Advanced Analysis and Visualization of Space Weather Phenomena

by

Joshua J. Murphy

B.S., University of Maine, 2008

M.S., University of Colorado Boulder, 2013

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2017
This thesis entitled:
Advanced Analysis and Visualization of Space Weather Phenomena
written by Joshua J. Murphy
has been approved for the Department of Computer Science

Prof. Xiao-Chuan Cai

Dr. Scot Elkington

Date ________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.
As the world becomes more technologically reliant, the more susceptible society as a whole is to adverse interactions with the sun. This “space weather” can produce significant effects on modern technology, from interrupting satellite service, to causing serious damage to Earth-side power grids. These concerns have, over the past several years, prompted an out-welling of research in an attempt to understand the processes governing, and to provide a means of forecasting, space weather events. The research presented in this thesis couples to current work aimed at understanding Coronal Mass Ejections (CMEs) and their influence on the evolution of Earth’s magnetic field and associated Van Allen radiation belts. To aid in the analysis of how these solar wind transients affect Earth’s magnetic field, a system named Geospace/Heliosphere Observation & Simulation Tool-kit (GHOSTkit), along with its python analysis tools, GHOSTpy, has been devised to calculate the adiabatic invariants of trapped particle motion within Earth’s magnetic field. These invariants aid scientists in ordering observations of the radiation belts, providing a more natural presentation of data, but can be computationally expensive to calculate. The GHOSTpy system, in the phase presented here, is aimed at providing invariant calculations based on LFM magnetic field simulation data. This research first examines an ideal dipole application to gain understanding on system performance. Following this, the challenges of applying the algorithms to gridded LFM MHD data is examined. Performance profiles are then presented, followed by a real-world application of the system.
Dedication

To my wife Amanda, and sons Jarod and Kaladin. Thank you for being so patient.
Acknowledgements

First and foremost, I would like to acknowledge my family. My wife, Amanda, and my sons Jarod and Kaladin. They have been an incredible support while working on this dissertation. I would also like to acknowledge my advisors: Dr. Scot Elkington, and Dr. Xiao-Chuan Cai from LASP and Computer Science respectively. They were both tremendous with their support.

The DoD Science, Mathematics, and Research for Transformation (SMART) scholarship for service program made this thesis possible through funding and job prospects. I would like to specifically thank Keisha Williams and Phillippe Reed at the US Navy Space and Naval Warfare Systems Center (SPAWAR) Atlantic in Charleston, South Carolina for allowing me extra time during internships to work on this dissertation. That extra time was invaluable; thank you.
Contents

Chapter

1 Space Weather

1.1 A Brief History of Space Weather ........................................... 1
1.2 Connecting the Sun and the Earth ............................................ 3
1.3 Radiation Belts ........................................................................ 5
  1.3.1 Trapped Particle Motion ...................................................... 5
  1.3.2 The Adiabatic Invariants ...................................................... 7
1.4 Analysis and Visualization of Particles in the Magnetosphere ............ 10
1.5 Numerical Models of the Earth’s Magnetosphere ............................ 10
  1.5.1 Dipole Geomagnetic Field Model ........................................ 11
  1.5.2 Empirical Geomagnetic Field Models ................................... 12
  1.5.3 Magnetohydrodynamic (MHD) Geomagnetic Field Models ........ 14
1.6 Thesis Format ........................................................................ 16

2 Observational Uncertainties and CME Transients ............................. 17

2.1 The Model Environment ........................................................... 20
  2.1.1 The WSA Model ................................................................ 20
  2.1.2 The Cone Model ................................................................ 21
  2.1.3 The Enlil Model ................................................................ 23
2.2 Mitigation of Uncertainties with Ensemble Modeling ......................... 24
2.3 Analyzing Effects of Uncertainty with CME Impact Domain Analysis

2.3.1 The 06 October 2013 CME Event

2.3.2 Defining an Impact

2.3.3 Finding and Populating the Impact Domain

2.3.4 Solar Wind Impact Domains and Quartiles

2.4 Conclusions

3 Adiabatic Invariant Methods and Error Analysis in a Dipole

3.1 Invariant Algorithms

3.1.1 Computing $\mu$

3.1.2 Computing $K$

3.1.3 Computing $L^*$

3.2 Error Response in a Dipole Field

3.2.1 Error Response due to Field Line Footprint Inaccuracies

4 Working with LFM Model Gridded Output

4.1 The LFM Grid

4.2 LFM Grid Dipole Response

4.3 Algorithm Adjustments for LFM Data

4.3.1 The New $K/B$ Model

4.3.2 Numerical Noise

4.3.3 Extremely Distorted Magnetic Fields and Search Progression

4.3.4 Dealing with Multiple $K$ Instances Per Line

4.3.5 Final Updated Drift Trajectory Search Algorithm

4.3.6 Accuracy and Convergence

4.4 Application of Methods

4.4.1 Van Allen Probes $L^*$ and $K$ Comparisons

4.4.2 LFM vs. LFM/RCM Full Field Comparisons
## Tables

### Table

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Enlil Variables</td>
<td>25</td>
</tr>
<tr>
<td>2.2</td>
<td>Impact Domain Plot Values</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>LFM grid resolutions</td>
<td>66</td>
</tr>
</tbody>
</table>
Figures

Figure

1.1 The Sun Earth Connection .................................................. 3
1.2 Diagram of Magnetosphere ............................................... 4

2.1 Overview of Enlil Simulation Space .................................... 17
2.2 Overview of the Cone Model ............................................. 22
2.3 WSA/Enlil Simulation of 06 October 2013 CME .................. 29
2.4 Illustration of Impact Definition ....................................... 30
2.5 Impact Domain Plots for max(DP) and max(\frac{dDP}{dt}) .......... 33
2.6 Impact Domain Plots for time to impact and CME Contact Duration ........ 36
2.7 Impact Domain Plots for Solar Wind Properties ................. 38
2.8 Quartile Envelopes for Solar Wind Properties ................. 42

3.1 Results of Re-sampling Field Lines ............................... 51
3.2 Analytic Dipole \(L^*\) vs. Intersection \(\lambda\) .................. 59
3.3 \(\lambda\) Error profiles ..................................................... 60
3.4 \(E_{\text{trace}}\) Error profile for 6.6\(R_E\) ....................... 61
3.5 \(E_{\text{trace}}\) Error profile for range of \(L^*\) values ........ 62

4.1 LFM Grid Side and Front View ..................................... 64
4.2 LFM Grid 2D Details ................................................... 65
4.3 \(L^*\) Errors on LFM Grid ............................................. 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.31</td>
<td>Van Allen Probes LFM/RCM $L^*$ Plot</td>
<td>97</td>
</tr>
<tr>
<td>4.32</td>
<td>LFM/LFMRCM $</td>
<td>B</td>
</tr>
<tr>
<td>4.33</td>
<td>LFM/LFMRCM Comparison $K$ equatorial</td>
<td>101</td>
</tr>
<tr>
<td>4.34</td>
<td>LFM/LFMRCM Comparison $K$ meridional</td>
<td>102</td>
</tr>
<tr>
<td>4.35</td>
<td>LFM/LFMRCM Comparison - $L^*$ equatorial</td>
<td>104</td>
</tr>
<tr>
<td>4.36</td>
<td>LFM/LFMRCM Comparison - $L^*$ meridional</td>
<td>105</td>
</tr>
<tr>
<td>4.37</td>
<td>LFM/LFMRCM Comparison - $L^*$ contours</td>
<td>106</td>
</tr>
<tr>
<td>4.38</td>
<td>LFM/LFMRCM Comparison</td>
<td>107</td>
</tr>
<tr>
<td>5.1</td>
<td>Point $L^*$ Histogram</td>
<td>109</td>
</tr>
<tr>
<td>5.2</td>
<td>Point Completion Time Histogram</td>
<td>109</td>
</tr>
<tr>
<td>5.3</td>
<td>$L^*$ vs. Point Completion Time</td>
<td>110</td>
</tr>
<tr>
<td>5.4</td>
<td>Graphical Computation Completion Time</td>
<td>111</td>
</tr>
<tr>
<td>5.5</td>
<td>GHOSTpy Performance Profile for Single $L^*$ Point</td>
<td>112</td>
</tr>
<tr>
<td>5.6</td>
<td>GHOSTpy Performance Profile for KB Line Search</td>
<td>113</td>
</tr>
<tr>
<td>5.7</td>
<td>GHOSTpy Performance Profile for K calculation</td>
<td>114</td>
</tr>
<tr>
<td>5.8</td>
<td>GHOSTpy Parallel Scalability</td>
<td>117</td>
</tr>
<tr>
<td>5.9</td>
<td>GHOSTpy Parallel Timing Profiles</td>
<td>118</td>
</tr>
<tr>
<td>5.10</td>
<td>GHOSTpy Parallel Comms Timing Profiles</td>
<td>119</td>
</tr>
<tr>
<td>5.11</td>
<td>GHOSTpy Performance Histogram for Parallel Execution Time</td>
<td>121</td>
</tr>
<tr>
<td>5.12</td>
<td>GHOSTpy Performance Histogram for Parallel Execution Time (Improved Binning)</td>
<td>122</td>
</tr>
</tbody>
</table>
Chapter 1

Space Weather

1.1 A Brief History of Space Weather

The effects of space weather have been observed for centuries. Records of auroral observation date as far back as 193 BC; verified observations from independent observers in multiple locations on the same night were seen as early as 1101 AD, and indications of possible intense geomagnetic storms arose by 1138 AD. (Willis and Stephenson, 1999). In the 1600s, Galileo Galilei postulated that the aurorae were caused by sunlight illuminating air as it escaped the Earth’s shadow. René Descartes suggested that they were caused by ice crystals forming high in the atmosphere (Kivelson and Russell, 1995). With the discovery of Earth’s magnetic field in 1700s, the aurorae were finally linked to the magnetosphere (Hiorter, 1747).

On September 1, 1859, Richard Carrington observed, in what would prove to be one of the most severe space weather events to this day, an intense white-light brightening of the sun (Carrington, 1859). The “intensely bright and white light,” what we now refer to as a solar flare, was followed 18 hours later by an intense magnetic storm. Baker et al. (2008), a report by the National Academy of Science on severe space weather events and their societal impacts, succinctly summarizes the news reports of the time. Campers in the Rocky Mountains reported “auroral light, so bright that one could easily read common print.” Henry C. Perkins, a physician in Massachusetts, reported “a perfect dome of alternate red and green streamers” over New England. The aurora were seen as far south as Central America in the northern hemisphere, and as far north as Santiago Chile in the southern hemisphere. Some telegraph operators reported fires started from induced currents.
in telegraph lines, while others reported being able to disconnect batteries from their systems and still send messages. By the mid 1860s, researchers Hermann Fritz in Zürich, and Elias Loomis at Yale made convincing arguments for a connection between auroral occurrences and the sunspot cycle (Schröder, 1998), but it wasn’t until the advent of radio astronomy in the 1930s that the significance of these connections was appreciated, and not until after the start of the space age did the concept of what we now call space weather emerge (Bleeker et al., 2001).

In the Early 1950s, Eugene Parker, following up on observations by Ludwig Biermann (Biermann and Schlüter, 1951), postulated the idea of a supersonic magnetized plasma being continuously ejected from the Sun, now known as the solar wind. This conclusion forever changed the way scientists perceive space, and led to a much greater understanding of how the Sun and Earth are connected magnetically.

In 1958, following the successful launch of the Explorer I and Explorer III spacecraft (Explorer II was a launch failure), James Van Allen and his team discovered what are, today, called the Van-Allen Radiation Belts. First announced on May 1, 1958 as “enormous intensities of geomagnetically trapped corpuscular radiation” (from James Van Allen’s Preface to Chapter 5 of Bleeker et al. (2001)), the radiation belts have become an extremely important aspect of space weather, as their changes can impact a great deal of today’s space based technology (Baker et al., 2008).

Another significant advance in the theories governing space weather came with the discovery of intermittently occurring events involving large volumes of sun-ejected plasma, called coronal mass ejections (CMEs), in the 1970s. This discovery brought with it the recognition that CMEs were the source of non-recurrent geomagnetic storms, not solar flares (Gosling, 1993).

The potential effects of CMEs on technology was dramatically demonstrated on March 13, 1989, when a large CME impacted the Earth (Baker et al., 2008). Some of the most notable effects were on the power grid. The Quebec grid was tripped off-line, which caused the Provence to experience a blackout for about 9 hours. The United States experienced more than 200 separate power grid related effects. In October of 2003, the disruptive potential of space weather was again made clear during what are now known as the “Halloween Storms.” These storms, following more
than a decade of technological advancements since the 1989 storm, generated a more varied array of effects, ranging from near complete Global Positioning System (GPS) distribution to the loss of entire satellites (Baker et al., 2008).

These storms had significantly more affect on the way of life on Earth than the relatively mild irritations caused by the Carrington event in 1859. Were the Carrington storm to occur today, the results could be catastrophic (Baker et al., 2008, 2013a). The need for the understanding of, as well as the need to be able to predict, space weather is clear; to maintain our way of life in this technological age, understanding how the sun affects our local space may be crucial.

1.2 Connecting the Sun and the Earth

The primary driver of space weather is the Sun; disgorging a continuous flow of supersonic magnetic plasma known as solar wind. Typical solar wind travels at speeds ranging from 350 – 500 km/s, has a number density around $3 - 5 \text{ cm}^{-3}$, and contains an embedded weak magnetic field, averaging about 5 nT, known as the Interplanetary Magnetic Field (IMF). Generally parallel to the ecliptic plane, various solar conditions can cause the IMF to deviate from its parallel course. This
deviation can then interact with the Earth’s magnetic field, generating the interactions associated with space weather.

Earth’s internal magnetic field, first suggested to be roughly a dipole by Gilbert (1958), interacts with the solar wind, giving rise to various currents within the near Earth region. These currents generate what can be thought of as “external contributions” to the magnetic field. This combination of magnetic field contributions generates the magnetosphere, and can be expressed as:

$$B_{\text{mag}} = B_{\text{int}} + B_{\text{ext}}$$

where $B_{\text{mag}}$ is the magnetic field of the magnetosphere, $B_{\text{int}}$ is the Earth’s intrinsic magnetic field, and $B_{\text{ext}}$ contains the external magnetic field contributions caused by the interaction with the solar wind.

The different external contributions to the magnetosphere can further be broken down by

Figure 1.2: Various regions and currents of the magnetosphere. Image Credit: Kivelson and Russell (1995)
each individual current system, thus giving the expression:

\[ B_{ext} = B_{RC} + B_{TC} + B_{FAC} + B_{INT} + B_{MP} \]

where \( B_{RC} \) is the magnetic field contribution of the ring current, \( B_{TC} \) is the contribution of the tail current, \( B_{FAC} \) is the contribution of the field-aligned current, \( B_{INT} \) is the contribution from the interconnection field, and \( B_{MP} \) is the contribution from the magnetopause. This breakdown is how several empirical magnetospheric models are computed (Tsyganenko, 2014) (This is discussed in more detail in a later section). Figure 1.2 provides a graphical representation of the different current systems that contribute to the magnetosphere.

The interaction of the IMF with the magnetosphere allows for a phenomenon known as reconnection (Dungey, 1961). Given a IMF field oriented southward, opposite that of the geomagnetic field, at the magnetopause boundary on the day-side of Earth, the IMF field will merge with the geomagnetic field lines. The solar wind will draw these field lines (now “open”, or connected to the Sun) to the night side of Earth. The field lines will reconnect in the tail, and magnetic tension will cause the field lines to draw back sunward.

1.3 Radiation Belts

1.3.1 Trapped Particle Motion

A trapped charged particle’s motion in an electromagnetic field is described by the Lorentz Force equation (Northrop, 1963; Roederer and Zhang, 2014), and is given in Equation (1.1).

\[ \frac{d\vec{p}}{dt} = q\vec{v} \times \vec{B} \]  

(1.1)

where \( \vec{p} \) is the particles momentum, and \( \vec{v} \) is the velocity in the “original frame of reference (OFR).”

The practical result of this is a gyration of the particle around its field line. The particle’s center of gyration is referred to as the “guiding center,” and moves along the field line in relation to its initial parallel velocity and any parallel force. If viewed from what Roederer and Zhang (2014) calls the
“guiding center system (GCS)” frame of reference, which is a frame that travels along the guiding center with a velocity:

\[ \vec{v}_\parallel = \text{const}, \quad \vec{v}_\perp = 0 \]

In this frame of reference, \( v_\perp \) is constant, and the particle’s gyroradius can be obtained with:

\[ \rho_C = \frac{v_\perp^2}{a_\perp} = \frac{m v_\perp^*}{q B} = \frac{p_\perp^*}{q B} \quad (1.2) \]

The cyclotron period can be defined as:

\[ \tau_C = \frac{2 \pi \rho_C}{v_\perp^*} = \frac{2 \pi m}{q B} \quad (1.3) \]

and the gyrofrequency can be expressed as:

\[ \omega_C = \frac{2 \pi}{\tau_C} = \frac{q B}{m} \quad (1.4) \]

When an external force is introduced perpendicular to the local magnetic field, the guiding center of a particle can drift across field lines. In the case of an external force generated by an electric field, this is given by Equation (1.5), and is often referred to as “E-cross-B drift.”

\[ \vec{v}_E = \vec{E} \times \vec{B} \quad (1.5) \]

Additionally, centrifugal force felt by a particle bouncing between magnetic poles leads to curvature drift, given as:

\[ \vec{v}_c = \frac{p_\parallel^2}{q \gamma m_0} \frac{\vec{B} \times (\hat{b} \cdot \nabla) \hat{b}}{B^2} \quad (1.6) \]

where \( \gamma \) is the relativistic correction factor, and \( \hat{b} \) is the unit vector in the direction of \( \vec{B} \). A final drift, the grad-B drift, caused by magnetic field gradients perpendicular to the local magnetic field is given by Equation (1.7).
\[ \vec{v}_{\nabla B} = \frac{M \vec{B} \times \nabla_{\perp} B}{q\gamma B^2} \] (1.7)

where \( M \) is the magnetic moment of the particle. Together, Equations (1.6) and (1.7) are known as the gradient-curvature drift. This drift, when considered in the magnetosphere, causes positively charged particles to drift clockwise around the Earth when viewed from the north pole, and negatively charged particles to drift counter-clockwise. These two drift populations, in combination with each other, generate the ring-currents within the magnetosphere. A more thorough treatment of this topic can be found in Roederer and Zhang (2014) and Northrop (1963).

1.3.2 The Adiabatic Invariants

The Lorentz equations describe three modes of motion for a particle trapped in a magnetic field: Gyro-motion, as described above, bounce motion (particle bouncing between two mirror points in a magnetic field), and a drift motion perpendicular to the magnetic field. Each type of motion has a quasi-conserved quantity called an adiabatic invariant. These quantities are conserved as long as the field changes slowly in time and space compared to the relevant properties of each motion (time and space). These three invariants are described in following sections.

1.3.2.1 The First Adiabatic Invariant

The invariant associated with the gyromotion of a charged particle in a magnetic field is given by Equation (1.8), and is often referred to as the magnetic moment of the particle.

\[ M = \frac{p^2}{2m_0 B} = \frac{p^2\sin^2\alpha}{2m_0 B} \] (1.8)

This invariant is conserved if the changes in the magnetic field occur on a time scale large compared to the gyroperiod (see Equation (1.3)), or the geometry of the field changes on a scale large compared to the gyroradius (see Equation (1.2)). If changes are not large compared to these properties, the invariant nature may be broken, and the particle motion can only be described by the complete Lorentz Equation.
The literature exhibits confusion over what symbol should represent the first adiabatic in-
variant. Walt (1994) and Kivelson and Russell (1995) use the term $\mu$ to signify the invariant, while
Roederer and Zhang (2014) uses $M$. Other sources utilize $\mu$ to denote the non-relativistic form
of the invariant, while reserving $M$ for the relativistic form. In this work, $\mu = M$ and are used
interchangeably.

### 1.3.2.2 The Second Adiabatic Invariant

The second invariant, which is associated with the bounce motion of a particle, is related
to the fact that the conservation of the first invariant ($M$) within a converging magnetic field,
particles are subject to mirroring. The mirror field for a particle is given in Equation (1.9). This
field strength can also be computed as a relationship between the local $B$ field and the particle’s
pitch angle, $\alpha$.

$$B_m = \frac{B}{\sin^2 \alpha} = \frac{p^2}{2m_0 \mu} \quad (1.9)$$

where $\alpha$ is given by:

$$\alpha = \tan^{-1} \left( \frac{v_\perp}{v_\parallel} \right) \quad (1.10)$$

Within dipole-like field geometries, mirroring leads to a bounce motion along the guiding
center of the particle, leading to a periodic motion between the two poles. The invariant associated
with this periodic motion is given in Equation (1.11).

$$J = \oint p_\parallel ds = \oint p \cos \alpha(s) ds = \oint p \sqrt{1 - \frac{B(s)}{B_m}} ds = 2pI \quad (1.11)$$

$$I = \int_{s_m}^{s'_m} \sqrt{1 - \frac{B(s)}{B_m}} ds \quad (1.12)$$

where the integral $I$ is along the field line intersecting with the particle’s position, $s_m$ and $s'_m$ are
the mirror points where $B(s) = B_m$. The integral $I$ is a purely field-geometry integral, and it is
conserved as long as the kinetic energy and \( J \) are conserved. In realistic magnetic field conditions, where the field is acted upon by an external force, \( I \) will no longer be constant. Utilizing the invariant nature of the first invariant, \( M \), a new field line invariant, \( K \), can be constructed (Roederer and Zhang, 2014). The equation for \( K \) is given in (1.13).

\[
K = \frac{J}{2\sqrt{2mM}} = I\sqrt{B_m} = \int_{s_m}^{s_m} \sqrt{(B_m - B(s))} \, ds = \text{const} \quad (1.13)
\]

The combination of \( M, K \), and the conservation of energy allow for the definition of a particle's guiding field line at any longitude, \( \phi \). The entire drift shell for a particle can be determined numerically by iteratively searching for field lines that satisfy the same values of \( B_m \) and \( K \) (Roederer, 1970; Roederer and Zhang, 2014).

