The electric and magnetic field data comes in a special coordinate system defined relative to the satellite and to Earth's geomagnetic field. Therefore, geomagnetic field models were required to transform the electric and magnetic fields data from the Berkeley Data Coordinate

System (DCS) to GEO which was the next step of data processing. Once all of the available field and position data was transformed into GEO, the final step was to find the cross product of the electric field and magnetic field intensity, H (found by multiplying the magnetic field by the permeability of free space). The resulting Poynting vector could be multiplied by an area to find the Poynting flux. Luckily for the students, the kinetic energy flux was already calculated by Berkeley researchers and posted in the summary files. The students simply multiplied by a scalar to obtain the desired units.

The final step in the project was for the students to plot and interpret the results. The students plotted using MATLAB and presented to Cosgrove and Bachivan. Because a thorough understanding of the results by the students is outside the scope of this project, the students could not readily verify their results, but upon initial inspection, were told by Cosgrove and Bachivan that their results were reasonable. Specifically, the values of the Poynting and kinetic energy flux were within the expected range and the directions were as expected as well.

In wrapping up the project, the students wrote this report and documented all of their efforts through video and supplemental text documents. The students also created a website as a resource for future students and advisers.

3. INTRODUCTION

As year one of a four-year project proposed to the National Science Foundation by Doctor Russell Cosgrove and Doctor Hasan Bahcivan of SRI International, this report outlines the processes created to extract and process data from the Fast Auroral SnapshoT Explorer (FAST) Satellite for Poynting Flux and kinetic energy flux.

The FAST Satellite collects data in the auroral regions of the atmosphere, specifically the thermosphere. This data is stored in data banks at the University of California at Berkeley. This project required the electric and magnetic field data, positional data, and ion and electron energy data. This data is available in several forms online and through special software that can be downloaded from the website. The data used in this project is located in Common Data Format (CDF) files, which are easily imported into MATLAB, the software which was used to process the data. These CDFs were downloaded from the Space Science Labs Berkeley website.

The data comes in various coordinate systems. The raw positional data is in the Geocentric Equatorial Inertial (GEI) coordinate system, and needed to be converted to the Geocentric Geographic (GEO) coordinate system for ease of understanding and to achieve the position in a location which is static to the Earth. The raw electric and magnetic field data is in a coordinate system that is called, in this report, the Data Coordinate System, and also needed to be converted to the GEO coordinate system as well. The raw kinetic energy data has a non-traditional coordinate system in that the flux can only be moving toward the Earth or away from it.

As the positional data and the field data are two different files, they have different amounts of samples per file, so interpolation in time was necessary to plot the data with respect to the position of the satellite.

Processing this data for Poynting flux required the conversion of the positional and the field data to the GEO coordinate system, and for the magnetic field (B) to be converted to magnetic field intensity (H). Following that, the cross product of the electric field (E) and the H results in the Poynting flux. The kinetic energy data was available as flux, and as such the units needed to be converted and no other processing was necessary.

This report contains appendices of all of the MATLAB files created and utilized. In addition to this report, supporting documents such as explanatory videos, e-mail correspondence with scientists at UCLA and UC Berkeley, and other relevant documentation is provided for the future researchers involved in completing Doctors Cosgrove and Bahcivan's goals outlined in his proposal.

4. BACKGROUND

4.1 THE ATMOSPHERE

The Earth's atmosphere is multiple layers of various gases that are held by the Earth's gravitational pull. It is principally composed of Nitrogen and Oxygen molecules with many other trace elements as well. It can also be divided into five main layers, depending on elevation and temperature. The temperatures of the various layers of the atmosphere are determined by atmospheric heating. Atmospheric heating comes from many sources and is an important aspect of this MQP. Between each of the layers of the atmosphere are "pauses". A pause is a temperature extrema (i.e., where the variation with altitude reaches a minimum or maximum), and can be thought of as the distinct line between each of the layers, and as such is where the maximum changes are observed in the various properties of the layers. In the section below, the five layers of the atmosphere will be explained in more detail. Shown below in Figure 1 is an illustration of the layers of the atmosphere.

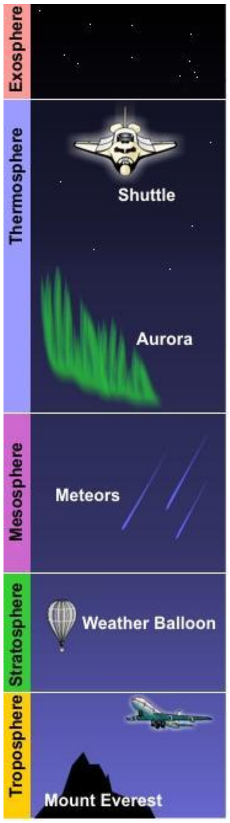


FIGURE 1: THE ATMOSPHERE BROKEN INTO FIVE LAYERS (NOT TO SCALE) (NATIONAL WEATHER SERVICE)

The layer of atmosphere at the lowest elevation is called the troposphere. The troposphere extends between the surface of the Earth to approximately ten kilometers above the Earth's surface depending on the location (the troposphere is thinner at the poles and thicker at the equator). Because it is the lowest layer of the atmosphere and it has four other layers of atmosphere sitting on top of it, it is very dense compared to the other four layers. As such, the troposphere makes up for approximately 75% of the total mass of the atmosphere. It is heated primarily from the shifting of energy from the surface of the Earth. Heating from the surface of the

Earth means that, for the troposphere, the warmest altitude is the lowest altitude (closest to the ground). This also means that, as elevation increases, temperature decreases. The dynamic of temperature decrease with elevation increase is the primary cause of climate-related weather. Warm air rises from the surface and cools at higher elevations, then falls down again. Hence, the vast majority of weather phenomenon occurs in the troposphere. The pause that separates the troposphere from the next layer of atmosphere is called the tropopause. In scientific circles, the term “lower atmosphere” describes the regions of the troposphere and tropopause.

The second layer of atmosphere is called the stratosphere. The stratosphere expands from the end of the troposphere (about ten kilometers) to approximately fifty kilometers. This layer contains less mass than the troposphere, but still more than the upper three layers; it contains approximately 20% of the total mass of the atmosphere. It is heated primarily through absorption of ultraviolet radiation, thus gets warmer as elevation increases. This is directly opposite of the troposphere, where temperature decreases as elevation increases. Because the temperature is already warmer at higher elevations, there is little convection in the stratosphere, so there is little weather activity. The pause that separates the stratosphere from the next layer of atmosphere is called the stratopause.

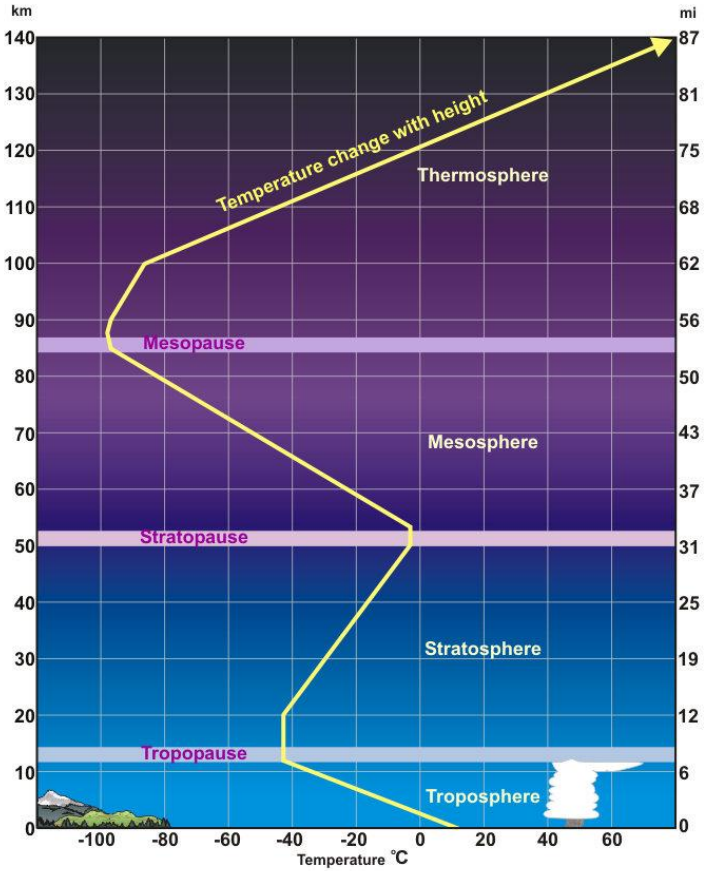


FIGURE 2: TEMPERATURE FLUCTUATIONS IN THE ATMOSPHERE (NATIONAL WEATHER SERVICE)

The third layer of the atmosphere is called the mesosphere. The mesosphere expands from the end of the stratosphere (about fifty kilometers) to approximately 85 km. This layer contains significantly less mass than the two lower layers of atmosphere, but still contains more mass than the upper two. In fact, this layer is dense enough to begin the slowing of incoming meteorites. The friction between the meteorite and the molecules in the mesosphere is typically large enough that it can begin burning up the meteorite. However, because the mesosphere has such low density compared to the two lower layers, low-energy ultraviolet radiation does not do a good job of heating at higher elevations. Thus, the temperature drops as the elevation increases. The temperature fluctuation with respect to elevation is similar to the case with the troposphere and opposite of the stratosphere. The pause that separates the mesosphere from the next layer of atmosphere is called the mesopause. In science jargon, the term "middle atmosphere" describes the regions of the stratosphere, stratopause, mesosphere and mesopause. Shown above in Figure 2, is an illustration of the temperature fluctuations between the various layers of the atmosphere. Note the slopes of the temperature versus altitude lines in the various layers.

The fourth and fifth layers of the atmosphere are called the thermosphere and exosphere. The thermosphere extends from the end of the mesosphere (about 85 km) to about 700 km and is considered the "upper atmosphere" in scientific circles. The exosphere extends from the end of the thermosphere (about 700 km) to about 10,000 km above the Earth's surface. The exosphere is the outermost part of the atmosphere and is where some particles escape the Earth's gravitational pull and float into free space. It should be noted that although these two are much larger (with respect to distance) than the three lower regions put together, they only account for about 5% of the total mass of the atmosphere. This means that the air is very thin; however, an interesting fact is that the temperature of the particles in the thermosphere become quite "hot" due to ultra-violet and x-ray heating from the sun. This MQP will attempt to explore some of those heating issues. What is known is that higher-energy ultraviolet and x-ray radiation absorbed from the sun causes a lot of energy to be dissipated into the thermosphere, and temperature increases with elevation (about 2000 C at the top of the layer). However, because the air is so thin at these altitudes, it would still feel very cold to human skin. A similar phenomenon can be thought to happen in the exosphere. Even though the particles are "hot" in a technical sense, to a human, it would feel cold because there are so few of them.

A very important part of the atmosphere, especially within the context of this MQP is a region called the ionosphere. It should be noted that the ionosphere is not one of the "5 layers of the atmosphere" determined by temperature as detailed in this section. The ionosphere (and others that are beyond the scope of this MQP) is a layer of the atmosphere that is determined by properties other than temperature. It actually sits in a region on the very edge of the stratosphere and extends through the last two layers of the atmosphere. Scientists argue about where the ionosphere actually ends, but the range is between 300 and 1,000 km above the Earth's surface. The ionosphere gets its name because it contains mostly particles that have either gained or lost electrons through interactions with solar waves and particles. It has particular importance because this MQP will be studying some of the interaction between incoming energy from the sun, the ions in the ionosphere, and the resultant electric and magnetic fields. The heating of the upper atmosphere happens when ions in the ionosphere are dragged rapidly through the atmosphere by electric fields. These fast moving ("hot") ions keep colliding with the neutral (and therefore unaffected by the electric field) atmospheric atoms, thereby heating them.

4.2 THE AURORA

Much of the research done in this project involves the aurora. The studies are focused on FAST satellite orbits which pass through the aurora and the fields and particle data that is collected in these regions. The aurora has been the focus of curiosity and studies for thousands of years, with legends and stories revolving around it from all areas of the world. Early hypotheses from Aristotle, Descartes, and Benjamin Franklin, among others, attempted to explain the aurora as a scientific event. These include Aristotle's theory of "rips" in the sky, other theories about reflected firelight from the edge of the world, sunlight reflecting off of ice, and Benjamin Franklin's idea about the electrical potential of clouds. Around the middle of the 19th century, scientific breakthroughs occurred and now more is known about the aurora, however there are still many unknowns. (Geophysical Institute)

The aurora is a luminous glow of the upper atmosphere. It is caused by energetic particles that enter the atmosphere from the geospace environment called the magnetosphere. These particles are composed of mostly electrons, with protons in lesser quantities. Electrons travel along the magnetic field lines. As the Earth's magnetic field originates and terminates at the poles, the electrons are guided to the higher latitudes. As they move into the upper atmosphere, they increase their chance of collision with an atom or a molecule, and these collisions cause the atom or molecule to acquire some of the energy from the particle. To release this energy, the atom or molecule releases a photon, which causes light.

FIGURE 3: THE AURORAL LIGHTS (LUMMERZHEIM)

The altitude of the aurora varies depending on the energy of the electrons. The bottom edge is usually around 100 kilometers, while the top edge can be seen up to 600 kilometers. It can also appear in various shapes and colors for different reasons. Arches and arcs are usually green, at times with red above and purple below, and are motionless and always present. These usually happen during a quiet phase, and during the growth of a substorm it will drift southward. Bands are like arcs, except they have structure, and appear as curtains or drapes. The colorings of bands are similar to that of arches and arcs. These usually occur during the "substorm expansion phase" or the substorm breakup phase. Coronas appear during the expansion phase of the substorm when an active, rayed curtain passes over the viewer's zenith. At the end of an auroral display, diffuse glows can be seen, and depending on the latitude they are green or red for high and low, respectively. (NASA)

Solar wind consists of a stream of protons and electrons, also known as plasma, and is the outermost atmosphere of the sun. Due to the temperature of the sun, the outer layers boil off which results in a very thin gas that expands outward. This solar wind contains the sun's magnetic field embedded within, with a density that is nearly as low as a vacuum. Solar wind must flow around planets, and when planets have magnetic fields, these are also seen as obstacles. Depending on the strength of the field in comparison to the solar wind, it will either be forced along with the field, or will bend the field. In terms of the Earth's magnetic field, this varies as the plasma approaches Earth. It will bend the field unless it becomes too strong. At about 10-12 Earth Radii from the Earth, the solar wind and the magnetic field balance each other out. (One Earth Radii is 6371 kilometers). To put this distance into perspective, the moon is approximately 60 Earth Radii away from the Earth, which demonstrates how strong the magnetic field in this solar wind is. (Lummerzheim)

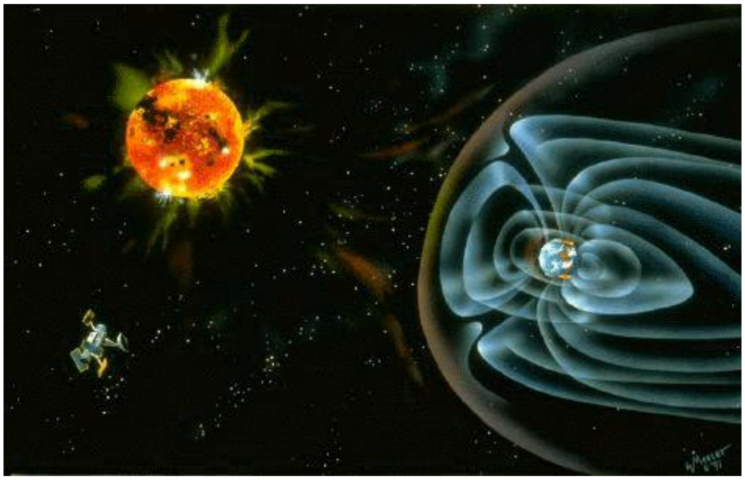


FIGURE 4: AN ARTISTS RENDITION OF THE EARTH'S MAGNETOSPHERE AND THE SUN (LUMMERZHEIM)

When the solar wind approaches the magnetosphere, it can transfer energy to it in a number of ways. The most effective way, reconnection, can occur when the magnetic field in the solar wind and the magnetic field of the magnetosphere are anti-parallel, and the fields melt together, with the solar wind dragging the magnetospheric field and plasma. The magnetosphere, in turn, releases electrons and protons into the upper latitude atmosphere, where it can release energy, which results in the aurora.

Aurora can have effects on the high altitude atmosphere. It can only directly affect the environment at 90-100 kilometers or above, however ionization can occur a bit lower and affect the propagation of radio waves. Another effect the aurora can have involves perturbations in the Earth’s magnetic field. Occasionally the Earth will experience a magnetic storm which changes the magnetic field; however the aurora can cause substorms which change the magnetic field in the Polar Regions. (Lummerzheim)

This project strives to understand more about the aurora and the matter in which these particles interact with Earth’s magnetic field. The space weather that causes the aurora can cause radiation exposure to people travelling in aircrafts, can damage satellites, can affect radio communications, and can even cause power outages by damaging transformers.

4.3 THE FAST SATELLITE

The Fast Auroral SnapshoT Explorer (FAST) is designed to carry out in situ measurements of acceleration physics and related plasma processes that are related to the Earth’s aurora. Initiated by scientists at the University of California at Berkeley, it is designed to carry out small and focused investigations. It takes high resolution data samples strictly in the auroral zones. The primary scientific objectives of the FAST mission are to understand how particles are accelerated to create the aurora, and investigate the microphysics of these and similar space plasma processes. The Earth’s auroral acceleration processes are very complex. Many of them result from a combination of magnetic fields, waves, and certain distributions of energetic ions and electrons. A large part of this mission is determining which of these processes are related to the cause and which to the effect of the auroral acceleration phenomena.

FAST launched August 21st, 1996, after a two year delay due to difficulties with the launch vehicle, into a high-inclination elliptical orbit. The orbit's inclination is 83 degrees from the equator. The apogee altitude, or point on the orbit farthest from Earth's center of mass, is 4175 kilometers, and the perigee altitude, or point on the orbit closest to Earth's center of mass, is 350 kilometers. This can be seen in Figure 5. The satellite crosses Earth's auroral zones four times per orbit. The light that produces the visual aurora is emitted in the upper atmosphere at low altitudes, around 100 to 200 kilometers above Earth. The acceleration of particles that produce the aurora is located at high altitudes, between 2000 and 10000 kilometers above Earth. Therefore, FAST required an apogee as high as possible within the auroral acceleration region. The orbit's alignment of perigee changes at -1.75 degrees per day, and the orbital period is 133 minutes.

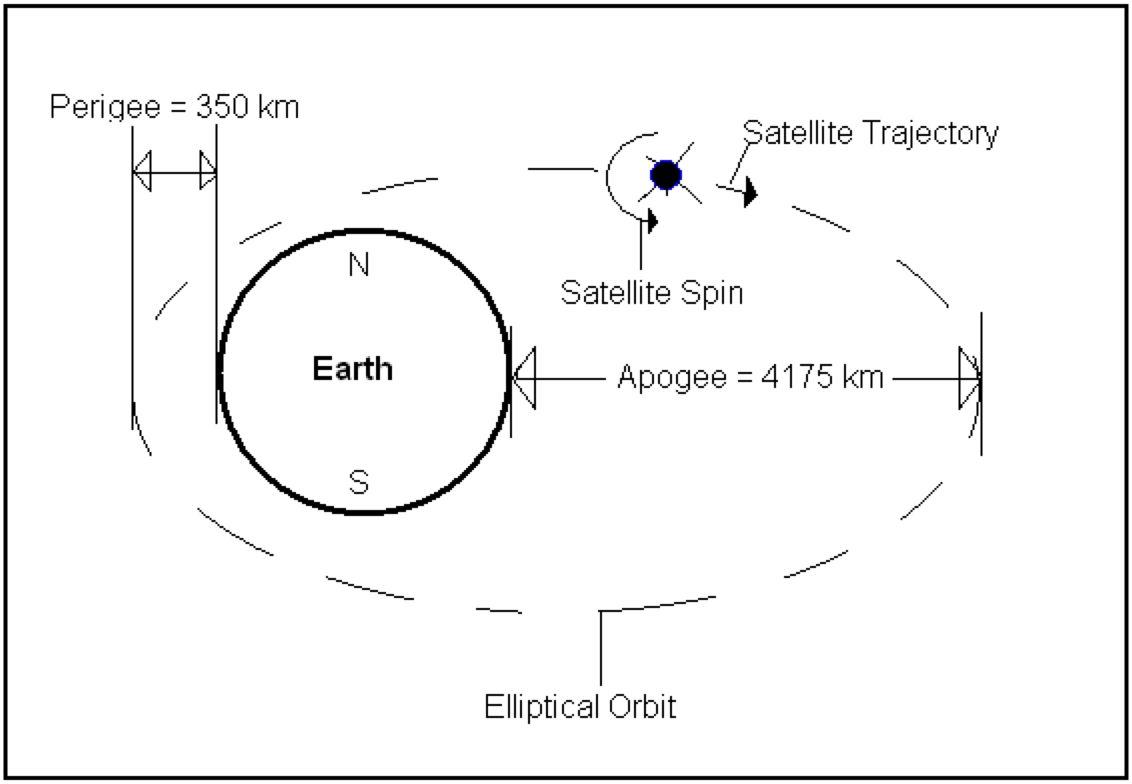


FIGURE 5: FAST SATELLITE ORBIT

FAST shows many improvements over similar missions. It is much faster than other satellites, with a short orbit period, and the orientation of the instruments in terms of the magnetic field helps provide continuous measurements of energetic particles of all angles between the particle and the magnetic field, independent of the spacecraft spin.

The general design approach of FAST includes several important factors. The mass of the spacecraft and instruments were created to be a light as possible, in order to achieve the highest possible apogee. The design and placement of the components was important in order to optimize the moment of inertia of the spacecraft, which in turn maximized the length of the spin axis and electric field booms. There is only one flight computer on FAST, which controls all instruments and data acquisition, provides one common memory for the burst and data storage, provide all regulated power, and house any electronics boards that are not needed at sensor locations. There is a separate electronics system, called Mission Unique Electronics (MUE) which handles the basic life support functions for FAST, such as attitude control, battery charge control, command ingest, and safing functions. The instruments are situated on the satellite to protect against radiation. There are no solar paddles or extended solar arrays, which can disrupt the in situ measurements, block energetic particle orbits, cause unwanted shadows, or create deleterious wake effects. A priority in the design was electrostatic and electromagnetic cleanliness; the solar array and all exposed surfaces must be conducting and kept at the same potential as the spacecraft's internal ground. The satellite has a large, one Gigabit solid state memory, and utilizes a variety of downlink rates, with the highest rate at 2.25 Mbps, and uplink commanding at 2 kbps. The design is single string, with no redundancy, and meets the qualifications set by NASA's "Class C" Quality Assurance Program. It was vigorously tested, and the scientific goals can only be met by repeatedly sampling the high altitude charged particle auroral environment.

FAST is made of aluminum, with a single deck where the instruments and electric boxes are mounted. There are two magnetometer booms, located 180 degrees relative to each other in the spin plane. For launch they are stowed along the spin axis. There are several instruments deployed after launch, the two axial stacer electric field booms and the four radial, equally spaced wire electric field booms. The solar array attaches to a hollow shell in order to maximize the solar array area within the limits of the launch vehicle's shroud. It is a conducting solar array with circuitry, sealed inside a Faraday box. Figure 6 shows the satellite before launch.

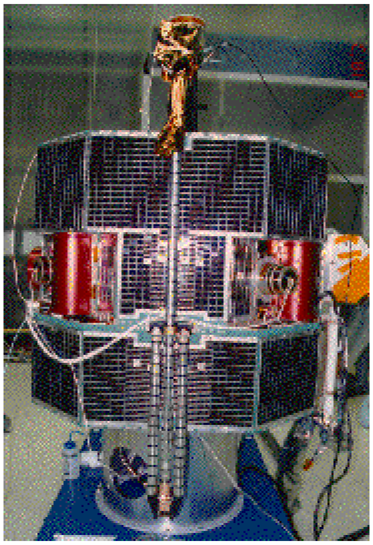


FIGURE 6: THE FAST SATELLITE BEFORE LAUNCH (FAST AURORAL SNAPSHOT EXPLORER)

There is an Attitude Control System on FAST which provides autonomous spin and precession control following separation from the launch vehicle, then provides spin and precession control during normal mode operations, in order to meet science imposed attitude requirements. It maintains the spin rate and the spin axis attitude consistent with the power and thermal requirement, and was designed to maintain the spacecraft attitude as a spinner with 12 revolutions per minute and to maintain the sun angle to less than 60 degrees if necessary.

Knowledge of the attitude is derived in a number of ways. The direction the spin axis is pointing is verified by orbit detrending between the modeled magnetic field and the observed magnetic field, with a nominal accuracy within 0.1 degrees. This allows accurate subtraction of the geomagnetic field from the measured magnetic field in order to obtain the perturbation magnetic field need for Poynting flux calculations. The spin phase error is much greater than 0.1 degrees in sunlight, due to frequent phase skips at the eclipse entry and exit.

FAST has multiple instruments to collect and store information. These instruments are the electron and ion electrostatic analyzers, an energetic ion instrument that distinguishes ion mass, and vector DC and wave electric and magnetic field instruments. Data collection occurs based on the idea that auroral processes only occur in limited bands, about five to ten degree wide sectors containing ovals of auroral light that surround the poles. The instruments are programmed to take "snapshots" of the auroral acceleration phenomena. The fields and the particle data are collected within the auroral acceleration region enabling cause and effect to be distinguished. One of the spin-plane electric field booms did not deploy correctly, however the satellite is still fully operational, with sensors on three booms which are sufficient to provide the spin plane electric field vector.

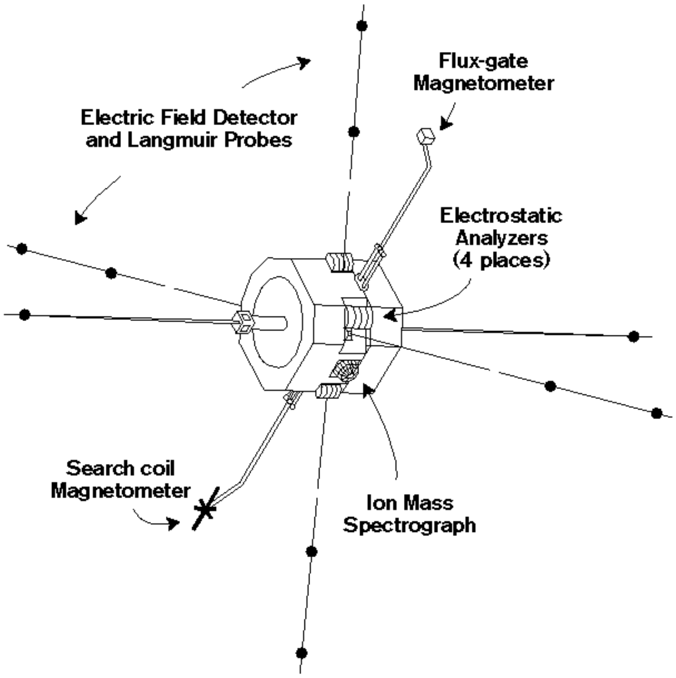

Flux-gate Magnetometer
Electric Field Detector and Langmuir Probes
Electrostatic Analyzers (4 places)
Search coil Magnetometer
Ion Mass Spectrograph

FIGURE 7: AN ILLUSTRATION OF THE FAST SATELLITE INSTRUMENTS (FAST AURORAL SNAPSHOT EXPLORER)

The electrostatic analyzers (ESAs) are quadrispherical and they measure ion and electron distribution functions. Particles enter the analyzer over a 360 degree field of view, are selected in energy, and imaged by the microchannel plate and discrete anodes. The 360 degree field of view is in the spacecraft spin plane, <u>aligned within six degrees of the magnetic field direction when the spacecraft is in the auroral zones.</u> The out of plane field of view is ten degrees for the electron sensor and twelve degrees for the ion sensor. The sensor heads are split into pairs of "half analyzers" which are located on opposite sides of the spacecraft. There are 16 of these half analyzers, arranged into four ESA stacks at 90 degree intervals around the spacecraft. Three of these are stepped ESA analyzers which gather high time resolution electron measurements at specific energies. The fourth is an ion or electron spectrometer which makes detailed, full pitch-angle distribution measurements.

The Time of flight Energy Angle Mass Spectrometer (TEAMS) is a high sensitivity, mass revolving ion spectrometer. It has an instantaneous 360 degree by 8 degree field of view, and was designed to measure the full three-dimensional distribution function of the major ion species within one half spin period of FAST.

The Magnetic Field instrument consists of a vector DC fluxgate magnetometer and a vector AC search coil magnetometer. The fluxgate magnetometer is a 3 axis instrument which uses highly stable low noise ring core sensors, boom-mounted at two meters from the body of the spacecraft in a shielded housing. The sensor electronics provide drive signals for the sensors, amplifies and detects the second harmonic signals that are proportional to the magnetic field, and digitizes the magnetic field information from DC to 100Hz using a 16-bit analog to digital converter. The search coil magnetometer is also a three axis sensor system, and it collects AC magnetic field data over a frequency range of 10Hz to 2.5kHz on two axes, and the third axis' range extends to 500kHz. The electronics further amplify the signals, and they are digitized using a 16-bit analog to digital converter.

The electric field instrument was designed to deploy ten spherical sensors, two each on four 28-meter radial wire booms, and one each on two axial stacers. One of the wire booms did not deploy properly. The electric field components are derived from the voltage difference between two spheres in voltage mode. In current mode, the sphere is biased to a fixed voltage and the preamplifier output represents the current in the sphere. There are many data acquisition modes: low frequency signal processing, high frequency signal processing, swept frequency analyzer, plasma wave tracker, broad band filters, wave-particle correlator, high speed burst memory, high frequency phase difference measurement, and fields data signal processor.

The collected data includes bursts, very high time resolution snapshots of data evaluated by the on-board flight computer. Only the events satisfying predetermined criterion are stored at maximum resolution.

The FAST Satellite is a highly-sophisticated satellite created to collect different forms of data in order to assist understanding of the auroral acceleration region and the particles, fields, and aspects of the atmosphere that create the lights and energy of the aurora. (Pfaff, Carlson and Watzin)

4.4 NSF PROPOSAL

In January of 2009, Doctors Russell Cosgrove and Hasan Bahcivan of SRI International submitted a proposal to the National Science Foundation (NSF). The NSF proposal laid out a 4 year plan for researchers at SRI to work with students at WPI to study the Earth's upper atmosphere. The aspect of the upper atmosphere that is of particular interest in this proposal is the complex interaction between the Earth's magnetic field and solar wind. Specifically, the Poynting flux and kinetic energy flux due to the interaction is what Cosgrove proposes to quantify. The Poynting flux and kinetic energy flux both cause heating in the upper atmosphere, and space weather models like Global Circulation Models (GCMs) require that this heating be accurately quantified in order to produce reasonably accurate outputs. It may be easiest to think of the heating in the upper atmosphere as the total energy that the upper atmosphere absorbs from solar wind. The goal of the NSF proposal is to find how much energy is actually absorbed by the atmosphere at any given time and attempt to fit a model to what is found.

In order to accurately quantify the Poynting Flux and kinetic energy flux (and hence the heating in the upper atmosphere), there are a number of measurements that need to be taken. Specifically, measurements of the electric field vector and magnetic field vector with respect to time are needed. The FAST satellite is the instrument that will supply these measurements. FAST researchers have accumulated a vast collection of data over the course of 20,000 orbits, most of which have data on the electric and magnetic fields. The majority of the work proposed for the NSF paper will be the study and processing of data from the FAST satellite.

Although there are already models that can give us approximations for the Earth's magnetic field, these are not the measurements that are needed to calculate the interaction. What is needed is an accurate measurement of the change in magnetic field due to outside sources. To start, measurements need to be taken in the upper atmosphere to get the total magnetic field (these are taken by FAST). Then, the Earth's magnetic field (which models already exist for) must be subtracted from the total magnetic field to get the magnetic field due to solar wind.

A problem also exists with models of electric fields in the Earth's atmosphere. Whereas the Earth's magnetic field is relatively constant with respect to space-time, electric-field models are highly probabilistic and can only give probabilities of how the electric field vector is aligned at any point in space-time. This aspect of electric field models is most commonly referred to as "electric field variability". Cosgrove and Bahcivan propose to overcome the problem of electric field variability by taking electric field measurements directly from the FAST satellite instead of from current electric field models.

It is important to note that although the FAST satellite measures both the magnetic and electric fields in the upper atmosphere, one cannot simply find the fluxes using those measurements alone. One important aspect of this proposal is the processing of the data. Because FAST is spin-stabilized, the measurements for the electric fields and magnetic fields appear to vary sinusoidally. This makes sense because even though FAST takes measurements continuously from its own constant reference frame (Satellite Reference Frame, SRF), from an outside reference frame, it is still spinning. So, once the data is collected, it must be processed to extract the sinusoidal component using algorithms provided by researchers at U.C. Berkeley.

As mentioned previously, Cosgrove and Bahcivan planned on this work being divided into four separate parts. Because they speculated that the majority of the work would be done by WPI students as part of their MQP, it was thought that these parts could be divided into four years. The first three years of the project will be done by students in conjunction with SRI researchers, and in the last year of the project, SRI researchers will compile all of the previous work to create models for Poynting flux and kinetic energy flux.

It should be noted that the first part of the NSF project will be WPI's Silicon Valley 2010 ECE MQP. The final deliverables for this year's project are values of the Poynting and kinetic energy flux as calculated from a small subset of orbits from the large data pool of approximately 20,000. Given that the majority of this report is written to detail this MQP, the remainder of this section will only be used to detail the last three parts of the project.

In the second year of the project, students and SRI researches will already have values for Poynting and kinetic energy flux for a small number of FAST orbits. The goal of the second part of the project is to develop a procedure for processing all of the FAST orbits up to a certain date. The processing of data includes the downloading, quality control, and computation of any required algorithms. The final deliverable for the second part of the project is to produce binned statistical models for the various fluxes in the upper atmosphere. SRI researchers will also help the students in finding parameters that describe geophysical properties and binning those as well. It is assumed that the second part of the NSF project will be done by WPI students as part of the Silicon Valley 2011 ECE MQP.

The third year of the project is assumed to be done by WPI students as part of the Silicon Valley 2012 ECE MQP. This group of students, given the binned statistical models produced by the second year students, will be required to expand on the statistical model by fitting an analytical model. They will also be finding the most expressive geophysical parameters as explained by SRI researchers. Given the difficulty with fitting such a large sum of data, it is thought that SRI researchers will be giving significantly more support to students than in parts one and two.

For the last part of the project, SRI researchers will finish the project. Finishing the project will include estimation of a spatial spectrum of the fluxes, and collaboration with researchers at other organizations in order to interpret the data. As a final deliverable, the models and interpretations of the Poynting flux data set will be made available to the public and shared with other researchers. It is not known whether any undergraduate students at WPI will be able to assist with the final part of the project. (Cosgrove)

4.5 SRI INTERNATIONAL

The project sponsor for this MQP is SRI International. Formerly known as Stanford Research Institute, SRI International is one of the world's largest contract research firms. SRI is an independent non-profit company dedicated to development of ideas for commercial entities, government agencies and other organization. SRI has conducted research for many US Government agencies including the Department of Defense and the National Institute of Health. Notable commercial business clients of SRI include Samsung, Toyota and VISA. SRI solves important problems through the use of science and technology. By solving these problems, SRI brings innovations to industry markets and major contributions to the lives of many people. These undertakings can be summed up in SRI's mission statement: "We are committed to discovery and to the application of science and technology for knowledge, commerce, prosperity, and peace."[1] SRI performs Research and Design (R&D) within many, many areas including communications and networks, education, energy within the environment, robotics and even homeland security and US national defense.

The MQP will be completed primarily within SRI's Engineering Systems Division (ESD) which is one of SRI's multiple R&D sectors. ESD is SRI's largest division and is in charge of performing various engineering research, system engineering and implementation, system testing, and system incorporation services. Because there are so many areas studied at ESD, the sector is further divided into the information systems, products & solutions, and sensing subdivisions. The Center for Geospace Studies (CGS) is the part of the company that concentrates on the research of the Earth's upper atmosphere and the environment just on the edge of space using radar diagnostics. Using a wide array of instruments and technologies, researchers at CGS will get a better understanding of natural phenomena relating to physics in the magnetosphere, middle atmosphere, lower thermosphere, stratosphere and troposphere that are poorly understood today. The MQP will be done mostly within CGS.

One of the more recent achievements within CGS is their creation of the Advanced Modular Incoherent Scatter Radar (AMISR). AMISR is a mobile radar diagnostic system that can track atmospheric measurements in real-time. CGS won SRI International's Presidential Achievement Award in 2004 for their success with this project. In February of 2008, a version of AMISR called the Poker Flat AMISR (PFISR) successfully completed its first year of nearly fully continuous operation. The previously unattainable amount of data collected was combined with that of other radar's get the high-latitude measurements.

4.6 REFERENCE FRAMES

A major part of the work for this MQP has to do with understanding reference frames. A reference frame is simply the way that an object is described in relation (in reference) to something else. For example, the two most common reference frames will be used to describe where and when this MQP report is being written: it is being written at SRI International in Menlo Park, California at 2:37 PM on Friday, January 22, 2010. Like the example, reference frames in this MQP describe two properties: position in time, and position in space. A reference frame transformation is needed when we are given data to describe an object in one reference frame and we need it in a different reference frame. There is a specific technique for transforming one reference frame to another, but this section will only describe the reference frames. Transformations will be discussed in a later section.