### 1.3.2.3 The Third Adiabatic Invariant

The adiabatic invariant associated with the third type of motion, drift motion, is given by Equation (1.14).

\[
\Phi = \iint \vec{B} \cdot d\vec{s} = \iint \vec{B} \cdot \hat{n} \, ds \quad (1.14)
\]

and represents the magnetic flux contained within a particle’s drift orbit. Numerically, we can calculate this with the flux through Earth’s polar cap, bounded by the surface intersection of the field lines making up the particle’s drift shell (Roederer and Zhang, 2014). As the time scale for the drift orbit of a particle is much larger than those for gyro-motion and bounce-motion, this invariant is the most likely to not be conserved. A convenient reference value, based on \( \Phi \), is the \( L^* \) parameter; a value that represents where a particle would be located if all external contributions to the magnetic field are slowly relaxed (see Section 1.5.2 for a description of external contributions). The relationship between \( \Phi \) and \( L^* \) is given by:

\[
L^* = \frac{2\pi B_E R_E^2}{\Phi} \quad (1.15)
\]
where $B_E$ is the value of $B$ at the equatorial point on the surface of the Earth, and is proportional to Earth’s dipole moment. $R_E$ represents the radius of Earth (Roederer and Zhang, 2014).

### 1.4 Analysis and Visualization of Particles in the Magnetosphere

The use of $L^*$ is prevalent and critical in the analysis and visualization of the radiation belts (Hudson et al. (1997, 2015); Elkington et al. (2002, 2004); Tu et al. (2012); Baker et al. (2004) to name a small selection of examples), yet the means of calculating the values is mostly relegated to empirical models of the magnetosphere.

Of the $L^*$ calculation libraries available, IRBEM (Boscher et al., 2012) utilizes by default the IGRF model, but can be set to use a user-defined model as long as it meets specific requirements, and implements the method for calculating $L^*$ found in Roederer (1970). LANLGeoMag (Henderson, 2010) solves the guiding center equations utilizing a selection from the Tsyganenko models. LANL* (Koller et al., 2009; Koller and Zaharia, 2011; Yu et al., 2012) takes the unique approach of utilizing an artificial neural network to predict $L^*$ on a Tsyganenko model. Min et al. (2013a,b) utilizes a technique, similar to Roederer (1970), which uses two conserved quantities to find drift trajectories in a Tsyganenko model, but utilizes iso-energy contour from a discrete 2-D energy space, defined by $M$ and $K$ to calculate drift orbits. None of these methods deal specifically with optimized calculations of $L^*$ for large gridded MHD fields. The research in this thesis intends to provide a solution to fill the need for determining the three invariants, $M$, $K$, and $L^*$, in these complex field geometries.

### 1.5 Numerical Models of the Earth’s Magnetosphere

As eloquently noted in Tsyganenko (2013), the geomagnetic field is “the principle agent connecting our planet’s ionosphere with the highly variable interplanetary medium, incessantly disturbed by dynamical processes at the Sun.” This reality has led scientists to devise models of how the magnetosphere behaves. The following three sections describe three different types of models used to aid in the understanding of how the Earth’s magnetosphere behaves. The first of the
sections covers the first order approximation of the magnetic field; the simple dipole. The second section covers the series of Tsyganenko empirical models, and the third section briefly discusses MHD models.

1.5.1 Dipole Geomagnetic Field Model

The first order approximation of the Earth’s magnetosphere is the basic dipole. The dipole magnetic field provides many advantages when developing analytic methods for more complex fields, as the field itself and many analyses of the field have analytic solutions, providing a baseline for error analysis of methods.

The basic equations for a dipole can be found in many introductory books on space physics. Kivelson and Russell (1995) provides equations to produce a dipole in Cartesian coordinates:

\[
B_X = 3xyM_zr^{-5} \\
B_Y = 3yzM_zr^{-5} \\
B_Z = (3z^2 - r^2)M_zr^{-5}
\]

where \(B_x, B_y,\) and \(B_z\) are components of the magnetic field, \(\langle x, y, z \rangle\) representing the location vector, and \(r\) is the radial distance from center of the Earth. \(M_z\) is the z component of the Dipole Moment. In reality, Earth exhibits a difference between the spin-axis and the magnetic axis. This difference induces what is called a dipole tilt. The basic model provided by Kivelson and Russell (1995) does not, however, take into account any form of dipole tilt. For a more complete solution, the equations provided in (1.17) incorporate the requisite tilt. These equations are derived from the Tsyganenko T96 model code ((Tsyganenko, 1995, 1996)), and generate a dipole field at the given location \(\langle x, y, z \rangle\) in the GSM coordinate system.
\[ B_x = \frac{M_z}{(\sqrt{x^2 + y^2 + z^2})^5} \left( (y^2 + z^2 - 2x^2) \sin(\psi) - 3 \ z \ x \ cos(\psi) \right) \]

\[ B_y = -3y \frac{M_z}{(\sqrt{x^2 + y^2 + z^2})^5} \left( x \times \sin(\psi) + z \cos(\psi) \right) \] (1.17)

\[ B_z = \frac{M_z}{(\sqrt{x^2 + y^2 + z^2})^5} \left( (x^2 + y^2 - 2z^2) \cos(\psi) - 3 \ z \ x \ sin(\psi) \right) \]

For these equations, \( M_z \) is again the \( z \) component of the Dipole Moment (we use \( 3.14 \times 10^4 nT \cdot R_E^3 \)) and \( \psi \) is the dipole tilt applied to the field (in radians). The vector \( \langle x, y, z \rangle \) again serves as spatial location (in units of \( R_E \)).

1.5.2 Empirical Geomagnetic Field Models

An improvement above and beyond the first order approximation requires additional contributions to the magnetic field. In empirical models, these contributions are determined by fitting a model to a large quantity of observational spacecraft data. These models are typically fit for individual contributions to the magnetic field. According to Tsyganenko (2013), the total magnetospheric magnetic field vector \( B \) can be represented as the sum of these individual contributions. Equation (1.18) gives an example of this combination methodology

\[ B = B_I + B_E \] (1.18)

where \( B_I \) represents the internal contribution to the magnetic field (the main geomagnetic field), and \( B_E \) represents the external contributions to the magnetic field (associated with electric currents flowing inside and outside Earth).

The external contribution, \( B_E \), can be further broken down into the sum of contributions from individual influences. This results in \( B_E \) being defined as in equation (1.19).

\[ B_E = B_{RC} + B_{TC} + B_{FAC} + B_{INT} + B_{MP} \] (1.19)

where \( B_{RC} \) is the magnetic field contribution of the ring current, \( B_{TC} \) is the contribution of the tail current, \( B_{FAC} \) is the contribution of the field-aligned current, \( B_{INT} \) is the contribution from
the interconnection field, and $B_{MP}$ is the contribution from the magnetopause field (also referred to as the shielding field). The Tsyganenko magnetic field models calculate, in varying combinations (depending on the specific model in question), external contributions, and combine them into a unified $B_E$. Adding these external contributions to an internal ($B_I$) field will result in a model of the magnetosphere.

The following outlines some of the more commonly used Tsyganenko models. One of the earlier empirical models used in the field was the Tsyganenko T89 Model (Tsyganenko, 1989)). As an input, this model utilizes the planetary K index ($K_P$) (Bartels et al., 1939). The $K_P$ index is a 3-hour global index based on a weighted average of local $K$ values. These values are, in turn, a quantification of the disturbances to the horizontal component of the Earth’s magnetic field. The index value ranges from 0 to 9. By Tsyganenko’s reckoning (Tsyganenko, 1995), the T89 model (and its earlier T87 iteration), suffered from three serious deficiencies: it had an unstable “de facto” magnetopause; a crude method of parameterizing, utilizing only $K_P$; and inaccuracies in the $B_z$ values of the equatorial magnetotail. The Tsyganenko (1995) model revisions were designed to alleviate some of these deficiencies through adding a realistic shape to the magnetopause, a fully controlled shielding component for all current contributions, a flexible tail and ring current representations, and a more accurate fitting of spacecraft data to the field. Tsyganenko (1996) describes the results of the completed T96 model. The inputs for the T96 model consist of solar wind dynamic pressure ($p_{dyn}$), IMF $B_y$, IMF $B_z$, and $D_{st}$.

The Tsyganenko (2002b) (T02) variant was built upon the T96 model, and updated the system with new methods. In addition to standard solar wind inputs, the model was adjusted to add latent terms that depend on past solar wind conditions. This model updates the ring current method from the simple empirical ring currents of the T96 and earlier models to a more accurate approximation based the electric current calculated from an observed distribution of particle pressure and anisotropy. The new ring current model also takes into account the strong dawn-dusk asymmetry of the ring current during times of disturbance (more on the new ring current model can be found in Tsyganenko (2000)). The Birkeland Currents simulation for this model were
completely redesigned to address a major deficiency of the old methods; insufficient flexibility. A
detailed description of the mathematical structure of the new Birkeland Currents module is given

In Tsyganenko and Sitnov (2007), a model is presented that expands the field to include
contributions from arbitrary azimuthal and radial variations of the geomagnetic field, allowing for a
more natural way to represent the field related to the tail and ring (partial and symmetric) currents.
In Tsyganenko (2014), model improvements were made that take into account the IMF dependent
shape of the magnetopause, and provided a physically more consistent global deformation of the
equatorial current sheet due to dipole tilt. The ring current module was updated to reflect a more
realistic background magnetic field, and the entire model valid range was extended to between 40
and 50 $R_E$. Most recently, Tsyganenko and Andreeva (2015) provided an update to the model to
include the Newell coupling function (see Equation (2.1) in Chapter 2) (Newell et al., 2007) as a
driver for the model.

1.5.3 Magnetohydrodynamic (MHD) Geomagnetic Field Models

MHD models take a different approach to solving the magnetosphere problem. Instead of
utilizing observational data to define a model, these models treat the system as a magnetized
plasma, governed by the magnetohydrodynamic equations for the magnetosphere. A combination
of Maxwell’s equations and magnetized fluid equations, the MHD equations can be expressed as
follows (from Kivelson and Russell (1995)):

Continuity Equation:

$$\frac{\delta \rho}{\delta t} = - \nabla \cdot (V \rho)$$  \hspace{1cm} (1.20)

Momentum Equation:

$$\frac{\delta V}{\delta t} = - (V \cdot \nabla) V - \frac{\nabla p}{\rho} + \frac{(J \times B)}{\rho}$$  \hspace{1cm} (1.21)
Pressure Equation:
\[ \frac{\delta p}{\delta t} = -(V \cdot \nabla)p - \gamma p \nabla \cdot V \]  
(1.22)

Faraday’s Law:
\[ \frac{\delta B}{\delta t} = \nabla \times (V \times B) + \eta \nabla^2 B \]  
(1.23)

Ampere’s Law:
\[ J = \nabla \times (B - B_I) \]  
(1.24)

where \( \rho \) is the plasma density, \( V \) is the velocity, \( p \) is the pressure, \( B \) is the magnetic field, and \( B_I \) is the Earth’s internal field. The \( \gamma \) term is the polytropic index, and \( \eta \) is the magnetic diffusivity.

The Lyon-Feddar-Mobarry (LFM) 3D MHD magnetospheric model has been in use for nearly three decades (Lyon et al., 2004). The model was designed to satisfy several criteria: maintain \( \nabla \cdot \vec{B} = 0 \) to within the roundoff error; put resolution where it is needed; use special methods where there is a high magnetic field; and use an integrated ionospheric model. The model utilizes a finite volume technique on a stretched spherical non-orthogonal adapted grid that places the majority of the resolution closest to Earth, and takes the upstream solar wind conditions as input. The model outputs its results in the SM coordinate system.

The LFM model solves a somewhat different form of the MHD equations. They are cast in the form:

\[ \frac{\delta \rho}{\delta t} = -\nabla \cdot \rho \vec{v}, \]  
(1.25)

\[ \frac{\delta \rho \vec{v}}{\delta t} = -\nabla \cdot (\rho \vec{v} \vec{v}) - \nabla \cdot \left( \frac{I B^2}{8\pi} - \frac{\vec{B} \vec{B}}{4\pi} \right), \]  
(1.26)

\[ \frac{\delta E_p}{\delta t} = \nabla \cdot \left( \vec{v} \left( \rho v^2/2 + \frac{\gamma}{\gamma - 1} P \right) \right) - \vec{v} \cdot \nabla \cdot \left( \frac{I B^2}{8\pi} - \frac{\vec{B} \vec{B}}{4\pi} \right), \]  
(1.27)

\[ \frac{\delta \vec{B}}{\delta t} = \nabla \times (\vec{v} \times \vec{B}), \]  
(1.28)
where $E_P = \rho v^2/2 + \frac{P}{\gamma - 1}$, $\rho$ is plasma density, $\vec{v}$, velocity, $P$, pressure, and $\vec{B}$, the magnetic field.

Lyon et al. (2004) provides in-depth detail on the construction of the model.

1.6 Thesis Format

This thesis is broken down into 5 major parts. First, in Chapter 2 covers interactions between CMEs and the solar wind, providing for analysis of observational errors vs. modeled transients. Chapter 3 covers ideal analytic analysis for the adiabatic invariant algorithms, covering errors associated with algorithms used to compute $L^*$. Chapter 4 describes the effects of the LFM grid on how the algorithms perform, and how they needed to be modified to work with LFM data, then presents a brief real-world analysis utilizing the new tools. Chapter 5 covers serial and parallel system timings, algorithm profiles, load balancing, and efficiency analysis. Finally, Chapter 6 possible improvements that can be implemented, and future directions of research.
The use of complex physics-based models is relatively new in space weather forecasting, and there is much to learn about how they operate in a production forecast environment. When predicting the transit of a coronal mass ejection (CME) through the solar system, space weather forecast centers often utilize the WSA/Enlil + Cone modeling suite (Parsons et al., 2011; Millward et al., 2013; Lee et al., 2013; Steenburgh et al., 2014; Mays et al., 2015). This suite is a complex
system that consists of three coupled models: the Wang-Sheeley-Arge (WSA) solar model (Arge and Pizzo, 2000), a geometric CME cone model (Zhao et al., 2002; Xie et al., 2004), and the Enlil ambient solar wind model (Odstrcil, 2003; Odstrcil et al., 2004; Odstrcil, 2004a,b). The WSA model takes synoptic maps as input and utilizes a hybrid empirical and physics-based approach to simulate the solar corona and solar wind through to the inner boundary of the Enlil Model. The cone model is used to estimate the morphological and kinematic properties of CME ejecta from coronagraph images; these properties are injected into the inner boundary of the Enlil model at the simulation time necessary to model the propagation of the CME through the heliosphere. Enlil is a 3-D magnetohydrodynamic (MHD) model that simulates the morphological changes to, and provides time dependent conditions of, the solar wind through the heliosphere (to the outer boundary of the simulation). As a unit, this coupled system attempts to provide a physically accurate representation of solar wind conditions as CMEs travel from the Sun to the Earth. It should be noted that the WSA/Enlil + Cone model suite does not provide a representation of the magnetic field associated the CME ejecta cloud, thus limiting the usefulness of $|B|$ values within the forecast. Figure 2.1 provides a large-scale overview of the Enlil simulation space, illustrating the scale and morphology of a CME as it propagates through the inner solar system. The small dot at the leading edge of the CME cloud represents the grid location containing Earth.

Although this coupled-model environment has seen increasing use over the past several years (e.g. Baker et al. (2009, 2011, 2013b); Millward et al. (2013); Lee et al. (2013); Dewey et al. (2015); Mays et al. (2015); Cash et al. (2015)) , the predictions of CME transits still suffer from uncertainties related to cone model fitting from incomplete data (Pulkkinen et al., 2009; Lee et al., 2013; Mays et al., 2015). It is noted in Pulkkinen et al. (2009) that when using a single spacecraft coronagraph source for model fitting an assumption on the structure of the CME is required. The assumptions that generate the cone model (i.e. the CME propagates as a geometric cone) are, at best, only rough approximations of the actual ejecta cloud geometry. Furthermore, natural uncertainties arise when manually fitting a cone to a CME due to ambiguities in identifying the ejecta structure. Together, these conditions lead to uncertain central origins for the modeled CMEs.
Eliminating the human factor from ejecta identification, by means of automated fitting, makes for more consistent cone estimations; however, uncertainties still exist in the modeled origins due to the nature of the machine vision algorithms used (Pulkkinen et al., 2009; Jacobs et al., 2013).

As cone parameter uncertainties must be addressed in order to accurately predict a CME transit, methods have been developed over time to help mitigate the associated issues. One such method is **ensemble modeling**; a technique whereby multiple simulations are conducted for a given event (with CME cone parameters generated with some statistical approach) to provide a probabilistic forecast for CME arrival time (Lee et al., 2013; Mays et al., 2015). While these ensemble predictions provide a means for generating a probabilistic forecast, a more detailed understanding of the distribution of effects due to observational uncertainties is desired.

It is posited that the error responses to these observational input uncertainties and their propagation within the numerical models can have a significant effect on a forecast outcome. To test this hypothesis we have devised a method, called **CME Impact Domain Analysis (CME IDA)**, for describing the distribution of possible forecast results when utilizing the WSA/Enlil + Cone modeling suite (the method should also work, with some modification, for any other heliospheric solar wind model). Several works in the literature point to the CME central origin (solar latitude and longitude) being one of the most critical parameters for generating an accurate forecast (Pulkkinen et al., 2009; Millward et al., 2013; Lee et al., 2013; Mays et al., 2015), therefore, this study places a primary focus on the CME central origin. With this focus, a domain is defined that is inclusive of model runs, with varying central origins, that will generate a CME impact at our target location, in this case, Earth. Called the **Impact Domain**, this abstract domain can be used to analyze the statistical distribution of CME effects on the solar wind, and how they vary with the uncertainty of input.

Utilizing a Community Coordinated Modeling Center prediction for a CME that occurred on October 06, 2013 at 1439 UTC, this work constructs, by example, a CME IDA. This method generates IDA analysis plots to describe the statistical distribution of the impact-time, impact-duration, and effect on various solar wind parameters. Additionally, an analysis similar to superposed epoch
analysis (SEA) (Chree, 1913) is used to construct time-dependent tendencies for the domain. Following a description of the model environment and example of the method, the information obtained from these analyses is discussed and a process for generalizing the method is described.

2.1 The Model Environment

In an effort to understand how uncertainties affect a forecast, a simulation modeling configuration was chosen for this study that parallels an actual forecast environment. Steenburgh et al. (2014) outlines the concept of operations (CONOPS) for the WSA-Enlil operational transition used for space weather forecasting at the Space Weather Prediction Center (SWPC), and provides a useful resource for configuring the simulated forecast environment.

The Steenburgh et al. (2014) CONOPS document, as well as those by Parsons et al. (2011) and Mays et al. (2015) indicate that, within an operational environment, the Global Oscillation Network Group (GONG) synoptic magnetograms are typically used as the input for the WSA model, and the cone model is used for CME kinematic parameter insertion into the Enlil model inner boundary. With these documents as reference, it was decided to utilize the resources of the CCMC to conduct simulation runs of the WSA/Enlil + Cone coupled system. The GONG magnetogram synoptic maps were used as input, and the cone model profile generated by the CCMC/SWRC for the 06 October 2013 event was used for baseline CME parameter insertion.

2.1.1 The WSA Model

In this study, the simulation of a CME begins with the Wang-Sheeley-Arge (WSA) solar model (Arge and Pizzo, 2000). This model, which is an improvement on the original Wang-Sheeley model (Wang and Sheeley, 1992), is a hybrid empirical and physics-based model of the sun’s corona and solar wind. The WSA model is composed of three parts. Ground based observations of the photospheric magnetic field provide synoptic maps as input to a magnetostatic potential field source surface (PFSS) model (Schatten et al., 1969) that determines the coronal field to 2.5 solar radii ($R_\odot$). Next, the PFSS provides input to the Schatten Current Sheet model (SCS) (Schatten, 1971)
which provides a magnetic field topology of the upper corona. Finally, the model uses an empirical velocity relationship to provide solar wind speed at the outer coronal boundary (truncated at $30R_{\odot}$) (Arge and Pizzo, 2000).

For input to the Enlil model, only the innermost portion of the SCS solution is utilized, providing Enlil with the radial magnetic field and solar wind speed at $21.5R_{\odot}$. Densities and temperatures at the boundary are not provided directly by WSA, but are inferred by assuming mass flux conservation and pressure balance (Odstrcil, 2003; Lee et al., 2013).

### 2.1.2 The Cone Model

Within the forecast environment, one of the primary means of determining kinematic and morphological properties of CMEs is through the application of the cone model (Millward et al., 2013; Steenburgh et al., 2014; Mays et al., 2015). In Zhao et al. (2002) it was observed that, for CMEs on the limb of the sun, most (but not all) coronal mass ejections propagate nearly radially for the first couple of solar radii. If the assumption is taken that earthward facing (halo) CMEs also exhibit this behavior, then geometric analysis of the CME images can produce a 3-dimensional representation of the CME, modeled as a geometric cone. Zhao et al. (2002) proposed an iterative method for fitting the cone to the observed picture. This was extended by Xie et al. (2004) into an analytic method, which reduced the time and complexity required to calculate a cone fit.

Figure 2.2, a figure from Pulkkinen et al. (2009), provides the relationships between various parameters of a CME as calculated via the cone model. Within this figure, the propagation direction of the CME cloud is defined on the $(x, y, z)$ coordinates, while the plane-of-sky is defined on the $(x', y', z')$ coordinates. The two ellipses represent the boundary of the CME as seen in two sequential coronagraph images. $\omega$ is the cone half-angle, and $\alpha$, along with $\theta$, defines the direction of propagation. With the assumption that CMEs propagate radially from the Sun center, the direction of propagation implies the central origin of the CME. The velocity is calculated from the relationship $x = x_0 + v \cdot \Delta t$, where $x$ and $x_0$ are determined by fitting the same cone profile to two sequential coronagraph images, and $\Delta t$ is determined by the elapsed time between the two images.
Figure 2.2: Cone Model: “The plane \((y', z')\) refers to the plane of sky \((x'\)-axis points toward the observer, \textit{i.e.} toward the reader). Angle \(\alpha\) defines the direction of the propagation of the CME in the \((y', z')\) plane, \textit{i.e.} the angle between the \(y'\)-axis and the \(x\)-axis as projected into the \((y', z')\) plane. Angle \(\theta\) defines the rotation of the cone off of the \((y', z')\) plane, \textit{i.e.} the angle between the \(x'\)- and \(x\)-axes. \(\omega\) is the opening half-angle of the cone, \(x_0\) the initial distance of the cone front in the rotated coordinates \((x, y, z)\) and \(v\) is the velocity of the cone front propagation. \(\Delta t\) indicates the time interval during which the cone front propagates from \(x_0\) to \(x\).” (Pulkkinen et al., 2009)
According to Millward et al. (2013), when faced with only a single coronagraph perspective, forecasters rely on the Xie et al. (2004) method. With this method an ellipse is drawn over the coronagraph image with the minor axis intersecting the center of the sun. A cone can then be calculated to determine the central origin (heliospheric latitude and longitude) and the cone angle in terms of the major and minor axis of the ellipse, its distance from origin, and the angle of the minor axis with respect to the horizontal. Their experimental results show, as a CME halo tends toward circular geometry, the parameters derived are extremely sensitive to the exact form of the ellipse. The paper explores this phenomenon through multiple fittings of two different CMEs. It was determined that the CME parameter most affected by the model uncertainties is the latitude of the central origin, giving greater uncertainty to the origin point and direction of CME propagation. Millward et al. (2013) further noted that the absolute direction of events is a critical parameter in assuring accurate arrival times.

It is obvious from these works that determining exact initial parameters for a CME is not possible from the limited information available. The assumption that a CME propagates as a geometric cone is only a rough approximation of the actual ejecta cloud morphology, and the natural uncertainties that arise when manually fitting a cone to a CME cause ambiguities in identifying the cloud edge, making an exact cone fit unlikely. Still, the cone model seems to be the best choice available for modeling a CME in near-real-time space weather prediction due to its simplicity and speed of calculation.

2.1.3 The Enlil Model

The Enlil solar wind model (Odstrcil, 2003; Odstrcil et al., 2004; Odstrcil, 2004a,b), a 3-D MHD model of the solar wind within the heliosphere, is used to provide, in conjunction with the cone model, the propagation of CMEs through the solar system. Kinematic properties, as determined by the cone model, are injected into the inner boundary of the Enlil model, and are propagated through the simulation space, interacting with the ambient solar wind.

The Enlil model solves a set of time-dependent MHD equations in conservative form on a
spherical grid to propagate solar wind conditions for the time period being simulated. The system solves eight coupled nonlinear partial differential equations utilizing a version of the Total Variation Diminishing (TVD) technique (Harten, 1983; Harten et al., 1987), known as Total Variation Diminishing Lax-Friedrich (TVDLF)(Tóth and Odstrčil, 1996), modified to include a two-step time-discretization (Odstrcil, 2004a). This method is second-order accurate away from shocks and discontinuities, and provides stability that ensures non-oscillatory solutions (Odstrcil, 2003).

To facilitate a more detailed description of the analyses performed in this study, Table 2.1 provides a breakdown of the variables tracked throughout the execution of the Enlil simulation. This information is similar to that found in Table 12 of Odstrcil (2004b), but reflects the state of the current version of the Enlil Model (v. 2.8f). The Name column denotes the name of the variables within the NetCDF output files, the Description column is the plain-language identification of the variables, and the Units column provides the units associated with the values within the variable.

Variables of interest, and that are also used in analyses in this paper, include $X_1$ (heliospheric position), $X_2$ (meridional position), and $X_3$ (azimuthal position), which define the spatial coordinates for the simulation grid (in the HEEQ+180 coordinate system). $B_1$, $B_2$, and $B_3$ are respectively the radial, meridional, and azimuthal magnetic field vector components. Similarly $V_1$, $V_2$, and $V_3$ are the radial, meridional and azimuthal velocity vector components. $D$ is the mass density, $T$ is the temperature and $DP$ traces the evolution of the CME cloud (cloud tracer).

2.2 Mitigation of Uncertainties with Ensemble Modeling

Mays et al. (2015), Lee et al. (2013) and Millward et al. (2013) all provide in-depth detail on the problems encountered when fitting the cone model to CMEs in coronagraph images. To summarize, the issues can be reduced to problems of identifying features for any given CME. It is necessary to accurately identify the ejecta boundary to obtain a proper cone fit; If the CME boundaries are unclear, it becomes difficult to identify the proper ellipse, leading to uncertain measurements. One method for mitigating these uncertainties is through the use of ensemble modeling; a technique whereby multiple simulations are conducted for a given event (with the
Table 2.1: Variables tracked by the Enlil Model

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>Heliospheric Position</td>
<td>m</td>
</tr>
<tr>
<td>$X_2$</td>
<td>Meridional Position</td>
<td>radians</td>
</tr>
<tr>
<td>$X_3$</td>
<td>Azimuthal Position</td>
<td>radians</td>
</tr>
<tr>
<td>$B_1$</td>
<td>Radial Magnetic Field</td>
<td>T</td>
</tr>
<tr>
<td>$B_2$</td>
<td>Meridional Magnetic Field</td>
<td>T</td>
</tr>
<tr>
<td>$B_3$</td>
<td>Azimuthal Magnetic Field</td>
<td>T</td>
</tr>
<tr>
<td>$V_1$</td>
<td>Radial Velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>$V_2$</td>
<td>Meridional Velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>$V_3$</td>
<td>Azimuthal Velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>$D$</td>
<td>Mass Density</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
<td>K</td>
</tr>
<tr>
<td>$D_P$</td>
<td>Cloud Tracer</td>
<td>-</td>
</tr>
<tr>
<td>$B_P$</td>
<td>Polarity Tracer</td>
<td>-</td>
</tr>
<tr>
<td>$D_T$</td>
<td>Physical Time Step</td>
<td>s</td>
</tr>
<tr>
<td>$T_M$</td>
<td>Physical Time</td>
<td>s</td>
</tr>
<tr>
<td>$N_S$</td>
<td>Numerical Step</td>
<td>-</td>
</tr>
</tbody>
</table>
CME cone parameters based on some statistical approach) to provide a probabilistic forecast for CME arrival time. The studies by Lee et al. (2013) and Mays et al. (2015) provide a look at two different approaches to mitigating uncertainties with ensemble modeling. While the approaches are different, they both seek to provide more realistic forecasts of arrival time through statistical means.

The Lee et al. (2013) study utilizes an ensemble of eight WSA/Enlil + Cone runs to generate a probabilistic forecast. A cone software routine (utilizing the Xie et al. (2004) method) developed at the NOAA/SWPC is used to calculate cone parameters in a way designed to capture different possible cone configurations. The authors utilized the SOlar and Heliospheric Observatory (SOHO) Large Angle and Spectrometric COronagraph (LASCO) C3 coronagraph images for the primary cone fit. The study also utilized Solar TErrestrial RElations Observatory (STEREO) B COR2-B coronagraph images to constrain the radial distance of the cloud and the cone angle. For each ensemble set member, a cone fit was conducted with a fresh initialization of the cone tool to ensure that any previous fit parameters were not transferred to the next member. The fit was conducted in eight different fitting sessions utilizing slightly different values for angular width and leading edge distance (guided by the STEREO observations). For the study in question, the resulting ensemble set generated fits with central origins varying by one to three degrees in both latitude and longitude. The study consistently produced early CME arrival times, leading the authors to explore the effects of background solar wind as a contributing factor. They concluded that predicting shock arrival times accurately depends not only upon the input cone CME parameters but also on the reliable specification of the background solar wind.

Mays et al. (2015) describes another method for real-time ensemble modeling utilizing a software tool, developed by the CCMC for real-time CME analysis, called Stereoscopic CME Analysis Tool (StereoCAT). This method utilizes stereoscopic triangulation to produce a cone fit from two different fields of view (e.g. STEREO A and STEREO B). Utilizing the StereoCAT system in “ensemble mode”, a forecaster measures CME leading edge height and angular width. The graphical interface is reset, and the measurements are repeated a number of times. These measurements
are combined in all possible combinations to generate the probabilistic ensemble set, resulting in \( n = m^2 \) cone fittings, where \( m \) is the number of independent measurements made. The ensemble set is then used to produce \( n \) WSA/Enlil + Cone model runs, resulting in a distribution of \( n \) predicted arrival times at the location of interest. The study showed that for 47% (8 out of 17) of the ensemble runs that generated impacts (of the 30 ensembles conducted), the observed CME arrival time was within the spread of ensemble predictions. The conclusion was drawn that the initial distribution of CME input parameters has significant influence on the accuracy of a CME prediction.

With both methods, the useful information obtained is a spread of possible arrival times based on the repeated “best guesses” of the forecasters. Both methods rely on independent cone fittings (no fixed variables), and do not fully analyze the behavior and sensitivity of individual cone properties (i.e. central origin, cone angle, velocity), or the sensitivity to variation within different portions of the domain space. The CME Impact Domain Analysis (CME IDA) method is an attempt to provide a means of examining the sensitivity of individual parameters, through the entire impact domain, utilizing the distribution of associated effects within the domain. As a starting point, the method is presented by examining the sensitivity of variation in central origins, as this parameter has been identified through forecasting experience to be one of the more important parameters (Millward et al., 2013).

### 2.3 Analyzing Effects of Uncertainty with CME Impact Domain Analysis

To gain a firm understanding of how uncertainties in CME central origins affect forecast results, it is important to understand how they affect the propagation of a CME through the Enlil model itself. Furthermore, it is necessary to understand how this propagation affects the potential geoeffectiveness of the event. To facilitate this understanding, a method was developed for analyzing the entire CME impact domain. The following sections outline, through example, how to construct a CME IDA. The discussion begins with the selection of the 06 October 2013 CME event, then explores the concept of impact definition. From there the mechanics of constructing the impact
domain are discussed. Finally, utilizing the discussion of solar wind - magnetospheric state variable coupling functions presented in Newell et al. (2007), an analysis of the domain, and the products produced, is presented.

2.3.1 The 06 October 2013 CME Event

As a starting point for this analysis, the 06 October 2013 CME event was selected. This event has the advantage of being the only Earthward-directed CME in the simulation time-frame, providing a relatively clean heliosphere. The event has also been studied by other researchers, providing additional information on the nature of the event (e.g. Foster et al. (2015); Hudson et al. (2015); Mays et al. (2015)).

Following the selection of the event, it was modeled utilizing the cone profile produced by the CCMC/Space Weather Research Center (SWRC) at the time the CME occurred (the cone profile is published in the CCMC Space Weather Database of Notifications, Knowledge, Information (DONKI)). Figure 2.3 shows a snapshot of the CCMC Runs on Request (RoR) results for this simulation. The figure presents views of the heliosphere in the ecliptic plane (a), the meridional plane (b), and on the 1 AU shell (c). Image (d) provides a comparison between observational data and the model results at Earth. This cone profile called for a CME ejection velocity of 790 km/sec, a cone half-angle of 25.0°, central origin longitude/latitude of 6.0°/−15.0° and a 21.5R⊙ crossing time of 18:19 UTC. These values were utilized as a baseline for all model runs within the impact domain. As an interesting side note, the Mays et al. (2015) ensemble study produced median parameters for this event of: Velocity = 747 km/s; Latitude = 1°; Longitude = 2°; Cone half-angle = 16°. This resulted in a CME transit time of 53.0 hours, compared to the CCMC-based prediction of about 76 hours, providing additional evidence to the uncertainty of cone fittings.

2.3.2 Defining an Impact

The definition of an impact is central to the results of two primary CME IDA analyses: *time to impact* and *impact duration*. Depending on how an impact is defined, different information will
Figure 2.3: Global view of the 06 October 2013 CME event, shown at 0600 on October 10: WSA-Enlil + Cone scaled density is shown in (a) constant Earth Latitude plane, (b) meridional plane of Earth, and (c) the 1 AU shell. CME cone model parameters are from the CCMC/SWRC event prediction (Velocity: 790 km/s; cone half-angle: 25.0°; lon/lat: 6.0°/−15°; 21.5 $R_\odot$ crossing: 18:19 UTC). Panel (d) compares the simulation (blue) compared to the measured (red) density values at Earth. The dashed blue line represents the background solar wind density, and the yellow shaded region indicates the CCMC/SWRC prediction for CME impact.
Figure 2.4: Illustration of the impact definition utilized in this study. The top panel provides normalized traces for $DP$ (gray line) and $\frac{dDP}{dt}$ (dot-dash red line). The defined impact region is shown in the shaded area. The bottom panel provides the relationship of the impact region to the change in solar wind parameters. The plot is normalized by $\frac{Y}{Y_{max}}$ where $Y$ is the original value for the trace, and $Y_{max}$ is the original maximum (provided in the plot legend).
be presented in these analyses. This concept is illustrated with Figure 2.4; an illustration of how
this study defines an impact. The top panel provides normalized traces for $DP$ (the cloud tracer)
and $\frac{dDP}{dt}$ (the rate of change of the cloud tracer). The defined impact region is shown in the shaded
area. The bottom panel provides the relationship of the impact region to the change in density,
radial velocity, and $|B|$. The plot traces are normalized by $\frac{Y}{Y_{\text{max}}}$ where $Y$ is the original value for
the trace, and $Y_{\text{max}}$ is the original maximum value. The values for $Y_{\text{max}}$ are provided in the plot’s
legend. It can be seen that the impact definition (shaded yellow region) reveals only a subset of
the information presented by the model runs. This information is reflected in the distributions
for arrival time and contact duration. This study utilizes the entire disturbance area, defined as
$\Delta t_{\text{da}} = t_{df} - t_{di}$ (where $t_{df}$ is the time of last disturbance, and $t_{di}$ is that of the first disturbance),
for solar wind CME IDA plots, as well as for quartile plots, so as to look at effects such as pileup
and rarefaction. The region of pileup that precedes a CME can clearly be identified in Figure 2.4
by the time-gap between the cloud tracer and the solar wind tracers. The rarefaction region can
be identified where the change in solar wind traces fall below zero. A more focused study of just
the impact region is possible by adjusting the $\Delta t_{\text{da}}$ definition.

For the remainder of this paper, the impact will be defined based on the cloud tracer provided
in the Enlil output. The rate of change of the cloud tracer is used to identify the upward trending
inflection point of $DP$ (the maximal increase). We consider the CME to impact the target when
$DP' > 0$ and $DP'' = 0$. To eliminate the outlying runs that would be tagged as an impact, but have
little or no impact on the solar wind, we restrict the impact set to runs where $\max(DP) > 1 \times 10^{-5}$.
This threshold value was chosen somewhat arbitrarily based on observed trends in the data, however
at only 0.000039% of the maximum cloud tracer value for the domain, it does reduce the impact
set to a meaningful domain. For simplicity, we define the end of CME contact to be the time at
which $DP$ falls back below the value of $DP$ at impact.

The authors acknowledge that there are other definitions of an impact. Mays et al. (2015), for
instance, define an impact as a sharp increase in modeled solar wind dynamic pressure. Lee et al.
(2013) define an impact as the first steep rise in the solar wind speed and magnetic field. From
these definitions it is clear that a standard definition of an impact is not in use. The method used in this paper utilizes the same idea (sharp increase in a solar wind parameter), but applies it to the Enlil variable responsible for tracing the CME ejecta cloud. It is argued that this definition, while different from others, is suitable to demonstrate the method being described. It is not expected that the choice of a different reasonable impact definition will change the effectiveness of the method.

2.3.3 Finding and Populating the Impact Domain

With the nature of an impact defined, it is fairly straightforward to find the boundaries of the impact domain. As mentioned in Section 2, the impact domain is defined as the set of Enlil runs where an impact is generated at the given target location. The free variables in this study are those that define the central origin of the CME (Latitude and Longitude). The impact domain is thus an abstract 2-Dimensional domain with Solar Latitude as one dimension and Solar Longitude as the second dimension.

Looking at Figure 2.5, the nature of the impact domain can be clearly seen. The plots in this figure are Impact Domain Plots. They are used to display information related to the domain as a whole. In this study, the Y-axis denotes the Solar Latitude, and the X-axis denotes Solar Longitude. The color bars indicate the values of the information being plotted. Contour lines are used to provide a better visual perspective of the distribution within the domain. Figure 2.5 (a) provides the distribution of \( \text{max}(DP) \) (The maximum value of the cloud tracer at Earth for each Enlil Run) over the impact domain for the 06 October 2013 event. The values are provided in percentages of the maximum, with the maximum (the \( \times \) symbol on the plot) located at \((0^\circ \text{lon}, 5^\circ \text{lat})\), which is essentially along the Sun-Earth line (within the resolution of the grid). Image (b) presents \( \text{max}(\frac{DP'}{dt}) \) (the maximum value of \( DP' \) at Earth for each Enlil run) within the impact domain, and is also plotted as a percentage of maximum, which is located at grid coordinates \((-5^\circ, 10^\circ)\).

Although the plots in Figure 2.5 were generated from the results of the IDA, they serve as a good reference for defining the domain boundaries. Recalling that the definition of impact utilized relies on the cloud tracer and its derivative, the extent of the impact domain can be seen with these
Figure 2.5: Impact domain plot (a) provides the distribution of $\max(DP)$ over the impact domain and plot (b) shows the distribution of $\max\left(\frac{dDP}{dt}\right)$ (both in % of maximum). The dots located at 6.7° latitude, 0° longitude represent the Sun-Earth line. The × identifies the maximum values within the domain.
plots. Plot (a) shows how the lower bound on $DP$ restricts the domain, while plot (b) provides the distribution of “impact intensity”, where the intensity is a measure of the fastest positive changing cloud tracer value.

To locate the impact domain boundaries, a simple linear search of the domain space can be conducted. For this study, runs were performed at $15^\circ$ intervals along the $+X$ axis of the domain until a CME miss was generated. Once a miss was generated, a search at $5^\circ$ intervals was conducted between the last CME hit and the first CME miss from the $15^\circ$ search space. The first miss in the $5^\circ$ interval space was then defined as the outer boundary in the $+X$ direction. The domain was then defined as a box around the origin at the distance determined from the search. Verification-runs were conducted in the $+Y$, $-Y$, and $-X$ directions to verify the extents of the domain. The size of the domain was increased as necessary.

Determining the spatial resolution for the domain is dependent upon the resolution of the Enlil Model. This study utilizes low-resolution Enlil runs, which provide a grid resolution of $256 \times 30 \times 60$. This translates to a $4^\circ$ separation between cells. To ensure a meaningful result from the simulation, and avoid excessive computational expense, a commensurate spatial resolution of $5^\circ$ was selected for this study.

Utilizing this resolution to define the free variables, and using outer boundaries of $-50^\circ$ and $50^\circ$ in both the latitude and longitude directions (as determined in the domain search), a RoR ensemble job was requested utilizing the forecast information from the 06 October 2013 CME for static parameters. This generated a total of 441 low-resolution WSA/Enlil v2.8f runs to populate the domain.

Populating the impact domain was a fairly straightforward task, thanks to a custom ensemble job submission form created by CCMC for the purpose of this research (the submission form is now available as part of the RoR system on the CCMC website). This RoR ensemble submission form allows for the specification of fixed parameters as well as providing a method for defining what parameters to vary, and how to vary them, between runs. It was a simple task to specify the domain conditions within the form and submit the jobs.
2.3.4 Solar Wind Impact Domains and Quartiles

Presentation of information within the impact domain is handled through two styles of plots. First, information is plotted on Impact Domain Plots (IDPs). Second, time-series statistical information is presented, in a style similar to superposed epoch analysis, on quartile plots.

IDPs take the form of a 2-Dimensional grid representing solar longitude on the X-axis and solar latitude on the Y-axis. Each node of the grid represents the cone model central origin for a model run. Values at the nodes represent model and/or calculated values at the target location (e.g. Earth), and are plotted against a color-bar to represent ranges. Contour lines are added to provide a visual perspective to the distribution of results across the domain.

Time dependent information is plotted in time-series quartile form. Conceptually, these plots are generated by arranging the time series data into a two-dimensional array with run number on the Y-axis and time on the X-axis. The X-axis is epoch-aligned to impact time and the array is sorted over the Y-axis, placing the largest values at the top of the array, and the smallest values at the bottom of the array. The quartile time-series lines can then easily be defined as the 25%, 50%, and 75% points on the Y-axis. The minimum value series resides at the bottom of the array, while the maximum value series resides at the top.