4.6.1 TIME FRAME

Of the two types of reference frame transformations, a time frame transformation is the easier than a spatial frame transformation. This is because time is described in only one dimension; therefore, the only calculations that need to be performed are scaling and offsetting; whereas, some of the spatial frame transformations require trigonometry and other higher level mathematic manipulation. We will begin the discussion of time frame transformations by first describing some of the various time frames that were used during this MQP. In discussing these time frames, we will note two important properties: the point of reference and the basic units. It should be noted that some of the constructs are used in multiple time-frames.

The most important thing to note about describing a point in time is that it is quite easy to do. Unfortunately, because of the nature of human perception, it is quite difficult to describe the same moment in time with consistency. Without digressing into a philosophy discussion, it should be stated that the time frames explained below (and many others) exist because of the variable nature of human perception and the simple fact that there is no universally measurable "beginning of time". Thus, different people in different eras of human history have decided on different conventions to describe the same concept: time.

4.6.1.1 COORDINATED UNIVERSAL TIME

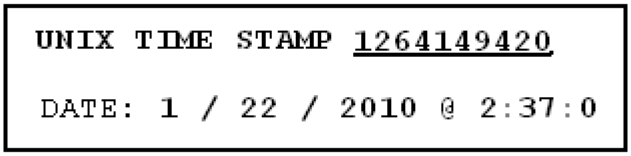
Coordinated universal time (UTC) is a time standard based on the passage of what scientists call "proper time." Proper time is a precise way of measuring the passage of time based on Einstein's theory of relativity. The theory explains that time passes at different rates dependent on the motion (speed) that a clock is moving. For example, let it be given that there are two clocks that are stationary with respect to one another, and synchronized in time. Then, if one of the clocks is accelerated for a certain amount of time, it can be seen that when the two clocks come together, the clock that was accelerated will show that less time has passed than the non-accelerated clock and the two clocks are no longer synchronized. UTC is the standard that best describes the passage of time on the surface of the Earth because it takes into account the Earth's rotation and Einstein's theories. It does this by adding "leap-seconds" at irregular intervals to balance for the slowing rotation of the Earth with respect to the cosmos.

UTC is also the basis for the use of the 24 time-zones used around the world. UTC is thought of as a zero-offset and all of the other time-zones as a positive or negative offset of UTC. A city commonly associated with UTC is Greenwich, England. Greenwich lies at 0 degrees longitude (the prime meridian) and is the site of the Royal Observatory. As can be seen now and will be seen later, Greenwich is an important reference point in both time and space. For the purposes of this MQP, it is important to take into account which time-zone a particular point in time is measured. Ideally, all of the data given and all of the calculations would be done with respect to UTC. An example: 2:37 PM in Menlo Park, California is in the Pacific Time Zone (PST), where PST = UTC -8, so 2:37 PM PST = 10:37 PM UTC.

There are a few other time standards that should be mentioned that are related to UTC. If one is not concerned with differences of less than a fraction of one second, then one could also use Greenwich Mean Time (GST), or one of the other variations of "Universal Time (UT)" such as UT1, where time is calculated from measurements taken from the Sun.

4.6.1.2 UNIX TIME

UNIX time is a format for describing time. It is the first system that will be described that uses a very specific “epoch” or point of reference in time. The epoch for UNIX time is midnight on January 1, 1970 UTC. UNIX time describes the number of seconds that have passed (in UTC) since the UNIX epoch. As shown in Figure 8, it looks just like a 10-digit number today (9-digits before September 9, 2001). If a UNIX time clock were to be watched today, a 10-digit number increase by 1 every second would be seen. It is important to note that UNIX time does not take into account the leap seconds that UTC takes into account. Therefore, leap seconds are periodically added into UNIX time manually. If one were to watch when such an insertion were added, one would see the UNIX clock repeat one second while the UTC clock inserts a “60th second.” Although most UNIX clocks only show an integer representation of the number of seconds since the UNIX epoch, some may show decimal points up to a certain resolution. The decimal points are the fractions of a second between seconds.



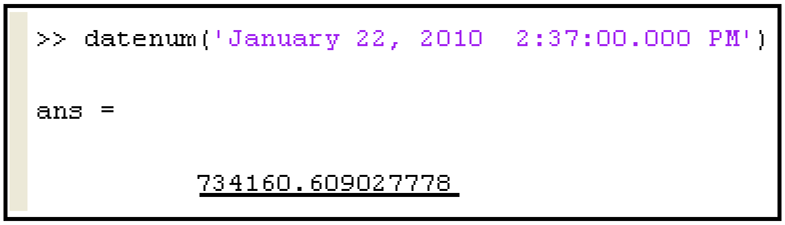
```
UNIX TIME STAMP 1264149420

DATE: 1 / 22 / 2010 @ 2:37:0
```

FIGURE 8: UNIX TIMESTAMP FOR 1/22/2010 2:37 AM UTC

UNIX time plays an important role in this MQP because it is necessary to correlate measurements taken from the FAST satellite with the position of the satellite in time, and the time that is given is measured in UNIX time.

4.6.1.3 MATLAB TIME

MATLAB time is another format for describing time. MATLAB time or MATLAB serial date number is similar to UNIX time in that it is only a number (along with any fraction of a number after the decimal place). It differs from UNIX time in two ways, though. First, the MATLAB epoch is the date January, 1 0000 at 00:00:00. Second, MATLAB serial date number describes the number of days since the epoch instead of the number of seconds like UNIX time. For example, the command “*datenum('January 22, 2010 2:37:00.000 PM')*” returns the offset for the number and fraction of the number of days between the MATLAB epoch and 2:37 PM on January 22 in the user’s specific time-zone (whether it be 2:37 PM Pacific Time or 2:37 UTC). This means that MATLAB time is not universally standardized like UNIX time, it is computed locally. This means that two computers running the “*datenum*” command at the exact same point in time, but in different time-zones, the command would return slightly different numbers. Given this information, extra care must be taken when converting between UNIX time and MATLAB time. Specifically, for any MATLAB serial date numbers converted to UNIX time, it must be made sure that the MATLAB serial date number is for UTC.



```
>> datenum('January 22, 2010  2:37:00.000 PM')

ans =

          734160.609027778
```

FIGURE 9: MATLAB SERIAL DATE NUMBER TIME-STAMP

MATLAB time is important in this MQP because MATLAB is used to convert UNIX times into different formats.

4.6.1.4 GREGORIAN AND JULIAN TIME

Although Gregorian and Julian time are both standards for measuring dates in time (like UTC, UNIX Time and MATLAB Time), they are different in the sense that they measure time on a longer scale. Gregorian and Julian time are calendar standards rather than time standards meaning that they are more concerned with naming conventions and long-term time-keeping than fractions of a second difference.

The Julian calendar was historically one of the first calendars to be put into use. It originates from the era of Julius Cesar around 46 B.C. The Julian calendar used Babylonian science to construct a solar year of 12 months divided into 365 days. In fact, the average year in the Julian calendar was exactly 365.25 days when taking leap years into account. This is slightly different from what scientists found to be the actual number of days in a solar year (about 365.24).

The Gregorian calendar was designed by astronomers in the 16th century to reconcile the slight error in Julian time-keeping. Since that time, it has become an international standard for calendars and the difference between the Julian Date and the Gregorian Date has continued to increase. For example, in the year 2000, the Julian and Gregorian calendars differ by about 13.5 days.

Julian and Gregorian calendars are important consideration for this MQP because even though the Gregorian calendar is the modern standard, some astronomical calculations require use of Julian time-keeping. Specifically, calculation of a “Julian Century” is required to find the “Greenwich Hour Angle.” This will be discussed in more detail later.

4.6.2 SPATIAL FRAMES

The second type of transformation that is done in this MQP is a spatial frame transformation. Spatial frame transformations are a little bit more difficult than time frame transformations because space is described in three dimensions instead of just one. Thus, there are more calculations necessary. This section of the report will begin by explaining the basic types of coordinate systems, and then will explore the various spatial frames of reference that are used in this MQP.

4.6.2.1 COORDINATE SYSTEMS

Before explaining the spatial reference frames that are used for the MQP, a distinction must be made between the basic types of coordinate systems that are used for this project. Cartesian and spherical coordinates are described below, as they are the most relevant systems to this MQP.

A 3-dimensional Cartesian coordinate system is the most common way to describe points in 3-d space. A Cartesian coordinate system is most powerful in describing lines and vectors that are straight (i.e. there is no curvature). The system is defined by three orthogonal axes which are usually labeled X, Y, and Z. All three axes intersect at exactly one point in space called the origin. The origin is what we may refer to as the reference. All points in space that are described by a Cartesian triple (i.e. {x,y,z}) are a certain distance away from the reference. It is important to keep this fact in mind when working with different Cartesian coordinate systems. Cartesian coordinate systems are powerful in describing straight lines, rectangular planes and less powerful in describing arcs, circles, semicircles and circular objects like spheres.

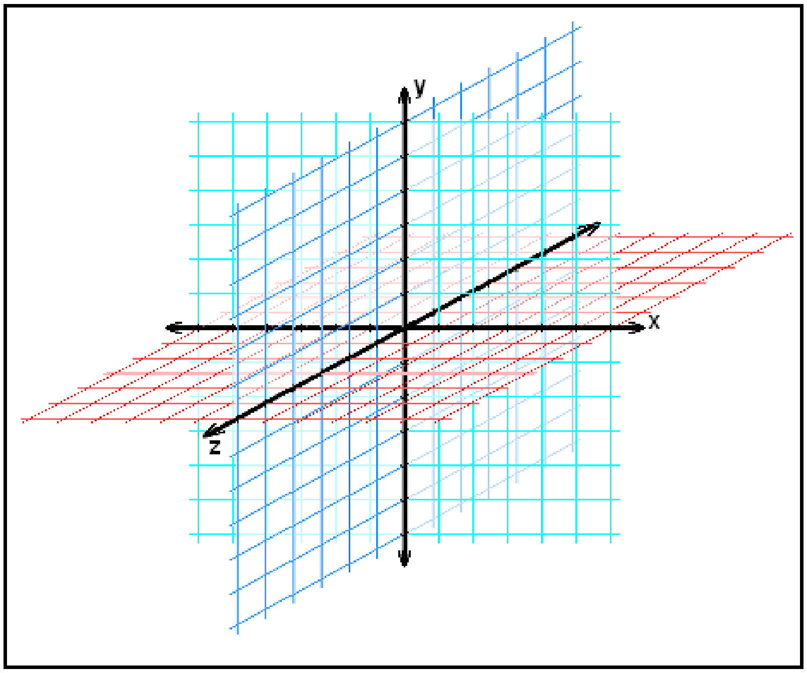


FIGURE 10: THREE-DIMENSIONAL CARTESIAN COORDINATE SYSTEM

The second type of coordinate system is most powerful when describing curved lines, circles and spheres. Hence it is called a spherical coordinate system. Spherical coordinates make use of the fact that circles and spheres can be easily described by a distance dimension and two angle dimensions; whereas, a Cartesian coordinate system does not require angles, but does require three distance dimensions. A spherical coordinate system contains an origin and a reference axis, A, and a plane through the reference axis. The distance between the origin and the point is the radius, rho. The angle from the reference axis, A, and the point is the elevation, theta. Finally, the last dimension is the angle from the reference axis plane and the point is called the azimuth, phi. Figure 11 shows an example of a spherical coordinate system.

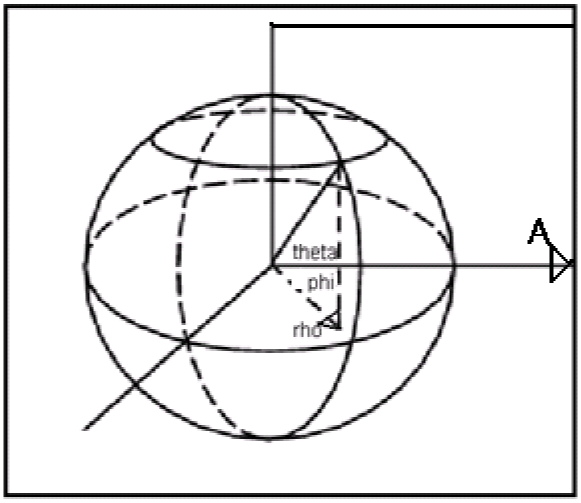


FIGURE 11: SPHERICAL COORDINATE SYSTEM

4.6.2.2 GEOCENTRIC EQUATORIAL INERTIAL

Now that the difference between a Cartesian coordinate system and a spherical coordinate system has been established, the process of explaining some of the common uses of those systems within the context of this MQP can be described, beginning with the Geocentric Equatorial Inertial (GEI) coordinate system. The GEI system is a Cartesian coordinate system. The origin of the GEI system is located at the center of the Earth, hence the name geocentric. The Z-axis is positive through the Earth's geographic North Pole. The X-axis is positive in the Earth's equatorial plane, toward the first point in the Aries constellation. The Y-axis is any axis which completes the Cartesian system. The GEI coordinate system can be thought of as more of a macroscopic system because it does not rotate with the Earth. In the GEI system, a point on the Earth's surface changes with the rotation of the Earth. Figure 12 shows a diagram of how the GEI system is laid out in relation to the Earth.

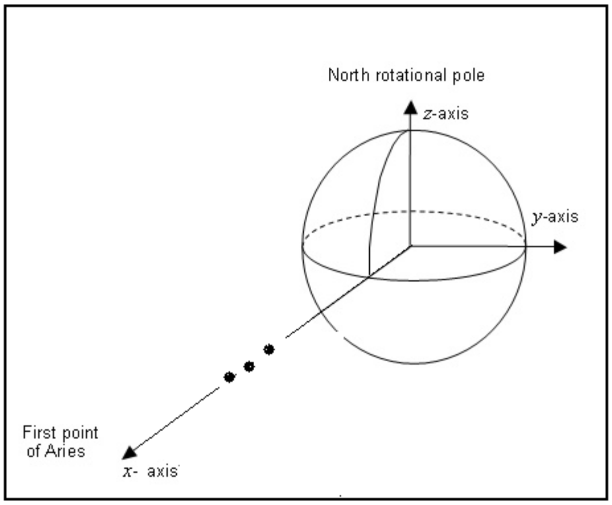


FIGURE 12: GEI COORDINATE SYSTEM

The GEI system is important because all of the positional data available for the FAST satellite is in the GEI coordinate system.

4.6.2.3 GEOCENTRIC GEOGRAPHIC (GEO)

Similar to the GEI system, the Geographic Geocentric (GEO) coordinate system is a Cartesian coordinate system with the origin located at the center of the Earth and the Z-axis is positive through the geographic North Pole. The main difference between the GEI and GEO systems is that in GEO, the X-axis is positive through the intersection of the equatorial plane and the prime meridian (0 degrees longitude). The Y-axis completes the coordinate system. Unlike the GEI system, the GEO system rotates along with the Earth; therefore, a point on the surface of the Earth will never change because the point and the system are moving together. It is important to realize that the GEI and GEO systems are very similar except for the fact that they are rotations of each other about the their Z-axes. The rotation angle has many names, but will be referred to here as the Greenwich Hour Angle (GHA). The GHA is the angle between the x-axis in the GEI system and the x-axis in the GEO system. Because the earth rotates about once every 24 hours, this angle increases at a steady rate from 0 and 360 degrees in those 24 hours.

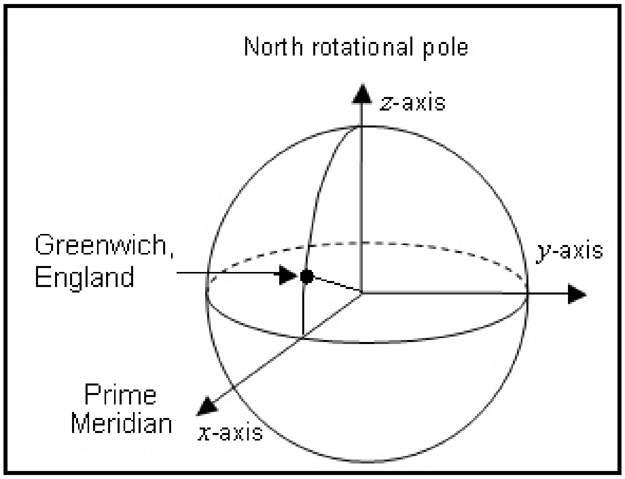


FIGURE 13: GEO COORDINATE SYSTEM

The GEO system is the system in which all of our final positional coordinate data will be. The GEO system is useful for this MQP because it moves with the rotation of the Earth, this is ideal because a goal is to define space in relation to the Earth as if Earth was a constant. All of the positional data for Poynting Flux and Kinetic energy flux will also be in the GEO system. In year 2 of the project the data will be transformed into magnetic coordinates, which have the z axis through the magnetic poles, however this transformation is beyond the scope of this MQP.

4.6.2.4 LATITUDE, LONGITUDE, ALTITUDE

The last coordinate system to be discussed is the geographic latitude, longitude, and altitude system, called the geographic system for short for this section of the report. Although it may not be apparent, this is actually a spherical coordinate system. In this system, altitude (from the center of the Earth) can be thought of as the radius, rho. Lines of latitude are like the angle between the reference axis (line going from center of earth through surface of Earth at intersection of Prime Meridian and Equator) and the point in question, theta (lambda in the picture). Finally, lines of latitude can be thought of as the angle between the reference plane (the Earth's Meridional plane) and the point in question, phi.

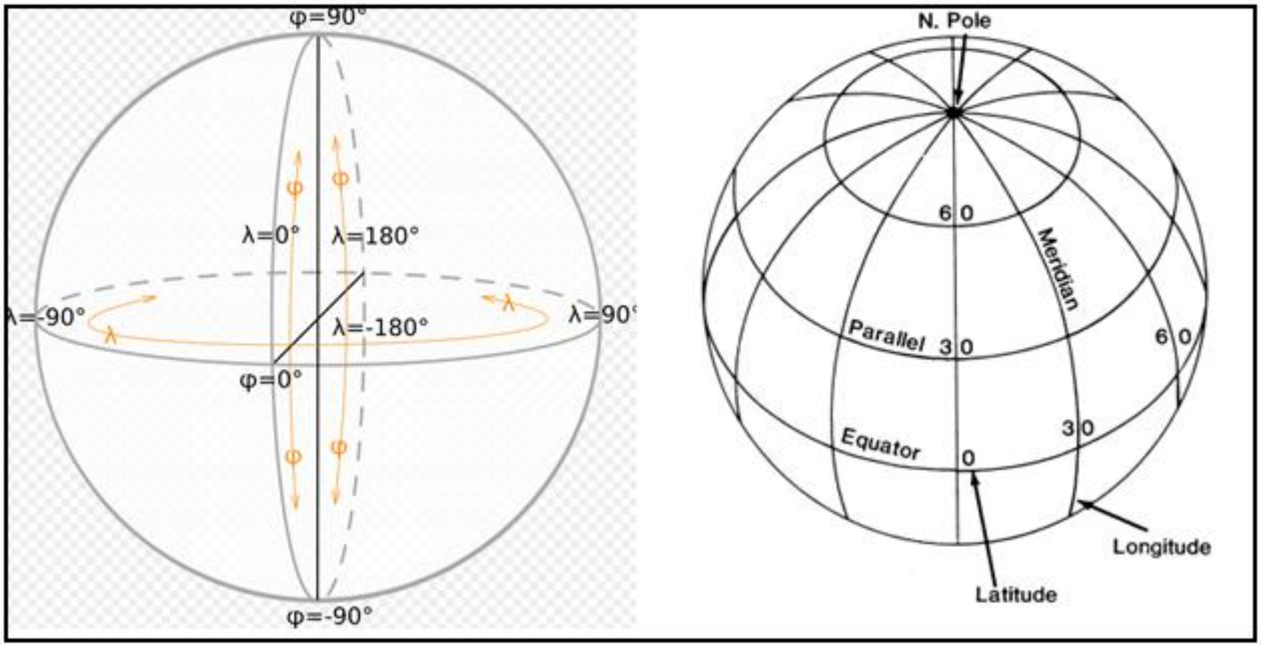


FIGURE 14: GEOGRAPHIC COORDINATE SYSTEM

Although the geographic coordinate system is not being used for this MQP, it is important to understand how it fits within the scheme of spatial reference frames.

4.6.2.5 FAST COORDINATE SYSTEM

The FAST coordinate system is used when the satellite telemeters data to the ground. The data sent is in its “raw” form meaning that it has had minimal processing between the phase where the actual measurements were taken to the time the data was sent to Earth. The FAST coordinate system is a Cartesian coordinate system with an origin at the center of the FAST satellite, seen in Figure 15. The X-axis of the system points positive almost perpendicularly away from the surface of the Earth. The Y-axis is almost parallel to the surface of the Earth and points positive in the opposite direction of the satellite’s trajectory. Together, the X and Y axes make up the satellite’s spin plane. By definition, the Z-axis is perpendicular to the spin plane, and is the axis around which FAST is spinning. The Z-axis is also almost parallel to the surface of the Earth. In the figure shown below, the Z-axis is coming out of the page.

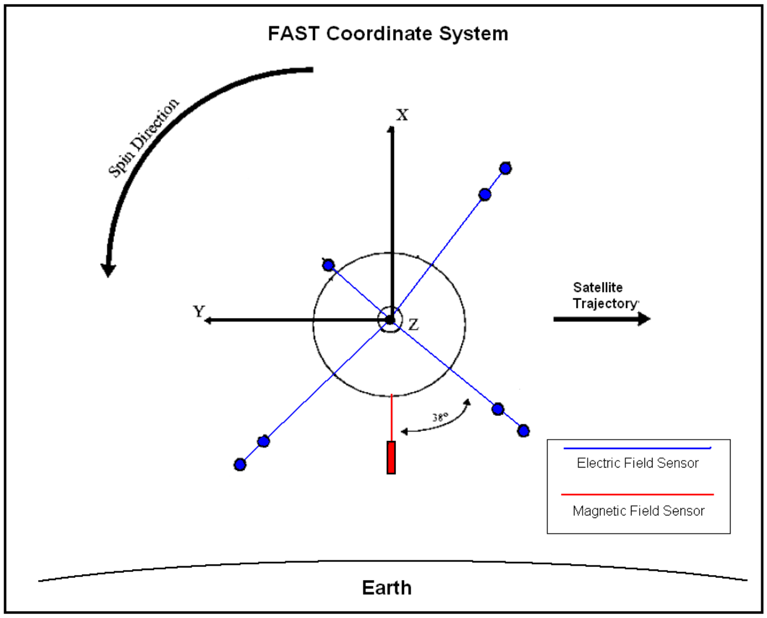


FIGURE 15: FAST COORDINATE SYSTEM WITH RESPECT TO SATELLITE

The reason that the X-axis is almost perpendicular to the surface of the Earth, and why the Y and Z axes are almost parallel is because the satellite coordinate system is based on the instantaneous trajectory of the satellite. Because the satellite follows an elliptical orbit, the instantaneous trajectory of the satellite is not always perfectly parallel to the ground. Figure 16 below shows FAST in three instances of its orbit. It is important to notice that the FAST coordinate system rotates with along with FAST's trajectory.

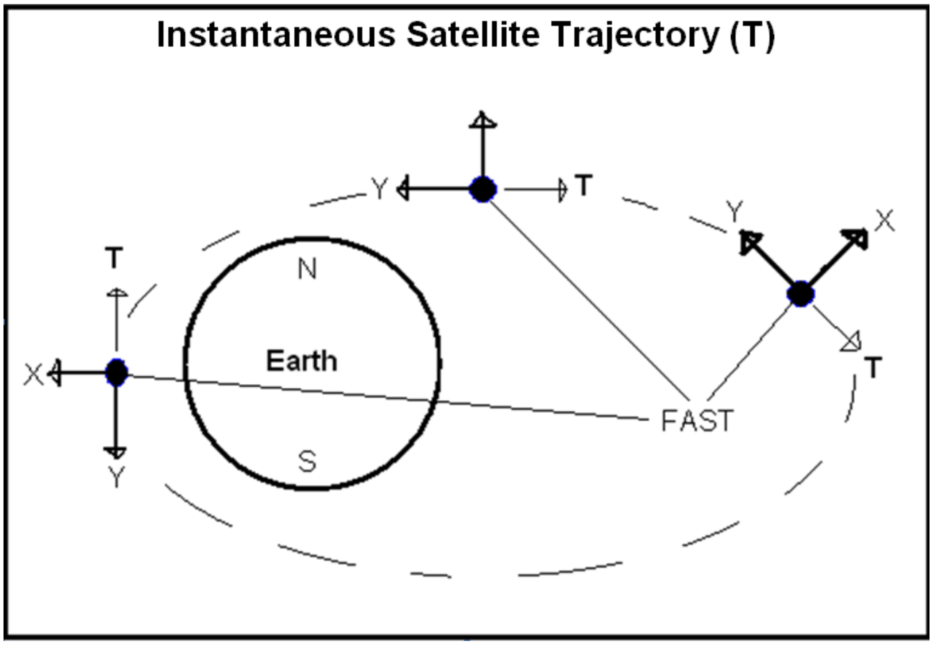


FIGURE 16: FAST COORDINATE SYSTEM WITH RESPECT TO TRAJECTORY

4.6.2.6 DATA COORDINATE SYSTEM

The last coordinate system is what students will call the "Data Coordinate System (DCS)," the data coordinate system is the system in which the summary data created by FAST researchers is expressed. Just as with all of the other Cartesian coordinate systems, it is defined by the orientation of the three axes. A visual representation of the DCS is shown in Figure 17.

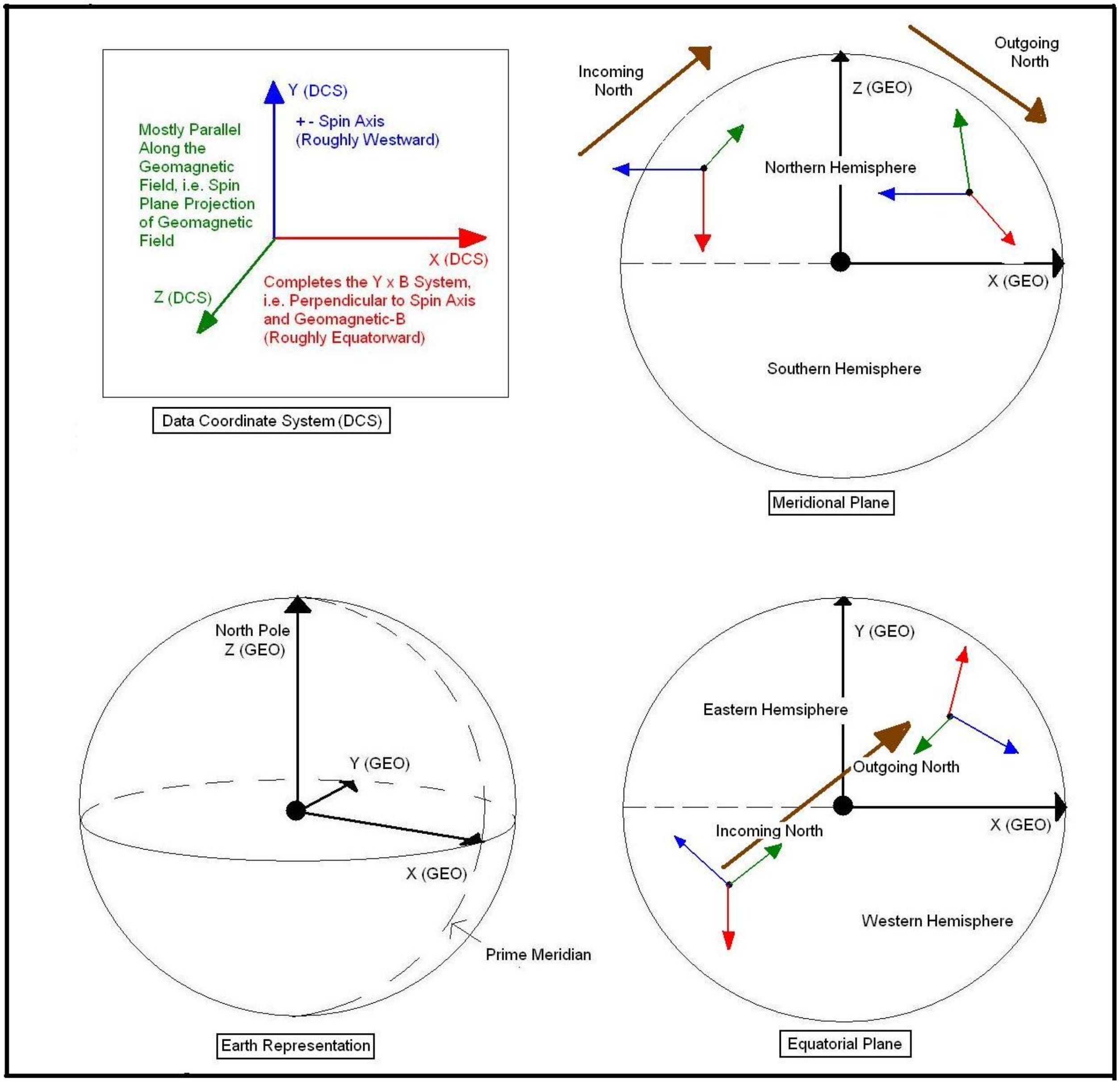


FIGURE 17: DATA COORDINATE SYSTEM

In the DCS, the X-axis is defined as the axis that lies in the spin plane of the satellite and perpendicular to the projection of the geomagnetic field onto the spin plane. The X-axis is mostly positive in an equator-ward direction (south at the North Pole and north in the South Pole) The Y-axis is defined as being along the spin axis of FAST. It's very important to note that the spin-axis is defined as positive along the Y-axis points in a westward direction (note this in Figure 17) when the satellite is outgoing north/incoming south. However, the spin-axis is defined as negative along the Y-axis when the satellite is moving incoming north/outgoing south. The Z-axis is the projection of the geomagnetic field onto the spin-plane. It mostly points positive in the direction of the geomagnetic field. It should be noted that, given the properties of the geomagnetic field, the Z-axis points mostly north, and as FAST instantaneously crosses the poles, gradually switches direction (note this in Figure 17), this is different from the X and Y axes, which actually contain discontinuities at the poles, where the switch directions. For more information on the Data Coordinate System and how it relates to the other systems, please see the short video submitted with this report entitled “Coordinate Systems”.

5. PROJECT EXPLANATION AND GOALS

5.1 PROBLEM STATEMENT

This project can be broken into two sections: Poynting Flux and Kinetic Energy Flux. Using these two models of energy flux in the atmosphere, Drs. Cosgrove and Bahcivan will be able to create new General Circulation Models to further understand near-Earth space. The main problem is electric field variability, which prevents scientists from using the electric field models to quantify the energy contributed by the solar wind, which interacts with the geomagnetic field in order to get to the thermosphere. In this, Year 1 of the project, the problem is to create routines to process a small number of FAST satellite orbits for Poynting Flux and Kinetic Energy Flux. These routines will be used in Year 2 of the project to process the rest of the orbits for this data.

5.1.1 ELECTRIC FIELD VARIABILITY

Electric Field Variability, as mentioned in the background section, is the main problem to solve in this project. To put it simply, one can take the average electric field and square it, or can square the electric field and then take the average of that, and those two averages will not be equal. The end result of the entire NSF proposal will involve solving this problem to appropriately create General Circulation Models.

5.1.2 EARTH'S MAGNETIC FIELD

The Earth's Magnetic Field is a strong field that points down at the magnetic South Pole and up at the magnetic north pole, which is slightly different than the geographic north and south poles. The geographic north and south poles are located at 90 degrees and -90 degrees latitude respectively, and the magnetic poles move over time. The magnetic north pole is currently located at approximately 82.7 degrees north and 114.4 degrees west (North Magnetic Pole- Wikipedia, the free encyclopedia), and the magnetic south pole is located at approximately 63.5 degrees south and 138 degrees east (South Magnetic Pole- Wikipedia, the free encyclopedia).

The Magnetic Field is modeled by what is called the International Geomagnetic Reference Field (IGRF) model. This model is update regularly. Due to the time span that the data for this project was collected, the IGRF 1995 model is the one used.

5.1.3 POYNTING FLUX

The Poynting Vector (or Poynting Flux) can be thought of as the amount of energy in an electromagnetic field. It is measured in units of Watts per square meter. It is calculated by taking the cross product of the Electric field (E) and the Magnetic Field Intensity (H). (Poynting Vector- Wikipedia, the free encyclopedia)

The data in this project is expressed in terms of the Electric field (E) and the Magnetic Field (B). The processing routines that allow the data to be used in MATLAB (more information in Section 6.1) remove the

Earth's magnetic field from the collected raw data from the satellite, which results in the δB, or fluctuation in the magnetic field, that is required for this project. The goal is to achieve the Poynting Flux provided by solar wind and other space plasmas, not including the magnetic field input from the Earth.

To convert the δB into H, all that is required is to divide the magnetic field by μ_0, the permeability of free space, which is equal to $4\pi*10^{-7}$ Henry per meter. B is measured in Tesla; H is measured in Amperes per meter.

It is quite transparent that by taking the cross product of E and H, one would be crossing the units of Volts per meter with Amperes per meter, which clearly results in Watts per meter squared.

5.1.4 KINETIC ENERGY FLUX

Kinetic energy flux is similar to Poynting Flux as they are both types of energy through a certain amount of area; however where the Poynting Flux involves the ways in which fields interact with one another, the kinetic energy flux involves the energy in electrons and ions caused by the interaction of Earth's magnetic field with the solar wind. The data available is energy flux of the electrons and ions, with the data being positive toward Earth. The units are ergs/cm^2-s, which will need to be converted into Joules/m^2-s. One erg is 10^{-7} Joules, and one cm^2 is 10^{-4} m^2, so to convert simply involves multiplying the values by 10^{-3}, resulting in the desired units.

5.1.5 COORDINATE SYSTEMS

As mentioned in Section 4.5, this project involves conversions between various coordinate systems. The positional data of the satellite is given in GEI coordinates and the fields' data is given in a different coordinate system based on the position of the satellite with respect to the Earth and the geomagnetic field, and these need to be converted into the GEO coordinate system in order for the calculations to make sense in reference to one another and in reference to the Earth. To extract altitude from this data, the GEO coordinates are converted to Geodetic coordinates, which gives the latitude and longitude of the satellite. The latitude is then used to extract the radius of the Earth at the different points, and then that radius is subtracted from the magnitude of the GEO coordinates which results in the altitude. These conversions are necessary to find the location of the field data to properly represent the various events that can happen due to Poynting Flux.

5.2 GENERAL CIRCULATION MODELS

The final goal of this project at the end of Year 4 is to create models that will help create general circulation models, which will help specify thermospheric dynamics and densities, which helps achieve a goal of the National Space Weather Program.

General Circulation Models are models of space weather involving thermospheric circulation and temperature. These models were originally created using only solar radiation, but have been improved to include energy from the interaction of ions with the magnetosphere. These models describe global temperature and density in the thermosphere, and model what is known about the upper atmosphere. (Cosgrove)

5.3 GOALS

The goal of Year 1 is to achieve a method for using the data available from the FAST satellite to compute Poynting Flux and Kinetic Energy Flux. These methods will be saved for use in future years of this project and will be instrumental in creating the GCMs that the final result of the NSF proposal has promised. The data from this project will greatly improve the models of energy input from the solar wind in these GCMs, in terms of using the Poynting flux and the kinetic energy flux separately or totaled. (Cosgrove)

6. METHODOLOGY AND IMPLEMENTATION

This section of the report will be covering the methods, processes, and different ways of achieving the top-level goals of the MQP. The first half of this section will cover the two separate but similar routes in calculating Poynting flux from the data taken from FAST, and all of the methodologies within those two routes. The latter half of this section will cover the methodologies for calculating kinetic energy flux. This section will describe, in detail, how all of the project's already existing resources were utilized, any new routines that were created, and how they all fit together.

As complements to the methodologies described, a short section on the students' setbacks will also be included. It should also be noted that all of the processes and methods described in this section were created as needed as the group moved ahead in the project.

6.1 CALCULATION OF POYNTING FLUX

This subsection of the report will go over the two ways to process for Poynting Flux. Although the two processes are different, they both share similar elements. The difference is in the way that the elements are executed. For the purposes of this report, the processes will be named after how the data from FAST was obtained, so the first process will be named "Downloading Raw Data Using Satellite Data Tool (SDT)," and the second process will be named "Downloading Summary Data from the Berkeley Web Archives." Figure 18 contains a flow chart of the two different methods by which the data can be obtained and processed.

It should be mentioned that these two processes rely on work already done by the FAST science team for archiving raw data from FAST. The FAST team took it upon themselves to create a process for obtaining and archiving all of the data from the FAST satellite. They have also made all of the data and processes for obtaining their archived data available to the public. The intricate details of how these processes work is outside the scope of this MQP, so it will suffice to say that the team has created a FAST "data bank" that holds all of the information from FAST. The data bank holds all of the electric field data, magnetic field data, satellite orbit information, electron density measurements, and many other data all of which come in various resolutions. Both the process for downloading raw data using SDT and downloading summary data from Berkeley rely on the fact that FAST data is available in the data banks.

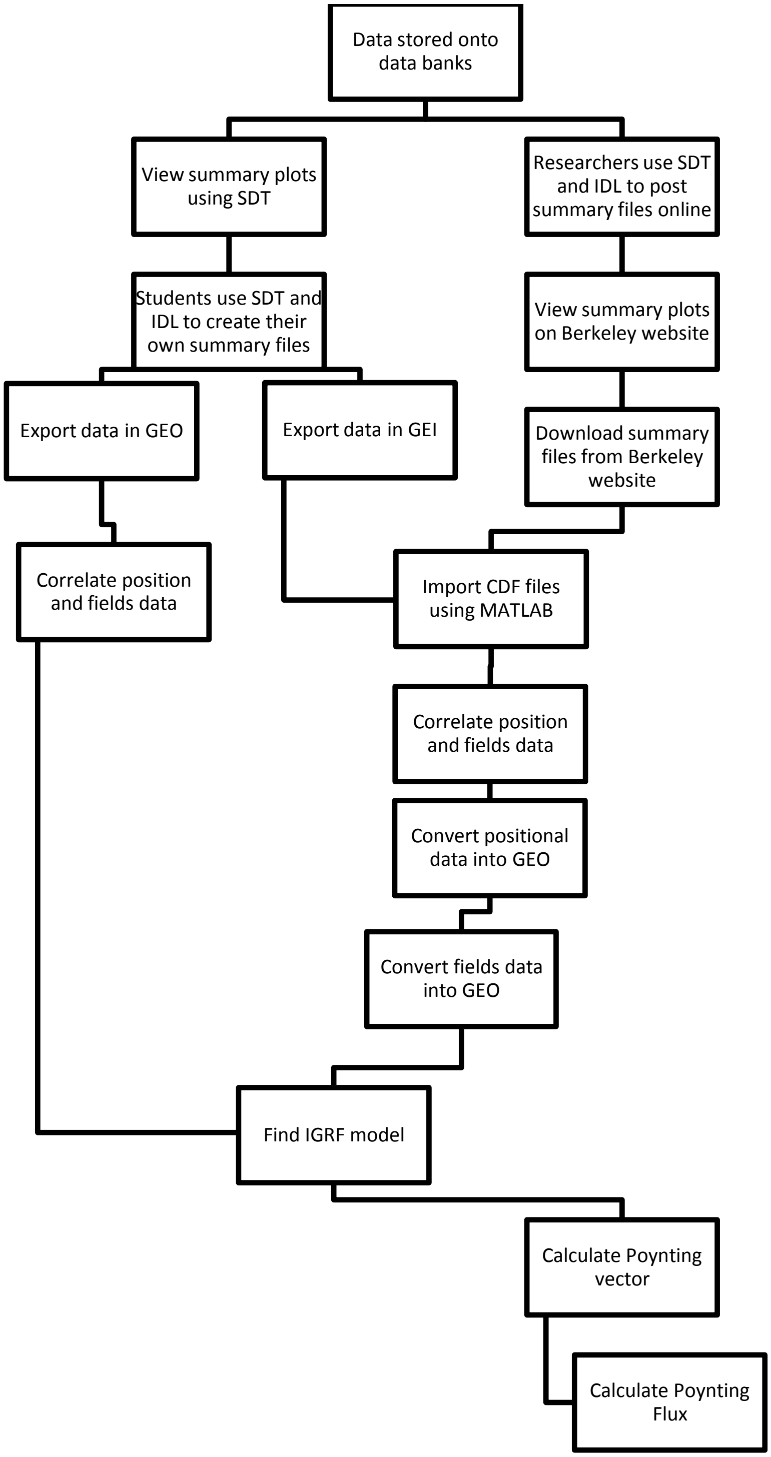

Data stored onto data banks
View summary plots using SDT
Researchers use SDT and IDL to post summary files online
Students use SDT and IDL to create their own summary files
View summary plots on Berkeley website
Export data in GEO
Export data in GEI
Download summary files from Berkeley website
Correlate position and fields data
Import CDF files using MATLAB
Correlate position and fields data
Convert positional data into GEO
Convert fields data into GEO
Find IGRF model
Calculate Poynting vector
Calculate Poynting Flux

FIGURE 18: DATA PROCESSING FLOW CHART

6.1.1 VIEWING RAW DATA USING SATELLITE DATA TOOL

The Satellite Data Tool (SDT) was developed as the primary method of viewing and downloading data. It is a resource for researchers working with many satellites including the CRRES, Polar, Cluster II as well as the FAST missions. SDT is a UNIX based software that interfaces with multiple other software packages to download data from the FAST data bank, view it, and export it. SDT can actually be considered a suite of utilities, all of which work together to allow users access to the raw measurements taken from the FAST satellite. In this subsection, the process of viewing raw data plots on SDT will be discussed. Viewing the plots allows users to examine a specific time period for interesting events. If a period contains an interesting event, the user can either export the data using SDT or download the summary data provided online.

Upon starting SDT, the user is given a choice to select a "configuration." In the context of SDT, a configuration is a default selection of desired raw data plots within a certain time-frame, usually chosen by orbit number. The plots are visual representations of the raw measurements taken by FAST. There are over one hundred plot types to choose from, but they can roughly be divided into 6 groups: electrons (EES), ions (IES), mass spectrometer readings (TEAMS), AC Fields, DC Fields, and orbit/ephemeris (positional) data. Once a configuration is chosen, the plots shown (as well as the time-frame) can be changed at any time. The configuration is simply a default.

The desired plots for finding Poynting Flux are the DC Fields (magnetic and electric). Shown in Figure 19, are the raw DC magnetic Field measurements taken by FAST, and as such are in the FAST coordinate system described above. Note that the sinusoidal waves for the X and Y components are a result of the spinning of the satellite. The raw data needs to be "despun" in order to get useable data. At this point the user can change/add/remove plots, or choose another time period to view using the Data Manager tool and selecting by different criterion, such as orbit number, date and time, altitude, and others. However, the process for all of these options is time-consuming because the data is stored off-site, so there is a delay when performing these operations. Thus, a much more time-effective way of viewing data is described in section labeled "6.1.3 Viewing Summary Plots from Berkeley Website". Once a period of time is decided to contain an interesting event, the user must obtain the data corresponding to that period of time. This is done by either exporting the raw data from SDT or by downloading the summary data from the Berkeley web archives.

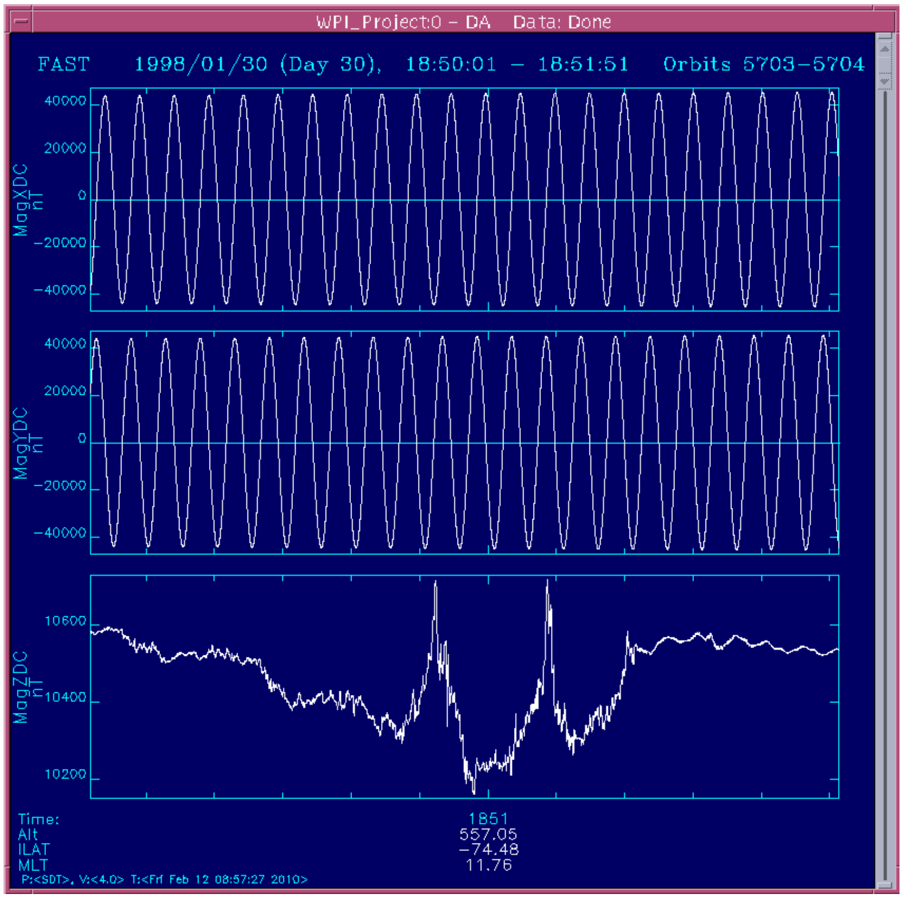

WPI_Project:0 – DA Data: Done
FAST 1998/01/30 (Day 30), 18:50:01 – 18:51:51 Orbits 5703–5704
40000
20000
0
–20000
–40000
MagXDC nT
MagYDC nT
MagZDC nT
10600
10400
10200
Time:
Alt
ILAT
MLT
1851
557.05
–74.48
11.76
P:<SDT>, V:<4.0> T:<Fri Feb 12 08:57:27 2010>

FIGURE 19: DC MAGNETIC FIELD PLOTS IN SDT

6.1.2 DOWNLOADING RAW DATA USING SATELLITE DATA TOOL

Once the user has decided that a period of time contains an interesting event (done by either viewing plots on SDT or viewing summary plots online), the user can obtain the actual data corresponding to that event using SDT. The user can choose to export the data for any period or length of time. To export data, the user must be viewing the desired plots and period of time. There are a number of export formats to choose from, but CDF files have been the choice for this MQP. The CDF exported files are downloaded and saved to the local UNIX workstation at SRI where they can be manipulated and changed in any fashion the user wishes.

One important note is that because the data is stored electronically off-site at Berkeley, there is a delay when exporting data using SDT. Also, because the data is raw (has not been processed), there is extra work to be done after exporting the data in order to calculate the Poynting flux. Thus, a more time-efficient way of obtaining FAST data is described in the section labeled "6.1.4 Downloading Summary Data from Berkeley Web Archives". However, by using summary data (data processed by Berkeley researchers), the user is forced to depend on processing techniques that may or may not be correct, per communication with Doctor Robert Strangeway, UCLA. Ideally, the user would obtain raw data and process it himself/herself.

6.1.3 VIEWING SUMMARY PLOTS FROM BERKELEY WEBSITE

The second method of viewing FAST data comes in the form of "summary plots" formatted as GIF images. These images are posted on the FAST website. The GIF images have been created by researchers at

Berkeley. The researchers used SDT in combination with IDL routines to process the raw data from FAST. Researchers then posted summaries of processed data online. The section titled "6.1.4 Downloading Summary Data from Berkeley Web Archives" explains the summary data in greater detail. Along with creating and posting summary data, the researchers created and posted summary plots. The summary plots are visual representations of processed FAST data similar to the plots provided by SDT of the raw data. By posting the summary plots online, Berkeley researchers made it easy to quickly browse through processed data and decide which orbits or which time-frames are interesting. An example of a summary plot is shown in Figure 20.

The reasons for viewing the summary data are the same as for viewing the raw data (in SDT): the user would like to see which periods of time contain interesting events. Once the user has decided which periods of time contain interesting events, they can move on to obtaining the data corresponding to those times. The user can obtain the data by either exporting the raw data from SDT or by downloading the summary data from the Berkeley web archives.

Choosing which orbits this year of the project would focus on was not done by any kind of scientific process. The top right corner of the summary plot page has buttons where one can easily scroll through each orbit by date or orbit number depending on criteria. The available summary plots were each looked at individually and certain orbit numbers were recorded as "interesting". "Interesting" orbits were determined by visual inspection of the electric and magnetic fields, and any orbits with strong fluctuations were noted.

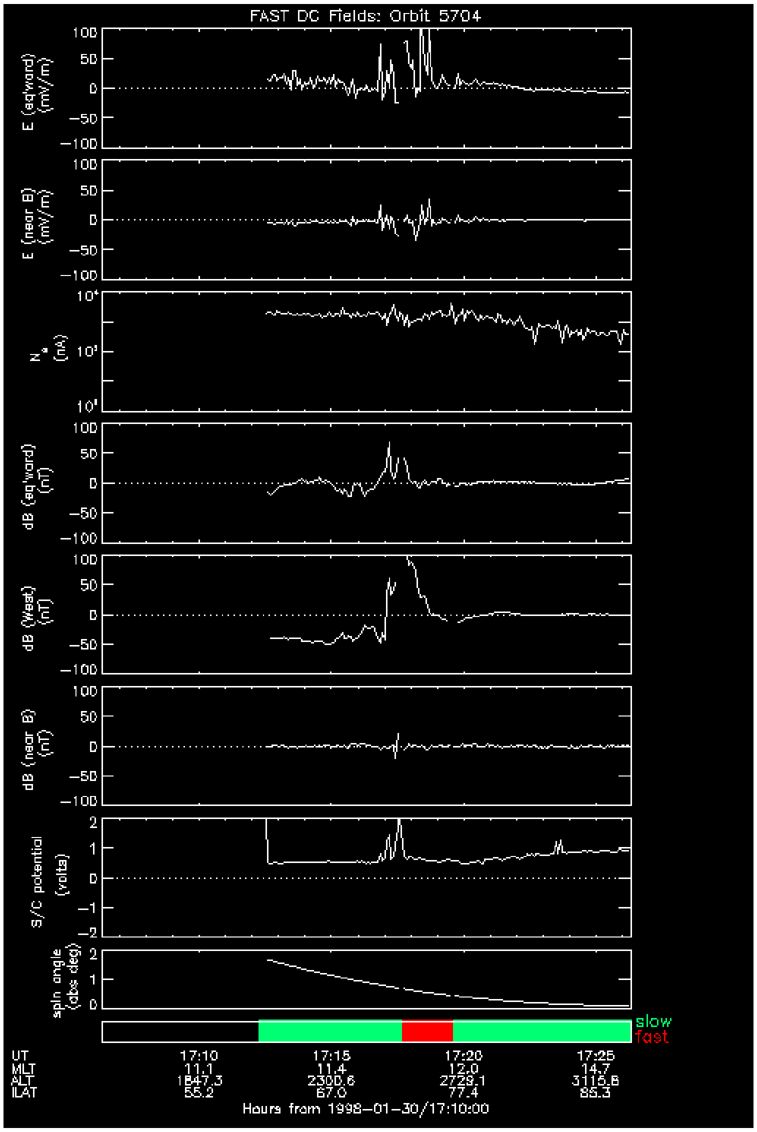


FIGURE 20: DC-FIELDS SUMMARY PLOT (ORBIT 5704, INCOING NORTH)

6.1.4 DOWNLOADING SUMMARY DATA FROM BERKELEY WEB ARCHIVES

Once the user has decided upon a period of time that he/she deems interesting (done by viewing raw data using SDT or viewing summary plots online), the user must obtain the data. One way to do this is by downloading the summary data from the Berkeley web archives. The summary data is a collection of CDF files that were created by Berkeley researchers to make the collection of data by other researchers easier. The intention was for the public to have easy access to the data without the overhead required by SDT. The summary data are divided into the same 6 categories that all of the plots data in SDT are configured (i.e. EES, IES, TEAMS, AC/DC Fields, and orbit). Summary plots (pictures) can be downloaded directly from the FAST website using. The summary data is compiled into CDF files that can be manipulated using many utilities like MATLAB and IDL. It should be noted that although downloading summary data seems like the obviously better choice than downloading raw data from SDT, it may be dangerous (Strangeway). By downloading the summary data, the user relies on processing techniques that may or may not be reliable. As previously mentioned, ideally the user would download the raw data and process it themselves.

6.1.5 LOADING DATA INTO MATLAB

The MATLAB function “load_data”, created specifically for use in this project, is used to read the CDF summary files created by Berkeley researchers. It uses the standard MATLAB function “cdfread”. It takes in 2 character arrays for the name of the CDF files being input. The function outputs two matrices. One matrix contains all of the field data and one matrix contains all of the satellite’s positional data. The function reads in two CDF files of the form "fa_k0_ttt_xxxxx_vyy" where “ttt” is the type of information included (either field data or positional data), “xxxxx” is the orbit number and “yy” is the revision number. The two CDF’s contain more information than is needed so the function only fills the MATLAB matrices with the data the MQP is concerned with. The two matrices that are outputted will allow other MATLAB functions to manipulate data within MATLAB. The function also creates a single plot with two lines. The lines are a function of the UNIX time vector (within the 2 separate CDF files) with respect to the number of samples. An example of this plot is shown below.

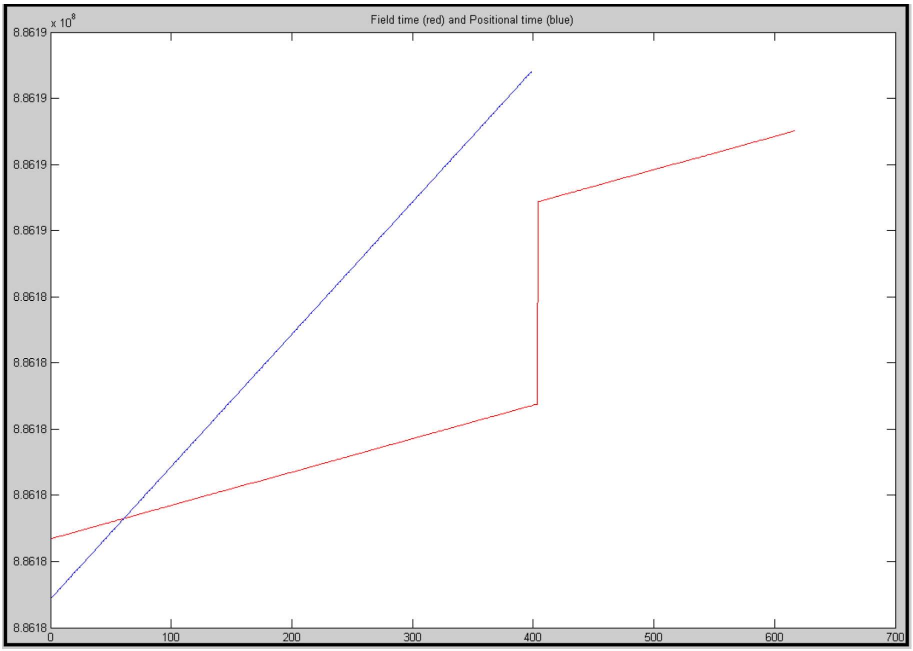


FIGURE 21: TIME (UNIX) VS. NUMBER OF SAMPLES FOR FIELD DATA AND POSITIONAL DATA (ORBIT 5704)

This plot is a representation of the timeframe of data that is in the CDF file. The blue line is the positional data and the red line is fields’ data. It can clearly be seen that the positional data covers a longer timeframe (from about 16:57 to 19:10 UT on January 30, 1998) while the fields data covers a shorter timeframe (from about 17:12 to 18:55 UT on January 30, 1998). It can also be seen that the fields’ data is discontinuous in time. There is a large jump between 17:46 and 18:37 where there is no fields’ data. The reason this plot is created is so that the user can manually delete all of the data after the discontinuity so that all of the data is continuous for one orbit. There is another problem, however. Because the fields data and positional data are in different resolutions (the fields data has more samples/second than positional data), it may not be known exactly where in space a particular fields event has occurred. Also, the two matrices may be of different lengths. For example, the fields’ data matrix might be 527 elements long and the positional data matrix may be 412 elements long. This means that it will be necessary to interpolate the positional data to a higher resolution.

6.1.6 INTERPOLATING POSITIONAL DATA FOR FIELDS DATA

The MATLAB function "interp_data", created specifically for this project, takes in the two matrices that are output from the "load_data" function. This routine uses the standard MATLAB "interp1" function. In special cases where the fields' data is discontinuous, the discontinuous data will have to be manually deleted from the matrix before the interpolation can be done correctly. Because the fields' data is at a higher resolution than the positional data, the positional data must be interpolated at every moment in time that fields' data exists. This is allowed because the sampling frequency is much larger than the switching frequency of the positional data. In other words, because the positional data changes at such a slow rate with respect to the number of samples taken, it is easy to interpolate "missing" data where it is needed. The picture below explains why the positional data needs to be interpolated. The blue dots signify the field data. The red dots signify the positional data. By interpolating, positional data can be found at every point of that fields data exists.

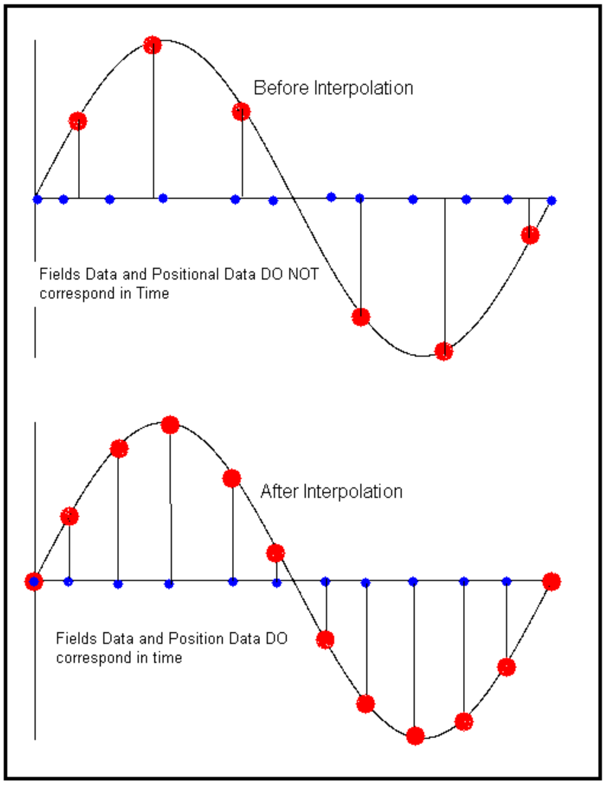

Before Interpolation
Fields Data and Positional Data DO NOT correspond in Time
After Interpolation
Fields Data and Position Data DO correspond in time

FIGURE 22: SIMPLIFIED INTERPOLATION EXAMPLE

The output of the function is a positional data matrix where every single point in time corresponds to a point in time that fields' data exists. Note that the fields' matrix is not changed in any way. At this point, both the positional data matrix and fields' data matrix are the same length. If the example in the previous section is the data being worked with, then both the fields' data and positional data matrices would be of length 527.

6.1.7 CONVERTING POSITIONAL DATA INTO GEOGRAPHIC COORDINATES

Now that every point of field data can be tied directly to a point in space (in GEI coordinates) and time (UNIX time), it is necessary to convert between GEI and GEO coordinates for positional data. GEI and GEO coordinates were discussed earlier in this report, but conversions between the two will be discussed here. The MATLAB function “gei2geo” takes in the interpolated positional data matrix that is output from the “interp_data” function. The first step in converting between GEI and GEO coordinates is finding the Greenwich hour angle (GHA). The Greenwich hour angle is the angle in the equatorial plane between the two x-axes in each coordinate system. Shown below is a representation of the GHA and what it signifies. Also refer to Figures 12 and 13.

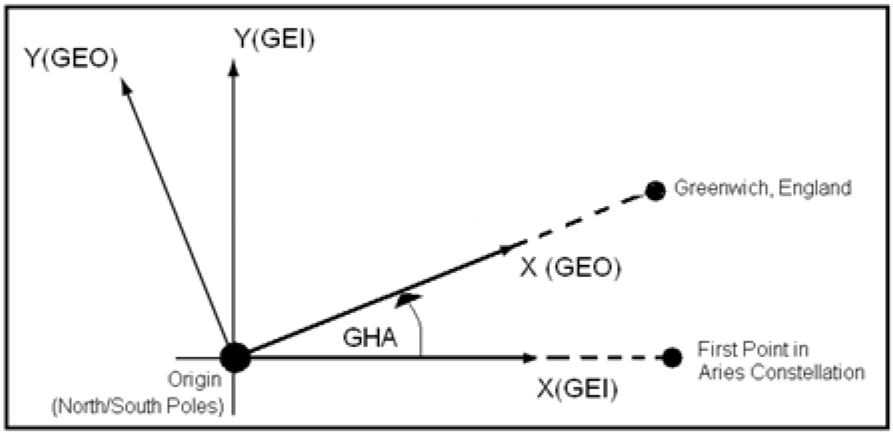


FIGURE 23: VISUAL REPRESENTATION OF GREENWICH HOUR ANGLE IN TERMS OF GEI AND GEO COORDINATES

In order to transform GEI coordinates to GEO coordinates, one must know the GHA at the exact time that the GEI coordinates were recorded. Unfortunately, there is no universally accepted equation for finding the GHA given a certain moment in time; instead there are a number of researchers who have published papers and submitted their own methods for finding the GHA at a moment in time. The gei2geo function actually uses two methods for finding the GHA, but only returns the result from using one of them. The equations which find the GHA are shown below. The first (Equation 1) is from Mike Hapgood at Rutherford Laboratories in England. (Hapgood, Rotation angles for space physics coordinate transformations) The second (Equation 2) is from a piece of IDL code written by researchers of FAST.

EQUATION 1

$$GHA\ (in\ degrees)\ =\ 100.461\ +\ (36000.770 * X) +\ (15.04107\ *\ Y)$$

In the first equation, X is the number of Julian Centuries since January 1, 2000, and Y is the number of hours since the preceding midnight UT. These values are calculated from UNIX time using the functions “unixtime2mat” and “since_j2000,” where “unixtime2mat” takes in a vector of UNIX times and produces a matrix of UT times in the format [year, month, day, minute, hour, second], and “since_j2000” takes in the UT matrix and outputs a vector which corresponds every UNIX time to the number of Julian Centuries since January 1, 2000. The number of hours since the preceding midnight is calculated in “gei2geo”. In the second equation, Z is the number of Julian days since the Julian epoch, and W is the fraction of the day since the preceding midnight UT. Both of these values are calculated in the “gei2geo” function.

EQUATION 2

$$GHA\ (in\ degrees) =\ modulus\ 360[(279.690983\ + (0.9856473354 * Z)\ + (360 * W\ +\ 180)$$

Once the GHA’s have been calculated for every point in time that field data exists, the transition matrices can be calculated which will transform three-dimensional GEI coordinates to three-dimensional GEO coordinates. The multiplication of transition matrices by the GEI coordinates are in the form shown below. (Kivelson and Russell)

EQUATION 3

$$\begin{bmatrix} Cos(GHA) & Sin(GHA) & 0 \\ -Sin(GHA) & Cos(GHA) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{matrix} V_x \\ V_y \\ V_z \end{matrix}$$

It should be noted that there is a separate transition matrix for every moment in time (every element) in the positional data. If the previous example matrix of length 527 GEI positional data points was used, there would be 527 separate transition matrices, and a result of 527 GEO points in GEO coordinates.

6.1.8 EXPORTING PROCESSED DATA USING SDT AND IDL

Another method of obtaining processed data is by using the Satellite Data Tool in conjunction with the IDL routines provided by Berkeley. These programs are both run on a Sun Solaris machine called "llama" that is available remotely to any machine that is connected to the SRI network. Using a UNIX terminal, SDT is started and run in the background so as to allow the user to also access IDL in the same terminal. The user would choose a preset configuration of plots, and then can use the "DataFromNetwork" option in the right-click menu to download data from Berkeley for any orbit desired.

In the meantime, IDL would be started in the terminal, using the 32-bit version as the 64-bit version does not work on llama. Once that is set, the user would typically be able to run the IDL routines provided by Berkeley, "fast_fields_summary.pro" for the field data, and "fast_ef_summary.pro" and "fast_if_summary.pro" for electron and ion data respectively, however this method proved to be a difficult process due to machine setup and other unforeseen circumstances, and had to be abandoned for this year of the project.

Unfortunately, processing files in such a manner will be necessary in future years due to a problem with the produced files by Berkeley in which there is an error in the orbits in the Southern Hemisphere (Strangeway).

6.1.9 CONVERTING FIELDS DATA INTO GEOGRAPHIC COORDINATES

As explained in an earlier section, the fields data written to the online summary files is in DCS. Because scientists outside of the FAST project are not interested in obtaining results for Poynting flux in DCS, the fields data needs to be converted into a different system. The "sat2geo" function converts fields data from DCS into GEO Cartesian coordinates. Once the Cartesian description of the fields data has been calculated, a spherical representation of the data is also found.

The sat2geo function requires four input matrices. The first and obviously most important matrix is the fields matrix which we would like to convert. It contains the three components of the electric and magnetic fields in the DCS. One important note is that 2 of the three components of the electric field are actually set to 0. The only measurement used in calculation is "E_X" the direction of the electric field that is equatorward. The "E_y" component is not present in the CDF files because it is assumed 0 given the definition of an electro-magnetic wave. The "E_Z" component is manually set to 0 because it is assumed to be primarily instrument noise (Strangeway). Note that E_x and E_z are the directions of the electric field in the DCS coordinate system. The next two matrices are the satellite positional matrix and satellite trajectory matrix (calculated from the "gei2geo" function). They each contain the three components that make up the position and trajectory of the satellite in GEO coordinates. The final matrix contains the three components of the IGRF model in GEO coordinates. This was covered in section 4.6.

The sat2geo function first uses the position and trajectory vectors to compute the spin axis vector using the cross product. The spin axis vector is the line in 3-D space that the satellite rotates about. The function also uses the position vectors to calculate the normalized vectors of the satellite in the spherical GEO system (ρ, θ, ϕ).

Now that the position, trajectory and spin axis vectors have been calculated, an exact mathematical description of the FAST coordinate system (as described in section 4.6.2) can be found. Using the mathematical description of the geomagnetic field model, the function can now find the projections of the geomagnetic field onto the axes of the FAST coordinate system. In the MATLAB code, these values are referred to as "u_spB" (projection onto spin-plane along B) and "u_spBp" (projection onto spin-plane perpendicular to B) . Both of these vectors lie in the spin plane and are perpendicular to each other.

Now that the projections of the geomagnetic field onto the FAST coordinate system as well as the unit vectors along the FAST coordinate axes expressed in the GEO coordinate system, the sat2geo function finds the exact moment in time when the DCS "flips" about the component along the geomagnetic field (Z component in DCS), thus ensuring that the Y and Z components in DCS always point west and equatorward respectively. This is done by finding the exact moment in time when the projection of the spin axis crosses the north pole.

Finally, given a description of the FAST coordinate system, a description of the geomagnetic field projections onto the FAST system, and by knowing the exact point in space that the DCS flips, the function can compute the electric and magnetic field components in Cartesian GEO coordinates by multiplying the components of the fields in DCS by unit vectors along the DCS coordinate axes expressed in GEO coordinates. From the Cartesian GEO coordinates, the function calculates the spherical GEO coordinates by multiplying the Cartesian coordinates of the fields by the normalized position vectors in the spherical GEO system.

The end result is two matrices containing all of the electric and magnetic field data in both spherical and Cartesian GEO coordinates. Now that the MATLAB suite has calculated the positional data and fields data in GEO coordinates and has correlated them in time, the actual calculation of the Poynting vector in GEO coordinates can be done.

6.1.10 FINDING THE IGRF MODEL

The 1995 IGRF Model is required in order to convert the fields data from the DCS into the GEO coordinate system. This is because the DCS is actually defined using the geomagnetic field and the FAST coordinate system. Luckily for this MQP team, a group of NASA scientists have developed a MATLAB function suite to calculate the geomagnetic field vector in GEO coordinates given GEO position.

The function "get_igrf" is a top-level function that calls the various functions in the MATLAB function suite. This is the only code written by the MQP team that dealt with the IGRF model. The next step would be to convert the fields data from the DCS into the GEO reference frame.

6.1.11 CALCULATING POYNTING VECTOR AND POYNTING FLUX

At this point in the MATLAB suite, the various functions have taken given non-time-corresponding positional and fields data in GEI and DCS coordinate systems respectively and converted them to time-corresponding positional and fields data in GEO coordinates. The calculation of the poynting vector can now be done.

First, the "H-Field" has to be calculated from the given "δB-Field". The H-Field is found by simply dividing the δB-Field by the constant for the permeability of free space, typically defined as $\mu_0 = 4\pi\times10^{-7}\ \mathrm{N{\cdot}A^{-2}}$. The Poynting vector is then defined as the cross product of the electric field vector and the H-Field vector. The next step would be to verify all of the results.

One simple, although non-exhaustive, approach to verifying the results for Poynting Flux in Cartesian GEO coordinates is by performing the operations in a different order. In the order of operations explained above, the fields data was first transformed into GEO coordinates and the cross product taken. A simple way of determining that there were no errors in writing code, would be to reverse the process (i.e. taking the cross product of the fields and finding a Poynting vector in DCS, and then transforming the Poynting vector into GEO coordinates). This method only really checks for coding mistakes. It does not take into account the overall methodologies described in this report.

Another simple, yet non-exhaustive, approach in determining if all of the conversions are right would be to compute the magnitude of the poynting vector in any given reference frame and compare it to the magnitude of the poynting vector in a different reference frame. The results should match almost perfectly because the magnitude of a vector is the same no matter what reference frame is it described in.

6.2 CALCULATION OF KINETIC ENERGY FLUX

As mentioned in Section 5.1.4 Kinetic Energy Flux, the calculation of Kinetic Energy Flux is a much simpler process than the calculation of Poynting Flux. The data can be obtained in a similar manner as Poynting Flux, either as summary data from the Berkeley website or as CDF files created through SDT and IDL.

The data available for both ion energy flux and electron energy flux is in ergs/cm^2-s, which as mentioned previously requires scaling to achieve Joules/m^2-s. The positive values are kinetic energy flux moving toward the Earth, and the negative values are moving away from the Earth.

The data is loaded into MATLAB in a similar manner as with the Poynting Flux, and the position is interpolated to match this data. The Kinetic Energy Flux is plotted along with the Poynting Flux.

7. DATA AND RESULTS

This section will outline the data and results for this project. First, plots of the given data will be discussed, followed by the correlation of this data to time. The positional data in GEO coordinates will be shown, and then the IGRF model in GEO coordinates. The final conversion of the electric and magnetic fields to GEO coordinates will be displayed, and then finally the two different ways to achieve the Poynting Vector will be plotted and the result will be shown in the Data Coordinate System as well as the GEO system. This section will finalize with a discussion on the Kinetic Energy Flux data. In each plot, the Latitude, Longitude, and Altitude are displayed as the x-axes for the different plots.

Altogether five different orbits will be shown with different characteristics of Poynting Flux: against the Earth's magnetic field, along the Earth's magnetic field, an orbit that shows characteristics of being both along and against this field, an orbit that produces what is suspected to be noise, and an orbit that produces what appears to be no Poynting Flux at all.

7.1 GIVEN DATA

The data for the position of the satellite is given in GEI coordinates. In Figure 24 through Figure 28 this data is displayed. These coordinates are difficult to understand unless the position of the first star in Aries is known for the particular points in time of the measurement, therefore the next step for this data is to convert it into GEO coordinates, which are simpler to understand by visual inspection with less outside information.