2.3.4.1 Impact Domain Plots for the 06 October 2013 Event

Two major items of interest in space weather forecasting are the CME arrival time at Earth (when to expect a storm), and the associated CME contact duration (length and strength of a storm). These categories are ideally suited to IDPs, and their distributions for the 06 October 2013 CME are provided in Figure 2.6. Plot (a) provides the distribution of time to CME impact at Earth. This time to impact is defined as $\Delta t = t_{1AU} - t_{21.5R_\odot}$ (the time from when the CME crosses 21.5$R_\odot$ to when it reaches 1AU; the impact time can instead be calculated in terms of absolute arrival time based on CME onset time, if desired). The distribution indicates that the earliest arrival is, to the resolution of the CME IDA, near the Sun-Earth line, with central origin
Figure 2.6: Impact Domain Plots showing the central origin uncertainty distributions for time to impact and CME Contact Duration at Earth for the 06 October 2013 CME event. The X-axis displays the central origin longitude and the Y-axis, latitude. Each node represents a single run’s CME central origin. The • located at 6.73° latitude, 0° longitude represents the Sun-Earth line at the time of the event. The × identifies the minimum time to impact within the domain (a) and the maximum CME contact duration (b).
Table 2.2: Minimum and Maximum values, and their central origin coordinates, for the Impact Domain Plots

<table>
<thead>
<tr>
<th>Property</th>
<th>Central Origin</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min Time to Impact</td>
<td>(0, 5)</td>
<td>76.0 (hours)</td>
</tr>
<tr>
<td>Max Impact Duration</td>
<td>(10, -15)</td>
<td>43.2 (hours)</td>
</tr>
<tr>
<td>Min Δ Density</td>
<td>(-15, 20)</td>
<td>-10.5 (N/cm³)</td>
</tr>
<tr>
<td>Max Δ Density</td>
<td>(-5, 10)</td>
<td>19.8 (N/cm³)</td>
</tr>
<tr>
<td>Min Δ Velocity</td>
<td>(10, 10)</td>
<td>-5.9 (km/s)</td>
</tr>
<tr>
<td>Max Δ Velocity</td>
<td>(0, 5)</td>
<td>78.6 (km/s)</td>
</tr>
<tr>
<td>Min Δ</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Max Δ</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Max DP</td>
<td>(0, 5)</td>
<td>0.255 (unitless)</td>
</tr>
<tr>
<td>Max dDP/dt</td>
<td>(-5, 10)</td>
<td>6.982e-03 (unitless)</td>
</tr>
</tbody>
</table>

grid coordinates of (0°, 5°) and an impact time of 76.0 hours. It is also clear that the distribution is not symmetric. Impact times are distorted across the domain, likely due to interactions of the CME with the ambient solar wind. The gradient of the arrival time is steepest toward the left of the plot, indicating that, for this event, uncertainties between 0° and −40° solar longitude will have a more profound affect on this forecast than would uncertainties toward +40° longitude.

Plot (b) provides the distribution of CME contact times across the domain as it propagates through space. Clearly the impact duration is asymmetric. The maximum occurs at central origin (10°, -15°), with a contact duration of 43.2 hours. The distortions in the contact duration distribution are likely due to the changing morphology of the CME as it interacted with the ambient solar wind during its transit to 1AU. The image in Figure 2.1 shows the general morphology (for this study) of a CME in transit. An extended limb can be clearly seen along the top and another down the left side of the CME cloud; features such as these would produce much longer contact time should they pass directly through the target location.

The need to understand the impacts of space weather at Earth is extremely important in space weather forecasting. Utilizing IDPs, information regarding the distribution of potential geoeffectiveness at Earth can be inferred. Going beyond time to impact and impact duration, Figure 2.7 graphically presents distributions of solar wind perturbations at Earth. Each plot represents
Figure 2.7: Impact Domain Plots showing the central origin uncertainty distributions for the Minimums and Maximums of solar wind properties: (a) & (b) ∆ Density; (c) & (d) ∆ Radial Velocity; (e) & (f) ∆ Total Magnetic Field (|B|). Values are minimum/maximum over the course of the event.
either the minimum or maximum value (over the course of the solar wind disturbance, $\Delta t_{da}$) of the property being displayed. Utilizing coupling functions, such as those found in Newell et al. (2007), it is possible to illicit understanding of the geoeffectiveness a CME.

Newell et al. (2007) discusses 20 Solar Wind-Magnetosphere Coupling functions and how well they correlate solar wind conditions to 10 magnetospheric state variables. A function is provided that correlates to 9 of the 10 state variables (the only exception is Dst, for which the Temerin and Li (2006) coupling function correlates best). The coupling functions can be used to infer geoeffectiveness of the event being examined. The Newell et al. (2007) function is given as

$$\frac{d\Phi_{MP}}{dt} = v^{4/3} B_T^2 \sin^{2/3} \left( \frac{\theta_C}{2} \right)$$

(2.1)

where $\frac{d\Phi_{MP}}{dt}$ is the rate magnetic flux is opened at the magnetopause, $v$ is the bulk solar wind speed, $\theta_C$ is the interplanetary magnetic field (IMF) clock angle, given by $\tan^{-1}(B_y/B_z)$ and the perpendicular component of the magnetic field (in GSM coordinates) is given by $B_T = (B_y^2 + B_z^2)^{1/2}$.

To facilitate a discussion on possible geoeffectiveness of the 06 October 2013 CME, Figure 2.7 presents the minimums and maximums for three of the solar wind properties: change in Density over background ($\Delta$ Density), change in radial velocity over background ($\Delta$ Radial Velocity), and change in total magnetic field over background ( $\Delta |B|$). The plots on the left hand side of the figure represent the maximums, and the plots on the right present the minimums. The maximum and minimum disturbances within the Impact Domain are denoted with the symbol $\times$, and the location of the Sun-Earth line is denoted with symbol $\circ$.

Plots (a) and (b) display the distributions for minimum and maximum disturbances to Number Density (N/cm$^3$). Though density does not figure directly into the Newell et al. (2007) coupling equation, it does figure weakly into the Temerin and Li (2006) coupling function which holds the best correlation to Dst. Plot (a) represents the maximum compression in density, and plot (b) indicates the maximum rarefaction (minimum $\Delta$ density). It can be seen from these plots that both the maximum and minimum density disturbances occur away from the Sun-Earth line, with the maximum rarefaction occurring at a considerable angular separation. The distribution for the
maximum density disturbance is slightly elongated toward the lower right of the plot, with a steeper gradient toward the upper left. The minimum density disturbances also show the elongation trend to the lower right, however there is a large stable central region within the distribution. Uncertainties in CME central origin around and toward the upper left of the plot appear to produce little variability from +5° to -20° longitude and -5° to +20° latitude. The magnitudes of both the minimum and maximum density disturbances are also similar; the disturbances minimize at -10.5 N/cm³ and maximize at 19.8 N/cm³. Given the coupling functions, it is unlikely that the distribution in changes of density will, by themselves, cause much change in geoeffectiveness of the CME.

The radial velocity effects (shown in plots (c) and (d)) have a much larger input into the coupling functions, and thus provide us with a better view of a distribution of effects. The velocity distribution peaks near the Sun-Earth line (grid location (0°, 5°)) with a ∆ Velocity of 78.6 km/s, and minimize at grid location (10°, 10°) with a ∆ Velocity of -5.9 km/s. The distribution of velocity perturbations is nearly symmetrical for maximum values, but are extremely distorted to the right for minimum values. At first glance it appears that the negative variations are extreme, however the values of negative perturbations are an order of magnitude lower than the positive, making the negative distribution less significant.

The Minimum and Maximum ∆|B| distributions can only provide a limited amount of information due to the lack of internal IMF in the modeled CME, though, as is suggested in Mays et al. (2015), it is possible to illicit some information utilizing the magnitude of the magnetic field and looking at a representative selection of IMF clock angles. For the sake of completeness, the ∆|B| IDA distributions are presented in plots (e) and (f), with maximum ∆|B| peaking at a central CME origin of (-5°, 10°) and a value of 7.8 nT. The distribution is slightly compressed to the left, with the maximum residing away from the Sun-Earth Line. The minimum values distribution presents itself as kidney-shaped with a large stable region near the center, with the minimum value at central origin (0°, 10°) with a value of -2.3 nT. The steepest gradients, and the CME origins of greatest uncertainty appear to be toward the top and left of the plot, though considerable vari-
ability presents itself in other regions. The magnitude of change in the negative direction is about one-third of the magnitude of disturbance in the positive direction.

### 2.3.4.2 Quartile Time-series Plots for the 06 October 2013 Event

To quantify the effects of an event, it is also necessary to examine the trends of the impact domain over time. Figure 2.8 provides a quartile breakdown for the same three solar wind properties as with the IDPs. Plots on the left-hand side of the figure are rendered with the background solar wind conditions intact. Plots on the right-hand side have the background conditions removed, isolating the behavior of the disturbance. Each plot distinctly outlines the envelopes of the disturbance, with the min/max envelope depicted with the dashed red lines. The 25%/75% envelope is outlined with the blue dash-dot lines. The median is provided with a solid black line. Plots (a) and (b) provide number density information (N/cm$^3$); plots (c) and (d) portray radial velocity activity (km/s), and plots (e) and (f) depict the total magnetic field (|B|, nT).

Looking at the density envelope plots, (a) is most useful when comparing the envelopes to observations of an event, or when determining trends of the full solar wind. With the background conditions removed, as in plot (b), trends of changes due to the CME are more apparent. The density pile-up region is well defined, and ranges from no pile-up in the minimum case to a pileup of 20 N/cm$^3$ in the maximum case. The pileup begins as early as 40 hours prior to CME impact, with the median pileup beginning roughly 10 to 15 hours prior to impact. Looking at plot (b), we can identify the rarefaction region from the CME passage easily as the trends drop below zero (0). It is interesting to note that nearly 75% of all impacts converge toward zero between 15 and 20 hours after initial impact. The earliest rarefaction trends start as early as impact, and 25% of impacts see little to no rarefaction. The solar wind radial velocity, depicted in plots (c) and (d) indicate that 25% of starting origins produce impacts between a small loss in velocity and small gain in velocity. The median $\Delta$ velocity peaks around 5 km/s, leaving around 90% of the velocity variation to 50% of the starting origins. Within the constraints of the available magnetic field information, |B| field trends, shown in plots (e) and (f), are very similar to the Density trends. There exists a build-up
Figure 2.8: Solar Wind properties (at Earth) quartile envelopes for the 06 October 2013 CME event: Density (a & b); Radial Velocity (c & d); Total Magnetic Field $|B|$ (e & f). Plots (a, c, e) provide the information with the background values intact while plots (b, d, f) provide the quartile information with the background values removed. The red lines represent the min/max envelope; The blue lines, the 25%/75% envelope; The solid black line represents the median.
trend prior to impact, and there is a convergence where 50% of runs approach zero disturbance of the total magnetic field. The rarefaction trends are also similar, with the earliest starting near impact, and about 25% seeing none. Without accurate magnetic field information for the CME ejecta cloud, however, conclusions about $|B|$ field trends must be treated with assumptions to extract any meaningful information.

2.4 Conclusions

Examining the distribution of possible CME effects utilizing CME Impact Domain Analysis can provide insight into how uncertainties in CME origination conditions can affect a space weather forecast. Through examination of distributions of impact time, impact duration, and changes in solar wind conditions at Earth, a clear visual can be created to highlight the CME central origin regions of greatest uncertainty in a forecast. In this study it was shown that the central origin can have a significant impact on solar wind conditions at Earth, and this, in turn, can have a non-trivial impact on the geoeffectiveness of the event.

The CME IDA method provides a means to explore the uncertainty in the forecast through impact domain plots, which visualize the impact domain distribution spatially, and through time-dependent quartile plots. These plots can be used as a basis for additional investigations as well: One could potentially integrate the Newell et al. (2007) function over the impact duration generate a total flux. Plotting the information on an IDP would provide a distribution of CME “effect magnitude”.

The quartile plots, taken as a whole, can provide information about the propagation of uncertainties through the model framework. For the analyses presented here, the total error due to uncertainties in CME central origins can be bound by the minimum/maximum envelope. Comparisons of the quartile plots to observations could provide further insights to the performance of a given forecast.

Overall, it may take a bit of investigation to illicit full meaning of the distributions presented in this paper, but they do provide a compelling glimpse as to how the uncertainties in central origin
can affect a space weather forecast. Additional work will be necessary to determine if general trends can be inferred from Impact Domain Plots such as these.

The process of generalizing the CME IDA should begin with extending the method to analyze the sensitivity of other dimensions in the cone parameter space (i.e. velocity, cone angle). From there the process of generating a set of baseline analyses for a CME injected into ideal steady-state solar wind conditions should be undertaken. Building upon the ideal case, a database can be built by systematically conducting analyses with one type of background solar wind phenomenon at a time. From this database, a correlation study could explore the possibility of generating an analytic model to determine how various background conditions affect the ideal solution. At that point, the study could be extended to explore the possibility of developing statistical models of forecasts based on the ambient solar wind conditions and the estimated cone parameters. If successful, this would provide a tool to more accurately bound uncertainties in all forecasts.
Chapter 3

Adiabatic Invariant Methods and Error Analysis in a Dipole

Ordering observed particles in the magnetosphere is important to understanding how processes in the radiation belts work. The three adiabatic invariants $M$, $J$, and $\Phi$ provide and important part in this ordering. $M$ represent the magnetic flux contained within the gyro-obit of a particle. $J$ represent the length of travel along a field line that a particle travels. Its companion invariant, $K$, is used to define the “drift shell” in which a particle is trapped. $\Phi$ is the flux contained within the drift shell defined by $K$, and is used to determine the radial distance, $L^*$, that a particle would cross the equator if all of the external components of Equation (1.18) were slowly relaxed to zero. (Roederer, 1970; Roederer and Zhang, 2014)

This chapter focuses on the first iteration of algorithms designed to calculate the adiabatic invariants of particles trapped within the Earth’s magnetosphere. These algorithms are designed to function with non-gridded magnetic field models, and provide a framework for error analysis of the gridded methods presented in Chapter 4. Section 3.1 reviews the equations for the adiabatic invariants, and provides the basic methodology necessary for computing the solutions numerically. Section 3.2 looks at the error responses of the algorithms with respect to the ideal solutions on a dipole, for the algorithms where analytic solutions exist.

3.1 Invariant Algorithms

This section is broken down into three parts. Each part deals with a single invariant and the methods necessary for calculation. Even though the first invariant is a simple analytic equation,
it is provided here for the sake of completeness in the discussion. Following a description of the
invariant, each subsection will look at the algorithms and methodologies required to produce the
solution on a non-gridded magnetic field model. These methods are utilized in Section 3.2 to
investigate the potential errors involved.

3.1.1 Computing $\mu$

Recall from Chapter 1 that the first adiabatic invariant is given by:

$$\mu = \frac{p^2}{2m_0 B} = \frac{p^2 \sin^2 \alpha}{2m_0 B}$$

It can be seen from this equation that $\mu$ can be solved directly, and thus does not require any
complicated algorithms to calculate.

3.1.2 Computing $K$

Again, recall from Chapter 1 that the second adiabatic invariant is given by $J$, which is given
by:

$$J = \oint p_{||} ds = m \oint v \mu ds. \quad (3.1)$$

Roederer and Zhang (2014) states that this integral is taken along the guiding field line for a
complete bounce cycle ($\oint = 2 \int_{s_m}^{s_m}$). With the specific case of equipotential field lines, $v$ is constant
during one bounce. Taking into account the definition of the purely field-geometric quantity $I$ (3.2),
the equation in (3.3) can be derived.

$$I = \int_{s_m}^{s_m} \sqrt{1 - \frac{B(s)}{B_m}} ds \quad (3.2)$$

$$J = 2pI \quad (3.3)$$
When dealing with a more realistic magnetic field, under the influence of external forces perpendicular to the magnetic field lines, the integral $I$ will no longer be adiabatically invariant due to the fact that $p$ is changing. To obtain a value that remains invariant, it is necessary to combine the invariance of $\mu$ (through the magnetic field mirror strength $B_m$) with (3.3) to arrive at the equation in (3.4).

$$K = \sqrt{B_m} I = \int_{s_m}^{s'_m} \sqrt{B_m - B(s)} ds$$ (3.4)

where $B_m$ is calculated by:

$$B_m = \frac{p^2}{2m_0\mu} = \frac{B}{\sin^2 \alpha}$$ (3.5)

The algorithms in this section, and the remainder of this work, will utilize $B_m$ and the invariant $K$ for computational purposes.

### 3.1.2.1 Methodology for Calculating K

There are three basic steps involved in calculating the second adiabatic invariant. First, the guiding field line of the particle must be traced. Second, the field line position/value pairs are re-sampled to aid in the integration of $K$, and finally $K$ must be integrated between mirror points (points of equal $B_m$).

To satisfy the first requirement, field lines, which are instantaneous streamlines within a magnetic field, can be calculated by solving the initial value ODE given in Equation (3.6).

$$L(0) = x_0, \quad \frac{dL(u)}{du} = v(L(u))$$ (3.6)

For the prototype implementation, the ODE solver chosen is the Runge-Kutta-Fehlberg Fehlberg (1969) 4th order method with automatic time-step adjustment in the 5th order. This method uses the 6 equations in (3.7) to generate 4th and 5th order approximations.
\[ k_1 = h f(x_k) \]
\[ k_2 = h f\left(x_k + \frac{1}{4} k_1\right) \]
\[ k_3 = h f\left(x_k + \frac{3}{32} k_1 + \frac{9}{32} k_2\right) \]
\[ k_4 = h f\left(x_k + \frac{1932}{2197} k_1 - \frac{7200}{2197} k_2 + \frac{7296}{2197} k_3\right) \]
\[ k_5 = h f\left(x_k + \frac{439}{216} k_1 - 8 k_2 + \frac{368}{513} k_3 - \frac{845}{4104} k_4\right) \]
\[ k_6 = h f\left(x_k - \frac{8}{27} k_1 + 2 k_2 - \frac{3544}{2565} k_3 + \frac{1859}{4104} k_4 - \frac{11}{40} k_5\right) \]

The actual 4th and 5th order approximations are constructed utilizing the equations below. The 4th order solution is given in Equation (3.8) and the 5th order solution is given in Equation (3.9)

\[ x_{k+1} = x_k + \frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4101} k_4 - \frac{1}{5} k_5 \]  

\[ z_{k+1} = x_k + \frac{16}{135} k_1 + \frac{6656}{12825} k_3 + \frac{28561}{56430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_6 \]

The step size can be adjusted every time step to an optimal value \((h_{\text{opt}})\) through a comparison of the 4th and 5th order solution to examine the truncation error. Specifying a maximum error tolerance, \(\tau\), allows us to adjust the time-step \(h\). This is given by Equation (3.10) and (3.11).

\[ s = \left(\frac{\tau h}{2|z_{k+1} - x_{k+1}|}\right)^{\frac{1}{4}} \approx 0.84 \left(\frac{\tau h}{|z_{k+1} - x_{k+1}|}\right)^{\frac{1}{4}} \]

\[ h_{\text{opt}} = s h \]

The basic approach for this method is presented in Algorithm 1 below.

This approach does a decent job finding the proper path in a dipole, as is shown in the error analysis in Section 3.2. It is of note that the field line trace should be terminated at the surface of the Earth (or just below to aid in calculations) for the methods in this section. When dealing with
Algorithm 1 Runge-Kutta-Fehlberg Method

1: procedure RK45TRACE(dir = forward, ib = 0.9, max_steps = 2000, x0)
2: \( x \leftarrow x_0 \)
3: \( \text{rad\_dist}_x \leftarrow \text{magnitude}(x_0) \)
4: \( \text{inner\_boundary} \leftarrow ib \)
5: \( path \leftarrow \text{empty array} \)
6: \( value \leftarrow \text{empty array} \)
7: \( step \leftarrow 0 \)
8: while \( \text{rad\_dist}_x > \text{inner\_boundary} \) do
9: \( s \leftarrow 0 \)
10: while \( s \) is not close to 1.0 do
11: \( k_1 \leftarrow \text{from Equation (3.7)} \)
12: \( k_2 \leftarrow \text{from Equation (3.7)} \)
13: \( k_3 \leftarrow \text{from Equation (3.7)} \)
14: \( k_4 \leftarrow \text{from Equation (3.7)} \)
15: \( k_5 \leftarrow \text{from Equation (3.7)} \)
16: \( k_6 \leftarrow \text{from Equation (3.7)} \)
17: \( x_4 \leftarrow \text{from Equation (3.8)} \)
18: \( x_5 \leftarrow \text{from Equation (3.9)} \)
19: \( s \leftarrow \text{from Equation (3.10)} \)
20: \( h \leftarrow sh \)
21: \( x \leftarrow x_4 \)
22: \( val_x \leftarrow \text{value}(x) \)
23: \( \text{rad\_dist}_x \leftarrow \text{magnitude}(x) \)
24: \( step \leftarrow step + 1 \)
25: if \( step > max\_steps \) then
26: break
27: append \( path \) with \( x \)
28: append \( value \) with \( val_x \)
return \( path, value \)
gridded data, the end point may not always be at a desirable location, as different models utilize
different inner boundaries. This problem will be addressed in Chapter 4 where modifications to
methods for gridded data are presented.

With the field line path established, the second step is to re-sample the trace so that, from
the minimum magnetic field value \( B_{\text{min}} \), both the north and south directions of the field line have
their locations defined by equal \( B \) values. In simpler terms, for every \( B \) value in the northern trace,
there should be an identical value in the southern trace, with its location dictated by this value. To
accomplish this, the prototype code utilizes a cubic-spline interpolation method. The procedure is
given in pseudo-code in Algorithm 2.

**Algorithm 2 Field Line Resample**

1: procedure FIELDLINERESAMPLE(traceForward, traceBackward)
2: \hspace{1em} northTrace ← traceForward
3: \hspace{1em} southTrace ← traceBackward
4: \hspace{1em} \( B_{\text{min}} \text{North} \) ← minimum of northTrace
5: \hspace{1em} \( B_{\text{min}} \text{South} \) ← minimum of southTrace
6: \hspace{1em} \( B_{\text{min}} \) ← minimum of \( B_{\text{min}} \text{North} \) and \( B_{\text{min}} \text{South} \)
7: \hspace{1em} recenter both traces on \( B_{\text{min}} \)
8: \hspace{1em} for \( B \) in northTrace do
9: \hspace{2em} interpolate new location in southTrace where \( B_{\text{south}} = B_{\text{north}} \)
10: \hspace{2em} append newSouth with new location/value pair
11: return northTrace, newSouth

Figure 3.1 provides the graphical result of re-sampling on a Tsyganenko T96 magnetic field.
This field is used for reference instead of a dipole, as the application of this process on a dipole
tends to produce symmetric results, making the re-sampling unnecessary.

Once the field line is re-sampled, it is a straightforward task to integrate for \( K \) (Equation 3.4).
This is accomplished by integrating from \( B_{\text{min}} \) to \( B_{\text{mirror}} \) in both the north and the south direction,
then adding the resultant values together. For the purpose of the prototype implementation, this
integration is done along the field line utilizing simple trapezoidal integration. The procedure for
calculating \( K \) is given in Algorithm 3.
Figure 3.1: This figure illustrates the results of re-sampling magnetic field lines. The south field trace is re-sampled to equal $B$ values of the north trace, thus allowing for more straightforward field line integration to determine $K$. This re-sampling is of a Tsygenenko T96 magnetic field (with dipole tilt). The field line being re-sampled is indicated by the blue and red dots along the line. The red dots signify the primary, and the blue dots signify the locations that were interpolated to provide the equal $B$ values. The black dots represent the $B_{min}$ on each field line. Bifurcated field lines can be easily identified on the sunward side. The black contour lines are of constant $|B|$.

### 3.1.3 Computing $L^*$

According to Roederer and Zhang (2014), in a dipole, the magnetic flux encompassed by a particle shell ($\Phi$) can be given by the equation in (3.12).

$$\Phi = \frac{2\pi k_0}{r} = \frac{2\pi B_E R_E^2}{L} = \frac{1.953}{L} \text{ Gauss} R_E^2$$  \hfill (3.12)

However, in a complex magnetic field, $\Phi$ must be numerically calculated from the flux integral as given in (3.13).

$$\Phi = \int \int \pi B \cdot dS$$  \hfill (3.13)
Algorithm 3 CalculateK

1: procedure CalculateK(traceForward, traceBackward)
2: \( K \leftarrow \) None
3: if field line is not bifurcated then
4: if \( B_{mirror} \) is in both field line traces then
5: \( K_f \leftarrow \text{trap_int} \) traceForward from \( B_{min} \) to \( B_{mirror} \) utilizing Equation (3.4)
6: \( K_b \leftarrow \text{trap_int} \) traceBackward from \( B_{min} \) to \( B_{mirror} \) utilizing Equation (3.4)
7: \( K \leftarrow K_f + K_b \)

return \( K \)

where \( \pi \) is the surface of the polar cap bounded by the drift shell field line intersection of the Earth’s surface.

Roederer and Zhang (2014) further states that, if the magnetic latitude of the drift trajectory for the particle is known, Stokes’ Theorem (Katz, 1979) can be utilized to transform the surface integral in (3.13) to the line integral in Equation in (3.14) to calculate \( \Phi \).

\[
\Phi \approx 2 B_E R_E^2 \int_0^{2\pi} d\phi \int_{\lambda_c(\phi)}^{\pi/2} \cos \lambda \sin \lambda \, d\lambda = B_E R_E^2 \int_0^{2\pi} \cos^2[\lambda_C(\phi)] d\phi \quad (3.14)
\]

where \( \lambda_C(\phi) \) is the dipole latitude of the intersection \( C \) at a given longitude \( \phi \).

Utilizing Equation (3.14) as a means of flux integration means very little on an ungridded dipole, as the solution reduces to the analytic solution for flux. It is, however, very useful in defining an analytic version of \( L^* \) that is dependent on only \( \lambda_C \). The definition of \( L^* \) (Equation 3.15) is the logical place to begin a derivation of equations for analyzing the numerical methods used to compute \( L^* \) on a field more complex than a dipole.

\[
L^* = \frac{2\pi B_E R_E^2}{\Phi} \quad (3.15)
\]

To perform a quality error analysis, the analytic form of \( L^* \), related to the field line intersection latitude (\( \lambda_C \)), is needed. To arrive at this analytic solution, start with substituting Equation (3.14) for \( \Phi \) in Equation (3.15) to arrive at Equation (3.16).

\[
L^* = \frac{2\pi}{\int_0^{2\pi} \cos^2[\lambda_C(\phi)] d\phi} \quad (3.16)
\]
When dealing with a dipole, the symmetric nature of the magnetic field reduces this to the analytic form in Equation (3.17).

\[ L_{\text{dipole}} = \frac{1}{\cos^2[\lambda_C]} \]  

(3.17)

With the analytic \( L^*(\lambda) \) function defined, the surface integration method can be examined.

### 3.1.3.1 Methodology for Calculating \( L^* \)

The first step for calculating \( L^* \) requires the finding of the particle drift trajectory. This relies on the invariant nature of the second invariant, \( K \), with a basic procedure that involves first tracing the guiding field line for the particle in question, and calculating \( K \) for the given \( B_m \) (or vice-versa). Next, field lines at other local-times around Earth must be found that conform to the same \( B_m/K \) pair. For the prototype system, this is done utilizing a basic bisection method to search for the field line in question. The procedure in Algorithm 4 describes how this is accomplished.

To find the entire drift trajectory (in a non-dipole field), more than one field line trace is needed. In a dipole, the symmetric nature of the field means that the entire drift trajectory can be defined by a single \( \lambda_C \), therefore only the initial trace is needed. In more complex fields, experimentation has shown that a minimum of 4 local times must be calculated (local times equating to midnight, dawn, noon, and dusk). More complex field geometries will be addressed, in greater depth, in Chapter 4.

Once the drift trajectory is known, the integration grid for the surface can be prepared, and the integration performed. Although Stokes’ Theorem eliminates the need for a full surface integration in many cases, there may still be times when the full surface integration is necessary. For the initial prototype, both the Stokes’ method and the full surface integration have been implemented. For the Stokes’ method, the boundary of the surface is integrated using a simple line integral. For surface integration, the grid is created as a simple subsection of a spherical shell. The grid is computed as is given in Algorithm 5.

With the grid defined, the integration can be performed. At a very basic level, a surface
Algorithm 4 Find New Drift Shell Field Line

1: procedure FINDNEWDRAFTHYLLINE($B_m$, $K$, $localtime$, $r_0$)
2:  $low_R_E \leftarrow 1.0$
3:  $start_unit \leftarrow$ Surface location at $localtime$
4:  $start_location \leftarrow start_unit \times r_0$
5:  $hi_R_E \leftarrow$ Radial Distance to Outer Boundary for $localtime$
6:  $new_R_E \leftarrow r_0$
7:  $B_{nl} \leftarrow$ None
8:  while $B_{nl}$ is None do
9:    $new_line \leftarrow$ field line originating at $start_location$
10:   $B_{nl} \leftarrow B_m$ from $new_line$ where $K_{nl} = K$
11:   if $B_{nl}$ is None then
12:      if $new_R_E$ is close to $low_R_E$ then
13:          return None
14:      $new_R_E \leftarrow 0.5 \times new_R_E$
15:      $start_location \leftarrow start_unit \times new_R_E$
16:  while $B_{nl}$ is not close to $B_m$ do
17:    $step \leftarrow$ Optimal Step Fraction () or 0.5 for bisection
18:    if ($B_{nl} < B_m$) or ($B_{nl}$ is None) then
19:      $hi_R_E \leftarrow new_R_E$
20:      $new_R_E \leftarrow low_R_E + step \times (hi_R_E - low_R_E)$
21:    else
22:      $low_R_E \leftarrow new_R_E$
23:      $new_R_E \leftarrow low_R_E + step \times (hi_R_E - low_R_E)$
24:    $start_location \leftarrow start_unit \times new_R_E$
25:    $new_line \leftarrow$ recompute field line with new $start_location$
26:    $B_{nl} \leftarrow B_m$ from $new_line$ where $K_{nl} = K$
27:  if $B_{nl}$ is None then
28:    if $new_R_E < r_0$ then
29:      while $B_{nl}$ is None do
30:        $redux \leftarrow 0.5 \times (hi_R_E - new_R_E)$
31:        $new_R_E \leftarrow new_R_E + redux$
32:        $new_line \leftarrow$ recompute field line with new $start_location$
33:        $B_{nl} \leftarrow B_m$ from $new_line$ where $K_{nl} = K$
34:      else
35:        while $B_{nl}$ is None do
36:          $redux \leftarrow 0.5 \times (new_R_E - low_R_E)$
37:          $new_R_E \leftarrow new_R_E - redux$
38:          $new_line \leftarrow$ recompute field line with new $start_location$
39:        $B_{nl} \leftarrow B_m$ from $new_line$ where $K_{nl} = K$
40:      return $new_line$
Algorithm 5 Defining the Integration Grid

1: procedure DEFINEPOLARGRID($R_E$, $num_{divisions}$, $base_{lat}_{fun}$)
2:     lon_points ← linear space from 0 to 360 with $num_{divisions}$ without endpoint)
3:     grid ← $num_{divisions} \times num_{divisions}$ array
4:     for lon in lon_points do
5:         lat_points ← linear space from $base_{lat}_{fun}(lon)$ to 90 with $num_{divisions}$
6:         for lat in lat_points do
7:             $\lambda$ ← lat in radians
8:             $\phi$ ← lon in radians
9:             grid[lat,lon] ← ($R_E$, $\lambda$, $\phi$)

return grid

The integral can be broken down into two parts: the area of a section, and the value of that section. More explicitly, a surface integral looks like:

$$\int_{\pi_{cell}} \int_{\pi_{cell}} = f(\pi_{cell}) \cdot a(\pi_{cell})$$

where $f(\pi_{cell})$ represents the cell value, and $a(\pi_{cell})$ the cell area.

The algorithm designed to calculate $\Phi$ takes advantage of this, allowing for testing of multiple different area and value functions. Algorithm 6 provides this procedure, and Algorithm 7 illustrates an area algorithm utilizing two triangles to calculate the area of a quadrilateral. A value function needs to have the ability, utilizing the same points arguments as the area algorithm, to return an optimal value for the given cell. Algorithm 8 provides an example that produces a cell value utilizing the average of the given point values.

The final step is to calculate $L^*$ itself. The procedure presented in Algorithm 9 combines the other two algorithms to generate a final $L^*$ value. The inputs to this algorithm are the location of the particle and its pitch angle, $\alpha$.

3.2 Error Response in a Dipole Field

The first invariant is an analytic calculation, and requires no analysis of errors. The second and third invariants, however, require numerical integration and thus should be analyzed to show their responses to error. Unfortunately, there is no analytic solution to the second invariant. The
Algorithm 6 Integrate Surface

1: procedure INTEGRATESURFACE(grid, valFun, areaFun)
2: \( \Phi \leftarrow 0 \)
3: lons \( \leftarrow \phi \) in grid
4: lats \( \leftarrow \lambda \) in grid
5: for index\( _\phi \) in range of lons do
6: if index\( _\phi \) + 1 is in range of lons then
7: \( i_1 \leftarrow \) index\( _\phi \)
8: \( i_2 \leftarrow i_1 + 1 \)
9: else
10: \( i_1 \leftarrow \) index\( _\phi \)
11: \( i_2 \leftarrow 0 \)
12: for index\( _\lambda \) in range of lats do
13: \( \vec{p}_{11} \leftarrow \) grid[lats\( _{i_1} \), lons\( _{i_1} \)]
14: \( \vec{p}_{12} \leftarrow \) grid[lats\( _{i_1} \), lons\( _{i_2} \)]
15: \( \vec{p}_{21} \leftarrow \) grid[lats\( _{i_2} \), lons\( _{i_1} \)]
16: \( \vec{p}_{22} \leftarrow \) grid[lats\( _{i_2} \), lons\( _{i_2} \)]
17: \( \Phi \leftarrow \Phi + \text{areaFun}(\vec{p}_{11}, \vec{p}_{12}, \vec{p}_{21}, \vec{p}_{22}) \times \text{valFun}(\vec{p}_{11}, \vec{p}_{12}, \vec{p}_{21}, \vec{p}_{22}) \)
return \( \Phi \)

Algorithm 7 Area Function using Triangles

1: procedure AREAFUN(\( \vec{p}_{11}, \vec{p}_{12}, \vec{p}_{21}, \vec{p}_{22} \))
2: \( \lambda_{11} \leftarrow \lambda(\vec{p}_{11}) \)
3: \( \lambda_{12} \leftarrow \lambda(\vec{p}_{12}) \)
4: \( \lambda_{21} \leftarrow \lambda(\vec{p}_{21}) \)
5: \( \lambda_{22} \leftarrow \lambda(\vec{p}_{22}) \)
6: \( \phi_{11} \leftarrow \phi(\vec{p}_{11}) \)
7: \( \phi_{12} \leftarrow \phi(\vec{p}_{12}) \)
8: \( \phi_{21} \leftarrow \phi(\vec{p}_{21}) \)
9: \( \phi_{22} \leftarrow \phi(\vec{p}_{22}) \)
10: \( r \leftarrow r(\vec{p}_{11}) \)
11: \( d\lambda_1 \leftarrow \lambda_{12} - \lambda_{11} \)
12: \( d\lambda_2 \leftarrow \lambda_{22} - \lambda_{21} \)
13: if \( \phi_{21} < \phi_{11} \) then
14: \( d\phi_1 \leftarrow (\phi_{21} + 2\pi) - \phi_{11} \)
15: else
16: \( d\phi_1 \leftarrow \phi_{21} - \phi_{11} \)
17: if \( \phi_{22} < \phi_{12} \) then
18: \( d\phi_2 \leftarrow (\phi_{22} + 2\pi) - \phi_{12} \)
19: else
20: \( d\phi_2 \leftarrow \phi_{22} - \phi_{12} \)
21: \( \text{area}_1 \leftarrow \frac{1}{2} (r^2 \cos \lambda_{11} \; d\lambda_1 \; d\phi_1) \)
22: \( \text{area}_2 \leftarrow \frac{1}{2} (r^2 \cos \lambda_{12} \; d\lambda_2 \; d\phi_2) \)
return \( \text{area}_1 + \text{area}_2 \)
Algorithm 8 Value Function using and Average of Points for $\vec{B} \cdot \hat{n}$

1: procedure \textsc{valueFun}(\vec{p}_{t11}, \vec{p}_{t12}, \vec{p}_{t21}, \vec{p}_{t22})
2: \quad $\vec{B}_{11} \leftarrow \text{value}(\vec{p}_{t11})$
3: \quad $\vec{B}_{12} \leftarrow \text{value}(\vec{p}_{t12})$
4: \quad $\vec{B}_{21} \leftarrow \text{value}(\vec{p}_{t21})$
5: \quad $\vec{B}_{22} \leftarrow \text{value}(\vec{p}_{t22})$
6: \quad $\hat{n}_{11} \leftarrow \vec{p}_{t11} / \text{magnitude}(\vec{p}_{t11})$
7: \quad $\hat{n}_{12} \leftarrow \vec{p}_{t12} / \text{magnitude}(\vec{p}_{t12})$
8: \quad $\hat{n}_{21} \leftarrow \vec{p}_{t21} / \text{magnitude}(\vec{p}_{t21})$
9: \quad $\hat{n}_{22} \leftarrow \vec{p}_{t22} / \text{magnitude}(\vec{p}_{t22})$
10: \quad $\text{dot}_{11} \leftarrow \vec{B}_{11} \cdot \hat{n}_{11}$
11: \quad $\text{dot}_{12} \leftarrow \vec{B}_{12} \cdot \hat{n}_{12}$
12: \quad $\text{dot}_{21} \leftarrow \vec{B}_{21} \cdot \hat{n}_{21}$
13: \quad $\text{dot}_{22} \leftarrow \vec{B}_{22} \cdot \hat{n}_{22}$
14: \quad \text{value} \leftarrow (\text{dot}_{11} + \text{dot}_{12} + \text{dot}_{21} + \text{dot}_{22})/4
\quad \text{return value}

Algorithm 9 Calculate $L^*$

1: procedure \textsc{calculateLstar}(\Phi, B_E = 3.15 \times 10^4, R_E = 1.0)
2: \quad $L^* \leftarrow \frac{2\pi B_E R_E^2}{\Phi}$
\quad \text{return} $L^*$
best that can be hoped for with any analysis of the second invariant involves determining the error associated with searching for field lines that maintain the invariant, plus the error response of the third invariant (which we can solve analytically). In the analysis of the third invariant, the fact that it can be solved analytically on a dipole can be used to generate profiles for error response and sensitivity. The remainder of this section demonstrates the error responses associated with the individual methods needed in solving the third invariant.

Calculation of the third invariant involves several steps, each of which can be analyzed for their error response. Using the method found in Roederer and Zhang (2014), the invariant nature of the second invariant \( K \) can be used to iteratively find the drift trajectory for a particle in the magnetosphere.

### 3.2.1 Error Response due to Field Line Footprint Inaccuracies

The Roederer and Zhang (2014) method requires that a particle’s bounce trajectory be traced to calculate \( K \), and the footprint of the field line must be determined (the footprint will hereafter be referred to as the \( \lambda \) intersection, or simply \( \lambda \)). With \( K \) and its associated \( B_{\text{mirror}} \) value, the space around Earth is systematically searched to find other field lines conforming to these now-known values. Once a sufficient number of field lines are located, the drift trajectory can be calculated and the flux through the drift shell \( (\Phi) \) can be calculated. With \( \Phi \) calculated, \( L^* \) can be easily calculated utilizing Equation (3.15). In a dipole, as is being examined here, the simplified version of the \( L^* \) computation given in Equation 3.17 can be used instead.

To fully understand how errors propagate in this system, it is necessary to look at each individual piece. First, it is important to note that for a dipole, field lines can be computed analytically with the following equation from Walt (1994):

\[
r = R_0 \cos^2 \lambda
\]

where \( \lambda \) is the latitude of the field line, and \( r \) is radial distance. \( R_0 \) is the equatorial crossing radial distance (where \( \lambda = 0 \))
Given this relationship, the surface $\lambda$ intersection point can be obtained. By fixing $r = 1R_E$ and utilizing the relationship in Equation (3.18), Equation (3.19) can be derived. Figure 3.2 shows this analytic relationship between $L^*$ and $\lambda$ on a Dipole.

$$
\lambda = \cos^{-1}\left(\frac{R_E}{R_0}\right)
$$

(3.19)

Knowledge of this profile will allow for future analysis of the accuracy of field line tracing on dipole gridded data (more in Chapter 4).

Given a starting $K$ and $B_{m_0}$, to find field lines on additional local times, bisect over field lines until one is found where $K_n = K_0$ for $B_{m_n} = B_{m_0}$. This search is conducted at intervals of longitude until a drift trajectory is defined. This provides an integration boundary for calculating magnetic flux ($\Phi$). If $K$ and $B_{mirror}$ are consistent over all field lines, all iteratively discovered field lines in the bounding set will fall on the $\lambda$ given by Equation (3.19).

To better understand the potential errors involved in this method, relying on the analytic solutions to $\lambda$ intersections as a guide (see Figure 3.2 and Equation (3.19)), the error associated
with tracing a field line through a magnetic field can be examined. Figure 3.3 shows this profile for a range of $L^*$ values. The traces in this profile are conducted utilizing a Runge-Kutta 4/5 method with a step error tolerance of $1E^{-5}$. The dashed line indicates the absolute error between the traced $\lambda$ intersection and the analytic solution.

![Figure 3.3: Error Profile for obtaining $\lambda$ from a field line trace vs. the analytic solution.](image)

Understanding the error in the field line trace provides a means of determining how the field line trace $\lambda$ affects the ideal solution for $L^*$ in general. Noting that the analytic solution for $L^*$ on a dipole is given by Equation (3.17), Equation (3.20) shows how the error in $L^*$ due to the field line trace ($E_{\text{trace}}$) can be calculated.

$$E_{\text{trace}} = L^*(\lambda_{\text{analytic}}) - L^*(\lambda_{\text{trace}})$$  \hspace{1cm} (3.20)

where $L^*(\lambda_{\text{analytic}})$ is the analytic solution for $L^*$ utilizing the analytic intersection for the field line, and $L^*(\lambda_{\text{trace}})$ is the analytic solution for $L^*$ utilizing the field line traced solution (or, in this case, a perturbed variation of the base $\lambda$). The profile for $E_{\text{trace}}$ is given in Figure 3.4. The green line shows the value of $E_{\text{trace}}$ for a given $\lambda_{\text{error}}$, and the blue line shows the ratio $\frac{E_{\text{trace}}}{\lambda_{\text{error}}}$ (the Error
Effect Ratio), which indicates the percentage of error from \( \lambda \) is seen in the final \( L^* \) calculation. For this plot \((L^* = 6.6R_E)\), it is obvious that the final \( L^* \) calculation sees about half of the error seen in \( \lambda \).