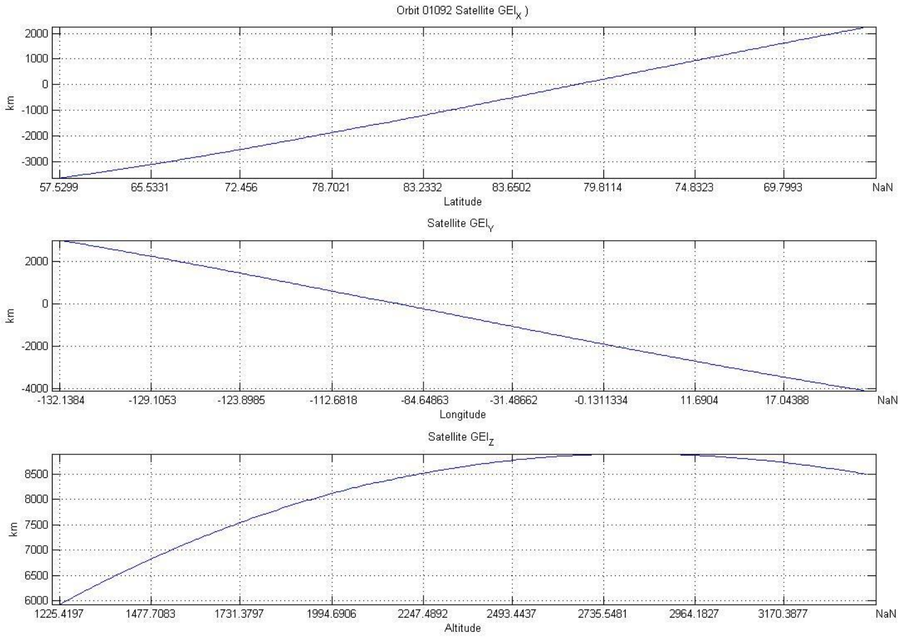


FIGURE 24: ORBIT 1092 SATELLITE POSITION IN GEI COORDINATES

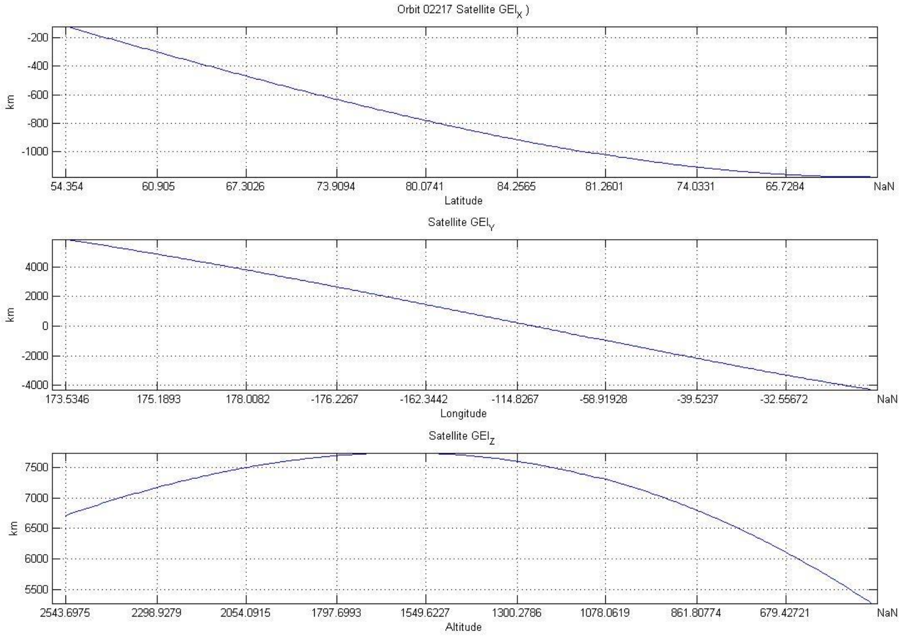


FIGURE 25: ORBIT 2217 SATELLITE POSITION IN GEI COORDINATES

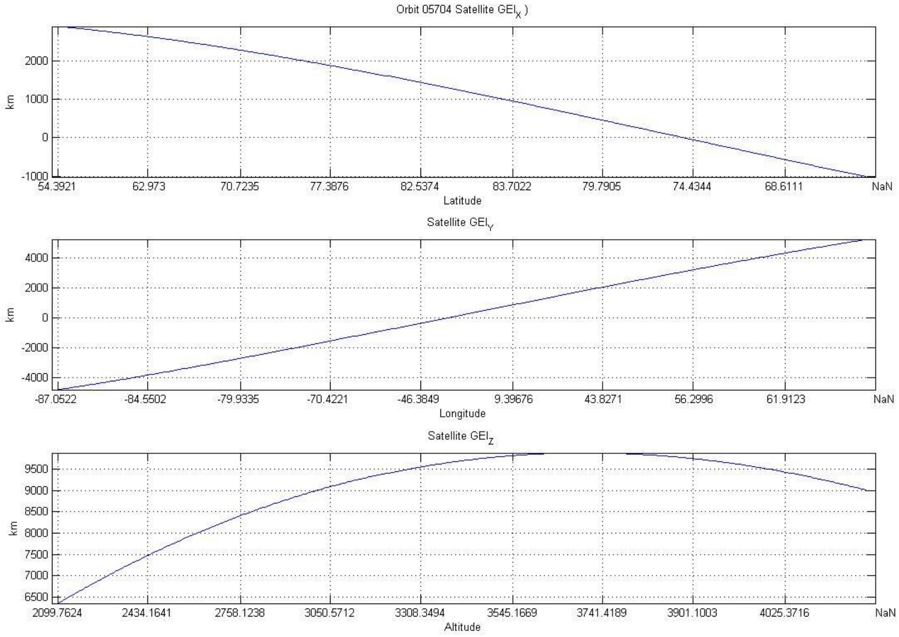


FIGURE 26: ORBIT 5704 SATELLITE POSITION IN GEI COORDINATES

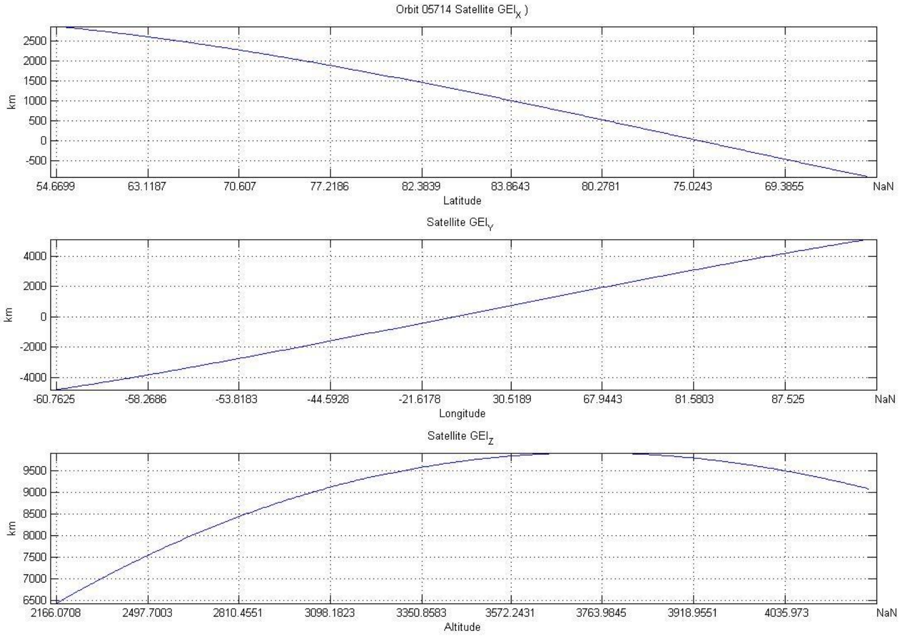


FIGURE 27: ORBIT 5714 SATELLITE POSITION IN GEI COORDINATES

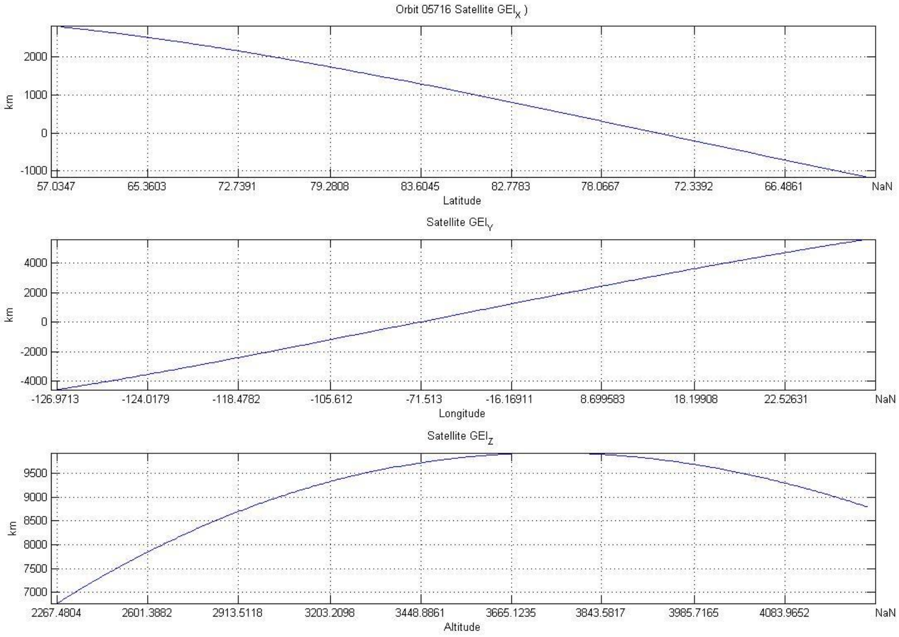


FIGURE 28: ORBIT 5716 SATELLITE POSITION IN GEI COORDINATES

The data for the Electric and Magnetic Fields is given in the Data Coordinate System. The plots display the electric field in the DCS X direction and the δB in the DCS X (blue), Y (red), and Z (green) directions.

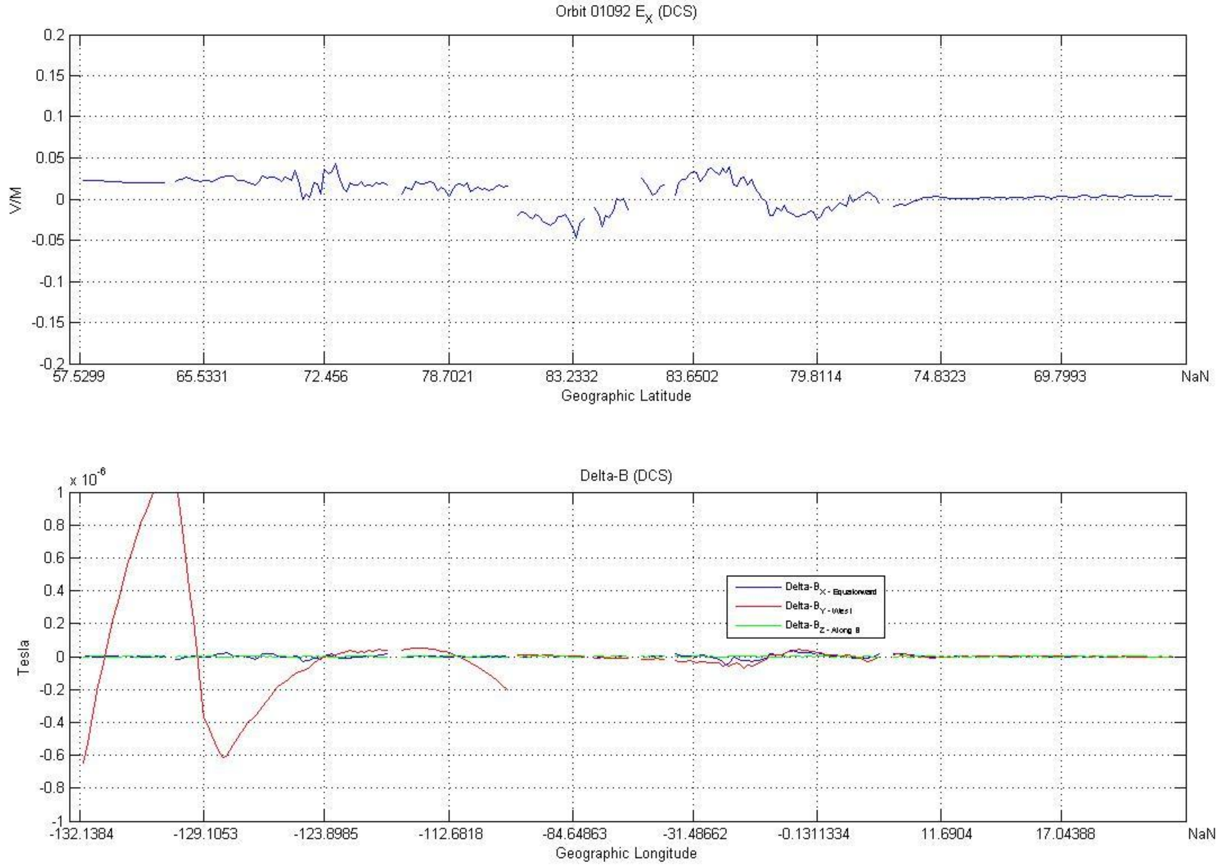


FIGURE 29: ORBIT 1092 ELECTRIC AND MAGNETIC FIELD DATA IN THE DATA COORDINATE SYSTEM

As shown in Figure 29, during orbit 1092 the electric field ranges from approximately -50 mV/m to 50 mV/m. The magnetic field is relatively low and constant in the X and Z directions; however there is a large amount of activity in the Y (westward) direction. The field increases greatly (over 1000 nT) in the westward direction, and then decreases down until there is a large magnetic field fluctuation (600nT) in the eastward direction. Note that the activity is only happening at certain latitudes; these latitudes are the location of the auroral zone in which the satellite is collecting data.

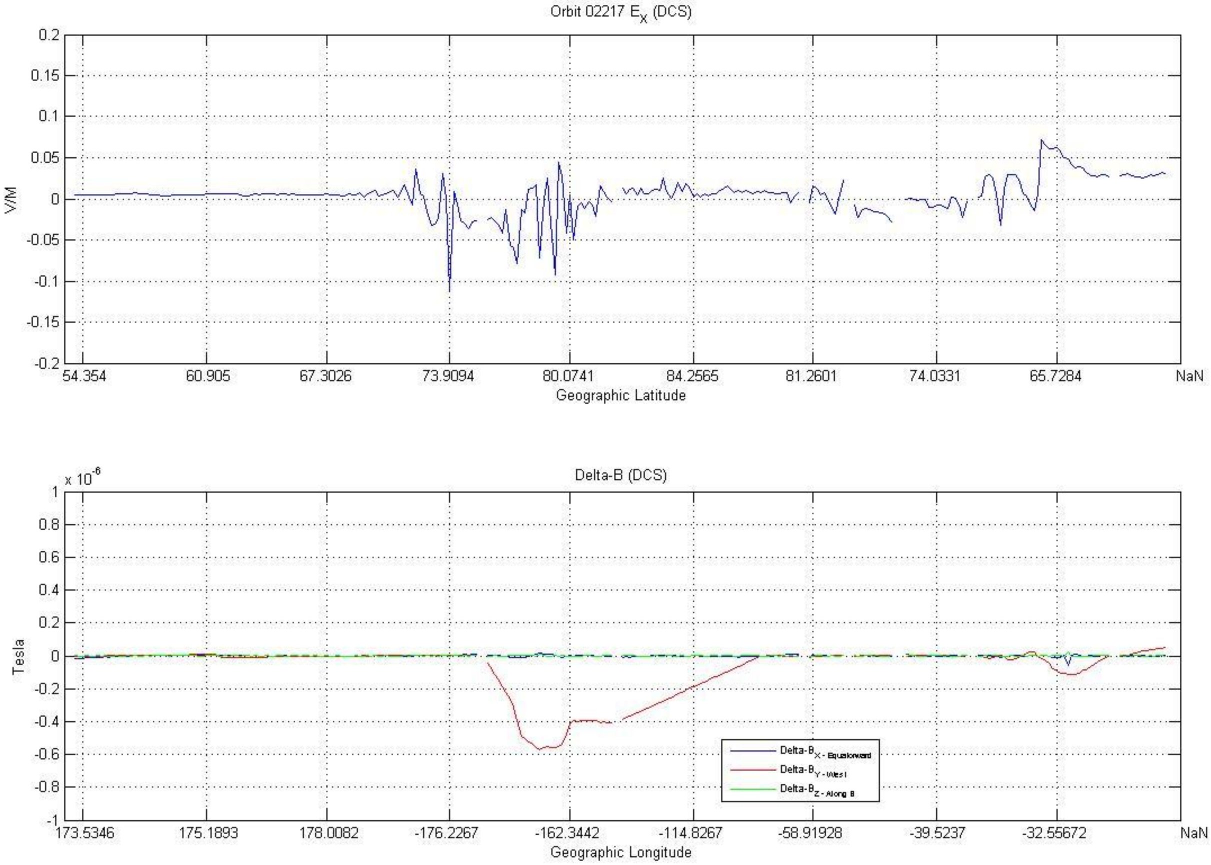


FIGURE 30: ORBIT 2217 ELECTRIC AND MAGNETIC FIELD DATA IN THE DATA COORDINATE SYSTEM

Above, Figure 30 shows the field data for orbit 2217. The electric field has large fluctuations, especially during the period where there are large fluctuations in the magnetic field. The magnetic field is pointing largely eastward during this orbit while the satellite is in the auroral zone.

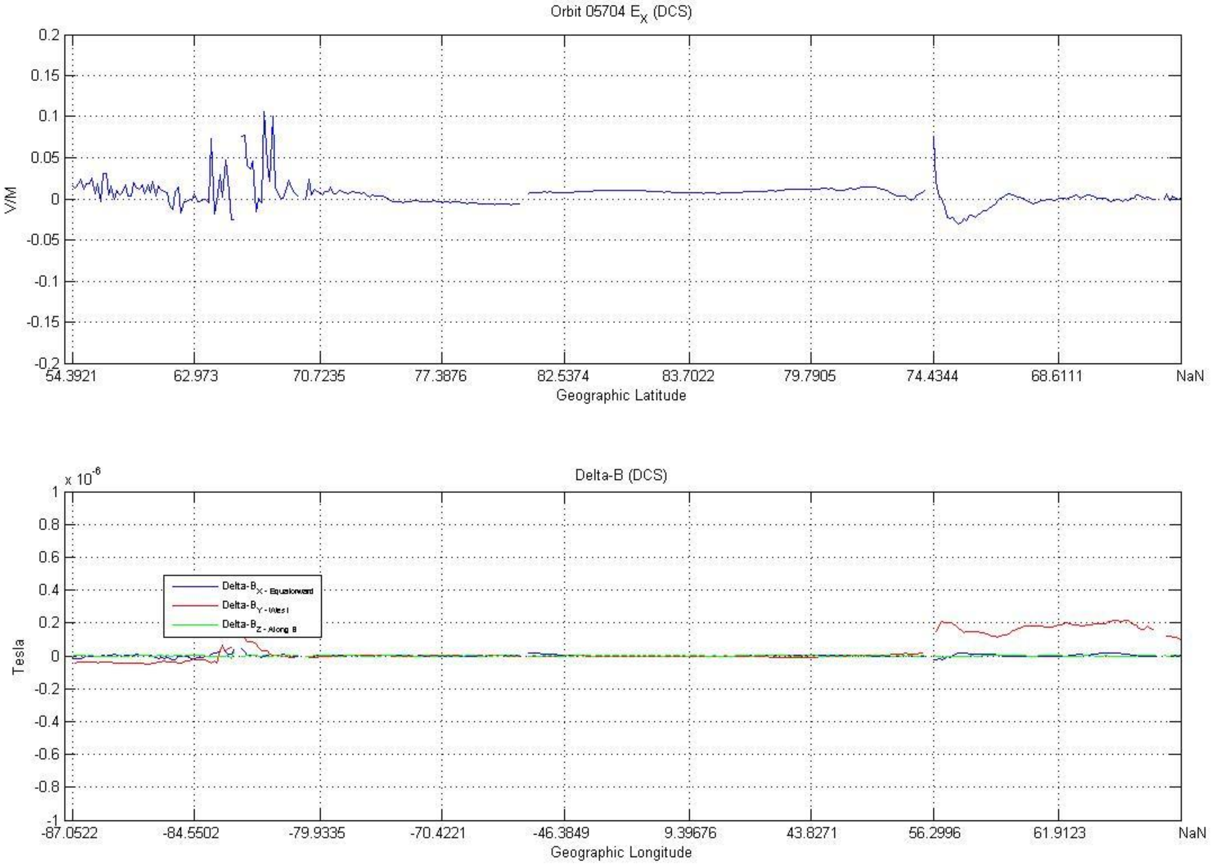


FIGURE 31: ORBIT 5704 ELECTRIC AND MAGNETIC FIELD DATA IN THE DATA COORDINATE SYSTEM

Orbit 5704, shown in Figure 31, appears to have some significant activity in the electric field but not much significant activity in the magnetic field fluctuations data in any direction.

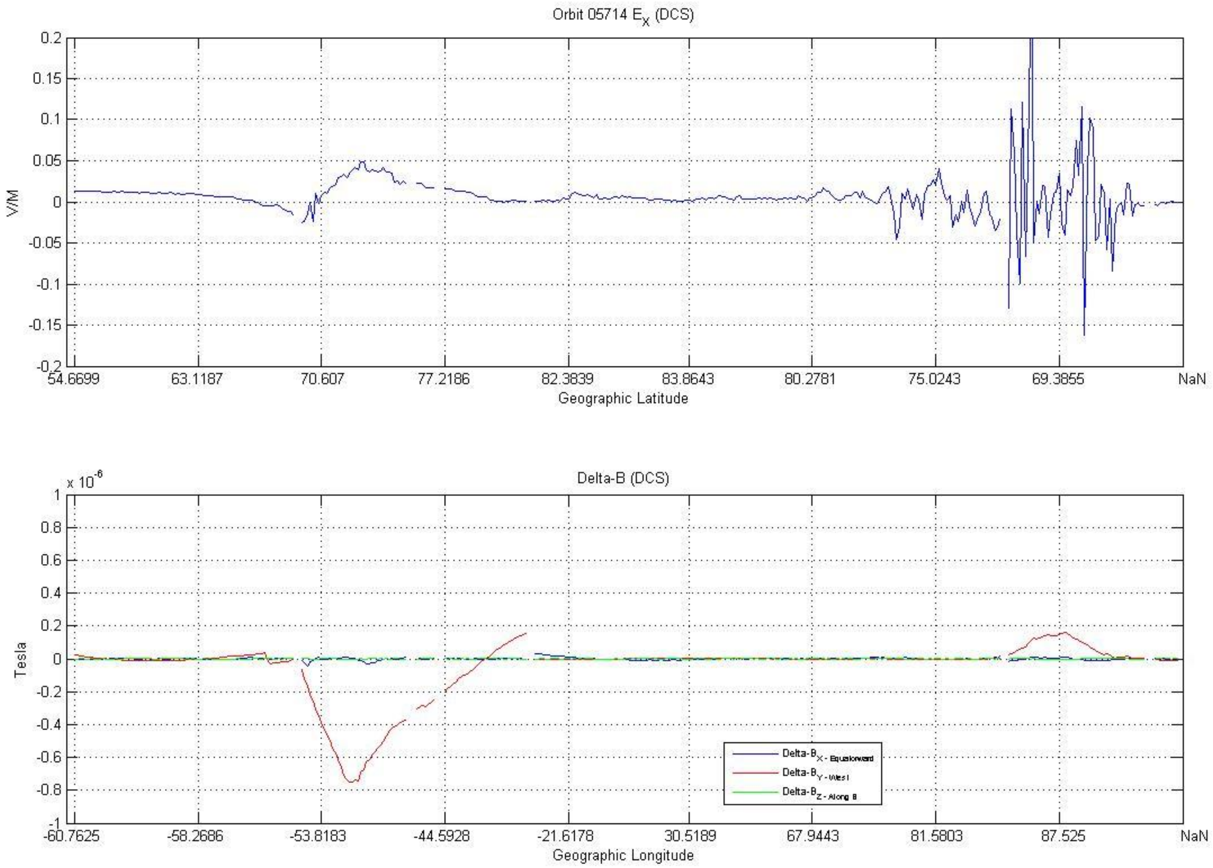


FIGURE 32: ORBIT 5714 ELECTRIC AND MAGNETIC FIELD DATA IN THE DATA COORDINATE SYSTEM

Figure 32 above shows orbit 5714. The places where the satellite enters and exits the auroral ring is apparent in this orbit, as there are two areas of significant activity at similar latitudes toward the beginning and the end of the data. As the satellite is incoming north, the fluctuations in the magnetic field are strongly eastward, and as it is outgoing north the field shows smaller but significant fluctuations westward.

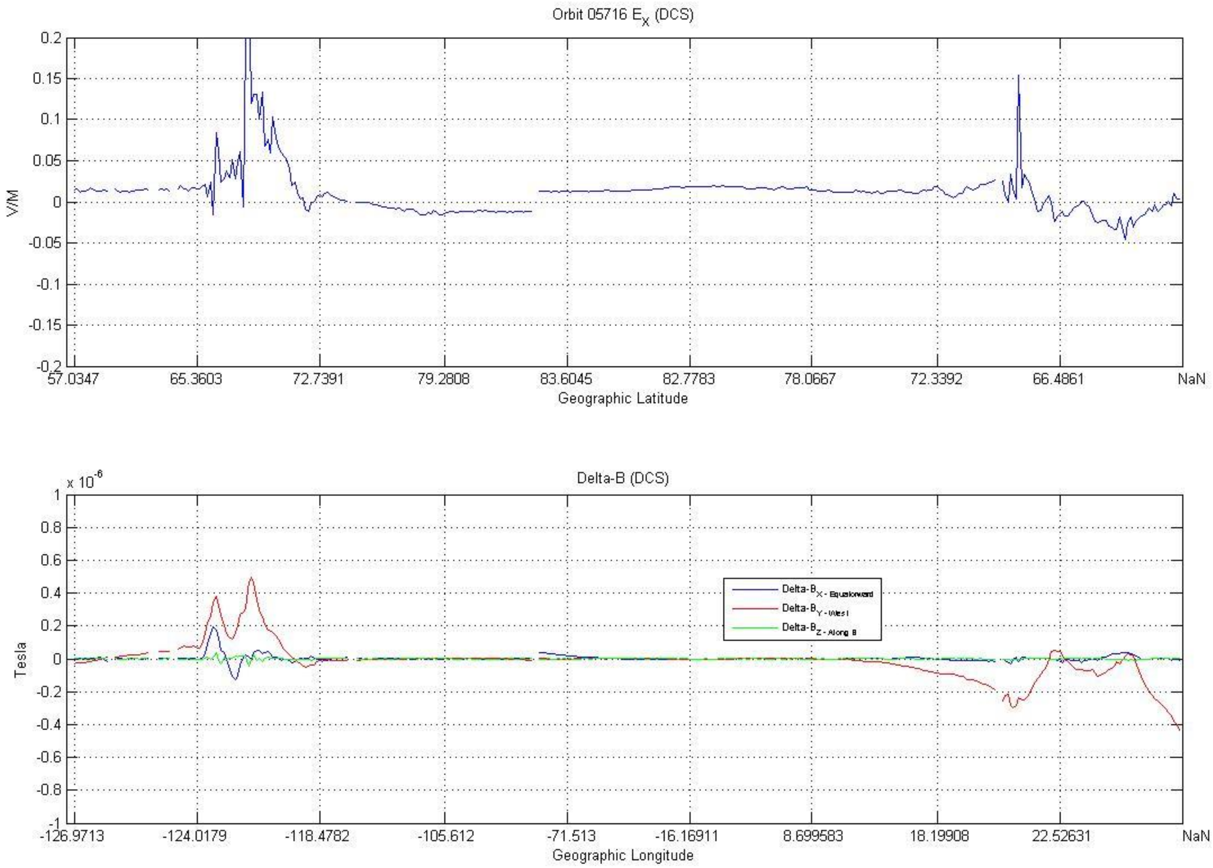
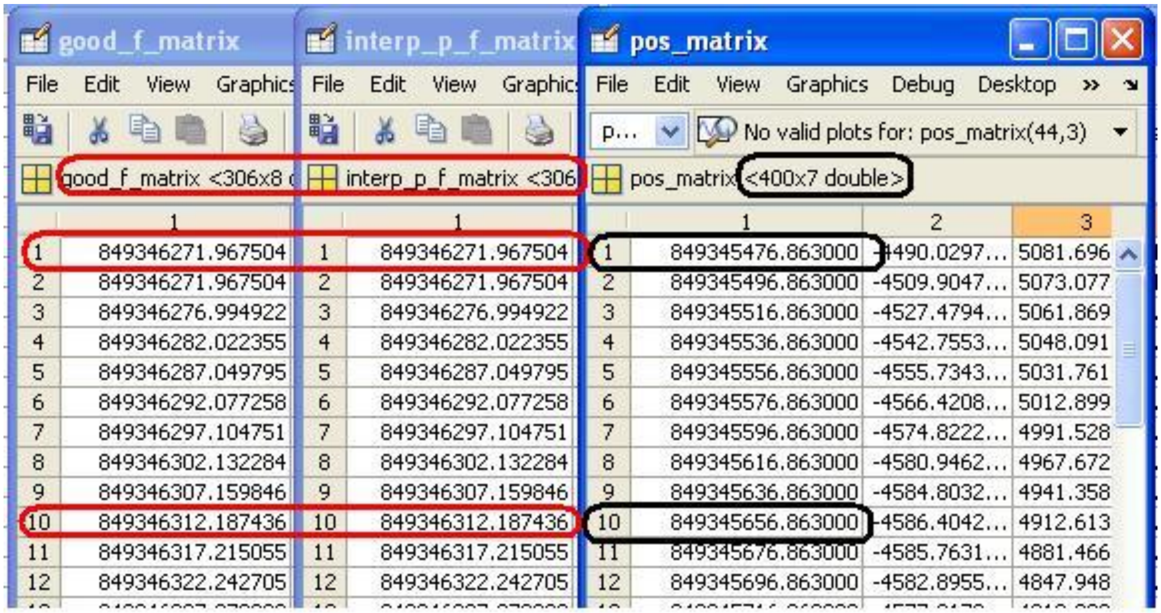


FIGURE 33: ORBIT 5716 ELECTRIC AND MAGNETIC FIELD DATA IN THE DATA COORDINATE SYSTEM

The field data for the final orbit this report will discuss, orbit 5716, is displayed above in Figure 33. This orbit has a significant electric field, exceeding 200 mV/m, and a significant magnetic field fluctuation in both the westward and equatorward directions.

7.2 CORRELATING DATA AND TIME CONVERSIONS

As mentioned previously, the data needs to be interpolated in order to accurately describe where the data was collected by the satellite. In Figure 34, the right-most matrix, “pos_matrix”, needed to be interpolated to match up the position in time to the data in the “good_f_matrix” on the left of the figure. The first column in each matrix is the UNIX timestamp of the data. It can be seen that initially, the times did not line up. After the interpolation function was applied, the matrix in the center resulted, and it is shown that the “interp_p_f_matrix” is lined up in time with the left-most matrix. It is also worth noting that the initial “pos_matrix” contained 400 samples, and the matrix it was interpolated with contained 306 samples. The resultant matrix contained the same amount of samples as the field matrix. Some of the positional data had to be removed due to interpolation, but no data was lost in this process.



good_f_matrix <306x8

	1
1	849346271.967504
2	849346271.967504
3	849346276.994922
4	849346282.022355
5	849346287.049795
6	849346292.077258
7	849346297.104751
8	849346302.132284
9	849346307.159846
10	849346312.187436
11	849346317.215055
12	849346322.242705

interp_p_f_matrix <306

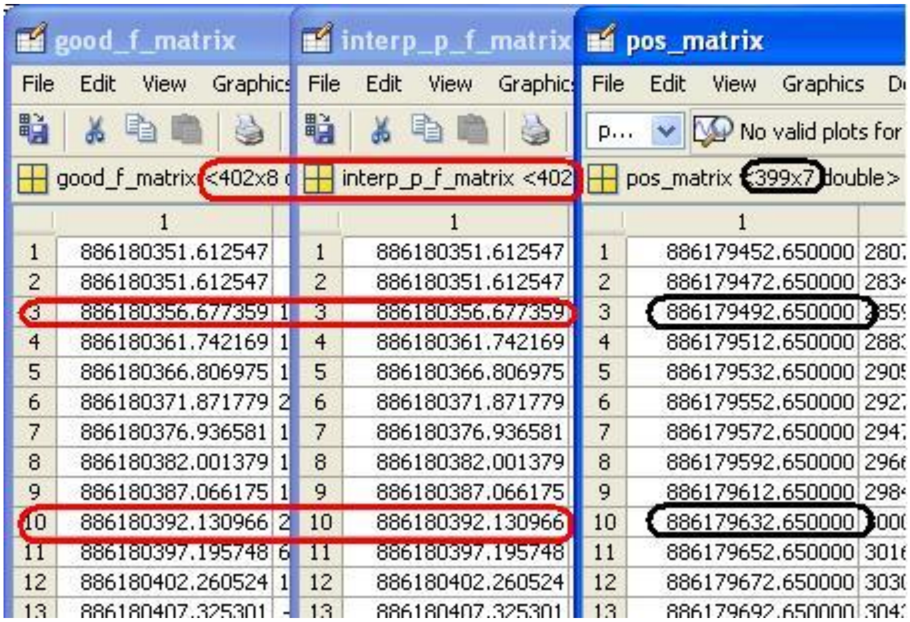
	1
1	849346271.967504
2	849346271.967504
3	849346276.994922
4	849346282.022355
5	849346287.049795
6	849346292.077258
7	849346297.104751
8	849346302.132284
9	849346307.159846
10	849346312.187436
11	849346317.215055
12	849346322.242705

pos_matrix <400x7 double>

	1	2	3
1	849345476.863000	-4490.0297...	5081.696
2	849345496.863000	-4509.9047...	5073.077
3	849345516.863000	-4527.4794...	5061.869
4	849345536.863000	-4542.7553...	5048.091
5	849345556.863000	-4555.7343...	5031.761
6	849345576.863000	-4566.4208...	5012.899
7	849345596.863000	-4574.8222...	4991.528
8	849345616.863000	-4580.9462...	4967.672
9	849345636.863000	-4584.8032...	4941.358
10	849345656.863000	-4586.4042...	4912.613
11	849345676.863000	-4585.7631...	4881.466
12	849345696.863000	-4582.8955...	4847.948

FIGURE 34: ORBIT 1092 INTERPOLATION OF POSITIONAL AND FIELDS DATA

Figure 35 shows another example of interpolation in orbit 5704. This is a different case than in orbit 1092, as the positional data contains fewer samples than the field data. As positional data is almost linear (if we know that at time t_0 the satellite is at position $p^0=(x^0,y^0,z^0)$, and at time t_1 the satellite is at $p^1=(x^1,y^1,z^1)$, it can be predicted that at a time between t_0 and t_1 the satellite will be somewhere between p^0 and p^1), it is possible to assume the position of the satellite in between samples.



good_f_matrix <402x8

	1	
1	886180351.612547	
2	886180351.612547	
3	886180356.677359	1
4	886180361.742169	1
5	886180366.806975	1
6	886180371.871779	2
7	886180376.936581	1
8	886180382.001379	1
9	886180387.066175	1
10	886180392.130966	2
11	886180397.195748	6
12	886180402.260524	1
13	886180407.325301	-

interp_p_f_matrix <402

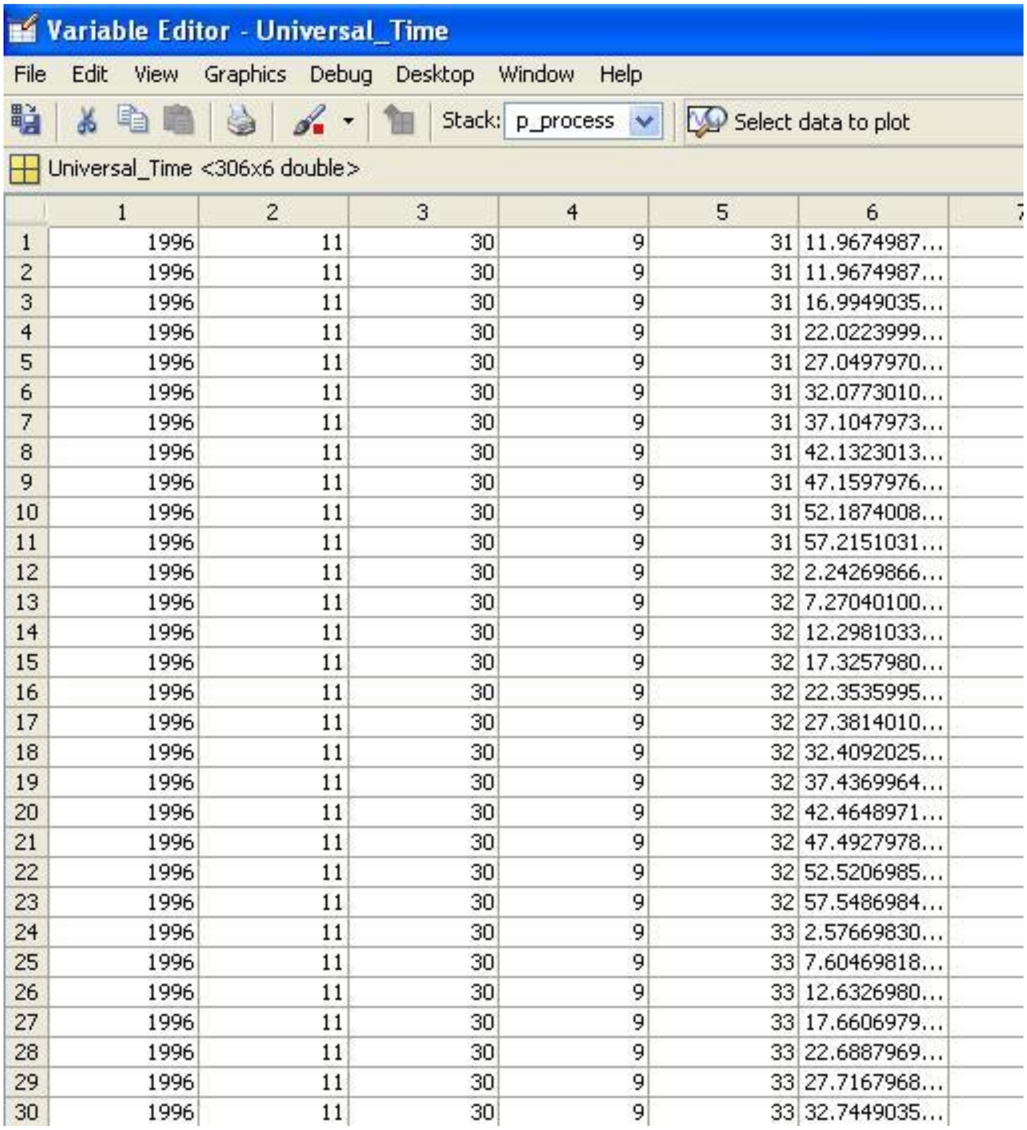
	1
1	886180351.612547
2	886180351.612547
3	886180356.677359
4	886180361.742169
5	886180366.806975
6	886180371.871779
7	886180376.936581
8	886180382.001379
9	886180387.066175
10	886180392.130966
11	886180397.195748
12	886180402.260524
13	886180407.325301

pos_matrix <399x7 double>

	1	
1	886179452.650000	280
2	886179472.650000	283
3	886179492.650000	285
4	886179512.650000	288
5	886179532.650000	290
6	886179552.650000	292
7	886179572.650000	294
8	886179592.650000	296
9	886179612.650000	298
10	886179632.650000	300
11	886179652.650000	301
12	886179672.650000	303
13	886179692.650000	304

FIGURE 35: ORBIT 5704 INTERPOLATION OF POSITIONAL AND FIELDS DATA

Using one of the MATLAB functions, the approximate universal time for each sample was found. This data was used to find the Greenwich Hour Angle needed to transform our data from GEI to GEO coordinates. The universal time matrix for orbit 1092 is shown in Figure 36.