![Error in \( L^* \) due to Error in Field Line Tracing (\( E_{\text{trace}} \)) on a dipole @ \( L^* = 6.6 R_E \)](image)

Figure 3.4: Profile for the error in \( L^* \) associated with the field line trace. The blue line indicates the ratio \( \frac{E_{\text{trace}}}{\lambda_{\text{error}}} \) for this \( L^* \) value \((6.6R_E)\). It can be seen that for an error in \( \lambda \), approximately 54.5% is passed to the overall \( L^* \) calculation. The instability seen on the left side of plot is due to floating point truncation error.

This error, however, is dependent on the starting \( \lambda \) (and the resulting desired \( L^* \)). Figure 3.5 shows the relationship of the Error Effect Ratio compared to the base \( \lambda \) solution. This provides insight on the sensitivity of the \( L^* \) calculation to errors in \( \lambda \). It can be seen in this plot that, as the base value of \( \lambda \) increases, so does the effect of error on \( L^* \).

This makes sense, as magnetic field strength gradients are much larger near the Earth’s poles. As \( \lambda \) increase (gets closer to the pole), so does the resulting \( L^* \) value. Thus, as the dipole equatorial radial distance increases (larger \( L^* \)), the closer \( \lambda \) gets to the pole, the faster the magnetic field changes, and the greater the effect due to any error in \( \lambda \).
Figure 3.5: Profile of the median Error Effect Ratio for a range of base $\lambda$ values.
Chapter 4

Working with LFM Model Gridded Output

4.1 The LFM Grid

Lyon et al. (2004) provides a succinct summary of the Lyon-Fedder-Mobarry (LFM) simulation grid. It is described as a non-orthogonal, non-adaptive, distorted spherical grid statically adapted to the problem. Within the grid, the $r$ and $\theta$ coordinates are fitted to the problem. The coordinates are derived by minimizing the weighted smoothness integral:

$$ I = \int dS \frac{(\nabla \mu)^2 + (\nabla v)^2}{w}, $$

(4.1)

Cell distribution is controlled by the weight ($w$), causing more cells to cluster wherever $w$ is large, and the reverse when $w$ is small. Weight is concentrated near the magnetopause, bowshock and tail; regions where it is expected that larger variations in field geometries will exist. The values for $w$ were calculated by hand. The $\phi$ grid coordinate is not adapted, although it is stretched in such a way that larger cells fall near the equatorial plane, with slightly smaller cells near the poles. The grid space covers a region around Earth that encompasses $+30R_E$ to $-300R_E$ along the $x$ axis, and a typical cylindrical radius of $100R_E$. The system is oriented in the solar magnetic ($SM$) coordinate system where the $+Z$ axis aligned to the north magnetic pole, the Sun-Earth line is contained in the $xz$ plane, and the $y$ axis is perpendicular to the other two (Laundal and Richmond, 2017). The LFM grid has a nominal inner boundary of $2.3R_E$.

Figure 4.1 provides a side and front view of the LFM grid. In the left hand image, the $X$ axis points to the top. The right hand view shows the grid looking toward the $+X$ face. The cylindrical
Figure 4.1: This image provides side and front views of the LFM grid. The graphics are annotated to illustrate the basic logical arrangement of the simulation grid.

Symmetry of the grid is clearly illustrated with radial components arranged in a fan line pattern around the central $x$-axis (notated by $\alpha$ angle in the figure). Images in Figure 4.2 provide two views of an individual grid blade. The top figure provides an overall overview of a blade, clearly showing the weighting of the grid cells. The bottom portion of the figure provides a zoomed view of the inner boundary. This view illustrates the distorted spherical nature of the grid.

Online documentation indicates that the grid can be configured to arbitrary dimensions (https://wiki.ucar.edu/display/LTR/MSETUP), but the most common resolutions used are given in Table 4.1. Also included in this table is a reference to the number of coordinate vertices in the grid. This number will become important in future sections when dealing with calculating the $K$ and $L$ invariants.

4.2 LFM Grid Dipole Response

Before delving into the specifics of dealing with MHD data on an LFM grid, it is important to understand how the grid itself affects the outcome of the $K$ and $L^*$ algorithms. To this end,
the following dipole analysis, similar to that conducted for a non-gridded dipole in Chapter 3, is presented, along with interpretations of potential impacts to the overall $K$ and $L^*$ solutions.

Recall from Chapter 3, Figure 3.3, that without a grid, tracing a field line through a dipole provided an end-point (footprint) accuracy at the surface of Earth on the order of $10^{-5}$ degrees. When a dipole is placed within an LFM grid, the controlling factor for accuracy becomes the resolution of the grid. Figure 4.3 provides, instead of footprint error, the relative $L^*$ error responses for three different grid sizes: Single, Double, and Quad (see Table 4.1 for more grid resolution detail). For an analysis of field lines along the $x$ axis, the resolution exhibiting the most erratic error response is, predictably, the single resolution grid. At low L values, $L^*$ accuracy is in the $10^{-2} - 10^{-3}$ range. As the radial distance increases, the accuracy improves, peaking at between 7.5 and 8.0 before falling back. The double precision grid is more stable in its error response, and performs, overall, about the same to slightly better than the single precision grid. The quad resolution grid response is well within the $10^{-4}$ range, though it fluctuates more than the double
Table 4.1: List of resolutions for common LFM grid sizes and their corresponding number of coordinate vertices.

<table>
<thead>
<tr>
<th>Name</th>
<th>Resolution</th>
<th>Number of Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>53 x 24 x 32</td>
<td>40,704</td>
</tr>
<tr>
<td>Double</td>
<td>53 x 48 x 64</td>
<td>162,816</td>
</tr>
<tr>
<td>True Double</td>
<td>106 x 48 x 64</td>
<td>325,632</td>
</tr>
<tr>
<td>Quad</td>
<td>106 x 96 x 128</td>
<td>1,302,528</td>
</tr>
</tbody>
</table>

resolution grid.

Figure 4.3: Profile for $L^*$ error by dipole $L^*$ generated down the +x axis.

This view along the $x$-axis is insufficient, however, to understand the complete behavior of the LFM grid. To gain a more complete understanding, it is necessary to look at sample field lines in all directions, as the $x$ axis has a unique geometry within the grids. To simplify this understanding, Figures 4.4 to 4.10 provide a statistical overview of the total and relative $L^*$ errors associated with tracing field lines within the different grid resolutions. Starting points for field line traces were constructed by randomly selecting starting radial distances (between $3.0R_E$ and $10R_E$) and starting $\lambda$ (from values between $-\frac{\pi}{4}$ and $\frac{\pi}{4}$ radians). These selections were then combined with 16 evenly spaced values of $\phi$ (covering the full range from 0 to $2\pi$ radians). Figures 4.4, 4.6, and 4.8
display the total $L^*$ error associated with every point in the sample set, ordered by dipole $L^*$. The three different resolution distributions were computed with the same set of sample starting points, to provide for easier comparison between resolutions. In all, the sample set contains 2400 unique staring points.

To determine the error, each point was used as an origin to produce a field line trace. From this line, the footprint $\lambda_c$ was determined and Equation (4.2) used to calculate $L^*$ based on the results of the trace ($L^*_t$). The ideal $L^*$ for the starting location ($L^*_{\text{ideal}}$) was then calculated using the same equation utilizing the starting location to supply $R$ and $\lambda$.

$$L^* = \frac{R}{\cos^2(\lambda)} \quad (4.2)$$

The total error was then computed with:

$$E_T = |L^*_{\text{ideal}} - L^*_t|. \quad (4.3)$$

Relative Errors were calculated by:

$$E_R = \frac{|L^*_{\text{ideal}} - L^*_t|}{L^*_{\text{ideal}}} \quad (4.4)$$

Once the errors were calculated, the data was sorted by $\phi$ to allow for examination of error distributions by local time. Figures 4.5 (single resolution), 4.7 (double resolution), and 4.9 (quad resolution) provide the local-time breakdown for relative $L^*$ errors.

Single resolution grids show the most variation in error throughout the grid (Figure 4.5). The variation becomes less pronounced as the grid resolution increases. The median error distribution for the single resolution grid sees the highest errors (on the order of $10^{-2}$) near noon (0°), with errors decreasing as field lines move toward midnight (180°).

Double resolution grids see less variation when compared to single resolution, while still exhibiting the highest errors on the noon side. The quad resolution grid shows similar variances to the double resolution grid, but with an overall lower median error rate. The noon-side error also presents as high in the quad resolution grid.
Figure 4.4: This figure provides a view of the dipole-L vs total $L^*$ error for 2400 random samples on a single resolution LFM grid.

Figure 4.5: This plot provides a statistical view of dipole $L^*$ errors within a single resolution LFM grid. Note the cyclical nature of the quartiles. The single resolution grid has the most variance in error. The sample consists of 2400 particle locations assembled from 150 randomly selected $\lambda$ and $r$ values combined with 16 evenly spaced values for $\phi$. 0° represents noon. Angles increase counter-clockwise looking down at the north pole.
Figure 4.6: This figure provides a view of the dipole-$L$ vs $L^*$ error rate for 2400 random samples on a double resolution LFM grid.

Figure 4.7: This plot provides a statistical view of relative dipole $L^*$ errors within a double resolution LFM grid. $0^\circ$ represents noon. Angles increase counter-clockwise.

Figure 4.10 breaks down the distributions by overall quartiles for each resolution. It can be seen that (as expected), the median error decreases as the resolution increases, giving confidence
Figure 4.8: This figure provides a view of the dipole-L vs $L^*$ error rate for 2400 random samples on a quad resolution LFM grid.

Figure 4.9: This plot provides a statistical view of dipole $L^*$ Errors within a quad resolution LFM grid.

that an increased grid resolution will produce a more accurate result when calculating $L^*$ with MHD simulation results.
Figure 4.10: This figure provides an overview of the overall error distribution for the three grid resolutions selected. It is clear that the grid resolution has a direct, if small, impact on the accuracy of the field line traces. This can be directly translated into an effect on the overall $L^*$ calculations.

The final factor to consider is the overall propagation of errors (due to grid resolution) to the final $L^*$ calculation. Figure 4.11 provides the error propagation rate from the grid based on dipole $L$ (i.e. the $\lambda_c$ point of a field line). The procedure for this was first outlined in Chapter 3 utilizing an un-gridded dipole field. Unlike Figure 3.5, which ordered data by the $1R_E \lambda_c$ intersection, Figure 4.11 orders the data by dipole $L^*$, making it valid for any radial-distance crossing point. This change was made due to the termination of the LFM grid at around $2.3R_E$.

It can be seen from Figure 4.11 that the median error propagation rates are nearly identical between grid resolutions. From this, it can be concluded that, regardless of the grid resolution, the LFM grid geometry produces an error amplification for field line footprint errors up to nearly $50 \times$, depending on the traced $L^*$ magnitude.

4.3 Algorithm Adjustments for LFM Data

Moving from a dipole to a more complex gridded magnetic field introduces several challenges to successfully computing $L^*$ for any given particle. Initial test runs utilizing LFM data revealed
several issues that had to be addressed in order to begin computing satisfactory $K$ and $L^*$ values. Based on the initial testing, it was discovered that the $K$ calculation algorithm was not robust enough to consistently and reliably calculate $K$ along a field line. Following intense investigation, it was determined that two primary issues were responsible for failure of the algorithm. First, the algorithm was making assumptions about magnetic field lines that did not hold in a complex field. Second, numerical noise in the MHD solution of the field causes fluctuations in some field lines that make identifying the correct $K/B$ pair on a field line very difficult. In addition to issues with the $K$ algorithm, problems were also identified in the method used to locate additional field lines with the same $K/B$ pair. In complex field geometries, specifically in areas where field lines are swept back toward the tail region, a linear bisection search becomes invalid. These issues are discussed in greater detail in the following sections.

Figure 4.11: This image shows the Error Effect Ratios for all three grid sizes based. This confirms that the Error transmission from from the grid is essentially the same when moving between grid resolutions.
4.3.1 The New $K/B$ Model

The $K/B$ relationship within a field line is one of the most important aspects to calculating $L^*$, as the field line footprints’ trajectory dictates the outcome of the magnetic flux integral used in calculating $L^*$. Failure to calculate a proper $K$ value for a given $B_m$ will result in the field line search routines finding incorrect lines, resulting in complete failure of the algorithm. Originally, the algorithm used to calculate $K$ had a critical flaw; it made the assumption that a field line would be symmetric about the minimum $|B|$ point on any given field line. The distribution of $K$ across the line was thus assumed to be symmetric. The process of re-sampling a line to calculate $K$ forced this assumption on the calculations. In complex field lines, symmetry of $K$ does not necessarily exist. Testing quickly revealed that a different method for calculating $K$ would be required.

The method eventually developed fell back on the physical interpretation of $K$. Recall from Equation (3.4), that $K$ is calculated with the integral:

$$K = \int_{s_m}^{s'_m} \sqrt{B_m - B(s)} \, ds$$

where $s_m$ is the first mirror point for a particle moving along the field line, $s'_m$ represents the opposite mirror point along the line, $B_m$ is the $|B|$ of the point corresponding to where the particle mirrors, and $B(s)$ represents the value of $|B|$ at points between $s_m$ and $s'_m$ along the field line. Figure 4.12 illustrates how $K$ is calculated. First, $K = 0$ will result wherever there is a local minimum, as there is no point for the particle to mirror on that does not first have to go through a higher intensity magnetic field. The blue tick, located at $B_m = 61.3$ in the figure, is an example of one such location. On a dipole, only one such location can exist for any given line. Continuing with the example, $K$ for $B_m = 500$ (red ticks) would be calculated by integrating Equation (3.4) where the two marks represent $s_m$ and $s'_m$, $K$ for $B_m = 2500$ would similarly be calculated between the gray marks, and so on. It can be seen from this illustration that an ideal dipole field will generate a $K$ profile that is symmetric, and this is indeed the result, as is shown in Figure 4.13. It is clear from this figure that both the north (left side) and south (right side) hemisphere $K$ calculations
are symmetric.

Figure 4.12: This figure illustrates how $K$ would be calculated for an ideal dipole field line. Note that, on a dipole line, all values of $B_m$ are symmetric between the north (left) and south (right) portions of the field line.

Figure 4.13: This image depicts an ideal K/B model, based on a dipole field line through L=8.0. Note that the model is symmetric about $B_m$. 
The first iteration $K$ algorithm broke down because it assumed this symmetry. In reality, the Earth’s dipole tilt, coupled with the interaction of the solar wind, induces non-symmetric field geometry, leading to a non-symmetric $K$ model. Figure 4.14 provides an illustration of such a situation. These asymmetries ultimately lead to an invalid model of $K$ when utilizing the original algorithm, which, in turn, results in poor $K/B$ matching in the search algorithm.

Note that, in this figure, the primary $|B|$ minimum is located off-center, and that the field line exhibits multiple $K = 0$ zones due to several local minima. These additional $K = 0$ regions may be either a valid $K = 0$ region, or, more likely, the result of numerical noise in the field line trace. Some of this noise, but not all, can be dealt with, and that will be addressed in a future section.

Adapting to the realities of complex field geometry required a complete re-write of the $K$ algorithm. To ensure that all points on the $K$ profile are properly calculated, the algorithm operates in two passes. The first pass calculates $K$ for every known $|B|$ value on the field line, moving from north to south. The second pass performs the same operation moving from south to north. The two
profiles are then joined. In most locations, a non-zero value is joined with a zero value, resulting in a non-zero value for that point (a location with a mirror point behind the trace progress will always yield a zero K), or a zero value with a zero value, which will yield $K = 0$ for that point. In some instances, where there is a local maximum in the field geometry, the algorithm may yield two slightly different values for $K$. In this instance, the final result is the maximum of the two.

Pseudo-code presented in Algorithms 10 and 11 walks through this process.

Algorithm 10 Calculate KB Model

1: procedure KB_MODEL(Trace_Data, Trace_Loc)
2:   $K_f, B_f \leftarrow K_{Forward}(Trace_Data, Trace_Loc)$
3:   $K_b, B_b \leftarrow K_{Forward}(\text{reverse}(Trace_Data), \text{reverse}(Trace_Loc))$
4:   $K \leftarrow \max(K_f, \text{reverse}(K_b))$
5:   $B \leftarrow B_f$
6: return $[K, B]$

Algorithm 11 Calculate K Forward Through Array

1: procedure K_FORWARD(Trace_Data, Trace_Loc)
2:   $K_f \leftarrow \text{Array}$
3:   $B_f \leftarrow \text{Array}$
4:   $IDX \leftarrow 1^{st} \text{Element}$
5: while $IDX < \text{len}(Trace_Data)$ do
6:   $B_m \leftarrow Trace_Data[IDX]$
7:   $S_m \leftarrow Trace_Loc[IDX]$
8:   $S'_{IDX2} \leftarrow \text{first index where } Trace_Data[IDX : END] > B_m$
9:   $S'_{IDX1} \leftarrow S'_{IDX2} - 1$
10:  $S'_m \leftarrow \text{Interpolated new point between } S'_{IDX1}, S'_{IDX2}$
11:  $K_s \leftarrow \int_0^{S'_m} \sqrt{B_m - B(s)} ds$
12:  $K_f[IDX] \leftarrow K_s$
13:  $B_f[IDX] \leftarrow B_m$
14:  $IDX \leftarrow IDX + 1$
15: return $[K, B]$

4.3.2 Numerical Noise

Figure 4.14 also exhibits characteristics of the second major problem that needed to be addressed: numerical noise in the MHD model output. In this case, numerical noise is defined as variations in areas where the grid-size is insufficiently small to produce a smooth result, or where interpolation errors produce random fluctuations. The apparently random breaks in the Figure
4.14 profile, where the value of $K$ intermittently drops to zero along an otherwise unbroken curve, are evidence of this issue. Figure 4.15 provides a more dramatic example of this phenomenon. Matching $K$ to $B$ properly in this environment is extremely difficult.

![K/B Model with LFM Data (Quad Res)
Line through [8.75, 0.0, 0.0] (w/out smoothing)](image)

Figure 4.15: This figure illustrates what happens to $K$ when a particle is in a bifurcation region within a field line that exhibits an excessive amount of noise. It is exceedingly difficult to match a $B$ value to its correct $K$ value within this configuration.

An issue can be confirmed as a noise problem in MHD output by progressively increasing the grid resolution, and noting if the fluctuations dissipate. For the example presented in Figure 4.17, this was accomplished through inspection of the same line in three different grid resolutions; single, double, and quad resolution. Numerical noise due to the MHD grid size can be seen in the progression. The blue line, representing the single resolution output, displays high-frequency oscillations through the bifurcation region. The black line, representing the double resolution output, also shows these oscillations, but at a lower frequency. Moving to a quad-resolution grid (red trace) shows that the oscillations virtually disappear, thereby confirming the presence of numerical noise. While the only means of removing the numerical oscillation at this scale is to increase the MHD grid resolution, local jitter can be smoothed through the filtering of high frequency noise in
the field line. This local jitter is due to errors associated with interpolating between grid points and can induce a large number of false $K = 0$ regions.

![Figure 4.16: This figure provides a zoomed view of $|B|$ with its associated $K$ model from Figure 4.15. This view clearly shows where the numerical jitter produces unwanted $K = 0$ regions.](image)

Figures 4.15 and 4.16 illustrate the effects of local jitter. Figure 4.15 shows a wide angle view of the relationship between $B_m$ and $K$. The compressed field line exhibits what, at first glance, appears to be a fairly uniform and smooth line. However, small fluctuations about the line have caused several false $K = 0$ regions to appear. This detail is revealed in Figure 4.16, with the magnetic field trace expanded to show the smaller fluctuations.

Figure 4.18 shows the same progression with a Sav-Gol 1-D digital signal filter (Savitzky and Golay, 1964) applied to the $|B|$ trace. This procedure removes the smaller oscillations without dramatically changing the original trace, leaving a smoothed version of the original. Figures 4.19 and 4.20 demonstrate this effectiveness on the K/B model calculation when the filter is applied. The number of $K = 0$ locations are greatly reduced, leaving only three remaining $K = 0$ instances. These regions may either be valid $K = 0$ zones, or be caused by noise due to the LFM grid resolution. Looking at Figure 4.20, the $K = 0$ region that exhibits at $B_{min}$ is the only guaranteed
Figure 4.17: This image depicts a common field line over 3 different grid resolutions. These lines have not been filtered. The blue line depicts the line on a single resolution grid, black on a double resolution grid, and red, a quad resolution grid. The numerical instability associated with the lower resolution grids is clearly visible.

Figure 4.18: This image depicts the same field lines as in Figure 4.17, but with a Sav-Gol filter (Savitzky and Golay, 1964) applied to the magnetic field trace, resulting in a far smoother line. This filtering greatly improves the chances of calculating a proper $K/B$ model.
true $K = 0$, while the leftmost $K = 0$ may be evidence of a bifurcated line, or may simply be an oscillation caused by noise. The central $K = 0$ is almost certainly induced by noise due to grid resolution.

![K/B Model with LFM Data (Quad Res)
Line through [8.75, 0.0, 0.0] (w/smoothing)](image)

Figure 4.19: This image shows the results of applying a Sav-Gol filter to a magnetic field trace. This is the same trace as is depicted in Figure 4.15. The resulting $K$ model is far more reasonable than that of the unfiltered line.