Universal_Time <306x6 double>

	1	2	3	4	5	6
1	1996	11	30	9	31	11.9674987...
2	1996	11	30	9	31	11.9674987...
3	1996	11	30	9	31	16.9949035...
4	1996	11	30	9	31	22.0223999...
5	1996	11	30	9	31	27.0497970...
6	1996	11	30	9	31	32.0773010...
7	1996	11	30	9	31	37.1047973...
8	1996	11	30	9	31	42.1323013...
9	1996	11	30	9	31	47.1597976...
10	1996	11	30	9	31	52.1874008...
11	1996	11	30	9	31	57.2151031...
12	1996	11	30	9	32	2.24269866...
13	1996	11	30	9	32	7.27040100...
14	1996	11	30	9	32	12.2981033...
15	1996	11	30	9	32	17.3257980...
16	1996	11	30	9	32	22.3535995...
17	1996	11	30	9	32	27.3814010...
18	1996	11	30	9	32	32.4092025...
19	1996	11	30	9	32	37.4369964...
20	1996	11	30	9	32	42.4648971...
21	1996	11	30	9	32	47.4927978...
22	1996	11	30	9	32	52.5206985...
23	1996	11	30	9	32	57.5486984...
24	1996	11	30	9	33	2.57669830...
25	1996	11	30	9	33	7.60469818...
26	1996	11	30	9	33	12.6326980...
27	1996	11	30	9	33	17.6606979...
28	1996	11	30	9	33	22.6887969...
29	1996	11	30	9	33	27.7167968...
30	1996	11	30	9	33	32.7449035...

FIGURE 36: ORBIT 1092 UNIVERSAL TIME MATRIX (YEAR, MONTH, DAY, HOUR, MINUTE, SECONDS)

7.3 TRANSFORMED POSITIONAL DATA

The positional needed to be transformed into GEO coordinates. First, the Greenwich Hour Angle was found using the universal time matrix in Section 7.2, and used in the transformation from GEI to GEO coordinates. This coordinate system is easier to understand, as it is static in relation to the Earth. Note that the Z direction is exactly the same in GEO as it is in GEI, as that axis goes through the geographic North Pole in both. The positional data for the five orbits discussed in this report are in Figure 37 through Figure 41.

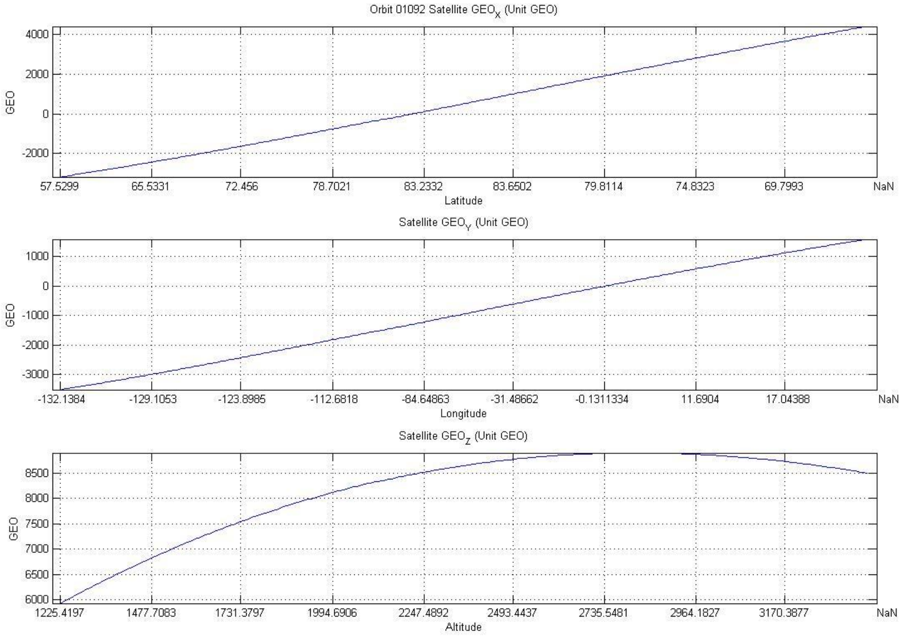


FIGURE 37: ORBIT 1092 SATELLITE POSITION IN GEO COORDINATES

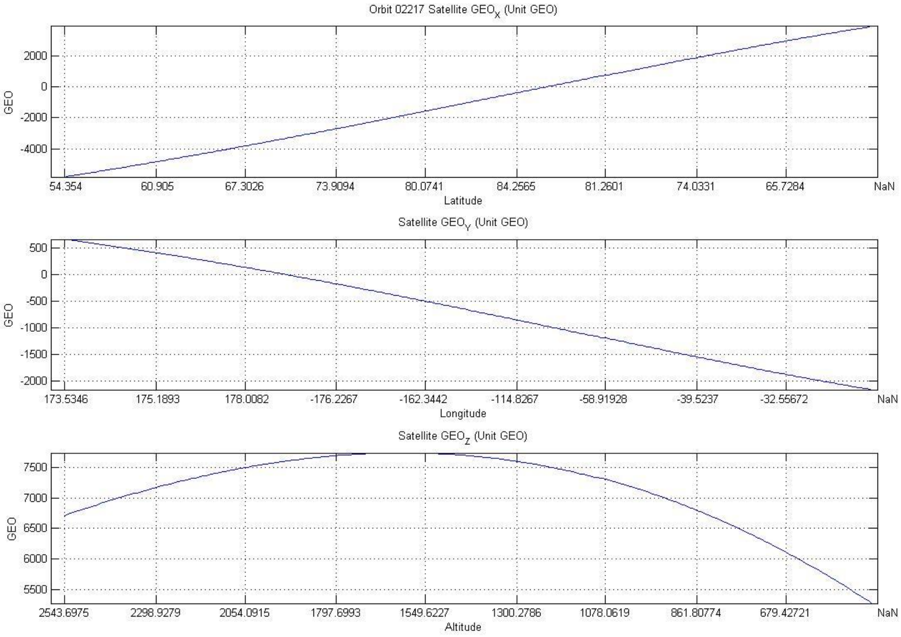


FIGURE 38: ORBIT 2217 SATELLITE POSITION IN GEO COORDINATES

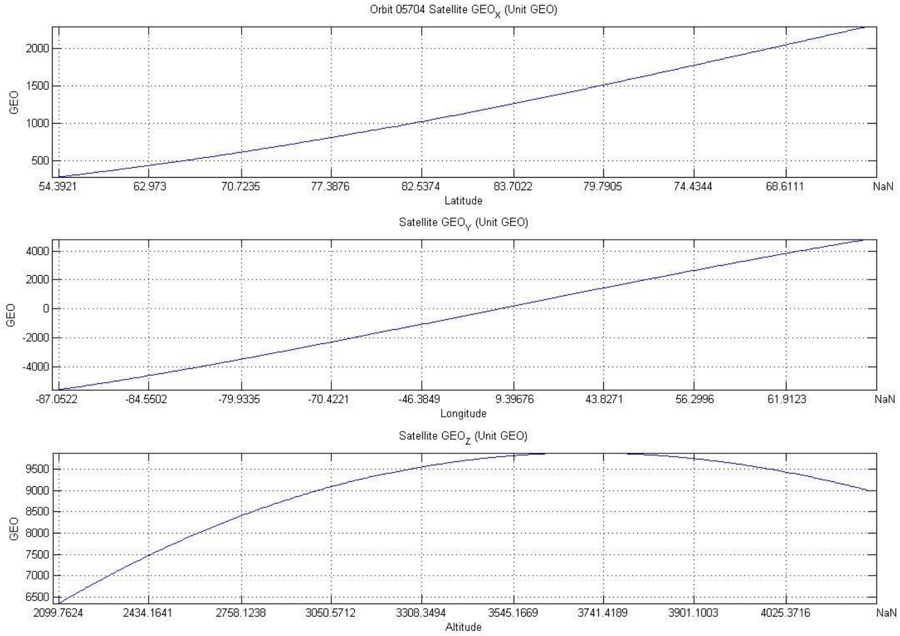


FIGURE 39: ORBIT 5704 SATELLITE POSITION IN GEO COORDINATES

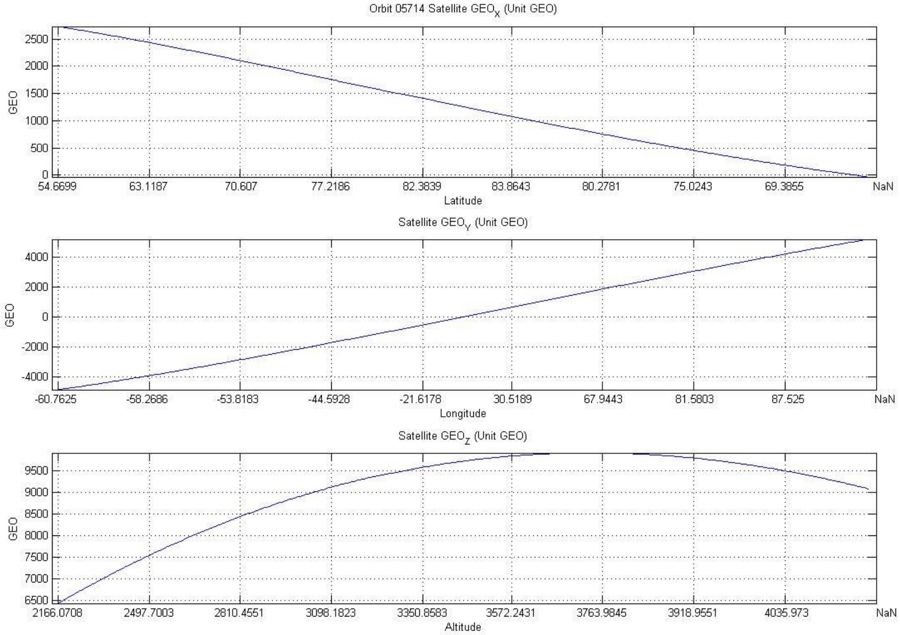


FIGURE 40: ORBIT 5714 SATELLITE POSITION IN GEO COORDINATES

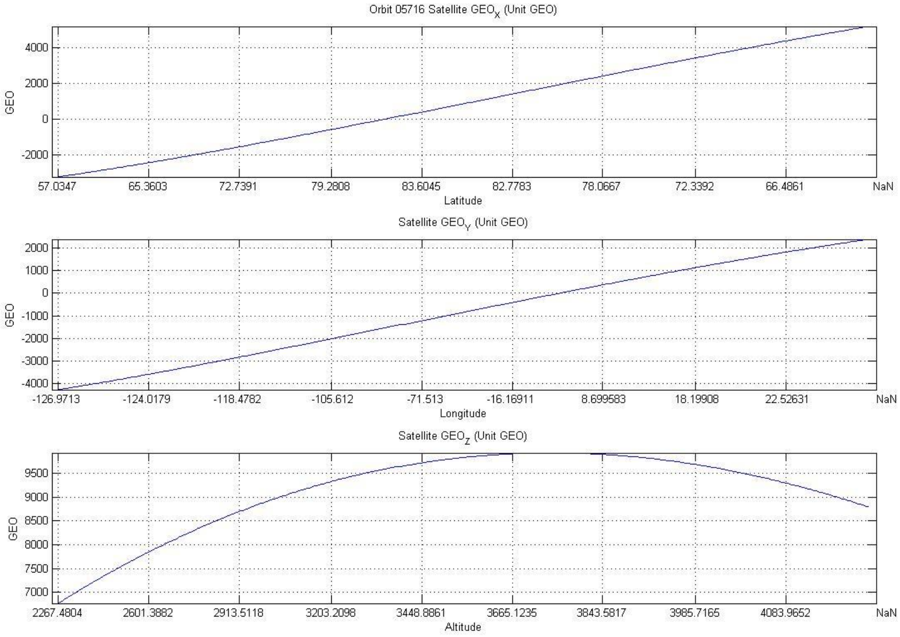


FIGURE 41: ORBIT 5716 SATELLITE POSITION IN GEO COORDINATES

7.4 THE IGRF MODEL

The model of the Earth's magnetic field is very important for this project. Each orbit will have a different IGRF model as they are all different in terms of position. The following figures, Figure 42 through Figure 46, display the IGRF model for each of the five orbits discussed, in GEO coordinates.

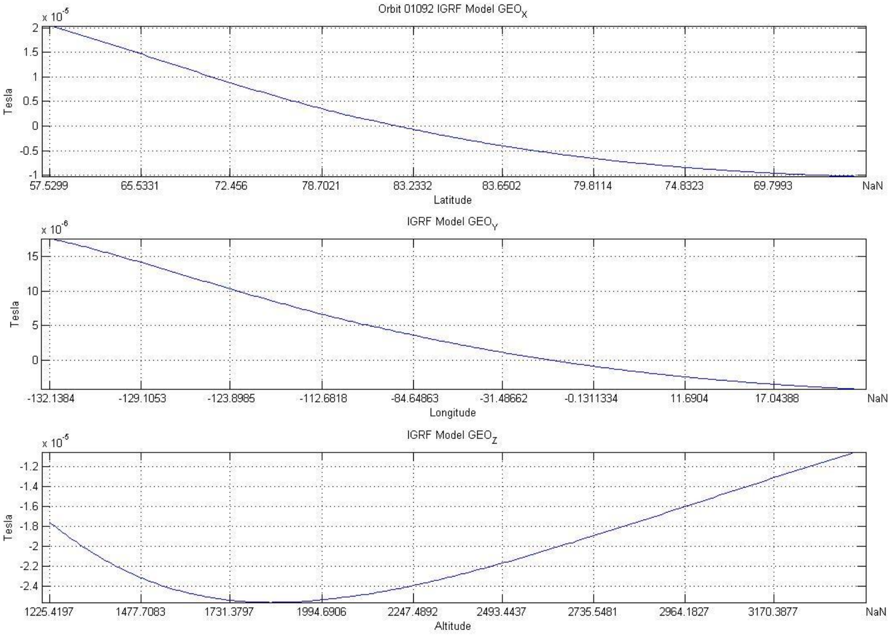


FIGURE 42: THE IGRF MODEL FOR ORBIT 1092

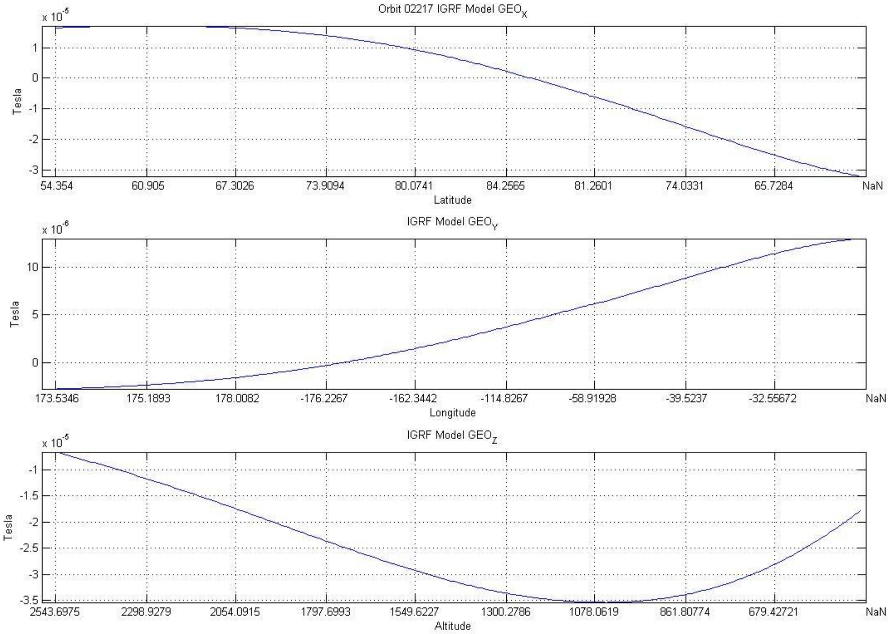


FIGURE 43: THE IGRF MODEL FOR ORBIT 2217

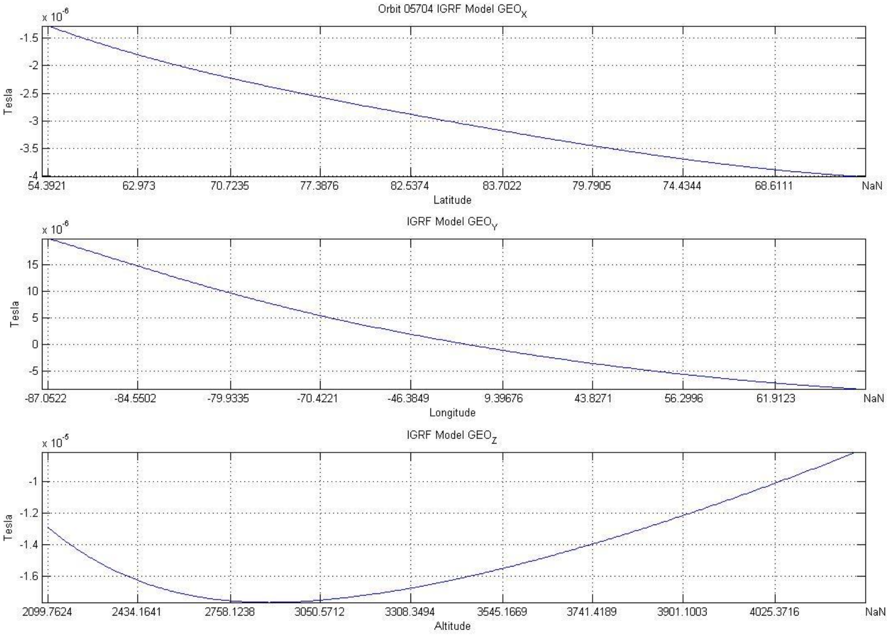


FIGURE 44: THE IGRF MODEL FOR ORBIT 5704

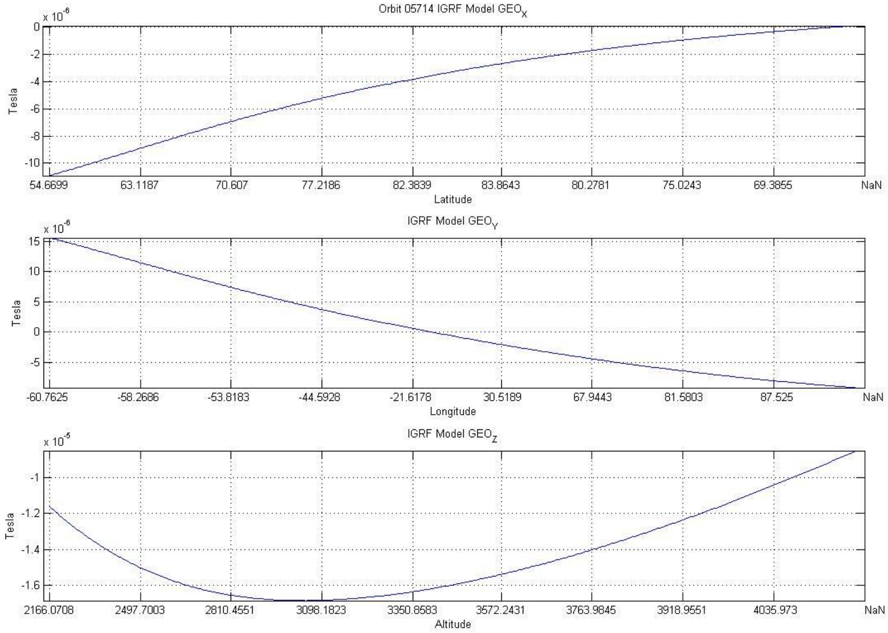


FIGURE 45: THE IGRF MODEL FOR ORBIT 5714

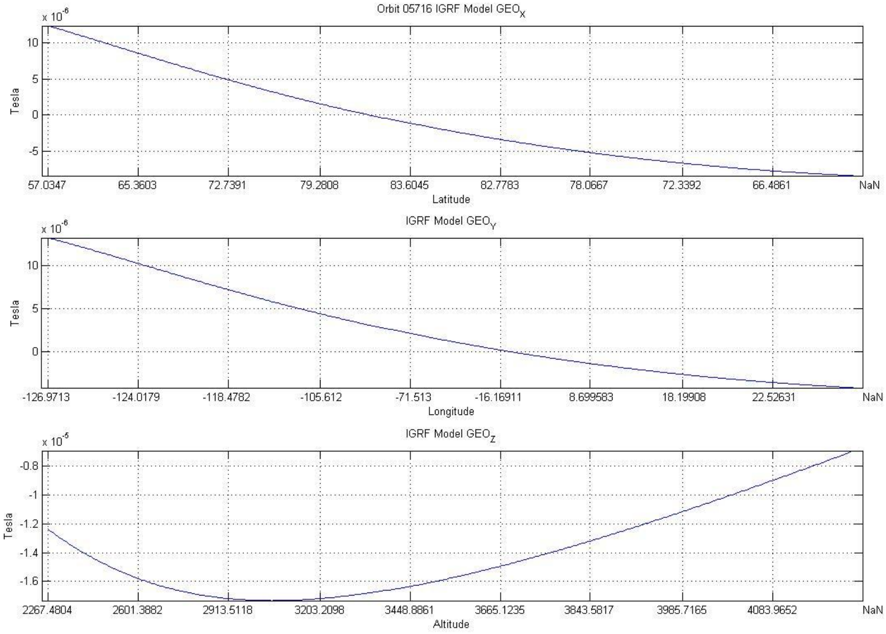


FIGURE 46: THE IGRF MODEL FOR ORBIT 5716

7.5 POYNTING VECTORS

There are two different methods to calculate the Poynting vector. One is to convert the raw magnetic field data to magnetic field intensity, and to cross that with the electric field, and then transform the resulting Poynting vector into Cartesian GEO and then spherical GEO coordinates. The other method is to convert all of the fields into Cartesian GEO coordinates first, then convert the δB to H, then perform the cross product resulting in the Poynting Vector in Cartesian GEO coordinates, and then converting this result into spherical GEO coordinates. Both methods result in the same Poynting Vector. The following plots demonstrate the electric and magnetic fields and the Poynting Vector in both the Data Coordinate System and in the spherical GEO coordinate system.

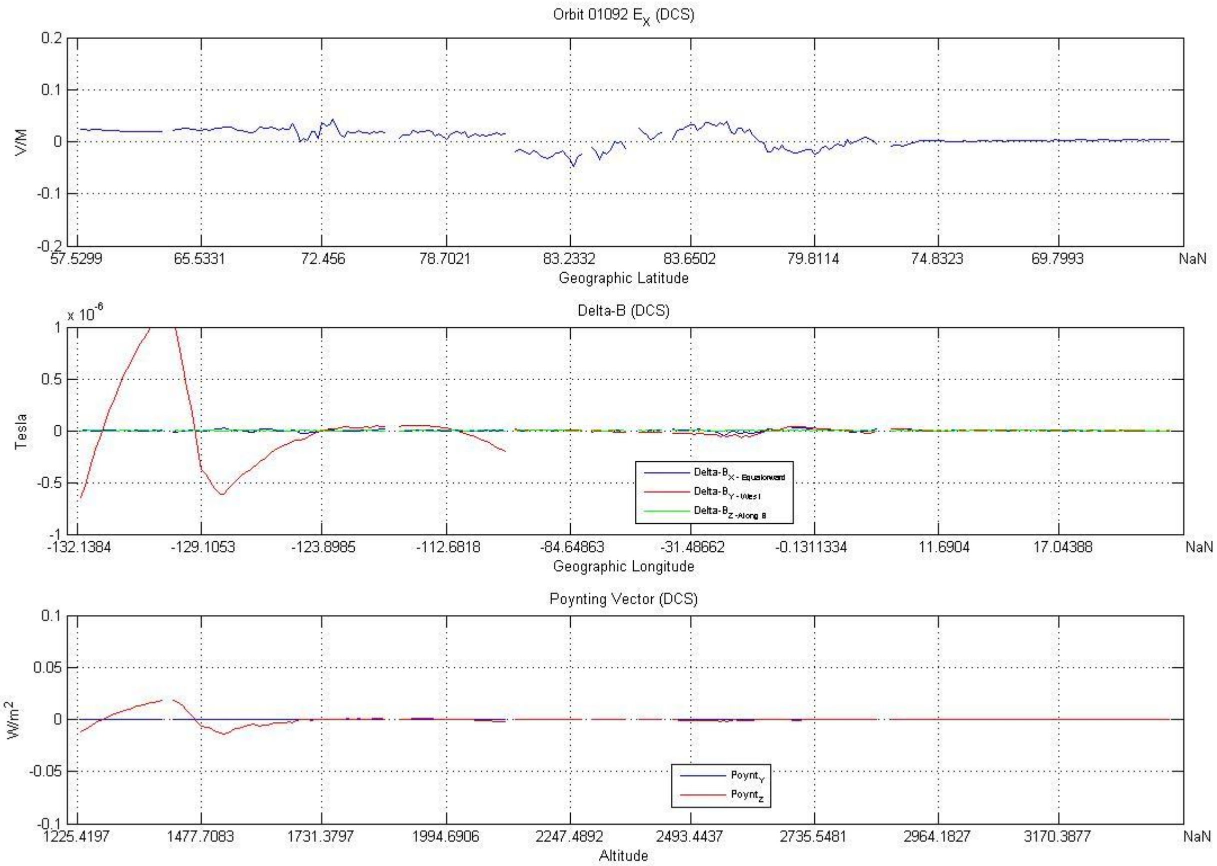


FIGURE 47: ORBIT 1092 ELECTRIC FIELD, MAGNETIC FIELD, AND POYNTING VECTOR IN DCS

The Poynting Vector for orbit 1092 as shown in Figure 47 follows a similar pattern as the magnetic field. The Z component of the Poynting vector is the component with the clear magnitude; when it is positive it is along the Earth's magnetic field and when it is negative it is against the Earth's magnetic field. It can be seen that the Poynting vector for orbit 1092 starts along the geomagnetic field and then changes its direction to be against.

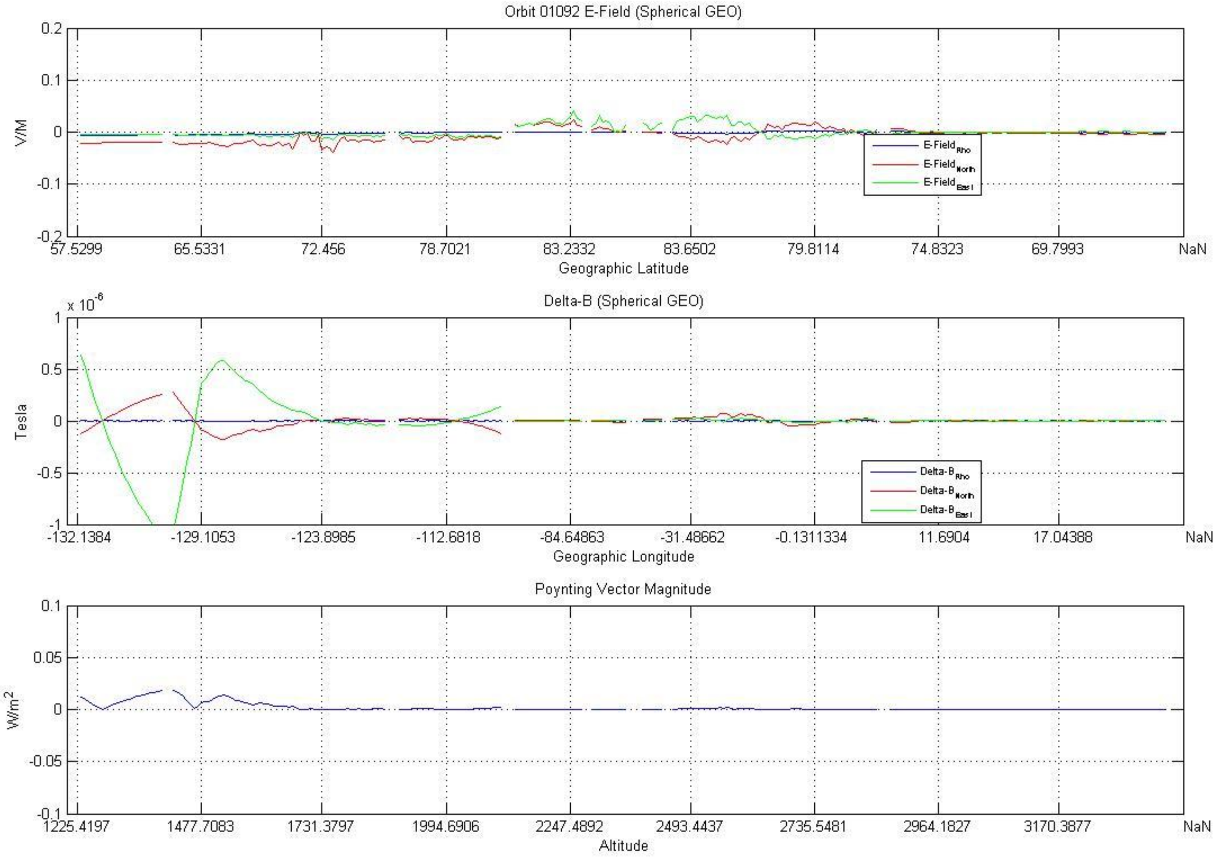


FIGURE 48: ORBIT 1092 ELECTRIC AND MAGNETIC FIELDS IN SPHERICAL GEO; POYNTING FLUX MAGNITUDE

After transforming the fields data to the spherical GEO coordinate system, the westward component gets transformed into an eastward and northward component. At very high latitudes, because there is a Cartesian system trying to describe a spherical orbit, the westward direction also has a north-south component that is extracted during the transformation to spherical coordinates, as can be seen very apparently in the magnetic field plot in Figure 48. The magnitude of the Poynting Flux is less than 50 mW/m, which makes it a small but significant event.

Below in Figure 49 the Cartesian coordinates of the Poynting Vector for orbit 1092 are displayed which was derived by converting the electric and magnetic fields to GEO coordinates before performing the cross product. In Figure 50 the same plot is shown, except that the data was derived by performing the cross product of the fields first and then transforming into GEO coordinates. The plots are identical, which shows that either method will produce the same results.

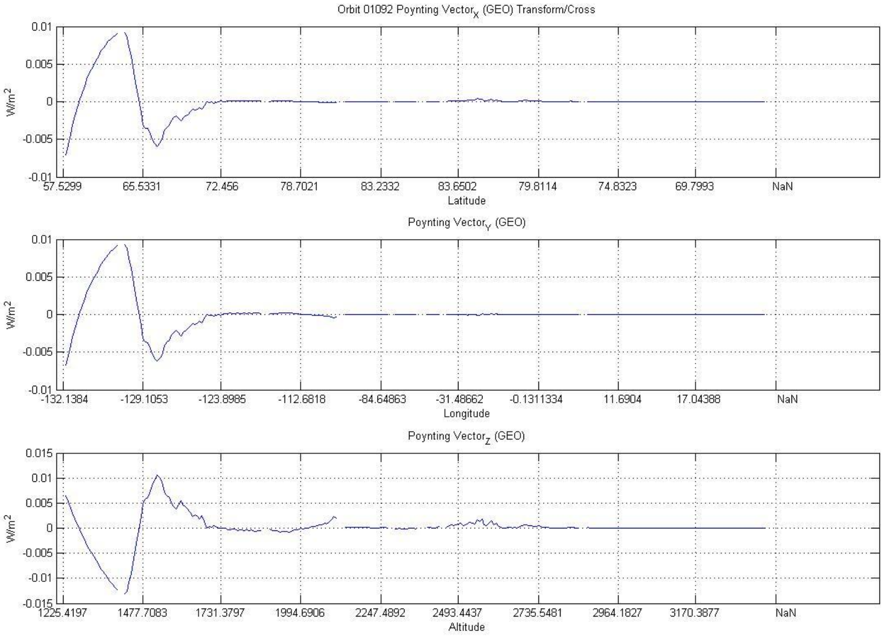


FIGURE 49: ORBIT 1092 POYNTING VECTOR IS CARTESIAN GEO COORDINATES- TRANSFORM THEN CROSS

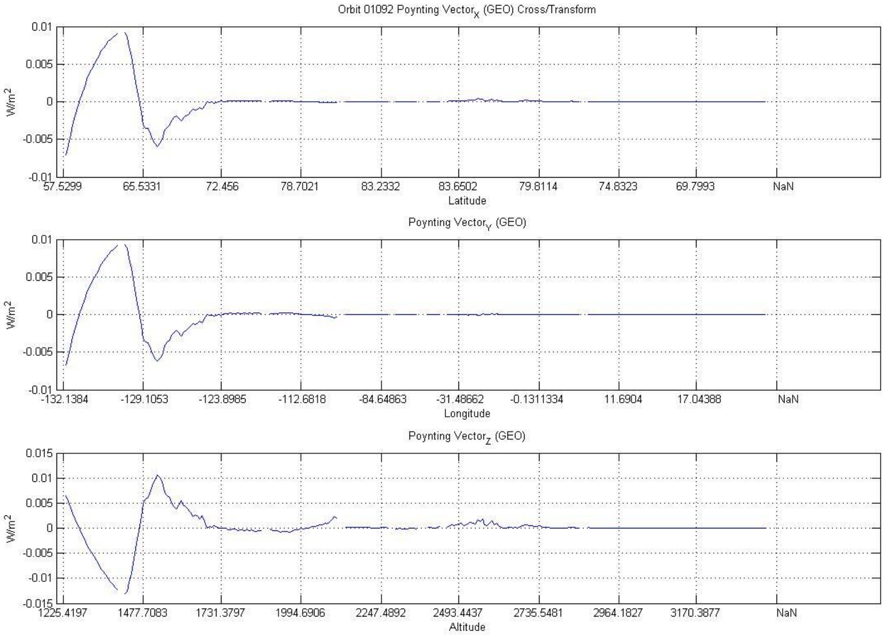


FIGURE 50: ORBIT 1092 POYNTING VECTOR IN CARTESIAN GEO COORDINATES- CROSS THEN TRANSFORM

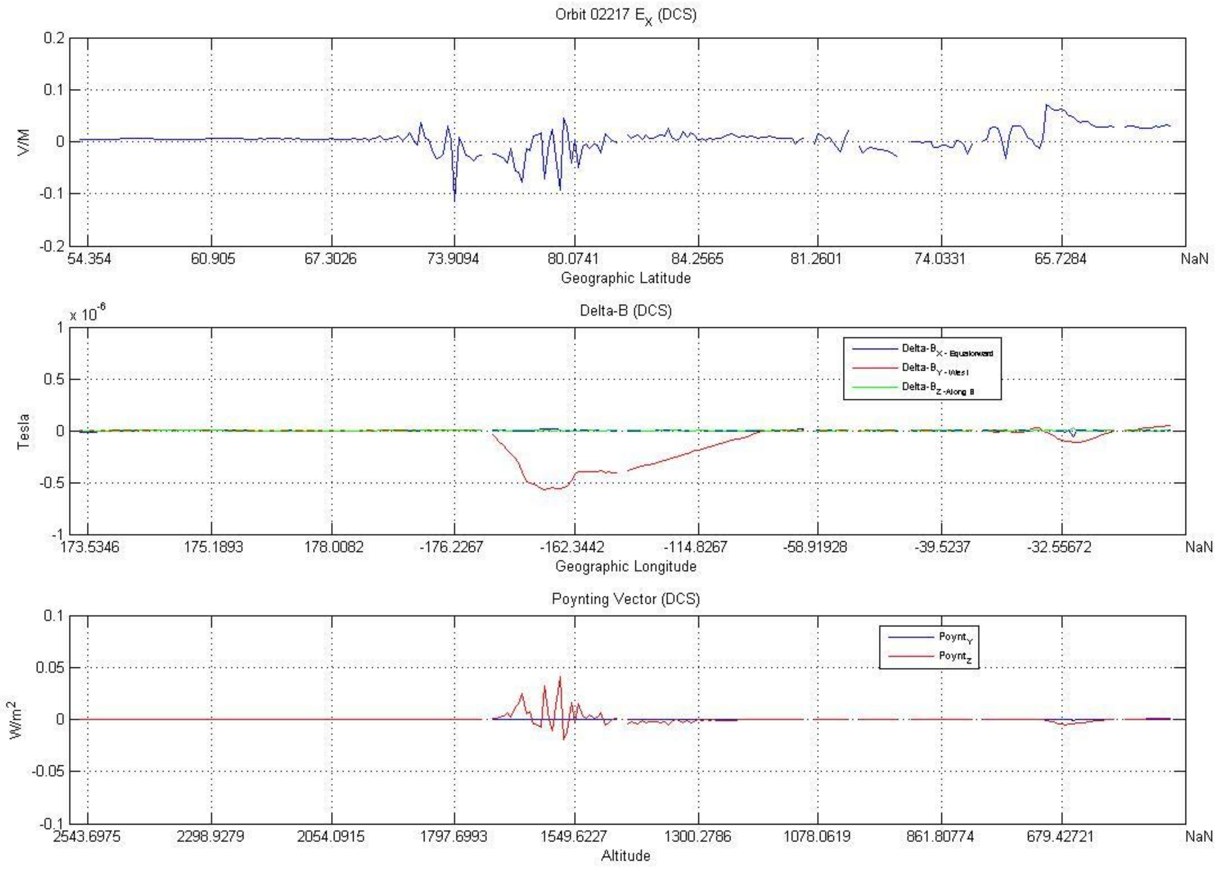


FIGURE 51: ORBIT 2217 ELECTRIC FIELD, MAGNETIC FIELD, AND POYNTING VECTOR IN DCS

Figure 51 shows orbit 2217. Toward the center of this plot there is significant activity for the electric and magnetic fields, as well as a significant Poynting vector in the Z direction. The rapid fluctuations of the vector mean that it is quickly changing direction to be along and against the geomagnetic field repeatedly. This may or may not be “noise” due to the instruments on the satellite- this may be investigated in future years of this project with a larger sample size of Poynting vectors.