### 4.3.3 Extremely Distorted Magnetic Fields and Search Progression

Searching for the correct $K/B$ matched field lines is a particularly difficult task in regions of extremely distorted field lines. Although $L^*$ is only defined in regions of stable particle trapping (Roederer and Zhang, 2014), and any calculated $L^*$ within the highly distorted areas would likely not be valid, it may still be necessary to actively move a search progression through these areas. If the target field line is near the boundary of distorted lines, the search algorithm will likely spend large quantities of time in these regions.

Searching in highly distorted field geometries causes the standard linear bisection search algorithm to fail. The extreme artificial situation outlined in Figure 4.21 illustrates this problem. In this example, an initial particle is specified with a high $B_m$ value (red dot). An initial field
Figure 4.20: This image provides a zoomed view of Figure 4.19, illustrating the effectiveness of the Sav-Gol filter in removing excessive jitter.

A line is then traced to calculate $K$ for the particle (solid black line). Finally, the search algorithm is applied in an attempt to find the specified line. With the standard linear search algorithm, which was based only on $\phi$, an effort to match the original trace fails. Looking at the dashed black line (the search path), the reason for this becomes clear; the field geometry is shifting the target field line away from the desired $\phi$ intersection ($\phi_c$). As the search progression jumps $\phi$ angles, the entire premise of the bisection search becomes invalid. The search relies on successive field lines along a path exhibiting predictable decreasing field intensity as the search progresses away from Earth, and increasing intensity as the search moves in. With the example illustrated in this figure, these conditions are not maintained. The relative intensity, as compared to the search path, of the distorted field line is unknown, and thus cannot be reliably used to continue the bisection.

In an effort to alleviate some of this issue, and find a valid line within distorted regions, an adaptive $\lambda$ condition was added to the search algorithm. Utilizing $\lambda$ of the original particle starting point provides for a considerably better chance of locating the necessary field lines when particles are physically closer to Earth, as the search tends to bisect along a path that is not as
Example of Line Search Failure with Linear $\lambda = 0$ Search

$K = 671.60, B_m = 234.24, \phi = 90.00^\circ$

Figure 4.21: The figure illustrates a linear search failure when utilizing a fixed $\phi$. The black line, labeled “LShell Line” is the line being sought. In this situation, as the search progresses along the fixed $\phi$, $\phi_{\tau}$ along the search path shifts away from the search $\phi_s$. The inset image is an angle view illustrating the 3 dimensional positioning of the associated field lines.

heavily distorted. This provides a greater chance for finding a valid line, and to calculate a valid $L^*$. Figure 4.22 illustrates the results of utilizing this technique for the same situation as described above. The adaptive $\phi$ search algorithm produces a fairly good, though not perfect match for the original particle line. As long as the $K/B$ match is within tolerance, the found line can be considered valid, and used for the $L^*$ calculation.
Figure 4.22: Utilizing and adaptive $\lambda$ search algorithm, a much better line fit is achieved. This search result illustrates a line that was not found with a simple fixed $\lambda$ linear search. To obtain a good $L^*$, it is not necessary to find the exact line, but the $K$ and $B$ parameters must be correct.

This method is not ideal, as it will still break down for initial particle positions physically located farther from Earth, where it is more difficult to avoid searching across distorted lines, or is too close to the Earth at extreme $\lambda$ positions where the field density is high. A possible solution to this problem, not yet attempted, would change the search progression to a bisection across $\lambda$ at

Example of Line Search with Adaptive $\lambda$ Search

$K = 671.60, B_m=234.24, \phi = 90.00^\circ, \lambda = 52.21^\circ$

$\lambda$ Error: 6.95e-03 radians
a fixed radial distance from the Earth, relative to the location in question, and in a region where field lines are not greatly distorted. This approach should lead to searching successive lines where search conditions are not violated, while avoiding direct interaction with regions of highly distorted field lines. This approach is being deferred as future research.

4.3.4 Dealing with Multiple $K$ Instances Per Line

Bifurcated field lines, or lines that exhibit multiple local $B_{min}$ conditions, provide a challenge to finding valid field lines for the $L^*$ calculation. In these areas, due to the definition of $K$, a given value of $K$ can appear multiple times along the arc of the line. This phenomenon is clearly illustrated in Figure 4.18 where any given value of $K$ appears no less than eight times along the arc of the line.

To calculate a correct $L^*$, it is necessary to correctly identify field lines that contain the correct $[K, B]$ pair. This requires tracing a certain path with the bisection search algorithm. This is a straightforward search for the best fit on non-bifurcated lines. The dilemma arises, however, when lines are bifurcated; which $[K, B]$ pair should be used for the search path? As most methods for calculating $L^*$ on empirical models ignore the bifurcation region entirely (Min et al., 2013b; Lejosne, 2014), the correct answer to this question is not at all clear. As $L^*$ may not be conserved in these regions (Min et al., 2013b; Roederer and Zhang, 2014; Ukhorskiy et al., 2017), and the behavior of $K$ for particles transiting this region is extremely complicated (Öztürk and Wolf, 2007; McCollough et al., 2012), only $L^*$ calculated in non-bifurcation regions should be considered valid, though instantaneous $L^*$ values will still be generated for these areas. The nature of noise in the gridded field line traces does makes it difficult to differentiate between field lines that are truly bifurcated, and those that only appear to be bifurcated because of the noise. Additionally, it may be necessary for the search to progress through these uncertain regions. Due to this need, the search algorithm must be able to navigate these areas. The approach taken with the GHOSTpy algorithm is to follow the path of minimum gap between $B_m$ and $B_k$, where $B_m$ is the mirror point being sought, and $B_k$ is the $|B|$ value associated with $K$ on the line. In regions that appear to be
bifurcated, this will return multiple \( B_m \) values for a given \( K \) value. Following the path of minimum \( B_{\text{gap}} \), where \( B_{\text{gap}} \) is defined as

\[
B_{\text{gap}} = |B_k| - |B_m|,
\]

will naturally lead to following a path through a pseudo-bifurcated line that is likely to ignore any remaining spurious \( K = 0 \) regions. \( B_{\text{gap}} \) is the primary means of navigation through the model space, with the algorithm checking this value on every iteration to decide which bisection path to take. The path can be thought of as “what are the search boundaries set to after each iteration.” If a gap is negative, the current search location becomes the outer boundary. If this gap is positive, it becomes the inner boundary. If the incorrect path is chosen the algorithm will terminate in a local minimum region, and will not converge to the optimal match, thus yielding a potentially invalid line match. This occurs primarily within the bifurcation region, providing an \( L^* \) value with uncertainty as to the overall validity of \( L^* \) calculations near these regions.

### 4.3.5 Final Updated Drift Trajectory Search Algorithm

The drift trajectory search algorithm is the most important component of the overall \( L^* \) calculation, and has undergone multiple and significant changes throughout the research process. Algorithm 12 provides the final updated search method on which all performance metrics are based. Some error checking routines have been removed for conciseness and ease of reference.

The algorithm starts by searching around the same radial distance of the original particle, and at the same \( \lambda \) angle. The area is bisected until a valid \( b_{\text{gap}} \) is found. If the search reaches the inner boundary and has only found an outer gap, the boundary is moved in. If the search reaches the outer boundary and has only found an inner gap, the boundary is moved out. If the search finds an invalid \( b_{\text{gap}} \), and the outer gap is defined, the inner boundary is moved out. If \( b_{\text{gap}} \) is invalid, and both inner and outer \( b_{\text{gap}} \) are unset, the outer boundary is moved in. When valid negative gaps are obtained, the outer boundary is moved in; the inner boundary is moved out for positive gaps. The search terminates when the line is found (in reality, when it is within the error tolerance), or
when the gap between lines reaches $\epsilon$. As noted earlier, the algorithm uses the $\min(b_{gap})$ path to advance the algorithm when more than one $b_{gap}(k)$ is present.

### 4.3.6 Accuracy and Convergence

Accuracy of the final $L^*$ calculation is directly related to how well the search algorithm finds the integration path for the flux calculation. The earlier part of this chapter focuses heavily on how errors propagate through the system, but does not provide a great deal of practical insight on accuracy within complex fields. As seen in the previous few sections, the search algorithm is not 100% accurate in finding field line footprints. The more accurate these footprints are, the more accurate the magnetic flux calculation (this can clearly be seen in a simple walk-through of Equation (3.16)), and thus the more accurate the final $L^*$ calculation. The question that needs to be asked then, is: How many field line traces are needed in order to obtain an accurate integration path?

![Figure 4.23](image)

Traditional methods usually involve adding field line traces until the final flux solution converges (see Roederer (1970) and Roederer and Zhang (2014)). The number of field line traces can be reduced, however, through the application of spherical linear interpolation between points. When applied to accurate field line traces, experimentation shows that, if field line fits are good enough, a 4-line convergence (lines at noon, midnight, dawn and dusk) can be accurate to the
Algorithm 12 Drift Path Search Algorithm ($\lambda$ adaptive)

1: \textbf{procedure} Search\_KB($\phi_0$, $\lambda_{pt}$)
2: \hspace{1em} $ol$ ← starting point radial distance
3: \hspace{1em} $\lambda_{\tau}$ ← $\lambda_{pt}$
4: \hspace{1em} $\phi_{\tau}$ ← $\phi_0$
5: \hspace{1em} $r_{in}$ ← $0.65 \times ol$
6: \hspace{1em} $r_{out}$ ← $1.85 \times ol$
7: \hspace{1em} $\tau$ ← $(r_{in} + r_{out})/2$
8: \hspace{1em} $b_{gap}$ ← $r_{out} - r_{in}$
9: \hspace{1em} $b_{gap_{in}}$ ← \textit{None}
10: \hspace{1em} $b_{gap_{out}}$ ← \textit{None}
11: \hspace{1em} \textbf{while} $b_{gap}$ > $\epsilon$ \textbf{do}
12: \hspace{2em} $\tau$ ← Trace line from $\text{Loc}$
13: \hspace{2em} $\text{loc} ← [r_{\tau}, \lambda_{\tau}, \phi_0]$
14: \hspace{2em} $r_{\tau}$ ← $b_{\tau}(k) - b_m$
15: \hspace{2em} \textbf{if} $b_{gap}$ is \textit{None} \textbf{then}
16: \hspace{3em} \textbf{if} $b_{gap_{in}}$ is \textit{None} AND $b_{gap_{out}}$ is \textit{not None} then
17: \hspace{4em} $r_{in}$ ← $r_{\tau}$
18: \hspace{3em} \textbf{else}
19: \hspace{4em} $r_{out}$ ← $r_{\tau}$
20: \hspace{4em} $\tau$ ← $(r_{in} + r_{out})/2$
21: \hspace{4em} \textit{Continue to next loop}
22: \hspace{2em} \textbf{else if} $b_{gap}$ < 0 \textbf{then}
23: \hspace{3em} \textbf{if} $r_{\tau}$ == $r_{in}$ \textbf{then}
24: \hspace{4em} $r_{in}$ ← $r_{in}/2$
25: \hspace{4em} $r_{out}$ ← $r_{\tau}$
26: \hspace{4em} $\tau$ ← $(r_{in} + r_{out})/2$
27: \hspace{4em} \textit{Continue to next loop}
28: \hspace{2em} \textbf{else}
29: \hspace{3em} $r_{out}$ ← $r_{\tau}$
30: \hspace{3em} $r_{\tau}$ ← $(r_{in} + r_{out})/2$
31: \hspace{3em} \textit{Continue to next loop}
32: \hspace{2em} \textbf{else if} $b_{gap}$ > 0 \textbf{then}
33: \hspace{3em} \textbf{if} $r_{\tau}$ == $r_{out}$ \textbf{then}
34: \hspace{4em} $r_{out}$ ← $r_{out} \times 1.5$
35: \hspace{4em} $r_{in}$ ← $r_{\tau}$
36: \hspace{4em} $\tau$ ← $(r_{in} + r_{out})/2$
37: \hspace{4em} \textit{Continue to next loop}
38: \hspace{2em} \textbf{else}
39: \hspace{3em} $r_{in}$ ← $r_{\tau}$
40: \hspace{3em} $\tau$ ← $(r_{in} + r_{out})/2$
41: \hspace{3em} \textit{Continue to next loop}
42: \hspace{2em} \textbf{else if} $b_{gap}$ is close to 0 \textbf{then return} $\tau$
43: \textbf{return} $\tau_{in}, \tau_{out}$
first decimal place for low $L^*$ particles. Figure 4.23 graphically illustrates the concept. In this figure, the red lines represent field lines traced with the T96 model. The black dots represent the four points used to calculate the value of $L^*$, and the yellow lines represent the integration path as determined through spherical linear interpolation. Visually, it can be seen that the boundary is well aligned with the field line footprints. The effect can be seen more quantitatively, utilizing LFM data, in the difference plots provided in Figure 4.24. In these images, the difference between a 4-line convergence, and a convergence to $10^{-4}$ is presented. In these quiet solar wind condition images, we can see that the difference between these two methods is roughly 0.025 where below $L^* \approx 7$. The differences are greatest where field line density is high, and where drift orbits move through heavily distorted fields.

These plots only show a single time-step comparison, and only for quiet time conditions. A more detailed examination, shown in Figure 4.25 as a comparison between $L^*$ converged to $10^3$ and a 4-point convergence. It can be seen that, in quieter times, the correlation is quite good, though not perfect. The biggest difference is the missing of high-disturbance events, such as the onset of the storm shown in the figure. Some of the disturbance is captured, but not to the magnitude as the higher level of convergence. It may be possible, through more extensive study, to develop a better set of directions to use as the interpolation base to possibly yield a better fit.

### 4.4 Application of Methods

The purpose of calculating adiabatic invariants is to order geomagnetically trapped particles. The distribution function for particles within the radiation belts is a six-dimensional problem, given by

\[ f = f(x, y, z, p_x, p_y, p_z) \] (4.5)

and completely characterizes the radiation belts for a point in time, providing the number of particles within a volume $(x + dx, y + dy, z + dz)$ with the momenta $(p_x + dp_x, p_y + dp_y, p_z + dp_z)$. 
Figure 4.24: These plots show the differences between a 4-line convergence with respect to a full convergence to $10^{-4}$. The contour lines are whole integer intervals of $L^*$ values from within the simulation.
This can be written equivalently in adiabatic invariant space as

\[ f = f(M, K, L, \phi_1, \phi_2, \phi_3) \]  \hspace{1cm} (4.6)

where \( M \) is the first invariant, \( K \) is the second invariant, and \( L \) the third. \( \phi_1, \phi_2, \) and \( \phi_3 \) represent their respective phases. If the phase distribution is uniform, the function can be reduced to a function of just the three invariants, thereby reducing a 6-dimensional problem down to a 3 dimensional problem, and would be given as

\[ f = f(M, K, L). \]  \hspace{1cm} (4.7)

With the GHOSTpy analysis system now in place, a real-world application example is the next logical step. The event chosen for a trial analysis is a CME event that impacted Earth on 02 Oct 2013. This storm is the immediate predecessor to the event examined in Chapter 2, and was chosen because it falls within the active time period of the Van Allen radiation belt storm probes, and quad-resolution LFM simulations already exist in two configurations; with and without
an integrated ring current model. This reality provides an opportunity to explore the difference between two simulation methods, and see how the modeled $L^*$ and $K$ compare to those originally computed for the spacecraft observations. The plots in Figure 4.26 provide an overview of the solar wind conditions as observed at L1. The storm is clearly identifiable in the early morning of October 2, with an initial impact time of around 0100 UT.

![Figure 4.26: Solar wind conditions for 02 October 2013. The CME event impact can be seen early in the morning, near 0100 UT.](image)

Figure 4.27 shows the Disturbance Storm Time ($D_{ST}$) index associated with the 02 Oct 2013 storm. The red dot indicates the peak negative $D_{ST}$ for the storm, and indicates the date and time that is utilized for the full field comparison in Section 4.4.2 ahead.

![Figure 4.27](image)
Figure 4.27: AER $D_{st}$ index for 01 Oct 2013 - 06 Oct 2013. The red dot indicates the peak negative $D_{st}$ and indicates the date that will be analyzed with full field comparisons.

4.4.1 Van Allen Probes $L^*$ and $K$ Comparisons

The top plot in Figure 4.28 shows $L^*$ calculation from the definitive data Las Alamos National Lab (LANL) produced for both Van Allen Probe spacecraft. $L^*$ values are plotted for both spacecraft, with the lines colored by orbit. The y-axis represents $L^*$ calculated from the Tsyganenko T89 model (see Chapter 2), and the x-axis reflects the time of day on 02 Oct 2013. The input driver for the T89 model, Kp, is plotted as a gray dashed line. The bottom plot shows the $L^*$ vs. $K$ relationship (again, calculated from the T89 model, and colored by spacecraft orbit) for each probe.

The plots in Figure 4.29 present the same spacecraft positions, with $L^*$ and $K$ invariants computed on an underlying dipole field, and calculated utilizing the GHOSTpy utilities developed as a byproduct of the research in this thesis. The dipole field is overlaid on a quad resolution LFM grid.

As would be expected with a dipole field with no external contributions, there is no distortion to the $L^*$ arcs along the spacecraft orbital trajectory. The $K$ values also present in a very regular
Figure 4.28: Van Allen Probes A and B $L^*$ and $K$ for observations on 02 October 2013, computed utilizing the T89 magnetic field model. These data were retrieved from the Coordinated Data Analysis Web (CDAWeb) data repository hosted by NASA Goddard Space Flight Center.

fashion with the largest $K$s appearing at lower $L^*$ values, though the $K$ values show scatter when compared to the definitive data. This scattering is due to the combined effect of interpolation and line trace errors within the LFM grid. In general, the trends of $L^*$ calculated with a dipole model agree those of the T89 model, though the magnitudes vary. This is to be expected as the
dipole is only a first order approximation of the magnetic field.

Figure 4.29: Dipole $L^*$ and $K$ for the Van Allen Probe positions on 03 October 2013. Calculated utilizing GHOSTpy.

Plots for $L^*$ and $K$ from LFM and LFM/RCM data are presented in Figures 4.30 and 4.31 respectively. The calculations for these plots were conducted in parallel utilizing the GHOSTpy tools, and values were converged to $10^{-3}$ after an initial 4-point convergence. Running on the
Summit supercomputer, the time to completion of each set (one for LFM and another LFM/RCM) was approximately 15 minutes on 192 cores (8 nodes). Position and epoch data were obtained from the definitive RBSP data product, and a script was written to match satellite times to model files. These matched file/point lists were then imported into a variant of the parallel GHOSTpy application for computation.

Figure 4.30: LFM $L^*$ vs. Time
When comparing these plots to either the T89- or dipole-calculated versions, the most obvious feature of the LFM calculated traces is the smoothness. The dipole $L^*$, being the first order approximation, has no variation in smoothness over course of the orbits. The T89 plots show a small degree of variation, with $L^*$ curve discontinuities near the tops of the curves. These discontinuities are a direct result of the 3-hour Kp index resolution used to drive the model (See the Kp plot in Figure 4.28). The LFM generated plots show a great deal of variation in the traces, as would be expected from the more dynamical MHD simulation fields. Beyond the smoothness differences, the LFM (non RMC) trace in Figure 4.30 shows relatively large $L^*$ values during the first orbit when compared to both Figure 4.28 and 4.29. This time period corresponds to the main phase of the storm, and stronger variations are expected.

With the addition of the ring current to the LFM, by way of the RCM model, the peak $L^*$ values, not including the early portion of the storm, are more in line with those seen in the other two iterations. Van Allen Probe B was outbound in its orbit at the time of CME impact (See Figure 4.32 for a view of the magnetic field around Earth near the time of impact), and appears to capture the onset of the event. It can be seen from roughly 0100 through 0900 that magnetosphere is exhibiting a large amount of compression, pushing $L^*$ values higher. As the field relaxes over the course of the storm, the $L^*$ values return to a more nominal value similar to those seen in the dipole- and T89-calculated $L^*$ values.

Comparing the $K$ values in Figure 4.30 with those in Figure 4.31, we see the majority of variation between between $L^* = 4$ and $L^* = 6$. This is as expected, as the LFM/RCM version shows more activity in those $L^*$ regions. Good agreement is seen with the T89 model produced values for orbits 2 & 3, with orbit 1 being the most different. Again, this difference is due to the magnetic field dynamics involved in the MHD simulation that exceed those seen in the relatively simple T89 model.

As a final note on the comparisons of $L^*$ output, it can be seen in the plots that a 4-line convergence with LFM MHD data, the general trend of $L^*$ is captured. With the addition of the ring current in the LFM/RCM runs, the fit is even better. Although a more thorough study of
Figure 4.31: LFM/RCM $L^*$ vs. Time

4-line convergence utilizing spherical-linear interpolation would be required, the method appears to have promise as a short-cut in calculating the computationally intensive $L^*$ values.
4.4.2 LFM vs. LFM/RCM Full Field Comparisons

Utilizing GHOSTpy in parallel, it is possible to process a large number of points in a relatively short period of time. This provides an opportunity for comparisons of $K$ and $L^*$ values within an entire 3-D region of the grid. This process was utilized to construct a comparison between LFM and LFM/RCM model results. Utilizing the Python mpi4py package, an application was written on top of GHOSTpy to handle a large quantity of points in parallel. The application takes as input the LFM grid file containing the $B$ field, a radial distance, used as a limit on which points to process, and a convergence value. The convergence value can be either -1 for only 4-point convergence, or a floating point value representing the torque for convergence. From these inputs, the application opens the supplied LFM data file, extracts a list of points, and applies the radial distance bounds, selecting all points within the radius as the computation set. Each process rank independently bins the points that it is responsible for calculating, and immediately begins processing.