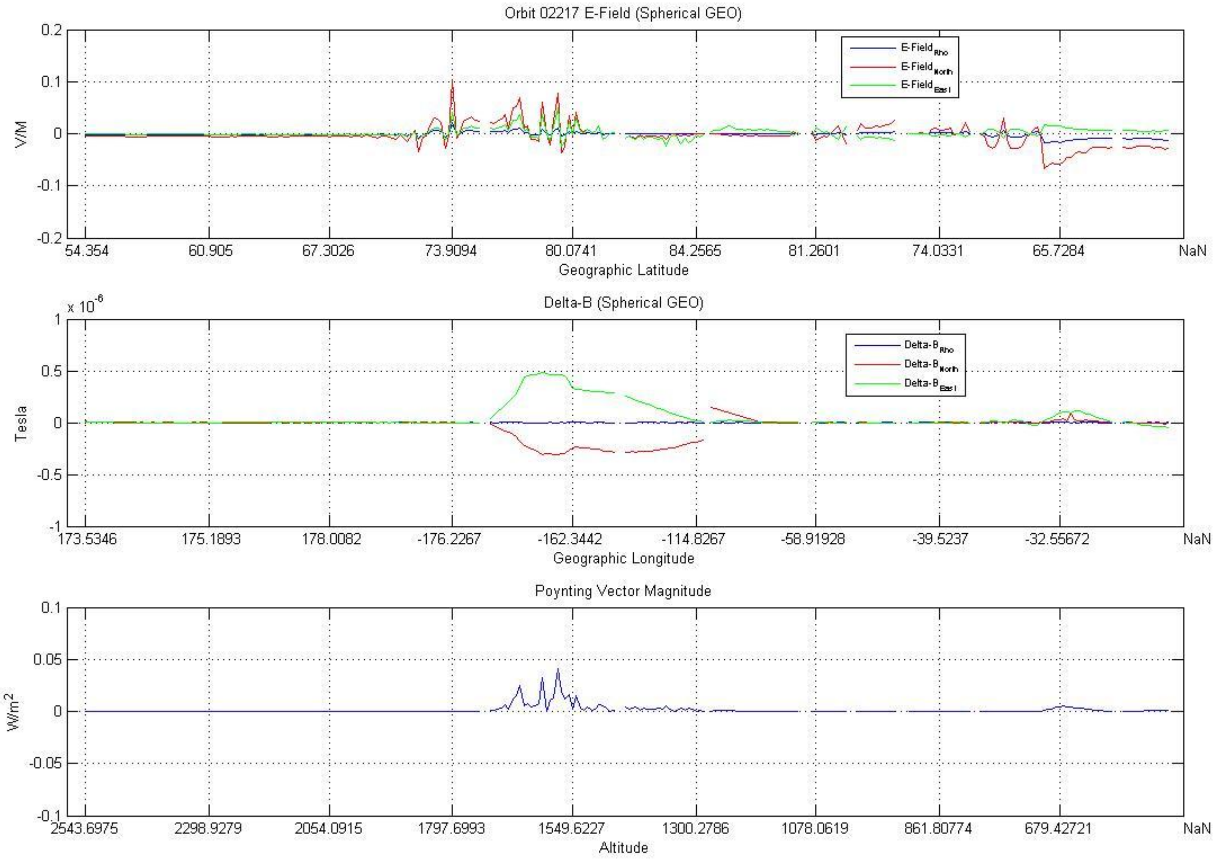


FIGURE 52: ORBIT 2217 ELECTRIC AND MAGNETIC FIELDS IN SPHERICAL GEO COORDINATES; POYNTING VECTOR MAGNITUDE

As mentioned previously, it is difficult to determine whether this Poynting vector is a significant value or is merely noise due to the instruments on the satellite. It does appear to have a significant magnitude at its peak however the rapid fluctuations of the vector make this difficult to classify.

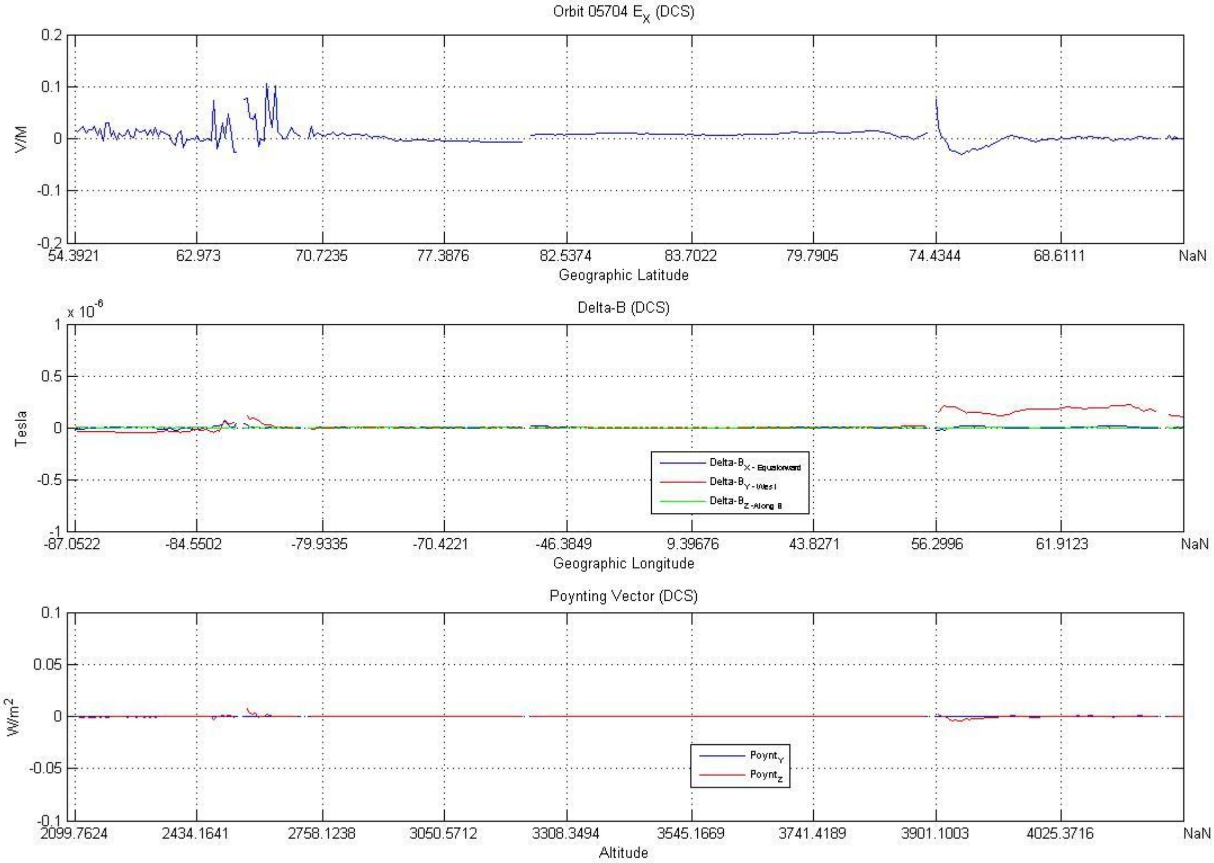


FIGURE 53: ORBIT 5704 ELECTRIC FIELD, MAGNETIC FIELD, AND POYNTING VECTOR IN DCS

Orbit 5704, shown in Figure 53, does not show any significant Poynting vector. This raises the question of whether orbit 2217 can actually be producing noise, as similar noise would be expected on each orbit regardless of data. This requires further study to determine, but for this report suffice it to say that orbit 5704 does not have a significant Poynting vector in any direction.

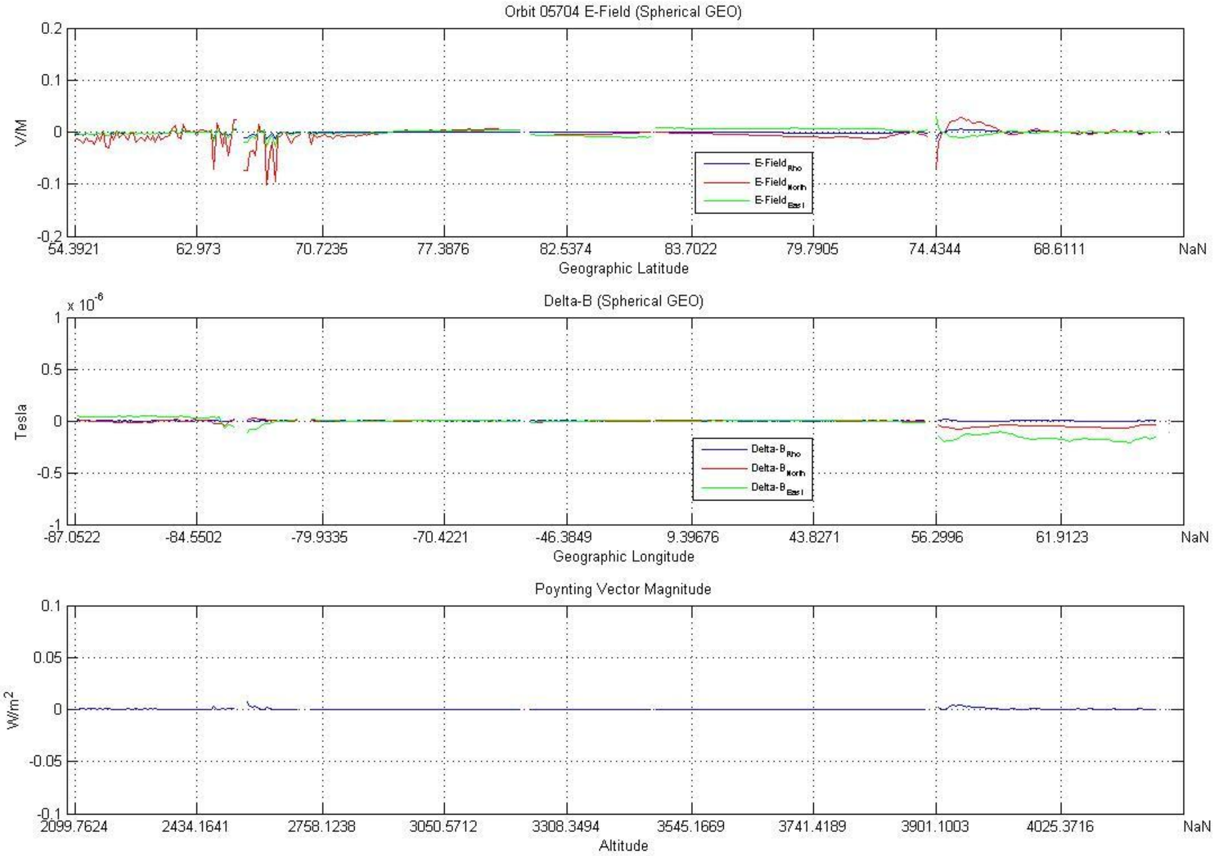


FIGURE 54: ORBIT 5704 ELECTRIC AND MAGNETIC FIELDS IN SPHERICAL GEO; POYNTING VECTOR MAGNITUDE

As discussed above and shown in a different way in Figure 54, orbit 5704 does not have a significant Poynting vector or any magnitude.

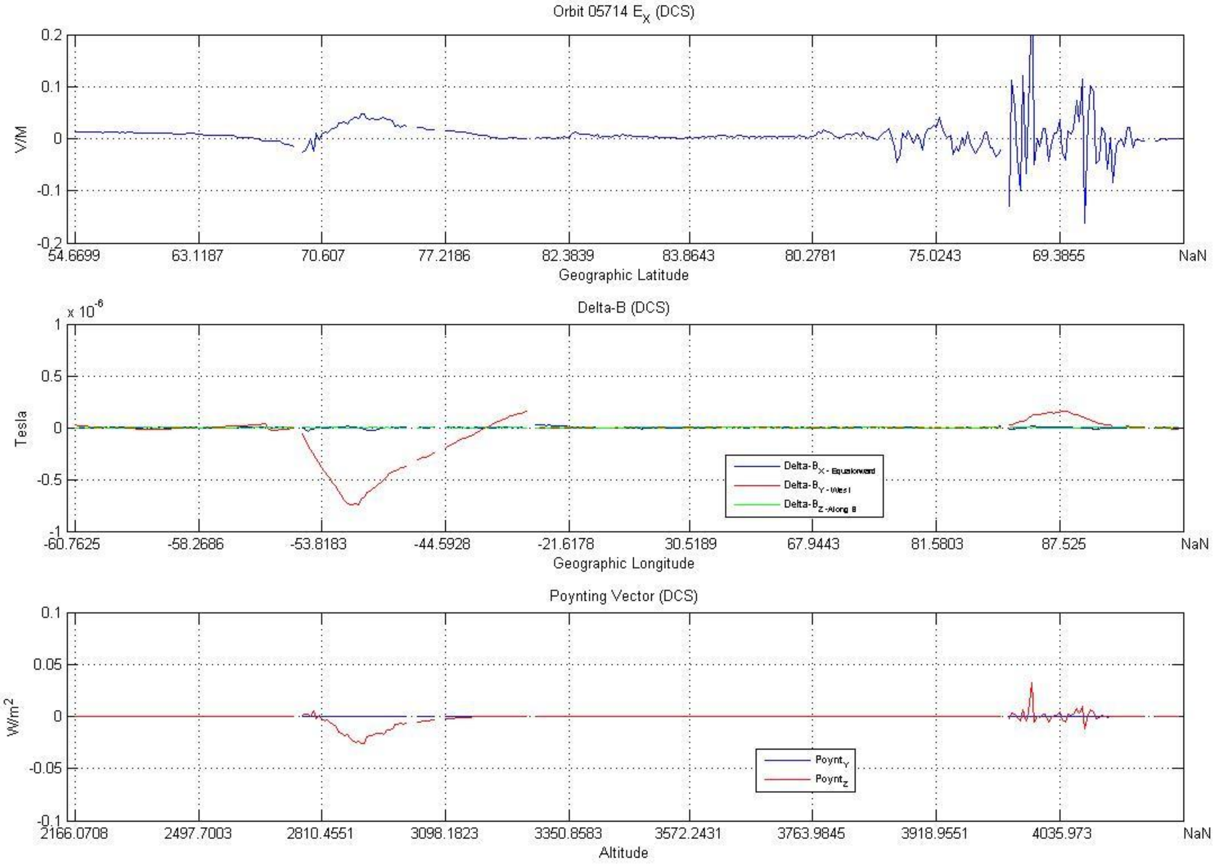


FIGURE 55: ORBIT 5714 ELECTRIC FIELD, MAGNETIC FIELD, AND POYNTING VECTOR IN DCS

Orbit 5714, shown in Figure 55, has a significant electric field in the equatorward direction and a significant magnetic field in the eastward direction. The Poynting vector is travelling against the Earth's magnetic field.

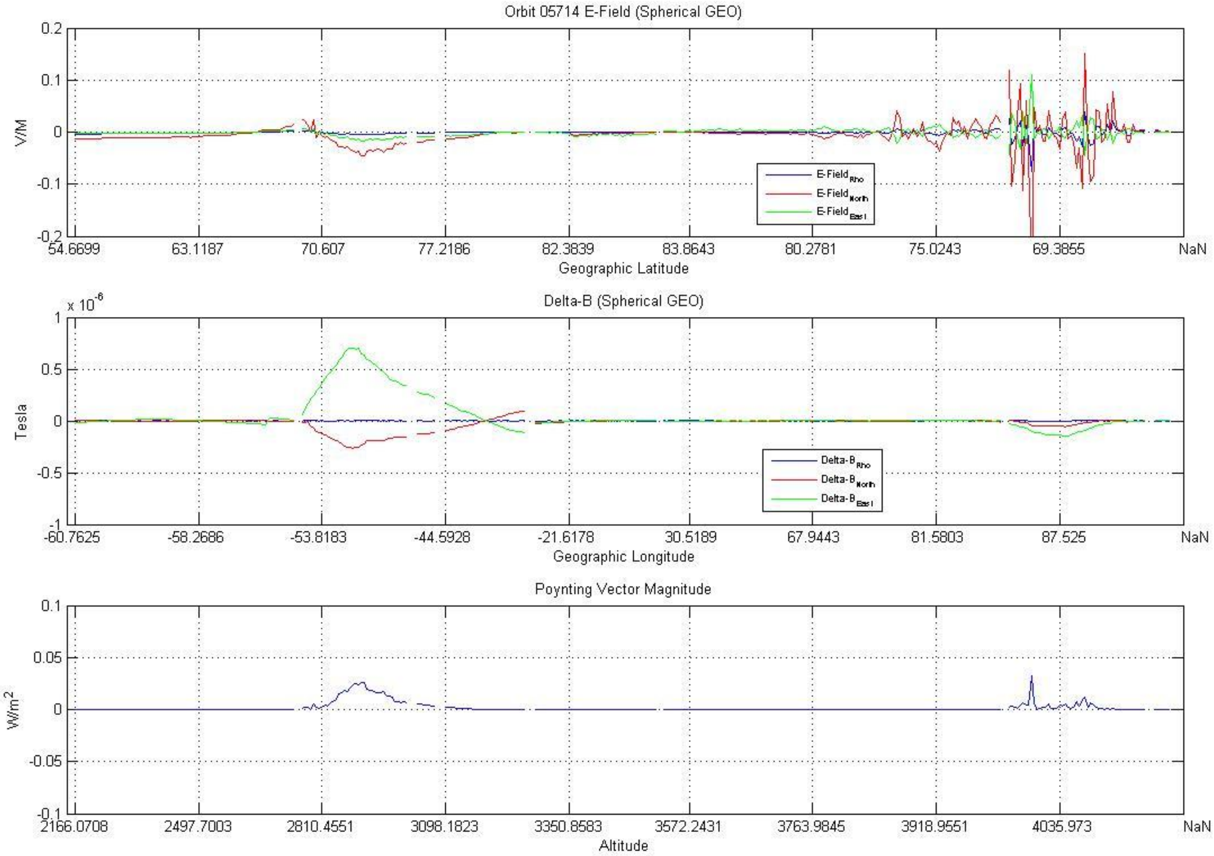


FIGURE 56: ORBIT 5714 ELECTRIC AND MAGNETIC FIELD IN SPHERICAL GEO; POYNTING VECTOR MAGNITUDE

The plots in Figure 56 show some significant observations about this orbit. This orbit shows clearly how the westward DCS component can be split into an eastward and a southward component in spherical GEO coordinates. The Poynting vector has a significant, consistent magnitude.

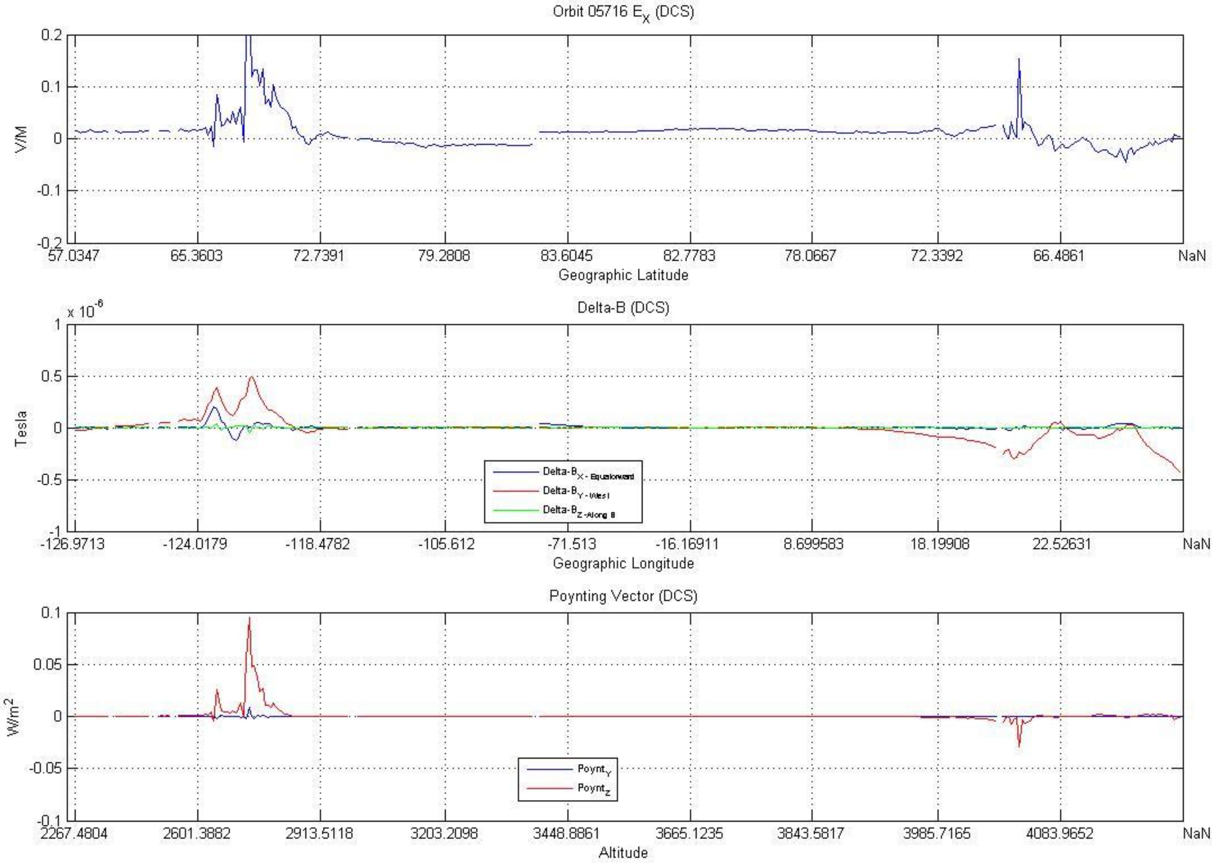


FIGURE 57: ORBIT 5716 ELECTRIC FIELD, MAGNETIC FIELD, AND POYNTING VECTOR IN DCS

Orbit 5716, in Figure 57, has interesting activity in both the electric and magnetic fields. It produces a Poynting vector along with the geomagnetic field in the incoming portion of the orbit, and a smaller Poynting vector during the outgoing portion of the orbit against the geomagnetic field. The Poynting vector at the beginning part of the orbit is similar to the Poynting vectors that researchers will be looking for in the future-energy travelling into the atmosphere during auroral activity, not out of the atmosphere such as in orbit 5714.

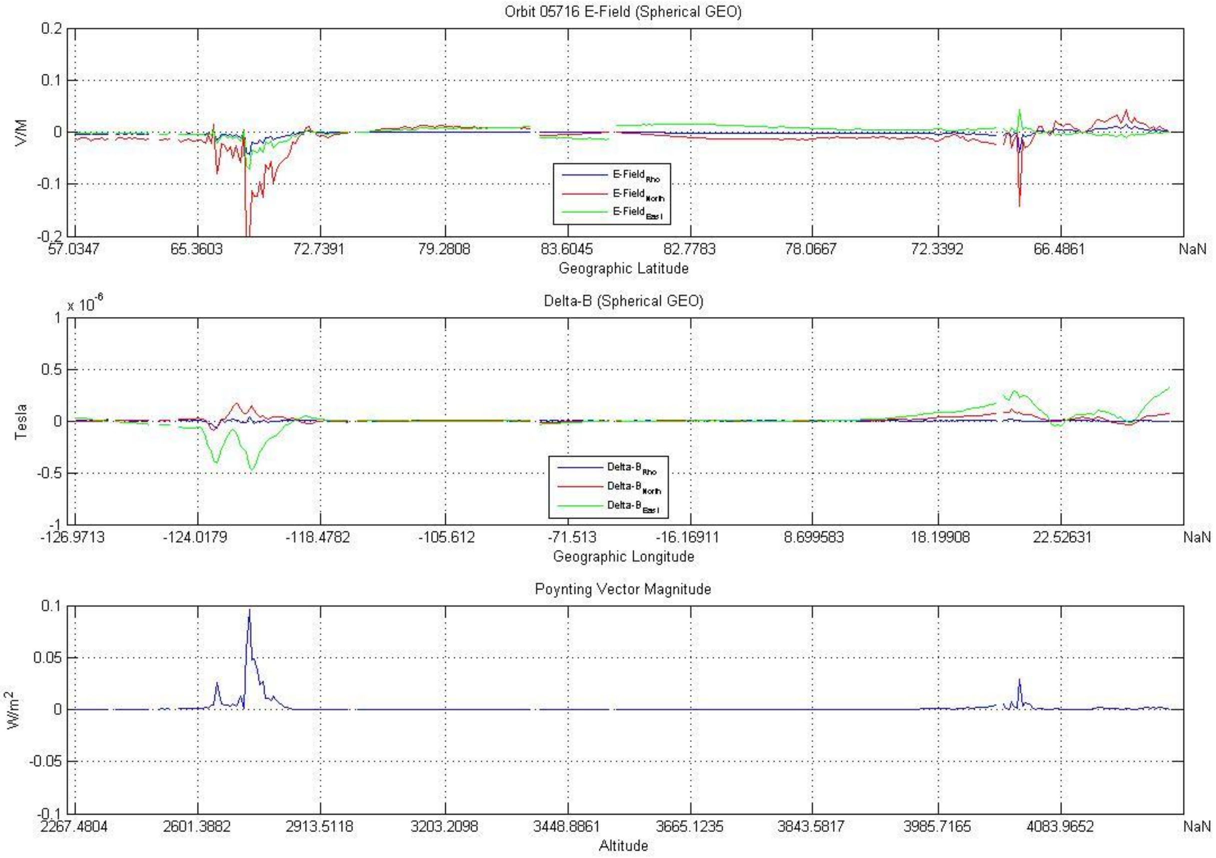


FIGURE 58: ORBIT 5716 ELECTRIC AND MAGENTIC FIELDS IN SPHERICAL GEO; POYNTING VECTOR MAGNITUDE

The magnitude of the Poynting vector, as seen in Figure 58, approaches 100 mW/m^2, which is a significantly large magnitude for the Poynting vector.

7.6 KINETIC ENERGY FLUX

The Kinetic energy flux was really simple to achieve. The magnitudes are much smaller, and any kinetic energy flux that is negative is travelling away from Earth, while the kinetic energy flux that is positive is travelling toward Earth. There are two different kinetic energy fluxes in each of the following plots, Figure 59 through Figure 63, energy that is in the electrons and energy that is in the ions from the solar plasma.

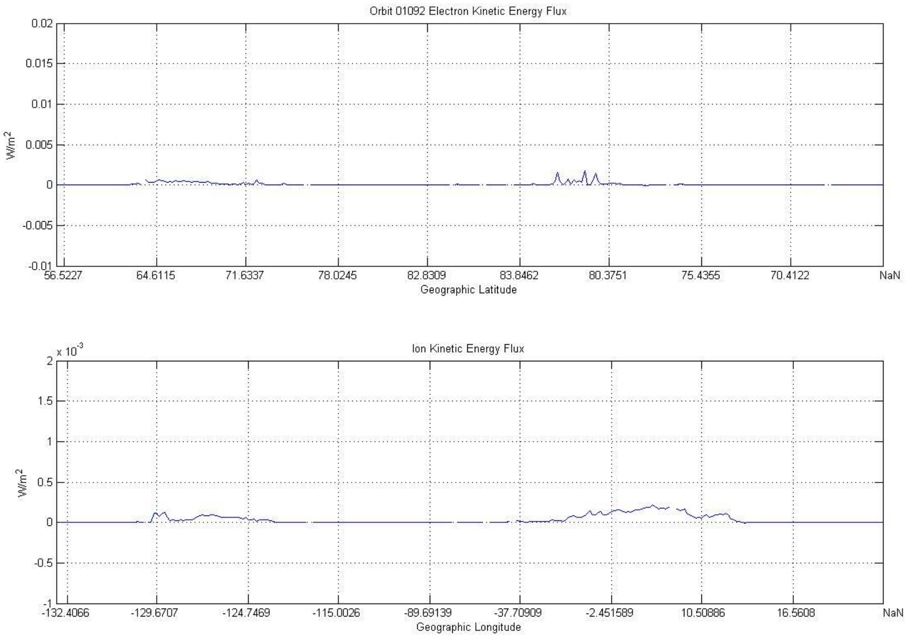


FIGURE 59: ORBIT 1092 ELECTRON AND ION KINETIC ENERGY FLUX

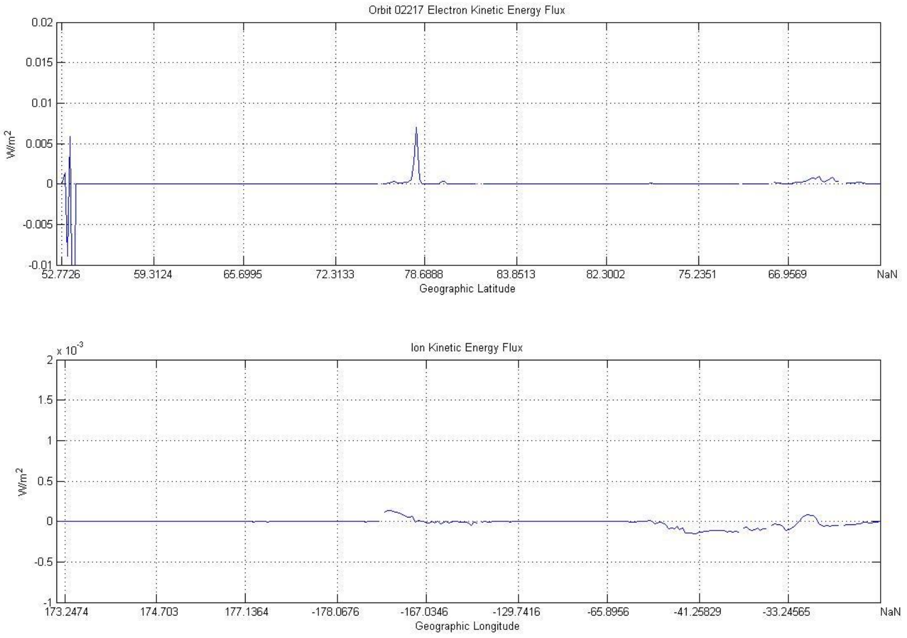


FIGURE 60: ORBIT 2217 ELECTRON AND ION KINETIC ENERGY FLUX

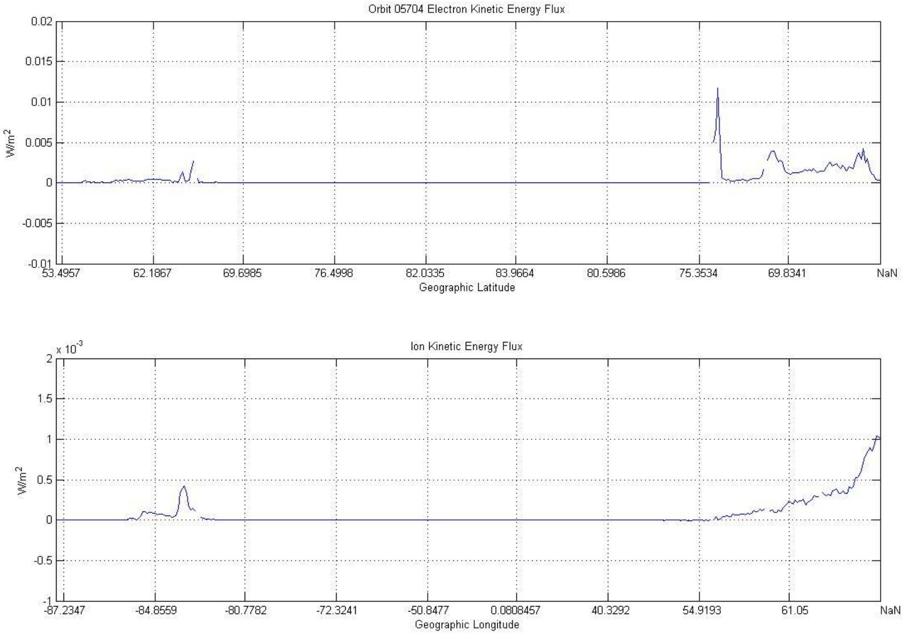


FIGURE 61: ORBIT 5704 ELECTRON AND ION KINETIC ENERGY FLUX

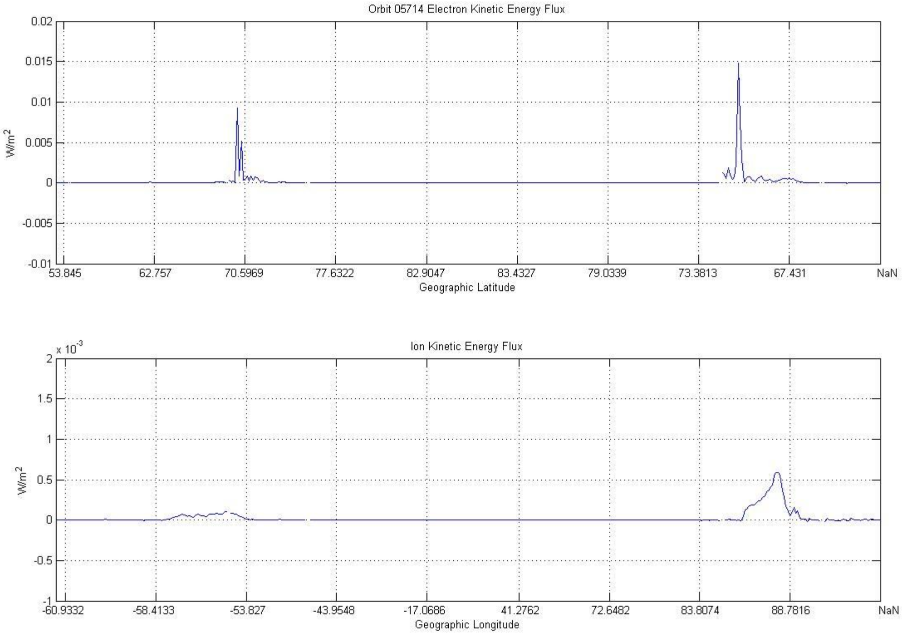


FIGURE 62: ORBIT 5714 ELECTRON AND ION KINETIC ENERGY FLUX

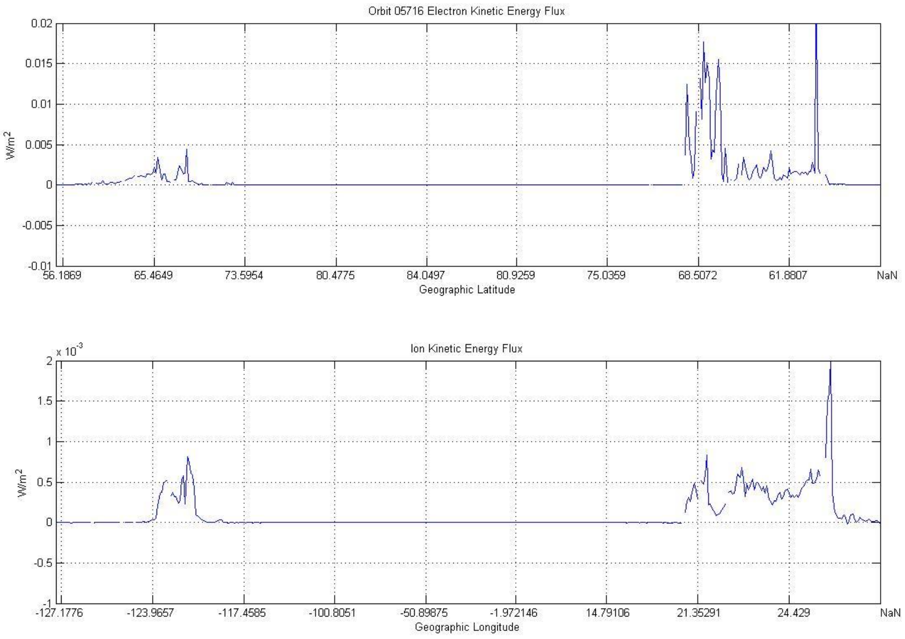


FIGURE 63: ORBIT 5716 ELECTRON AND ION KINETIC ENERGY FLUX

7.7 SUMMARY OF DATA

The tables below are a summary of the data from this entire section. Table 1 shows a summary of the Poynting Flux, the peak for each orbit, the sign for the peak, the width of the activity, and the altitude of the satellite at the time of the peak. Table 2 shows a summary of the kinetic energy flux data for both the ion and electron energy flux.

TABLE 1: POYNTING FLUX SUMMARY

Orbit	Peak Poynting Flux (mW/m^2)	Sign of peak Poynting Flux	Width of Poynting Flux (km)	Altitude (km)
1092	18.4	+	683.9	1300
2217	41.1	+	377	1600
5704	4	-	1912	3950
5714	26.4	-	1087	3000
5716	95.8	+	467.9	2750

TABLE 2: KINETIC ENERGY FLUX SUMMARY

	Kinetic Energy Flux					
	Electron			Ion		
Orbit	Peak(mW/m^2)	Sign	Width(km)	Peak(mW/m^2)	Sign	Width(km)
1092	1.74	+	393.7	0.22	+	1075
2217	6.98	+	232.9	1.56	-	1896
5704	11.7	+	1154	1.03	+	912.9
5714	14.77	+	572.2	0.59	+	399.5
5716	27.11	+	1303	2.07	+	1303

8. RECOMMENDATIONS FOR FUTURE WORK

The first year of this project was designed to pave the path for the future of the NSF project. As mentioned in Section 4.4 NSF Proposal, the second year of this project will be devoted to using the methods outlined in this report to calculate Poynting and Kinetic Energy flux for approximately 20,000 FAST orbits of data and create an average, and create statistical models. The third year will be using these statistical models to create analytical models, and the fourth year will produce General Circulation Models.

9. CONCLUSION

In closing, this report has explained the Silicon Valley 2010 Electrical and Computer Engineering (ECE) Major Qualifying Project (MQP). The goal of the project was to develop processing techniques and routines for calculating the Poynting flux and kinetic energy flux from measurements taken from the Fast Auroral SnapshoT Explorer (FAST) Satellite.

The background section of this report went over all of the topics and concepts required for understanding of the rest of the report. Specifically, sections on the Earth's atmosphere and aurora explained very basically the interaction between the Earth and solar wind. A section on the FAST satellite and the National Science Foundation (NSF) proposal explained that this project is, in fact, only a small part of a much larger set of projects being done by SRI, NSF, and NASA. Finally, a section on reference frames and their relation to this project gave the reader an important lesson on the various definitions of describing time and space. At this point, the reader should have all of the basic knowledge required to understand the remainder of the report.

In explaining the project and the final goals and deliverables, it is important to reiterate that this is only part of a larger scale project proposed by Doctors Russell Cosgrove and Hasan Bahcivan of SRI. Not only did the final deliverables include actual calculations of the Poynting flux and kinetic energy flux for a limited number of orbits, but all of the processes and routines required to do such calculations. This report also give details as to how the final results of the large scale project fit into other areas of research like general circulation models.

One of the most important sections of this report detailed how the students actually performed the tasks of calculating Poynting flux and kinetic energy flux. The methodologies section of the report went over how to obtain raw and processed data from the FAST satellite as well as how to view plots of the measured magnetic field, delta-magnetic field, electric field, kinetic flux, and positional data. The methodologies section continued onto how to correlate-in-time or "line-up" positional, fields, and kinetic energy data before clarifying how to convert data into different reference frames. The methodologies of this project are actually so important that a number of supporting sources outside of the report were created in order to better explain its various intricacies. Some external video sources include videos on how to run the MATLAB routines listed in the Appendices, how to view and export data using SDT and coordinate systems. There is also external documentation of correspondence between students of the MQP and researchers at University of California schools.

Another important section of the report included some of the actual deliverables of the project. The data and results section of the report displayed graphs of data for 5 different orbits. The different graphs show the small steps leading up to the final calculation of the Poynting flux and kinetic energy flux. Graphs of data include: given data, transformed positional data, the IGRF model, and finally Poynting and kinetic energy flux. The remainder of the report focused on recommendations of future work and wrapping up any questions that the reader may have about the project.

10. BIBLIOGRAPHY

3 January 2010 <http://sprg.ssl.berkeley.edu/fast/>.

Acklam, Peter J. MATLAB Utilities- Time and date utilities. 14 April 2008. January 2010 <http://home.online.no/~pjacklam/matlab/software/util/timeutil/>.

Atmosphere of Earth. 14 January 2010 <http://en.wikipedia.org/wiki/Atmosphere_of_Earth>.

Cosgrove, Russell. "NSWP: Poynting and Kinetic-Energy Flux Derived from the FAST Satellite." Proposal. 2009.

Epoch Converter. January 2010 <http://www.epochconverter.com/>.

Fogt, Robert. Online Conversion- Unix time conversion. 1997-2008. 22 January 2010 <http://www.onlineconversion.com/unix_time.htm>.

Geophysical Institute. Frequently Asked Questions about the Aurora; . August 2003. 22 January 2010 <http://www.gi.alaska.edu/asahi/>.

Hapgood, Mike. Coordinate transformations between geocentric systems. 4 June 1997. January 2010 <http://sspg1.bnsc.rl.ac.uk/SEG/Coordinates/geo_tran.htm>.

—. Rotation angles for space physics coordinate transformations. 6 October 1999. January 2010 <http://sspg1.bnsc.rl.ac.uk/SEG/Coordinates/angles.htm#gst>.

Kivelson, Margaret and Christopher Russell. Introduction to Space Physics. Cambridge: Cambridge University Press, 1995.

Lummerzheim, Dirk. Aurora FAQ. 12 5 2009. 22 January 2010 <http://odin.gi.alaska.edu/FAQ/>.

NASA. The Mystery- Aurora History. 2004. 22 January 2010 <http://ds9.ssl.berkeley.edu/themis/mission_auroraexplain.html>.

National Weather Service. NWS Jetstream - Introduction to the Atmosphere. 5 January 2010. 14 January 2010 <http://www.srh.noaa.gov/jetstream//atmos/atmos_intro.htm>.

North Magnetic Pole- Wikipedia, the free encyclopedia. 6 February 2010 <http://en.wikipedia.org/wiki/North_Magnetic_Pole>.

Pfaff, R., et al. "An Overview of the Fast Auroral SnapshoT (FAST) Satellite." Fast Auroral SnapshoT Explorer. NASA/Goddard Space Flight Center; University of California, Berkeley. 14 January 2010 <http://sprg.ssl.berkeley.edu/fast/scienceprod/papers/pfaff/FAST_Overview_formatted.pdf>.

Poynting Vector- Wikipedia, the free encyclopedia. 6 February 2010 <http://en.wikipedia.org/wiki/Poynting_vector>.

Roithmayr, Carlos. MATLAB routine to load Schmidt-normalized coefficients retrieved from ftp://nssdc.gsfc.nasa.gov/pub/models/igrf/igrf95.dat. 22 January 1997.

Schmitt, Stephen R. Sidereal Clock. 2004. January 2010.

South Magnetic Pole- Wikipedia, the free encyclopedia. 6 February 2010 <http://en.wikipedia.org/wiki/South_Magnetic_Pole>.

Strangeway, Robert. "E-mail Correspondance about FAST CDF Files."

Umland, Henning. Almanac. 2001-2008. January 2010 <http://www.celnav.de/longterm.htm>.

APPENDICES: MATLAB FUNCTIONS AND M-FILES

APPENDIX A: PROCESS_FLUX.M

```
% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International


% This is a top-level m-file which calls multiple other functions and
% m-files in order to process summary CDF files to find the Poynting
% vector and plot relevant quantities

clear;
clc;
close all;

%% Load Data
[field_files,positional_files,electron_files,ion_files] = file_names;


%% Define which Orbits to Process from "field_files" function
for orbit = 7:7

    clearvars -except field_files positional_files electron_files ion_files orbit

    [field_matrix, pos_matrix, electron_matrix,ion_matrix] =
load_data(field_files{orbit},positional_files{orbit},electron_files{orbit},
ion_files{orbit});
    %[fmat_5704,pmat_5704]=load_data('fa_k0_dcf_07641_v01','fa_k0_orb_07641_v01');
    %;load_data('fa_k0_dcf_05704_v02','fa_k0_orb_05704_v01');

    %% Define which values to plot
    e_sat_plot=0;
    b_sat_plot=0;
    sat_geo_plot=0;
    sat_gei_plot = 0;
    sat_traj_plot=0;
    spin_plot=0;
    uspb_plot=0;
    uspbp_plot=0;
    b_geo_plot=0;
    e_geo_plot=0;
    igrf_plot=0;
    poynt_plot1=0; % transform/cross
    poynt_plot2=0; % cross/transform
    poynt_sat_plot =0;
    poynt_down_b_plot1=0;% transform/cross
    poynt_down_b_plot2=0; % cross/transform
    final_plot1 =0;  %E-Fields,B-fields, and Poynting Vector (DCS)
    final_plot2 =0; %E-Fields,B-fields (Spherical GEO), and Poynting Vector Magnitude']
    final_plot3 =1;  % Kinetic Energy flux
    %% Interpolate Data
    [good_f_matrix,good_e_matrix, good_i_matrix,
interp_p_f_matrix,interp_p_e_matrix,interp_p_i_matrix]= interp_data(field_matrix,
pos_matrix, electron_matrix,ion_matrix);


    % Find Time Vector (Approximately UT)
    for i=1:length(interp_p_f_matrix)
        [m(i,1),m(i,2),m(i,3),m(i,4),m(i,5),m(i,6) ] =
unixtime2mat(interp_p_f_matrix(i,1));
    end


    %% Find Positional Data of Satellite in GEO Coordinates from GEI
    % Also finds latitude, longtitude, and altitude (GEODETIC)

    % Find GEO position of Field data
    [field_pos_geo,field_vel] = gei2geo(interp_p_f_matrix);
    [field_lat_long_alt] = satalt(field_pos_geo);

    % Find GEO position of Electron Data
    [elec_pos_geo,elec_vel] = gei2geo(interp_p_e_matrix);
    [elec_lat_long_alt] = satalt(elec_pos_geo);

    % Find GEO Position of Ion data
    [ion_pos_geo,ion_vel] = gei2geo(interp_p_i_matrix);
    [ion_lat_long_alt] = satalt(ion_pos_geo);

    %% Find IGRF Model
    % We only need to find this for the positions of the satellite for the
    % fields data

    field_igrf = get_igrf(field_pos_geo);
    % Normalized vectors of model
    norm_field_igrf = normr(field_igrf(:,2:4));

    %% Convert data from Data Reference Frame to GEO

[delta_b_geo,e_geo,u_spin,u_traj,u_spB,u_spBp,e_other,u_other,u_west,crossing,incomingnor
th] = sat2geo(good_f_matrix,field_pos_geo,field_vel,field_igrf);

    % Permiability of free space u_0
    pfs = 4*pi*1e-7;

    % Find H in the GEO frame (both cartesian and Spherical)
    h_geo = delta_b_geo/pfs;

    % Find H in the satellite data frame
    h_sat = (good_f_matrix(:,5:7)/pfs)*1e-9;

    % Find E in the satellite data frame
    e_sat(:,1:3) = good_f_matrix(:,2:4);
    e_sat = e_sat*1e-3; % output in Volts/meter

    %% Calculte the Poynting Vector
    % Done in 2 ways: 1) find cross product of field data data reference
    % frame to find P-Vector in data reference frame and convert the
    % p_vector from DRF to GEO OR  2) Convert field data into GEO reference
    % frame first and then find cross product

    % Find the poynting vector in satellite data frame
    poynt_sat = cross(e_sat,h_sat);

    % Poynting vector in cartesian GEO (transform/cross)
    poynt_geo1 = cross(e_geo(:,1:3),h_geo(:,1:3));

    % find the Poynting Vector in GEO (Cross/transform)
    for i = 1:length(poynt_sat)
        poynt_geo2(i,1:3) = (u_spBp(i,1:3)*poynt_sat(i,1)) +
(u_west(i,1:3)*poynt_sat(i,2)) + (u_spB(i,1:3)*poynt_sat(i,3));
    end


    %poytnting vector down b-field
    % find projection of poynting vector along the Normalized IGRF
    for i = 1:length(poynt_geo1)
        % transform/cross
        poynt_down_b1(i) = dot(poynt_geo1(i,:), norm_field_igrf(i,:)); % scalar
        fraction_down_b1(i) = poynt_down_b1(i) /(sqrt((poynt_geo1(i,1)^2 +
poynt_geo1(i,2)^2 + poynt_geo1(i,3)^2)));
        % cross/transform
        poynt_down_b2(i) = dot(poynt_geo2(i,:), norm_field_igrf(i,:)); % scalar
        fraction_down_b2(i) = poynt_down_b2(i) /(sqrt((poynt_geo2(i,1)^2 +
poynt_geo2(i,2)^2 + poynt_geo2(i,3)^2)));
    end

    % find the index's within frac_down_b that DONT have NaN
    % frac_down_b_index=find(~isnan(fraction_down_b));
    % mean(fraction_down_b(frac_down_b_index));

    %% Find Magnitudes of B-Feild, E-Field, IGRF, and Poynting Flux
    for i = 1:length(delta_b_geo);
        mag_b(i) = sqrt((delta_b_geo(i,1)^2)+ (delta_b_geo(i,2)^2)+
(delta_b_geo(i,3)^2));
        mag_igrf(i) = sqrt((field_igrf(i,2)^2)+ (field_igrf(i,3)^2)+
(field_igrf(i,4)^2));
        mag_e(i) = sqrt((e_geo(i,1)^2)+ (e_geo(i,2)^2)+ (e_geo(i,3)^2));
        mag_poynt1(i) = sqrt((poynt_geo1(i,1)^2)+ (poynt_geo1(i,2)^2)+
(poynt_geo1(i,3)^2));
    end


    for j=1:length(delta_b_geo);
        angle(j) = acosd((dot(u_spB(j,1:3),field_igrf(j,2:4))/mag_igrf(j)));
    end;


    %% Plot Results

    % Find first lat_long_alt value that does NOT contain a NaN
    field_first = 1;
    elec_first = 1;
    ion_first = 1;
    while isnan(field_lat_long_alt(field_first)) == 1,
        field_first = field_first+1;
    end
    while isnan(elec_lat_long_alt(elec_first)) == 1,
        elec_first = elec_first+1;
    end
    while isnan(ion_lat_long_alt(ion_first)) == 1,
        ion_first = ion_first+1;
    end
    field_x = round(linspace(field_first,length(field_lat_long_alt),15));
    field_time = 1:length(field_pos_geo);
    elec_x = round(linspace(elec_first,length(elec_lat_long_alt),15));
    elec_time = 1:length(elec_pos_geo);
    ion_x = round(linspace(ion_first,length(ion_lat_long_alt),15));
    ion_time = 1:length(ion_pos_geo);

    if (e_sat_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Electric Fields in DCS
Coordinates'])
        subplot(3,1,1); plot(good_f_matrix(:,2));title(['Orbit '
field_files{orbit}(11:15) ' E-Field_X (Satellite Coordinate System)']);ylabel('mV/m');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(good_f_matrix(:,3));title('E-Field_Z (Satellite Coordinate
System)');ylabel('mV/m');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3); plot(e_other);title('E-Field_{other} (Satellite Coordinate
System)');ylabel('mV/m');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end



    if (b_sat_plot == 1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Magnetic Fields in DCS
Coordinates'])

        subplot(3,1,1); plot(field_time,good_f_matrix(:,5));title(['Orbit '
field_files{orbit}(11:15) ' B-Field_X (Satellite Coordinate System)']);ylabel('nT');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');
        grid on;axis([0 length(field_time) min(good_f_matrix(:,5))
max(good_f_matrix(:,5))])

        subplot(3,1,2); plot(field_time,good_f_matrix(:,6));title('B-Field_Y (Satellite
Coordinate System)');ylabel('nT');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');
        grid on;axis([0 length(field_time) min(good_f_matrix(:,6))
max(good_f_matrix(:,6))])
```

```
        subplot(3,1,3); plot(field_time,good_f_matrix(:,7));title('B-Field_Z (Satellite
Coordinate System)');ylabel('nT');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');
        grid on;axis([0 length(field_time) min(good_f_matrix(:,7))
max(good_f_matrix(:,7))])
    end

    if (sat_geo_plot == 1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Satellite Position in GEO
Coordinates'])

        subplot(3,1,1);plot(field_time,field_pos_geo(:,2));title(['Orbit '
field_files{orbit}(11:15) ' Satellite GEO_X (Unit GEO)']);ylabel('GEO');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');
        grid on;axis([0 length(field_time) min(field_pos_geo(:,2))
max(field_pos_geo(:,2))])

        subplot(3,1,2);plot(field_time,field_pos_geo(:,3));title('Satellite GEO_Y (Unit
GEO)');ylabel('GEO');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');
        grid on;axis([0 length(field_time) min(field_pos_geo(:,3))
max(field_pos_geo(:,3))])

        subplot(3,1,3);plot(field_time,field_pos_geo(:,4));title('Satellite GEO_Z (Unit
GEO)');ylabel('GEO');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');
        grid on;axis([0 length(field_time) min(field_pos_geo(:,4))
max(field_pos_geo(:,4))])

    end

    if (sat_gei_plot == 1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Satellite Position in GEI
Coordinates'])

        subplot(3,1,1);plot(field_time,interp_p_f_matrix(:,2));title(['Orbit '
field_files{orbit}(11:15) ' Satellite GEI_X )']);ylabel('km');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');
        grid on;axis([0 length(field_time) min(interp_p_f_matrix(:,2))
max(interp_p_f_matrix(:,2))])

        subplot(3,1,2);plot(field_time,interp_p_f_matrix(:,3));title('Satellite GEI_Y
');ylabel('km');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');
```

```
        grid on;axis([0 length(field_time) min(interp_p_f_matrix(:,3))
max(interp_p_f_matrix(:,3))])

        subplot(3,1,3);plot(field_time,interp_p_f_matrix(:,4));title('Satellite GEI_Z
');ylabel('km');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');
        grid on;axis([0 length(field_time) min(interp_p_f_matrix(:,4))
max(interp_p_f_matrix(:,4))])
    end

    if (sat_traj_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Satellite Trajectory in GEO
Coordiantes'])
        subplot(3,1,1);plot(field_time,u_traj(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Satellite Trajectory_X (Unit GEO)']);ylabel('Unit GEO m/s');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(field_time,u_traj(:,2));title('Satellite Trajectory_Y (Unit
GEO)');ylabel('Unit GEO m/s');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,u_traj(:,3));title('Satellite Trajectory_Z (Unit
GEO)');ylabel('Unit GEO m/s');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

%     if (spin_plot ==1),
%         figure('name',['Orbit ' field_files{orbit}(11:15) ': Spin-Axis Vector in GEO
Coordiantes'])
%         subplot(3,1,1);plot(field_time,u_spin(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Spin Axis Vector_X (Unit GEO)']);ylabel('Unit GEO m/s');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
%         subplot(3,1,2);plot(field_time,u_spin(:,2));title('Spin Axis Vector_Y (Unit
GEO)');ylabel('Unit GEO m/s');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
%         subplot(3,1,3);plot(field_time,u_spin(:,3));title('Spin Axis Vector_Z (Unit
GEO)');ylabel('Unit GEO m/s');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
%     end
%
%     if (uspb_plot ==1),
%         figure('name',['Orbit ' field_files{orbit}(11:15) ': Spin Plane Projection of
B-Field'])
%         subplot(3,1,1);plot(field_time,u_spB(:,1));title(['Orbit '
field_files{orbit}(11:15) ' U-spb_x']);
```

```
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
%         subplot(3,1,2);plot(field_time,u_spB(:,2));title('U-spb_Y');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
%         subplot(3,1,3);plot(field_time,u_spB(:,3));title('U-spb_Z');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
%     end
%
%     if (uspbp_plot ==1),
%         figure('name',['Orbit ' field_files{orbit}(11:15) ': Vector Perpendicular to
Spin Plane Projection of B-Field'])
%         subplot(3,1,1);plot(field_time,u_spBp(:,1));title(['Orbit '
field_files{orbit}(11:15) ' U-spbp_x, out N']);
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
%         subplot(3,1,2);plot(field_time,u_spBp(:,2));title('U-spbp_Y, out N');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
%         subplot(3,1,3);plot(field_time,u_spBp(:,3));title('U-spbp_Z, out N');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
%     end

    if (b_geo_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Delta-B Field in GEO
Coordinates'])
        subplot(3,1,1);plot(field_time,delta_b_geo(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Delta-B_X (GEO)']);ylabel('Tesla');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(field_time,delta_b_geo(:,2));title('Delta-B_Y
(GEO)');ylabel('Tesla');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,delta_b_geo(:,3));title('Delta-B_Z (GEO)');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (e_geo_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Electric Fields in GEO
Coordinates'])
        subplot(3,1,1);plot(field_time,e_geo(:,1));title(['Orbit '
field_files{orbit}(11:15) ' E_X (GEO)']); ylabel('V/M');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
```

```
        subplot(3,1,2);plot(field_time,e_geo(:,2));title('E_Y (GEO)'); ylabel('V/M');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,e_geo(:,3));title('E_Z (GEO)'); ylabel('V/M');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (igrf_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': IGRF in GEO Coordinates'])
        subplot(3,1,1);plot(field_time,field_igrf(:,2));title(['Orbit '
field_files{orbit}(11:15) ' IGRF Model GEO_X']);ylabel('Tesla');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');
        grid on;axis([0 length(field_time) min(field_igrf(:,2)) max(field_igrf(:,2))])

        subplot(3,1,2);plot(field_time,field_igrf(:,3));title('IGRF Model
GEO_Y');ylabel('Tesla');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');
        grid on;axis([0 length(field_time) min(field_igrf(:,3)) max(field_igrf(:,3))])

        subplot(3,1,3);plot(field_time,field_igrf(:,4));title('IGRF Model
GEO_Z');ylabel('Tesla');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');
        grid on;axis([0 length(field_time) min(field_igrf(:,4)) max(field_igrf(:,4))])

    end

    if (poynt_plot1 ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Poynting Vector in GEO
Coordinates'])
        subplot(3,1,1);plot(field_time,poynt_geo1(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Poynting Vector_X (GEO) Transform/Cross']);ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(field_time,poynt_geo1(:,2));title('Poynting Vector_Y
(GEO)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,poynt_geo1(:,3));title('Poynting Vector_Z
(GEO)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (poynt_plot2 ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Poynting Vector in GEO
Coordinates'])
```

```
        subplot(3,1,1);plot(field_time,poynt_geo2(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Poynting Vector_X (GEO) Cross/Transform']);ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(field_time,poynt_geo2(:,2));title('Poynting Vector_Y
(GEO)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,poynt_geo2(:,3));title('Poynting Vector_Z
(GEO)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (poynt_sat_plot ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Poynting Vector in SAT
Coordinates'])
        subplot(3,1,1);plot(field_time,poynt_sat(:,1));title(['Orbit '
field_files{orbit}(11:15) ' Poynting Vector_X (Sat) Cross/Transform']);ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(3,1,2);plot(field_time,poynt_sat(:,2));title('Poynting Vector_Y
(DCS)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Longitude');grid on;
        subplot(3,1,3);plot(field_time,poynt_sat(:,3));title('Poynting Vector_Z
(DCS)');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (poynt_down_b_plot1 ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Poynting Vector Along B-
Field'])
        subplot(2,1,1);plot(field_time,poynt_down_b1);title(['Orbit '
field_files{orbit}(11:15) ' Poynting Vector Along IRGF (GEO)
Transform/Cross']);ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
        subplot(2,1,2);plot(field_time,fraction_down_b1);title('Fraction of Poynting
Vector Along IRGF (GEO)');ylabel('W/m^2');axis([ 1 length(poynt_down_b1) -1.1 1.1]);

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    end

    if (poynt_down_b_plot2 ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Poynting Vector Along B-
Field'])
```

```
    subplot(2,1,1);plot(field_time,poynt_down_b2);title(['Orbit '
field_files{orbit}(11:15) ' Poynting Vector Along IRGF (GEO)
Cross/Transform']);ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Latitude');grid on;
    subplot(2,1,2);plot(field_time,fraction_down_b2);title('Fraction of Poynting
Vector Along IRGF (GEO)');ylabel('W/m^2');axis([ 1 length(poynt_down_b2) -1.1 1.1]);

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
  end

  if (final_plot1 ==1),
    figure('name',['Orbit ' field_files{orbit}(11:15) ': E-Fields,B-fields, and
Poynting Vector (DCS)']);

    subplot(3,1,1);
    plot(field_time,e_sat(:,1));
    title(['Orbit ' field_files{orbit}(11:15) ' E_X (DCS)']);
    ylabel('V/M');
    set(gca,'XTick',field_x);
    set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});
    xlabel('Geographic Latitude');grid on;
    axis([0 length(field_time) -.2 0.2])

    subplot(3,1,2);
    plot(field_time,good_f_matrix(:,5)*1e-9,'blue');title('Delta-B
(DCS)');ylabel('Tesla');hold;
    plot(field_time,good_f_matrix(:,6)*1e-
9,'red');plot(field_time,good_f_matrix(:,7)*1e-9,'green');hold;
    EBP_legend1 = legend('Delta-B_{X - Equatorward}','Delta-B_{Y - West}','Delta-B_{Z
-Along B}','Location','Best');
    set(EBP_legend1,'FontSize',7);
    set(gca,'XTick',field_x);
    set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});
    xlabel('Geographic Longitude');grid on;hold;
    axis([0 length(field_time) -1*1e-6 1*1e-6])

    subplot(3,1,3);
    plot(field_time,poynt_sat(:,2),'blue');title('Poynting Vector
(DCS)');ylabel('W/m^2');hold;
    plot(field_time,poynt_sat(:,3),'red');ylabel('W/m^2');hold;
    EBP_legend2 = legend('Poynt_Y','Poynt_Z','Location','Best');
    set(EBP_legend2,'FontSize',7);

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
    axis([0 length(field_time) -.1 0.1])
  end

  if (final_plot2 ==1),
    figure('name',['Orbit ' field_files{orbit}(11:15) ': E-Fields,B-fields (Spherical
GEO), and Poynting Vector Magnitude']);

    subplot(3,1,1);
    plot(field_time,e_geo(:,4),'blue');title(['Orbit ' field_files{orbit}(11:15) ' E-
Field (Spherical GEO)']);ylabel('V/M');hold;
```

```
        plot(field_time,e_geo(:,5),'red');plot(field_time,e_geo(:,6),'green');
        EBP_legend3 = legend('E-Field_{Rho}','E-Field_{North}','E-
Field_{East}','Location','Best');
        set(EBP_legend3,'FontSize',7);

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,1))});x
label('Geographic Latitude');grid on;
        axis([0 length(field_time) -.2 0.2])

        subplot(3,1,2);
        plot(field_time,delta_b_geo(:,4),'blue');title('Delta-B (Spherical
GEO)');ylabel('Tesla');hold;

plot(field_time,delta_b_geo(:,5),'red');plot(field_time,delta_b_geo(:,6),'green');
        EBP_legend4 = legend('Delta-B_{Rho}','Delta-B_{North}','Delta-
B_{East}','Location','Best');
        set(EBP_legend4,'FontSize',7);

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,2))});x
label('Geographic Longitude');grid on;hold;
        axis([0 length(field_time) -1*1e-6 1*1e-6])

        subplot(3,1,3);
        plot(field_time,mag_poynt1,'blue');title('Poynting Vector
Magnitude');ylabel('W/m^2');

set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;
        axis([0 length(field_time) -.1 0.1])
    end


    if (final_plot3 ==1),
        figure('name',['Orbit ' field_files{orbit}(11:15) ': Kintetic Energy Flux']);

        subplot(4,1,1);
        plot(elec_time,good_e_matrix(:,2));title(['Orbit ' field_files{orbit}(11:15) '
Electron Kinetic Energy Flux']);ylabel('W/m^2');

set(gca,'XTick',elec_x);set(gca,'XTickLabel',{num2str(elec_lat_long_alt(elec_x,1))});xlab
el('Geographic Latitude');grid on;
        axis([0 length(elec_time) -.01 0.02])

        subplot(4,1,2);
        plot(elec_time,good_e_matrix(:,2));title(['Orbit ' field_files{orbit}(11:15) '
Electron Kinetic Energy Flux']);ylabel('W/m^2');

set(gca,'XTick',elec_x);set(gca,'XTickLabel',{num2str(elec_lat_long_alt(elec_x,2))});xlab
el('Geographic Longitude');grid on;
        axis([0 length(ion_time) -.001 0.002])



        subplot(4,1,3);
        plot(ion_time,good_i_matrix(:,2));title('Ion Kinetic Energy
Flux');ylabel('W/m^2');
```

```
set(gca,'XTick',ion_x);set(gca,'XTickLabel',{num2str(ion_lat_long_alt(ion_x,1))});xlabel(
'Geographic Latitude');grid on;
        axis([0 length(elec_time) -.01 0.02])

         subplot(4,1,4);
        plot(ion_time,good_i_matrix(:,2));title('Ion Kinetic Energy
Flux');ylabel('W/m^2');

set(gca,'XTick',ion_x);set(gca,'XTickLabel',{num2str(ion_lat_long_alt(ion_x,2))});xlabel(
'Geographic Longitude');grid on;
        axis([0 length(ion_time) -.001 0.002])

%          subplot(3,1,3);
%          plot(field_time,(good_e_matrix(:,2) + good_i_matrix(:,2)));title('Poynting
Vector Magnitude');ylabel('W/m^2');
%
set(gca,'XTick',field_x);set(gca,'XTickLabel',{num2str(field_lat_long_alt(field_x,3))});x
label('Altitude');grid on;

    end

    load handel;
end
max_electron = max(good_e_matrix(:,2))
min_electron = min(good_e_matrix(:,2))
max_ion = max(good_i_matrix(:,2))
min_ion = min(good_i_matrix(:,2))
%sound(y,Fs);
```

APPENDIX B: FILE_NAMES.M

```
function [field_files, positional_files, electron_files, ion_files] = file_names

%  These files were collected online from the Berkeley website
% the files can be found at the following websites:
%http://sprg.ssl.berkeley.edu/data/misc/fast/cdf/dcf_old/ (for Field data)
%http://sprg.ssl.berkeley.edu/data/fast/k0/orb/03000/  (for Position data)
%http://sprg.ssl.berkeley.edu/data/fast/k0/ees/ (for Electron data)
%http://sprg.ssl.berkeley.edu/data/fast/k0/ies/ (for Ion data)

%Orbit/Ephemeris Files
positional_files{1} = 'fa_k0_orb_00922_v01';
positional_files{2} = 'fa_k0_orb_00923_v01';
positional_files{3} = 'fa_k0_orb_01092_v01';
positional_files{4} = 'fa_k0_orb_02217_v01';
positional_files{5} = 'fa_k0_orb_02679_v01';
positional_files{6} = 'fa_k0_orb_05704_v01';
positional_files{7} = 'fa_k0_orb_05716_v01';
positional_files{8} = 'fa_k0_orb_05714_v01';
positional_files{9} = 'fa_k0_orb_02989_v01';
positional_files{10} = 'fa_k0_orb_03237_v01';

%DC Fields Files
field_files{1} = 'fa_k0_dcf_00922_v02';
```

```
field_files{2} = 'fa_k0_dcf_00923_v02';
field_files{3} = 'fa_k0_dcf_01092_v01';
field_files{4} = 'fa_k0_dcf_02217_v01';
field_files{5} = 'fa_k0_dcf_02679_v01';
field_files{6} = 'fa_k0_dcf_05704_v02';
field_files{7} = 'fa_k0_dcf_05716_v01';
field_files{8} = 'fa_k0_dcf_05714_v01';
field_files{9} = 'fa_k0_dcf_02989_v01';
field_files{10} = 'fa_k0_dcf_03237_v01';

%Electron Files
electron_files{1} = 'fa_k0_ees_00922_v04';
electron_files{2} = 'fa_k0_ees_00923_v04';
electron_files{3} = 'fa_k0_ees_01092_v04';
electron_files{4} = 'fa_k0_ees_02217_v04';
electron_files{5} = 'fa_k0_ees_02679_v04';
electron_files{6} = 'fa_k0_ees_05704_v04';
electron_files{7} = 'fa_k0_ees_05716_v04';
electron_files{8} = 'fa_k0_ees_05714_v04';
electron_files{9} = 'fa_k0_ees_02989_v04';
electron_files{10} = 'fa_k0_ees_03237_v04';

%Ion Files
ion_files{1} = 'fa_k0_ies_00922_v04';
ion_files{2} = 'fa_k0_ies_00923_v04';
ion_files{3} = 'fa_k0_ies_01092_v04';
ion_files{4} = 'fa_k0_ies_02217_v04';
ion_files{5} = 'fa_k0_ies_02679_v04';
ion_files{6} = 'fa_k0_ies_05704_v04';
ion_files{7} = 'fa_k0_ies_05716_v04';
ion_files{8} = 'fa_k0_ies_05714_v04';
ion_files{9} = 'fa_k0_ies_02989_v04';
ion_files{10} = 'fa_k0_ies_03237_v04';
```

APPENDIX C: LOAD_DATA.M

```
function [field_matrix, pos_matrix, electron_matrix,ion_matrix] =
load_data(file1,file2,file3,file4)
% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% Sebastian.Musielak@Gmail.com

% Inputs two strings which correspond to two CDF files
% First string is of type "fa_k0_dcf_xxxxx_yyy"
% Second string should be of type "fa_k0_orb_xxxxx_yyy",
% where xxxxx is orbit number and yyy is version

% Note that these two files are the "summary data" files as described in
% numerous sources including the MQP Report, and in FAST documentation
%
% Outputs two matrices, the first matrix contains 7 columns corresponding
% to all of the Fields data: time (UNIX); e-field in X and Z directions;
% magnetic field in X,Y, and Z directions; and the angle (in degress)
% between the FAST spin plane and the geomagnetic field.
% The second matrix contains data corresponding to positional information
% of FAST. There are 7 columns: time(UNIX); GEI position X,Y,Z; GEI
% velocity (trajectory) x,y,z.


%% Code Starts

% check that both files are from the same orbit
% If files are not from the same orbit, return error
if ((~strcmp(file1(11:15),file2(11:15))) || (~strcmp(file2(11:15),file2(11:15)))
||(~strcmp(file3(11:15),file2(11:15))))
    'Input CDF Files do not come from the same orbit'
    'Please make sure that CDFs come from the same orbit'
    return
end

%% Load CDF files, find lengths
[field_data] = cdfread(file1);
data1 = field_data(:,1);
len1 = length(data1);

[pos_data] = cdfread(file2);
data2 = pos_data(:,1);
len2 =length(data2);

[electron_data] = cdfread(file3);
data3 = electron_data(:,1);
len3 = length(data3);

[ion_data] = cdfread(file4);
data4 = ion_data(:,1);
len4 =length(data4);

%% Extract only Relevant Data
%Preallocating space
field_matrix = zeros(len1,7);
pos_matrix = zeros(len2, 7);
electron_matrix = zeros(len3,2);
ion_matrix = zeros(len4, 2);


% extract fields data from Cells into matrices
for i = 1:len1
    field_matrix(i,1) = field_data{i,2}; %time
    field_matrix(i,2) = field_data{i,3}; %ex
    field_matrix(i,3) = 0;               % Set Ey to 0 bc its noise
    field_matrix(i,4) = 0;%field_data{i,4}; %ez
    field_matrix(i,5) = field_data{i,6}; %magx
    field_matrix(i,6) = field_data{i,7}; %magy
    field_matrix(i,7) = field_data{i,8}; %magz
    field_matrix(i,8) = field_data{i,10}; %spin angle
end;

% extract position and velocity Vectors into seperate Matlab Cells
for j = 1:len2
%     pos_matrix(j,1) = unixtime2mat(pos_data{j,9});
    pos_matrix(j,1) = pos_data{j,9};
    GEI_pos_vector{j} = pos_data{j,13};
    GEI_vel_vector{j} = pos_data{j,15};
end;

% extract seperate matlab cells to seperate matlab matrices
for k = 1:length(GEI_pos_vector)
    sep_pos_mat(k,1:3) = GEI_pos_vector{1,k};
    sep_vel_mat(k,1:3) = GEI_vel_vector{1,k};
end

% combine 2 seperate matlab matrices to 1 matrix
for l = 1:length(GEI_pos_vector)
    pos_matrix(l,2) = sep_pos_mat(l,1); %gx
    pos_matrix(l,3) = sep_pos_mat(l,2); %gy
    pos_matrix(l,4) = sep_pos_mat(l,3); %gz
    pos_matrix(l,5) = sep_vel_mat(l,1); %velx
    pos_matrix(l,6) = sep_vel_mat(l,2); %vely
    pos_matrix(l,7) = sep_vel_mat(l,3); %velz
end

% Electron Data matrix contains time and kintetic energy flux (W/m^2)
for m = 1:len3
    electron_matrix(m,1) = electron_data{m,2};
    electron_matrix(m,2) = electron_data{m,11}*1e-3;
end

% Ion Data matrix contains time and kintetic energy flux (W/m^2)
for n = 1:len4
    ion_matrix(n,1) = ion_data{n,2};
    ion_matrix(n,2) = ion_data{n,11}*1e-3;
end

return
```

APPENDIX D: INTERP_DATA.M

```
function [good_f_matrix,good_e_matrix, good_i_matrix,
interp_p_f_matrix,interp_p_e_matrix,interp_p_i_matrix ] = interp_data(field_matrix,
pos_matrix, electron_matrix,ion_matrix)
%pos_time,low_res_altitude,low_res_latitude,low_res_longtitude
% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% Sebastian.Musielak@Gmail.com

% Inputs are 4 matrices:
% 1) Fields data... includes electric and magnetic fields data and a time
% vectors
% 2) Positional data... includes the satellite position and satellite
% velocity in GEI coordinates and a time vector
% 3) Electron data... includes the kinetic energy data due to electrons and
% its time vector
% 4) Ion data... includes the kinetic energy data due to ions and its time
% vector
% First, the function "trims" 3 pairs of data (field/position,
% electron/postion, and ion/position).  This trimming cuts all data where
% the pairs of data do NOT have overlapping times.  This creates 6
% matrices, the "good field", "good electron", and "good ion" matrices and
% three copies of the position matrix with start and stop times at points
% in time corresponding to the other three matrices.

% Then, using the interp1 function,  the positional data for the fields
% data is interpolated to fit the same length as the fields data and be at
% the same points in time as the fields data.  This, effectively, gives a
% satellite position for every single measurment taken of the fields.  This
% is repeated for the electron and ion data.

% The function outputs 6 matrices:  the trimmed field, electron, and ion
% matrices and the three positional matrices that give satellite positions
% for each of the readings of fields, electron and ion data.




%% Code Starts

%just for 5704
%field_matrix = field_matrix(1:402,:);


%put time from matrices individual vectors
f_time = field_matrix(:,1);
p_time = pos_matrix(:,1);
e_time = electron_matrix(:,1);
i_time = ion_matrix(:,1);

% find indices of Field/Position vectors that contain overlapping times
overlap_fp_position = p_time<=f_time(end) & p_time>=f_time(1);
overlap_fp_field = f_time<=p_time(end) & f_time>=p_time(1);

% find indices of Electron/Position vectors that contain overlapping times
overlap_ep_position = p_time<=e_time(end) & p_time>=e_time(1);
overlap_ep_electron = e_time<=p_time(end) & e_time>=p_time(1);

% find indices of Ion/Position vectors that contain overlapping times
overlap_ip_position = p_time<=i_time(end) & p_time>=i_time(1);
overlap_ip_ion = i_time<=p_time(end) & i_time>=p_time(1);


% Find 3 Position Matrices that correspond to overlap for 1) position/field
% 2) postion/electron 3) position/ion
good_pos_field_matrix = pos_matrix(overlap_fp_position,:);
good_pos_electron_matrix = pos_matrix(overlap_ep_position,:);
good_pos_ion_matrix = pos_matrix(overlap_ip_position,:);

% Find the 3 other matrices that correspond to
good_f_matrix = field_matrix(overlap_fp_field,:);
good_e_matrix = electron_matrix(overlap_ep_electron,:);
good_i_matrix = ion_matrix(overlap_ip_ion,:);
```

```
%% Interpolate a new positional matrix for every data matrix

interp_p_f_matrix=interp1(good_pos_field_matrix(:,1),good_pos_field_matrix,good_f_matrix(
:,1));
interp_p_e_matrix=interp1(good_pos_electron_matrix(:,1),good_pos_electron_matrix,good_e_m
atrix(:,1));
interp_p_i_matrix=interp1(good_pos_ion_matrix(:,1),good_pos_ion_matrix,good_i_matrix(:,1)
);




% this copies the time vector from F matrix to p matrix to avoid NaN's
interp_p_f_matrix(:,1) = good_f_matrix(:,1);
interp_p_e_matrix(:,1) = good_e_matrix(:,1);
interp_p_i_matrix(:,1) = good_i_matrix(:,1);
```

APPENDIX E: UNIXTIME2MAT.M

```
function [year,month,day,hour,minute,second] = unixtime2mat(unix_time)
% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% Sebastian.Musielak@Gmail.com

% Modified from code provided by Val Schmidt
% Center for Coastal and Ocean Mapping 2007

% Inputs unix time stamps (seconds since Jan 1, 1970)UTC.  This is a
% 9-digit number before Sept 9,2001 and a 10-digit number after.

% Outputs the approximate year, month, day, hour, minute, second in a
% vector format.
%
% IMPORTANT NOTE: The function may not handle leap years or leap seconds
% appropriately.
%
%

%% Code starts here
% Declare Constants
unix_reference = [1970 1 1 0 0 0]; % Reference date of January 1,1970 (12:00 AM)
seconds_per_day = 60*60*24;

% Convert unix timestamp to date vector
unix_epoch = datenum(unix_reference);
matlab_time = unix_time./seconds_per_day + unix_epoch;
time_vector = datevec(matlab_time);
year = time_vector(1);
month = time_vector(2);
day = time_vector(3);
hour = time_vector(4);
```

```
minute = time_vector(5);
second = time_vector(6);
```

APPENDIX F: GEI2GEO.M

```
function [p_GEO2,VEL_GEO2] = gei2geo(p_GEI)

% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International

% Inputs are all of the positional data (positional time vector,
% GEI(x,y,z), and Velocity(x,y,z) and the field data time vector
% Note that the two time vectors should be different lengths, but
% start and end at the same time (different resolutions)
% unix_time_vector is an (n x 1) vector
% GEI is an (n x 3) matrix
% VEL is an (n x 3) matrix

% Outputs two seperate data sets for the conversion of GEI to GEO
% To convert, multiply the GEI coordinates by a transition matrix.  (i.e.
% GEO = T x GEI, where GEO and GEI are nx3 matrices with n 1x3 vectors
%  (x,y,z) and T is the 3x3 transition matrix).

% GEO1_xyz and VEL1_xyz is calculated using a Greenwich Hour angle
% calculated using thefollowing formula:
% theta = 100.461 + 36000.770T + 15.04107H degrees
% T = Time in Julian centuries from 1 January 2000 12:00 pm UT on
% H = the time in hours since that preceding UT midnight

% GEO2 is calculated using a Greenwich Hour angle calculated using the
% following formula:
% 279.690983 + 0.9856473354*dj + 360*fracday + 180
% dj = Julian day number (julian days since January 1, 4713 BC 12:00 pm UT
% fracdays = fractions of a day since midnight of that day

% Once a transition matrix is found, just multiply a GEI (x y z) vector by
% the matrix to find a GEO (x1 y1 z1) vector


%% Code Starts Here
% Preallocate vectors and matrices to 0
len = length(p_GEI);
t_matrix1 = zeros(3,3,0);
t_matrix2 = zeros(3,3,0);
cents = zeros(len,1);
days = zeros(len,1);
mattime = zeros(len,6);
theta1 = zeros(len,1);
theta2 = zeros(len,1);

%% Find number of julian centures and julian days since (Jan 1, 2000 @ 12PM)
for i = 1:len
```

```
    [mattime(i,1),mattime(i,2),mattime(i,3),mattime(i,4),mattime(i,5),mattime(i,6) ] =
unixtime2mat(p_GEI(i,1));
    [cents(i)] = since_j2000(mattime(i,1),mattime(i,2),mattime(i,3),0,0,0);
end

% find fraction of hours since midnight that day
for k  =1:len
fracminute(k) = mattime(k,6)/60;
newmin(k) = fracminute(k) + mattime(k,5);
frachour(k) = newmin(k)/60;
newhour(k) = mattime(k,4) + frachour(k);
fracday(k) = newhour(k)/24;
newday(k) = mattime(k,3) + fracday(k);
dj(k) = (365*(mattime(k,1)-1900) +  (mattime(k,1)-1901)/4 +  mattime(k,3) + fracday(k)) -
0.5;
end

%% Find Greenwhich Mean Sidereal Time (Greenwhich hour angle) in degrees
% Use angle to find transition matrices
% Transition Matrices have the form

% |  cos(theta)   sin(theta)  0 |
% | -sin(theta)   cos(theta)  0 |
% |     0             0       1 |

% Find Transition Matrices
for j = 1:len
    theta1(j) = 100.461 + (36000.770*cents(j)) + (15.04107 * newhour(j)); % from Mike
Hapgood, Rutherford labs,UK
    theta2(j) = mod((279.690983 + 0.9856473354*dj(j) + 360*fracday(j) + 180),360); %from
FAST (in degress)
% Find Transition Matrix using Theta1
    t_matrix1(1,1,j) = cosd(theta1(j));
    t_matrix1(1,2,j) = sind(theta1(j));
    t_matrix1(1,3,j) = 0;
    t_matrix1(2,1,j) = -sind(theta1(j));
    t_matrix1(2,2,j) = cosd(theta1(j));
    t_matrix1(2,3,j) = 0;
    t_matrix1(3,1,j) = 0;
    t_matrix1(3,2,j) = 0;
    t_matrix1(3,3,j) = 1;
% Find Second matrix Using Theta 2
    t_matrix2(1,1,j) = cosd(theta2(j));
    t_matrix2(1,2,j) = sind(theta2(j));
    t_matrix2(1,3,j) = 0;
    t_matrix2(2,1,j) = -sind(theta2(j));
    t_matrix2(2,2,j) = cosd(theta2(j));
    t_matrix2(2,3,j) = 0;
    t_matrix2(3,1,j) = 0;
    t_matrix2(3,2,j) = 0;
    t_matrix2(3,3,j) = 1;
end

%% Multiply transition matrix by GEI Vectors to get GEO Vector
 for x = 1:len

     for y = 1:3
    % Insert Time Vectors
    p_GEO1(x,1) = p_GEI(x,1);
    p_GEO2(x,1) = p_GEI(x,1);
    VEL_GEO1(x,1) = p_GEI(x,1);
    VEL_GEO2(x,1) = p_GEI(x,1);
    %Insert GEO data
    p_GEO1(x,y+1) = (t_matrix1(y,1,x)*p_GEI(x,2)) + (t_matrix1(y,2,x)*p_GEI(x,3)) +
(t_matrix1(y,3,x)*p_GEI(x,4));
    p_GEO2(x,y+1) = (t_matrix2(y,1,x)*p_GEI(x,2)) + (t_matrix2(y,2,x)*p_GEI(x,3)) +
(t_matrix2(y,3,x)*p_GEI(x,4));
    VEL_GEO1(x,y+1) = (t_matrix1(y,1,x)*p_GEI(x,5)) + (t_matrix1(y,2,x)*p_GEI(x,6)) +
(t_matrix1(y,3,x)*p_GEI(x,7));
    VEL_GEO2(x,y+1) = (t_matrix2(y,1,x)*p_GEI(x,5)) + (t_matrix2(y,2,x)*p_GEI(x,6)) +
(t_matrix2(y,3,x)*p_GEI(x,7));
     end
end
```

APPENDIX G: SAT2GEO.M

```
function
[delta_b_geo,e_geo,u_spin,u_traj,u_spB,u_spBp,e_other,u_other,u_west,crossing,incomingnor
th] = sat2geo(good_f_matrix,field_pos_geo,field_vel,field_igrf)

% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% Sebastian.Musielak@Gmail.com

% This program is designed to convert delta-B vectors from satellite
% reference frame coordinates to Geocentric Geographic Coordinates (GEO).
% This requires the positional information (in GEO)on the satellite and
% the model geomagnetic field (IGFR model) data.

% Inputs are 4 matrices:
% 1)"good_f_matrix" contains 6 columns: Time,Ex,Ez,d-Bx,d-By,d-Bz all in Satellite
%   reference frame
% 2)"field_pos_geo" contains 4 satellite position columns: time,GEOx, GEOy,GEOz
% 3)"field_vel" contains 4 satellite trajectory columns: time, VELx,VELy,VELz
% 4)"field_igrf" contains 4 model Magnetic field columns: time, IGRFx,
%    IGRFy,IGRFz

% Outputs are:


%% Code Starts here
%% Begin by Normalizing all available data (DONE)
len = length(good_f_matrix);

% Satellite Trajectory
u_traj = normr(field_vel(:,2:4));

% Local Vertical (Line from Center of earth to satellite)
u_vert = normr(field_pos_geo(:,2:4));

% Find spherical GEO projections of position: rho, theta,phi
```

```
u_rho = u_vert;
for i = 1:len
 u_phi(i,:) = normr(cross([0 0 1],u_rho(i,:)));
 u_theta(i,:) = normr(cross(field_pos_geo(i,2:4),u_phi(i,:)));
end



% Find Spin Axis Vector
u_spin = normr(cross(u_traj,u_vert));

% Find X,Y,X Vector (Satellite Reference Frame)
x_vect = cross(u_spin,u_traj);
y_vect = -u_traj;
z_vect = u_spin;
%% Find the Spin Plane Projections of Magnetic Field

% By in CDF along space spin axis (WEST)
% Bx in CDF given by Y x B Model (equatorward)
% Bz in CDF parallel to model mag field (points north)

% spin-plane projection of the geomagnetic field (model-B)
for i = 1:len
    %find x-projection
    spB_x(i,:) = (dot(field_igrf(i,2:4),x_vect(i,1:3))*x_vect(i,1:3));
    %find y-projection
    spB_y(i,:) = (dot(field_igrf(i,2:4),y_vect(i,1:3))*y_vect(i,1:3));
    %find x and y projections totaled and normalize
    u_spB(i,:) = normr(spB_x(i,:) + spB_y(i,:));
end



new_spin = normr(u_spin(:,1:2)); % spin axis in x,y plane(equatorial plane)
new_geo = normr(field_pos_geo(:,2:3)); % geo coordinates in x,y plane



% Find crossing angle between axis and local vertical
% When crossing angle is zero, spin axis crosses pole
for x = 1:len
crossing(x) = acosd((dot(new_spin(x,1),new_geo(x,1)) + dot(new_spin(x,2),new_geo(x,2))));
end
crossing(length(crossing)+1) = crossing(length(crossing));

% Pre-allocate incoming north vector
incomingnorth = zeros(1,len+1);

% When incomingnorth =1, satellite is incoming north
% When incomingnorth =0, satellite is outgoing north
% When incomingnorth =2, other
for y = 1:len
    if (abs(crossing(y)) < 1),
        incomingnorth(y) = 2;
    else if (abs(crossing(y+1)) - abs(crossing(y)) < 0),
            incomingnorth(y) = 1;
        else if (abs(crossing(y+1)) - abs(crossing(y)) > 0),
                incomingnorth(y) = 0;
            else incomingnorth(y) = 2;
            end
        end;
    end;
end
incomingnorth(length(crossing)) = incomingnorth(length(crossing)-1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% find vector perpendicular to spin plane projection of model-B and
% spin-axis vector. u_SpBp changes sign wether it's incoming north or
% outgoing north
for x = 1:len
    if (incomingnorth(x) == 1),
    u_spBp(x,:) = normr(cross(u_spB(x,:), u_spin(x,:)));  % ingoing north/outcoming south
    u_west(x,:) = -u_spin(x,:);
    else if (incomingnorth(x) == 0),
            u_spBp(x,:) = normr(cross(u_spin(x,:), u_spB(x,:))); % outgoing
north/incoming south
            u_west(x,:) = u_spin(x,:);
        else u_spBp(x,:) = [NaN NaN NaN];u_west(x,:) = [NaN NaN NaN];
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% u_spBp(:,3) = -abs(u_spBp(:,3));

%% Find Delta-B in Cartesian GEO coordinate system

% First Calculate Cartesian GEO, the Spherical GEO
for j = 1:len
delta_b_cart(j,:) = (good_f_matrix(j,7)*u_spB(j,:))+ (good_f_matrix(j,5)*u_spBp(j,:)) +
(good_f_matrix(j,6)*u_west(j,:));  %(+uspin for outgoing,-uspin for incoming)
%delta_b_sper(j,:) = (dot(delta_b_cart(j,:),u_rho(j,:))*u_rho(j,:)) +
(dot(delta_b_cart(j,:),u_theta(j,:))*u_theta(j,:)) +
(dot(delta_b_cart(j,:),u_phi(j,:))*u_phi(j,:));
delta_b_sper(j,:) = [u_rho(j,:); u_theta(j,:);u_phi(j,:)]* [
delta_b_cart(j,1);delta_b_cart(j,2);delta_b_cart(j,3)]  ;

% b_rho(j) = u_rho(j,:)*[delta_b_cart(j,1) ;delta_b_cart(j,2) ;delta_b_cart(j,3)];
% b_theta(j) = u_theta(j,:)*[delta_b_cart(j,1) ;delta_b_cart(j,2) ;delta_b_cart(j,3)];
% b_phi(j) = u_phi(j,:)*[delta_b_cart(j,1) ;delta_b_cart(j,2) ;delta_b_cart(j,3)];
end



% Combine Cartesian and Spherical GEO into 1 matrix
delta_b_geo = [ delta_b_cart delta_b_sper];

% output in Tesla from nanoTesla
delta_b_geo = delta_b_geo*1e-9;




%% Find Electric Field in GEO Coordinate system

%
u_other = cross(u_spBp,normr(field_igrf(:,2:4)));
```

```
% Find Cartesian GEO
for k = 1:len
%e_other(k,:) = good_f_matrix(k,3)/(dot(u_other(k,:),u_spB(k,:)));
e_other(k,:) = 0;
e_cart(k,:) = (e_other(k)*u_other(k,:)) + (good_f_matrix(k,2)*u_spBp(k,:));
%e_sper(k,:) =
((e_cart(k,1))*u_rho(k,:))+((e_cart(k,2))*u_theta(k,:))+((e_cart(k,3))*u_phi(k,:));
e_sper(k,:) = [u_rho(k,:); u_theta(k,:);u_phi(k,:)]* [
e_cart(k,1);e_cart(k,2);e_cart(k,3)]  ;
end

%e_rho =

e_geo = [ e_cart e_sper];



%output in volts from milivolts
e_geo = e_geo*1e-3;
```

APPENDIX H: GET_IGRF.M

```
function [igrf_geo] = get_igrf(p_geo)
% Sebastian Musielak
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% Sebastian.Musielak@Gmail.com

% Modified from and supplemented by Code suite: "Magnetic Field Model"
% provided Carlos Roithmayr
% NASA Langley Research Center
% Spacecraft and Sensors Branch (CBC)
% 757 864 6778
% c.m.roithmayr@larc.nasa.gov

% and

% N. Papitashvili, NASA/GSFC/NSSDC, code 633/STX,
% Greenbelt, Maryland 20771, U.S.A.,
% DECNET:  NCF::NATASHA
% INTERNET: natasha@nssdca.gsfc.nasa.gov


% The input to this function is a matrix containing the positional
% data of the satellite. The matrix must be an (n x 4) matrix with the
% first column being a vector of UNIX times, and the last three columns
% being the satellite position in GEO coordinates.  This function calls 4
% other functions that are a part of the "Magnetic Field Model" suite.

%% Code starts here
global R_mean

R_mean = 6371.2;              % Mean radius for International Geomagnetic
```

```
                                  % Reference Field (6371.2 km)

[G,H] = IGRF95;                   % IGRF coefficients for 1995

nmax = 10;                        % max degree of geopotential
mmax = 10;                        % max order  of geopotential

Kschmidt = schmidt(nmax,mmax);

for k = 1:length(p_geo)

igrf_geo(k,1) = p_geo(k,1); % copy the UNIX times
[A,ctilde,stilde] = recursion(p_geo(k,2:4),nmax,mmax);
igrf_geo(k,2:4) = bfield(p_geo(k,2:4),nmax,mmax,Kschmidt,A,ctilde,stilde,G,H);


end
```

APPENDIX I: BFIELD.M

```
function bepe = bfield(repe,nmax,mmax,K,A,ctilde,stilde,G,H)
global R_mean

%+======================================================================+
%
%     Programmers:  Carlos Roithmayr                            Feb 1997
%
%           NASA Langley Research Center
%           Spacecraft and Sensors Branch (CBC)
%           757 864 6778
%           c.m.roithmayr@larc.nasa.gov
%
%+----------------------------------------------------------------------+
%
%     Purpose:
%
%     Compute magnetic field exerted at a point P.
%
%+----------------------------------------------------------------------+
%
%     Argument definitions:
%
%     repe    (km)      Position vector from Earth's center, E*, to a
%                       point, P, expressed in a basis fixed in the
%                       Earth (ECF): 1 and 2 lie in equatorial plane
%                       with 1 in the plane containing the prime
%                       meridian, in the direction of the north pole.
%
%     nmax              Maximum degree of contributing spherical harmonics
%
%     mmax              Maximum order of contributing spherical harmonics
%
%     K                 coefficients that relate Schmidt functions to
%                       associated Legendre functions.
%
```

```
%     A                 Derived Legendre polynomials
%
%     ctilde            See pp. 4--9 of Ref. [1]
%
%     stilde            See pp. 4--9 of Ref. [1]
%
%     G, H     Tesla    Schmidt-normalized Gauss coefficients
%
%     R_mean   km       Mean radius for International Geomagnetic
%                       Reference Field (6371.2 km)
%
%     bepe     Tesla    Magnetic field at a point, P, expressed in ECF
%                       basis
%
%+----------------------------------------------------------------------+
%
%     References:
%
%     1. Mueller, A. C., "A Fast Recursive Algorithm for Calculating
%        the Forces Due to the Geopotential", NASA JSC Internal Note
%        No. 75-FM-42, June 9, 1975.
%
%     2. Roithmayr, C., "Contributions of Spherical Harmonics to
%        Magnetic and Gravitational Fields", EG2-96-02, NASA Johnson
%        Space Center, Jan. 23, 1996.
%
%+----------------------------------------------------------------------+
%
%     Conversion factors:
%
%       1 Tesla = 1 Weber/(meter-meter) = 1 Newton/(Ampere-meter)
%               = 1e+4 Gauss   =  1e+9 gamma
%
%+======================================================================+

% The number 1 is added to degree and order since MATLAB can't have an array
% index of 0.

e1=[1 0 0];
e2=[0 1 0];
e3=[0 0 1];

rmag = sqrt(repe*repe'); % Magnitude
rhat = repe/rmag;        % Unit Vector

u = rhat(3);                    % sin of latitude

bepe = [0 0 0];

% Seed for recursion formulae

scalar = R_mean*R_mean/(rmag*rmag);

for n = 1:nmax

% Recursion formula
  scalar = scalar*R_mean/rmag;
  i=n+1;

  for m = 0:n

    j=m+1;

    if m <= mmax
      ttilde(i,j) = G(i,j)*ctilde(j) + H(i,j)*stilde(j);

%     ECF 3 component {Eq. (2), Ref. [2]}
      b3(i,j) =  -ttilde(i,j)*A(i,j+1);

%     rhat component {Eq. (2), Ref. [2]}
      br(i,j) = ttilde(i,j)*(u*A(i,j+1) + (n+m+1)*A(i,j));

%     Contribution of zonal harmonic of degree n to magnetic
%     field.  {Eq. (2), Ref. [2]}

      bepe = bepe + scalar*K(i,j)*(b3(i,j)*e3 + br(i,j)*rhat);
    end

    if ((m > 0) & (m <= mmax))

%     ECF 1 component {Eq. (2), Ref. [2]}
      b1(i,j) = -m*A(i,j)*(G(i,j)*ctilde(j-1) + H(i,j)*stilde(j-1));

%     ECF 2 component {Eq. (2), Ref. [2]}
      b2(i,j) = -m*A(i,j)*(H(i,j)*ctilde(j-1) - G(i,j)*stilde(j-1));

%     Contribution of tesseral harmonic of degree n and order m to
%     magnetic field.  {Eq. (2), Ref. [2]}
      bepe = bepe + scalar*K(i,j)*(b1(i,j)*e1 + b2(i,j)*e2);
    end

  end
end
```

APPENDIX J: IGRF95.M

```
function [G,H] = IGRF95

% MATLAB routine to load Schmidt-normalized coefficients
% retrieved from ftp://nssdc.gsfc.nasa.gov/pub/models/igrf/

% igrf95.dat                  1 Kb    Mon Nov 13 00:00:00 1995

% ? C.E. Barton, Revision of International Geomagnetic Reference
% Field Released, EOS Transactions 77, #16, April 16, 1996.


% The coefficients are from the 1995 International Geomagnetic Reference Field
% Carlos Roithmayr, Jan. 22, 1997.

%+++++++++++++++++++++++++++++++++++++++++

% The number 1 is added to ALL subscripts since MATLAB can't have an array
% index of 0.  Units of Tesla

G(2,1) = -29682e-9;
G(2,2) = -1789e-9; H(2,2) =  5318e-9;
G(3,1) = -2197e-9; H(3,1) =     0.0;
G(3,2) =  3074e-9; H(3,2) =   -2356e-9;
G(3,3) =  1685e-9; H(3,3) =  -425e-9;
G(4,1) =  1329e-9; H(4,1) =     0.0;
G(4,2) = -2268e-9; H(4,2) =  -263e-9;
G(4,3) =  1249e-9; H(4,3) =   302e-9;
G(4,4) =   769e-9; H(4,4) =  -406e-9;
G(5,1) =   941e-9; H(5,1) =      .0;
G(5,2) =   782e-9; H(5,2) =   262e-9;
G(5,3) =   291e-9; H(5,3) =  -232e-9;
G(5,4) =  -421e-9; H(5,4) =    98e-9;
G(5,5) =   116e-9; H(5,5) =  -301e-9;
G(6,1) =  -210e-9; H(6,1) =      .0;
G(6,2) =   352e-9; H(6,2) =    44e-9;
G(6,3) =   237e-9; H(6,3) =   157e-9;
G(6,4) =  -122e-9; H(6,4) =  -152e-9;
G(6,5) =  -167e-9; H(6,5) =   -64e-9;
G(6,6) =   -26e-9; H(6,6) =    99e-9;
G(7,1) =    66e-9; H(7,1) =      .0;
G(7,2) =    64e-9; H(7,2) =   -16e-9;
G(7,3) =    65e-9; H(7,3) =    77e-9;
G(7,4) =  -172e-9; H(7,4) =    67e-9;
G(7,5) =     2e-9; H(7,5) =   -57e-9;
G(7,6) =    17e-9; H(7,6) =     4e-9;
G(7,7) =   -94e-9; H(7,7) =    28e-9;
G(8,1) =    78e-9; H(8,1) =     -.0;
G(8,2) =   -67e-9; H(8,2) =   -77e-9;
G(8,3) =     1e-9; H(8,3) =   -25e-9;
G(8,4) =    29e-9; H(8,4) =     3e-9;
G(8,5) =     4e-9; H(8,5) =    22e-9;
G(8,6) =     8e-9; H(8,6) =    16e-9;
G(8,7) =    10e-9; H(8,7) =   -23e-9;
G(8,8) =    -2e-9; H(8,8) =    -3e-9;
G(9,1) =    24e-9; H(9,1) =      .0;
G(9,2) =     4e-9; H(9,2) =    12e-9;
G(9,3) =    -1e-9; H(9,3) =   -20e-9;
G(9,4) =    -9e-9; H(9,4) =     7e-9;
G(9,5) =   -14e-9; H(9,5) =   -21e-9;
G(9,6) =     4e-9; H(9,6) =    12e-9;
G(9,7) =     5e-9; H(9,7) =    10e-9;
G(9,8) =     0e-9; H(9,8) =   -17e-9;
G(9,9) =    -7e-9; H(9,9) =   -10e-9;
G(10,1) =     4e-9; H(10,1) =      .0;
G(10,2) =     9e-9; H(10,2) =   -19e-9;
G(10,3) =     1e-9; H(10,3) =    15e-9;
G(10,4) =   -12e-9; H(10,4) =    11e-9;
G(10,5) =     9e-9; H(10,5) =    -7e-9;
G(10,6) =    -4e-9; H(10,6) =    -7e-9;
G(10,7) =    -2e-9; H(10,7) =     9e-9;
```

```
G(10,8) =     7e-9; H(10,8) =     7e-9;
G(10,9) =     0e-9; H(10,9) =    -8e-9;
G(10,10) =    -6e-9; H(10,10) =     1e-9;
G(11,1) =    -3e-9; H(11,1) =      .0;
G(11,2) =      -4e-9; H(11,2) =     2e-9;
G(11,3) =       2e-9; H(11,3) =     1e-9;
G(11,4) =      -5e-9; H(11,4) =     3e-9;
G(11,5) =      -2e-9; H(11,5) =     6e-9;
G(11,6) =       4e-9; H(11,6) =    -4e-9;
G(11,7) =       3e-9; H(11,7) =     0.;
G(11,8) =       1e-9; H(11,8) =    -2e-9;
G(11,9) =       3e-9; H(11,9) =     3e-9;
G(11,10) =      3e-9; H(11,10) =    -1e-9;
G(11,11) =      0e-9; H(11,11) =    -6e-9;
```

APPENDIX K: SCHMIDT.M

```
function K = schmidt(nmax,mmax)

%+======================================================================+
%
%     Programmers:  Carlos Roithmayr                    Feb 1997
%
%           NASA Langley Research Center
%           Spacecraft and Sensors Branch (CBC)
%           757 864 6778
%           c.m.roithmayr@larc.nasa.gov
%
%+----------------------------------------------------------------------+
%
%     Purpose:
%
%     Compute coefficients that relate Schmidt functions to associated
%     Legendre functions.
%
%+----------------------------------------------------------------------+
%
%     Argument definitions:
%
%     nmax              Maximum degree of contributing spherical harmonics
%
%     mmax              Maximum order of contributing spherical harmonics
%
%     K                 coefficients that relate Schmidt functions to
%                       associated Legendre functions (Ref. [1]).
%
%+----------------------------------------------------------------------+
%
%     References:
%
%     1. Haymes, R. C., Introduction to Space Science, Wiley, New
%        York, 1971.
%
%     2. Roithmayr, C., "Contributions of Spherical Harmonics to
%        Magnetic and Gravitational Fields", EG2-96-02, NASA Johnson
%        Space Center, Jan. 23, 1996.
```

```
%
%+======================================================================+

% The number 1 is added to degree and order since MATLAB can't have an array
% index of 0.


% Seed for recursion formulae
K(2,2) = 1;

% Recursion formulae

for n = 1:nmax
    i=n+1;

  for m = 0:n
     j=m+1;

    if m == 0
    % Eq. (3), Ref. [2]
      K(i,j) = 1;

    elseif ((m >= 1) & (n >= (m+1)))
    % Eq. (4), Ref. [2]
      K(i,j) = sqrt((n-m)/(n+m))*K(i-1,j);

    elseif ((m >= 2) & (n >= m))
    % Eq. (5), Ref. [2]
      K(i,j) = K(i,j-1)/sqrt((n+m)*(n-m+1));
    end

  end
end
```

APPENDIX L: RECURSION.M

```
function [A,ctilde,stilde] = recursion(repe,nmax,mmax)

%+======================================================================+
%
%     Programmers:  Carlos Roithmayr                            Dec 1995
%
%           NASA Langley Research Center
%           Spacecraft and Sensors Branch (CBC)
%           757 864 6778
%           c.m.roithmayr@larc.nasa.gov
%
%+----------------------------------------------------------------------+
%
%     Purpose:
%
%     Recursive calculations of derived Legendre polynomials and other
%     quantities needed for gravitational and magnetic fields.
%
```

```
%+----------------------------------------------------------------------+
%
%     Argument definitions:
%
%     repe    (m?)      Position vector from Earth's center, E*, to a
%                       point, P, expressed in a basis fixed in the
%                       Earth (ECF): 1 and 2 lie in equatorial plane
%                       with 1 in the plane containing the prime meridian,
%                       3 in the direction of the north pole.
%                       The units of length are not terribly important,
%                       since repe is made into a unit vector.
%
%     nmax              Maximum degree of derived Legendre polynomials
%
%     mmax              Maximum order of derived Legendre polynomials
%
%     A                 Derived Legendre polynomials
%
%     ctilde            See pp. 4--9 of Ref. [1]
%
%     stilde            See pp. 4--9 of Ref. [1]
%
%+----------------------------------------------------------------------+
%
%     References:
%
%     1. Mueller, A. C., "A Fast Recursive Algorithm for Calculating
%        the Forces Due to the Geopotential", NASA JSC Internal Note
%        No. 75-FM-42, June 9, 1975.
%
%     2. Lundberg, J. B., and Schutz, B. E., "Recursion Formulas of
%        Legendre Functions for Use with Nonsingular Geopotential
%        Models", Journal of Guidance, Control, and Dynamics, Vol. 11,
%        Jan--Feb 1988, pp. 32--38.
%
%+======================================================================+

% The number 1 is added to degree and order since MATLAB can't have an
% array index of 0.

clear A;
A=zeros(nmax+3,nmax+3);         % A(n,m) = 0, for m > n

R_m = sqrt(repe*repe');
rhat = repe/R_m;

u = rhat(3);                    % sin of latitude

A(1,1)=1;                       % "derived" Legendre polynomials
A(2,1)=u;
A(2,2)=1;
     clear ctilde
     clear stilde
ctilde(1) = 1; ctilde(2) = rhat(1);
stilde(1) = 0; stilde(2) = rhat(2);

for n = 2:nmax
  i=n+1;
```

```
% Calculate derived Legendre polynomials and "tilde" letters
% required for gravitational and magnetic fields.

% Eq. (4a), Ref. [2]
  A(i,i) = prod(1:2:(2*n - 1));

% Eq. (4b), Ref. [2]
  A(i,(i-1))= u*A(i,i);

  if n <= mmax
%   p. 9,     Ref. [1]
    ctilde(i)  = ctilde(2) * ctilde(i-1) - stilde(2) * stilde(i-1);
    stilde(i)  = stilde(2) * ctilde(i-1) + ctilde(2) * stilde(i-1);
  end

  for m = 0:n
    j=m+1;


    if (m < (n-1)) & (m <= (mmax+1))
%     Eq. I, Table 1, Ref. [2]
      A(i,j)=((2*n - 1)*u*A((i-1),j) - (n+m-1)*A((i-2),j))/(n-m);
    end

  end
end
```

APPENDIX M: SATALT.M

```
function [lat_long_alt] = satalt(geo)
% Nicole Cahill
% Worcester Polytechnic Institute
% Silicon Valley MQP 2010
% at SRI International
% NCahill320@Gmail.com

%   Inputs X,Y,Z coordinates in GEO, converts to Geodetic then uses the
%   Geodetic latitude to find the Earth's radius at that spot.

%% Code Starts here

len=length(geo);
a=geo(:,2); %X-coordinate in GEO
b=geo(:,3); %Y-coordinate in GEO
c=geo(:,4); %Z coordinate in GEO
%ellip=[6378137.0, 0.0818191908426 22]; %ellipsoid= semi major axis of the earth,
eccentricity of the earth
ellip=[6378137.0, 0.016710219]; %ellipsoid= semi major axis of the earth, eccentricity of
the earth
[phi, lambda, h]=ecef2geodetic(a,b,c,ellip); %converts GEO X, Y, Z, to Geodetic lat,
long, height
alpha=6378.1370;    %Earth's equatorial radius
beta=6356.7523; %Earth's Polar radius
```

```
alpha2=alpha^2;
beta2=beta^2;
radtodeg = 180/pi;

guy=alpha2*cos(phi);
gal=beta2*sin(phi);
for i=1:len
    guy2(i,1)=guy(i,1)^2;
    gal2(i,1)=gal(i,1)^2;
end



thing=alpha*cos(phi);
thang=beta*sin(phi);
for i=1:len
    thing2(i,1)=thing(i,1)^2;
    thang2(i,1)=thang(i,1)^2;
end


almost=(guy2+gal2)./(thing2+thang2);
Erad=sqrt(almost);

for i=1:len
    a2(i,1)=a(i,1)^2;
    b2(i,1)=b(i,1)^2;
    c2(i,1)=c(i,1)^2;
end


mag=sqrt(a2+b2+c2);

alt=mag-Erad;

lat_long_alt(:,1) = radtodeg*phi;
lat_long_alt(:,2) = radtodeg*lambda;
lat_long_alt(:,3) = alt;

end
```

APPENDIX N: SINCE_J2000.M

```
function [jcent_since_j2k, newhour] = since_j2000(year,month,day,hour,minute,second)

%
% To find the number of days from J2000.0 for 2310 hrs UT on
% 1998 August 10th, do the following;
%
% 1. divide the number of minutes by 60 to obtain the decimal
% fraction of an hour, here 10/60 = 0.1666667
%
% 2. add this to the hours, then divide the total by 24 to obtain
% the decimal fraction of the day, here 23.166667/24 = 0.9652778
% This is the first number used below
%
% 3. find from table A the number of days to the beginning of
% August from the start of the year, here 212 days
%
% 4. write down the day number within the month, here 10 above
%
% 5. find from table B the days since J2000.0 to the beginning of
% the year, here -731.5
%
% 6. add these four numbers.
%
% For the date above;
%
%    0.9652778 + 212 + 10 - 731.5 = -508.53472 days from J2000.0

%
% Table A                 |  Table B
% Days to beginning of    |  Days since J2000 to
% month                   |  beginning of each year (* is leap year)
%                         |
% Month   Normal   Leap   |  Year   Days    |  Year   Days
%         year     year   | *1996   -1462.5 |
%                         |  1997   -1096.5 |
% Jan       0        0    |  1998   -731.5  |   2010  3651.5
% Feb      31       31    |  1999   -366.5  |   2011  4016.5
% Mar      59       60    | *2000     -1.5  |   2012  4381.5
% Apr      90       91    |  2001    364.5  |   2013  4747.5
% May     120      121    |  2002    729.5  |   2014  5112.5
% Jun     151      152    |  2003   1094.5  |   2015  5477.5
% Jul     181      182    | *2004   1459.5  |   2016  5842.5
% Aug     212      213    |  2005   1825.5  |   2017  6208.5
% Sep     243      244    |  2006   2190.5  |   2018  6573.5
% Oct     273      274    |  2007   2555.5  |   2019  6938.5
% Nov     304      305    | *2008   2920.5  |   2020  7303.5
% Dec     334      335    |  2009   3286.5  |   2021  7669.5


fracminute = second/60;

newmin = fracminute + minute;

frachour = newmin/60;

newhour = hour + frachour;

fracday = newhour/24;


switch year
  case 1996
    j2k_to_year_days = -1462.5; %change
  case 1997
    j2k_to_year_days = -1096.5; %change
  case 1998
    j2k_to_year_days = -731.5;
  case 1999
```

```
    j2k_to_year_days = -366.5;
  case 2000
    j2k_to_year_days = -1.5;
  case 2001
    j2k_to_year_days = 364.5;
  case 2002
    j2k_to_year_days = 729.5;
  case 2003
    j2k_to_year_days = 1094.5;
  case 2004
    j2k_to_year_days = 1459.5;
end

switch month
  case 1
    yeartomonth_days = 0;
  case 2
    yeartomonth_days = 31;
  case 3
      if ((year == 2000) || (year == 2004) || (year == 1996)), yeartomonth_days = 60;
      else yeartomonth_days = 59;end;
  case 4
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 91;
      else yeartomonth_days = 90;end;
  case 5
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 121;
      else yeartomonth_days = 120;end;
  case 6
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 152;
      else yeartomonth_days = 151;end;
  case 7
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 182;
      else yeartomonth_days = 181;end;
  case 8
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 213;
      else yeartomonth_days = 212;end;
  case 9
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 244;
      else yeartomonth_days = 243;end;
  case 10
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 274;
      else yeartomonth_days = 273;end;
  case 11
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 305;
      else yeartomonth_days = 304;end;
  case 12
      if ((year == 2000) || (year == 2004)|| (year == 1996)), yeartomonth_days = 335;
      else yeartomonth_days = 334;end;
end

days_since_j2k = fracday + day + yeartomonth_days + j2k_to_year_days + .5;
jcent_since_j2k = days_since_j2k/36525;
```