For each point, $B_m$ is set to the value of $|B|$ at the grid location. $K$ is then calculated from this $B_m$ and both values saved. A 4-point convergence is then performed and the resultant $L^*$ value saved. If a convergence value was initially specified, each point is subjected to a convergence routine until the specified tolerance is reached. The resultant $L^*$ value is then saved. Once all points have been calculated within a rank, a “gather-all” operation is initiated, sending all saved data to the rank 0 process. Rank 0 then re-assembles the points onto a grid and saved to disk in a VTK file format suitable for use with ParaView.

The time-step of 02 Oct 2013 at 07:16:30 UT was chosen for the following analyses, corresponds to the peak $D_{st}$ index value during the main phase of the storm. Figure 4.32 provides an overview of the magnetic fields conditions (for this time step) around Earth for both the LFM and LFM/RCM model output. These are provided as a reference while comparing invariant plots.

In Figures 4.33 through 4.36, equatorial and meridional cut planes are proved to illustrate the $K$ and $L^*$ configurations that exist within the time step for both the LFM and LFM/RCM runs. Figures 4.33 and 4.34 provide views of $K$ for pitch angles of $90^\circ$ at all grid points. The top
Figure 4.32: Side by side comparison of magnetic field magnitude, in the X/Y plane, between LFM without RCM (left) and LFM with RCM (right)

images represent the straight LFM runs, with the bottom images presenting the results utilizing LFM/RCM runs. Contour lines indicate successive whole-value L-shells ranging from $L^* = 3$ near
the inner boundary to $L^* = 10$ at the outer boundary.

A visual inspection of the two $K$ plots reveals little difference discernible with the naked eye. The northward swing of the $K = 0$ region of Figure 4.34 appears to be sharper in the LFM/RCM model than it does with the straight LFM, and a small bulge appears at around $-8R_E$ in the low $K$ region.

In the same character as the $K$ plots, Figures 4.35 and 4.36 represent the $L^*$ distributions based on the $K$ values presented in the previous figures. Again, contour lines represent the whole-number $L^*$ shells from 3 to 10. The $L^*$ distributions in Figure 4.35, for the LFM-calculated values, appears to be fairly smooth and regular, outside of a discontinuity that appears along the axis, to roughly $L^* = 8$. Beyond this point the continuity of $L^*$ seems to break down. Similar behavior is seen in the LFM/RCM generated plots, but with far less smoothness in the constant $L^*$ shells. Due to expansion outward of the $L^*$ regions, as compared to the non-RCM model version, the breakup of coherent $L^*$ regions begins just past $L^* = 6$.

Figure 4.36 shows that the smoothness of constant $L^*$ shells appears to be much more regular when moving from north to south along the $x$ axis. The discontinuity seen in the previous figures is evident along the $z = 0$ line of the $x$ axis. The north-south breakup of shell coherence can be seen in the extreme with the LFM/RCM results in this figure.

Figure 4.37 provides a view of the relative differences in $L^*$ between the two model configurations, with black contour lines representing straight LFM, and red depicting the results from using LFM/RCM data. Note that these contours show a discontinuity along the $+X$ axis. This discontinuity appears in some straight LFM runs, and in most LFM/RCM runs, with it being more pronounced in the LFM/RCM results. The nature of this discontinuity is currently unknown, but should be investigated further.

Of interest in this plot is the sunward bulging tendency of the LFM/RCM results compared to the LFM results. It can be seen, starting just past $L^* = 3$, that the forward edge of the L-shell moves radially out, with the differences being the greatest between $L^* = 5$ and $L^* = 6$. The LFM/RCM results between these two $L^*$ values completely encompass the $L^* = 6$ and $L^* = 7$
Figure 4.33: Equatorial cut-plane views of the distribution of $K$ for a pitch angle of $90^\circ$) within full field results generated with the LFM model (top), and the LFM with RCM model (bottom)
Figure 4.34: Meridional cut-plane views of the distribution of $K$ for a pitch angle of $90^\circ$ within full field results generated with the LFM model (top), and the LFM with RCM model (bottom)
sunward regions computed with the straight LFM model. It can also be seen quite clearly that the LFM/RCM constant $L^*$ shells are far more turbulent than those of the LFM model results. Areas of apparent disorder appear in the LFM/RCM tail region past $L^* = 6$. While some disorder appears in the LFM-result tail region, this phenomena is far more pronounced in the model containing a ring current.

Figure 4.38 provides a set of plots representing the actual quantitative differences between the results obtained using the LFM vs. the LFM/RCM model data. These difference plots show the least variation between the two exists inside of $L^* = 5$ as calculated with the Non-RCM model (The contour lines are the constant $L^*$ values from the straight LFM results). As distance from Earth increases, the differences progressively increase, with the largest differences seen in the tail region.
Figure 4.35: Equatorial cut-plane views of the distribution of $L^*$, generated from $K$ within the previous figures, for full field results generated with the LFM model (top), and the LFM with RCM model (bottom)
Figure 4.36: Meridional cut-plane views of the distribution of $L^*$, generated from $K$ within the previous figures, for full field results generated with the LFM model (top), and the LFM with RCM model (bottom)
Figure 4.37: Contour plots showing constant $L^*$ shells in an equatorial view. Black lines represent values calculated utilizing LFM without RCM, and the red lines depict values obtained utilizing LFM with RCM.
Figure 4.38: Difference plots showing the absolute $L^*$ differences between values computed using LFM vs. LFM/RCM as a model base.
Chapter 5

Parallel Operations and Timings

5.1 Serial Single Point Algorithm Timings

The following timings were conducted on a 12-core, 2.4 GHz, 2009 Mac Pro with 64 GB of RAM running macOS Sierra (10.12.4). The python subsystem was installed via the Homebrew package manager, and consists of the most recent version 2.7 libraries available at the time of writing. The timings consist of the averages from 500 randomly selected points (restricted to a radial distance of $8.5 \, R_E$ and bounded by $\lambda$ between $-\frac{\pi}{4}$ and $\frac{\pi}{4}$) within a quad resolution LFM simulation, with $B_m$ being set by the $|B|$ value at the selected point. $K$ is defined by this $B_m$, and $L^*$ calculated based on these values.

Figure 5.1 provides a histogram with the distribution of final $L^*$ calculated values. Figure 5.2 provides an overview of the completion time for all 500 samples. More interestingly, Figure 5.3 provides the relationship between execution time and final calculated $L^*$ values. A clear correlation exists, with an increase in computation time seeing very little variation in increases between $2.3R_E$ and $7R_E$. Beyond $7R_E$, the variation increases due to the greater complexities in outer field lines, and the associated divergence in field line trace times. This correlation provides for a good understanding to estimate execution time for $L^*$ and $K$ computations based on the approximate particle distance and latitude. For calculations with particles at low $R_E$ and low $\lambda$, it can be assumed that execution time will be in the lower end of the spectrum. Any particles with high $R_E$ or high $\lambda$ starting points will result in higher execution times. This correlation, along with the dipole analytic model for $L^*$ (Equation 3.17), can be utilized for initial load balancing when
Figure 5.1: This histogram shows the distribution of final calculated $L^*$ values within the sample set.

Figure 5.2: This histogram shows the distribution of execution time per calculation within the sample set.

calculating a large number of particles in parallel. This will be covered in sections on parallel operations.
Figure 5.3: This plot shows the relationship between execution time and calculated $L^*$ value for the sample set.

Figure 5.4 provides a graphical representation of the effects described by Figure 5.3. Notice that the largest execution times exist in the tail region (the left side of the left-hand plot). High-time regions can also be seen, to a more limited effect on north and south edges of the field depicted in the y-z plane in the right-hand image. It can also be seen from this figure that the majority of points have a relatively low computation time, indicating that a relatively few number of particles will have an undue influence on the load balancing of parallel computations. The values depicted in this figure represent calculations that have had their values converged to $10^{-3}$, thus the individual point calculation times are larger than would be expected for a 4-point convergence.

5.1.1 Single Point Computation - $L^*$ Overview

Before dealing with parallel operations, the serial execution for a single particle should be examined. Figure 5.5 provides a timing overview for the major components of a single-point 4-point convergence $L^*$ calculation. The analysis is refined to a lower level in Figure 5.6 with a breakdown of the major components of the $KBSearch$ methods. The most computationally expensive component
Figure 5.4: Graphical illustration of time required to compute $L^*$ based on location. This example originates from the main phase of the storm depicted in Chapter 4.

is then further refined with a breakdown of the field-line/KB model routines.

Looking at the top level overview in Figure 5.5, the “_init_” method represents the 500 sample average timing for a complete $L^*$ calculation converged on 4-lines. The “$l_{\text{star}}$” method represents the drift boundary magnetic flux integral plus the fixed computation for obtaining $L^*$, “add_trace” is the timing for adding a $\phi$ angle interpolation point to the drift trajectory computation. The bar representing the “_trace_from_location_” method indicates the time necessary to compute a single field line trace. Within all of the profile charts, blue bars represent cumulative time over the course of a single point execution; the orange bars represent time per call.

The average for a 4-point convergence is skewed high because points with high $B_m$, high $K$ combinations tend to take considerably more time to execute than do points of lower $B_m$ and lower $K$. This is due to the field geometry; the former situation requires extensive searching through long and distorted field lines. These timings make it clear that the majority of execution time is spent searching for the proper K/B field line fit. It is important to note that “_add_trace_” utilizes both
“\texttt{Lstar}” and “\texttt{trace_from_location}” as integral parts of the computation. Logically, “\texttt{init}” is comprised of 4 line traces and a single \( L^* \) computation. Each search progression utilizes \( L^* \) for boundary calculations, and a considerably number of field line traces to complete the search. The “\texttt{trace_from_location}” method represents the smallest grained timing.

![Bar chart]

Figure 5.5: Average execution time for calculating a single point (in serial), broken down by computational component. The blue bars indicate cumulative time per point. The orange bar indicates the execution time per call.

5.1.2 Single Point Computation - KB Line Search Routine

Focusing on the component that takes the most time to complete, Figure 5.6 provides a timing breakdown for a single line search operation, where “\texttt{search_B_adaptive}” represents the
Figure 5.6: Average execution for a KB search and its component methods. “search_B_adaptive” represents the total time (avg) for a single execution of a search. The remaining bars represent time for the given methods within a single search. Blue represents cumulative time; orange is time per call.

execution of a single line search. Again, the timings are skewed high due to the nature of high $K$, high $B_m$ points. Looking specifically at “search_B_adaptive” removes the $L^*$ component from the calculation, allowing for a more focused analysis of the search.

Within a single search the primary drivers for time expenditure are the tracing of field lines, and calculating the minimum $b_{gap}(k)$ for the lines. The smallest non-system time expenditure is “get_raw_path”, which is the method used in calculating the $\lambda$ start position for a search.
5.1.3 Single Point Computation - K/B Model

Figure 5.7 represents a refinement of timing granularity, focusing on the calculation of a field line and its associated $K/B$ model. In this figure, once again "\_init\_" represents the total time to completion for tracing the complete calculation. The method "\_trace_line\_" actually performs the Runge-Kutta 4/5 integration to calculate the field line; "\_k_mod\_" calculates $K$ values for every point on the line. This method is executed twice per line; once forward, once backward. It is clear from this plot that calculating $K$ requires the vast majority of the single line trace.

![Average Execution Time per Method Call for Field Line Trace and K Model](image.png)

Figure 5.7: Average execution for a Field line trace with associated $K$ model computation. "\_init\_" represents the full execution time for a single computation, with the other two being the major sub-components. Blue bars represent cumulative time; orange is time per call.
5.2 Parallel Operations

Using the GHOSTpy tools to calculate invariants in parallel is a fairly straightforward task. As alluded to in Chapter 4, an application was written on top of the GHOSTpy toolkit, utilizing the mpi4py message passing interface, in order to process large numbers of points. It is important to note that this application is a sample of what can be done with the system. Analyses of trapped particles may take on many different forms, requiring different sets of particles. The application written for this thesis calculates the invariants for particles within a specific radial distance of the Earth’s center. Other possible applications might include finding the $K = 0$ surface, a task that would require an iterative process to create a grid while calculating the invariants. Other tasks include, as is demonstrated in Chapter 4, the calculation of invariants for spacecraft locations. These other tasks would require a different driver application for the worker classes that have been written, for the application outlined here, to process particles in parallel.

5.2.1 Application Configuration

The application to process particles in parallel is functionally divided into two parts: a driver application that partitions the points for processing, and a worker module that oversees the computations on parallel nodes. As each particle can be computed entirely independent of all other processes, interprocess communication is, ideally, not required until nodes complete their processing tasks. Though, there may be situations where communications may be necessary to properly balance the computational load for optimal performance.

5.2.2 Parallel Scaling Efficiency

There are two types of efficiency that are normally computed for parallel applications; strong scaling and weak scaling. Strong scaling is utilized when a problem size remains fixed, and the number of processing units (processes) are increased to improve performance. This is typically the case where a problem is processor bound (takes a long time to compute a single data unit).
The goal behind strong scaling is to reduce the time required to compute a fixed size solution to a reasonable number. These problems generally incur a great deal of communications overhead, and the efficiency analysis provides a means of finding the best performance possible for a given problem. Strong scaling efficiency is given by:

\[ E_s = \frac{t_1}{N t_N} \]  

(5.1)

where \( t_1 \) represents the time to compute a solution for a work unit utilizing a single process, and the time required to compute a solution for the same work unit utilizing \( N \) processes is \( t_N \).

The concept of weak scaling involves problems where the computational workload for a single data unit is fixed to a single process, and speedup is obtained when additional data units are added, each assigned to its own process. This efficiency of weak scaling is given by:

\[ E_w = \frac{t_1}{t_N} \times 100\% \]  

(5.2)

where \( t_1 \) represents the time to complete one work unit utilizing a single process, and \( t_N \) is the length of time required to complete \( N \) work units utilizing \( N \) processes.

Due to the nature of the invariant problem (where a particle \( K \) and \( L^* \) values can be computed independently of any other particle), the best parallel model to use is weak scaling. The efficiency plots in Figure 5.8 provide an overview of the weak scaling efficiency for the parallel GHOSTpy operations running on Summit.

The efficiency plots in this figure are broken down by test run and the overall average. The tests were conducted 3 independent times, with each run selecting its particles to calculate at random. All tests ran the same number of points on all ranks. The nature of the application is such that all ranks hand their computed results back to rank 0 upon completion. Rank 0 then organizes and saves the data to disk.

From these tests, we can see that the application scales fairly well; it maintains > 60% efficiency while running on 1,536 processes. The overall degradation in efficiency is due to the final
Figure 5.8: Parallel efficiency profiles for GHOSTpy running in parallel on the Summit supercomputer. The efficiency profiles were run 3 independent times, represented by the blue, orange and green lines. The average efficiency is plotted on the heavy black line.

data gather and I/O. If these aspects were to be removed, the application would see even higher efficiency ratings.

5.2.3 Parallel Timing Profile

The time to complete execution can be viewed in 5 different segments. These include parallel communications time, final data I/O, invariant computations, point binning and point loading. In the following timings, data from the efficiency analysis above was utilized. Each core self-bins an identical set of 40 work units, loads the field data, then executes the computations with a 4-point convergence, then returns the data to rank 0 for sorting and writing to disk. Timings are taken at each point, and returned separately so as not to interfere with the communications timings. The timings for I/O include the time required to re-assemble the data into its proper grid.

Figure 5.9 provides an example timing for 1,536 cores. In this test, invariant computations make up 69.33% of the total execution time. This correlates well with the average efficiency for 1,536 cores seen in Figure 5.8. Communication time accounts for 7.84% of total execution time,
while final assembly of data and output I/O represent 12.08%. Binning comes in at 7.28% and loading grid data at 3.47%.

Figure 5.9: Average parallel execution timing breakdown for GHOSTpy parallel jobs running on 1,536 processes.

According to the profile in 5.8, efficiency decreases as the number of cores increase. The cause can be inferred by looking at the change in communications and I/O requirements as the number of processes increase. This effect is profiled in Figure 5.10. The combined communication and final I/O timings increase at a rate commensurate with the decrease in average efficiency. This indicates that, to improve overall efficiency, a parallel I/O scheme would need to be implemented. Utilizing parallel I/O would completely eliminate the need for interprocess communications, and the system would exhibit theoretical linear scaling to an unlimited number of work units. In reality, an active load balancing scheme would need to be implemented for extremely large jobs, requiring a degree of interprocess communications, albeit less than a full gather operation would require.
5.2.4 Load Balancing

For the initial GHOSTpy parallel application, passive load balancing is utilized. This type of load balance relies on each process to determine its own optimal set of points, without input from any other process. With the system profiled here, execution begins with each process rank computing its own working set of particles based on a simple modulus operation. The application is directed via command line arguments to restrict the selection of points within the LFM grid to a specified radial distance. Once the proper data file is loaded, the grid points are extracted and filtered to meet the radial distance restriction. Points are then binned and distributed to worker processes.

As not all points within the specified sphere will be calculable, there are likely to be a relatively large number of points that will be immediately rejected by the $K/B$ algorithm. This is typically caused when field particles reside on field lines that do not return to the inner boundary of the model (they terminate at the outer boundary). These particles will immediately return a $NaN$ value, and the system will proceed to the next point. Other non-calculable points include
points that reside on the inner boundary AND do not have a valid mirror point at the other pole. This will generate a $NaN$ $K$ value in the $K/B$ algorithm for the particle, and a $NaN$ $L^*$ value be immediately returned.

The final failure mode for a computation involves the initial convergence on 4 points. This occurs if a particle has a valid initial field line trace and associated $K$ and $B_m$ values, but valid lines cannot be found in all four of the convergence $\phi$ angles: $0$, $\frac{\pi}{2}$, $\pi$, and $\frac{\pi}{4}$. This failure is typical on the inner boundary and on the outer edge of computable points (where lines transition from closed to open). This failure can take considerably more time to determine that $L^*$ cannot be computed, and thus return a $NaN$.

These failures can cause the computational load to become unbalanced at high node counts, as the partitioned particles can finish at vastly different times. The $L^*$ computation timings shown in Figure 5.3 also contribute heavily to load imbalance. The figure clearly shows that, as the $L^*$ value of a particle increases, so does the complexity of calculating the invariants. Any computation bin that contains a large number of high-$L^*$ particles will exhibit longer execution times. The final load-balance is also affected by the convergence of a particle calculation. Depending on the level of convergence desired, this can add considerable expense to the calculations. This added execution time is directly related to magnetic field complexity of the region being searched, which in turn is related to its distance from Earth.

Figure 5.11 shows the practical results of this imbalance. This histogram depicts a parallel run on 1,536 cores processing 379,002 particles (all grid locations within $9.5 R_E$ on a quad resolution grid) with the application set to converge each particle to $10^{-4}$. Through a simple $(n \mod rank)$ operation, the particles were binned sequentially based on the grid data structure. No effort was made to organize particles by physical properties. The result is a clearly skewed normal distribution of execution times per process.

It can be inferred from the failure modes and calculation timing conditions, that when a particle exhibits a high dipole $L^*$ value it will also exhibit a high computed $L^*$ value, and possibly the lengthy compute-to-fail time associated. These relationships can be used for more efficient
Figure 5.11: Normalized histogram representing initial load balance for a quad resolution LFM grid, processing 379,002 particles on 1,536 cores, and calculating to a convergence of $1 \times 10^{-4}$.

binning of particles, and be used to reject particles, prior to calculation, that would exhibit a dipole $L^*$ value that exits the simulation space. This improved binning and associated filtering provides for a more efficient passive load balancing solution.

Figure 5.12 shows the results of modified binning based on the dipole $L^*$. The dipole $L^*$ value was analytically calculated for each particle and placed in a bin based on its truncated integer value. Bin contents were then evenly distributed over the available processes. It can be seen that, by utilizing this process, the execution time has been reduced from nearly 6.5 hours down to just over 3.75 hours.

A variant of this method, not yet attempted, would involve partitioning out bins in their entirety starting with the smallest bins first. Currently, assignment of points is accomplished by sequentially assigning the first point in each bin, then repeating until all bins have been emptied. By distributing the bins from smallest to largest, emptying each bin before moving on to the next, bunching of computationally intensive points can be avoided when working with a smaller number of processors.
It is important to note that these profiles represent a solution converged down to $10^{-4}$. In these convergence conditions, the amount of time required to converge any given solution is not known. The differences in convergence times lead to the normal distributions seen in these load balance plots. To ensure an optimally balanced execution while utilizing convergence, load balancing would need to be shifted from passive to active. An active solution can be achieved by designating a control processor to be responsible for handing out computation assignments to all worker processes. As each rank completes its initial set of computations, it requests a new set from the control process. If managed correctly, the load balance in this situation would be considerably improved, but at the expense of more communications during execution. The added communications expense will adversely affect overall scaling efficiency if workers request additional points for processing too often. This would have to be managed carefully.
Chapter 6

Closing Remarks

The development of this first-of-its-kind system for calculating the adiabatic invariants of geomagnetically trapped radiation within LFM MHD fields provides an easy and extensible means for scientists to interact with LFM simulation data. The understanding of how the invariant calculation behaves within this complex gridded environment has led to new questions; questions that can be answered through application of the tools within the GHOSTpy framework.

The unique 4-line convergence method outlined in this thesis is one aspect that can benefit from further exploration. It was shown that, for a single storm, the 4-line method performed well against the full convergence results for quiet times, and was in good agreement with official $K$ and $L^*$ values put out by the RBSP mission team. An exploration study should be conducted to examine the accuracy of the 4-point method over a much larger sample set. Now that the GHOSTpy tools are available, this should be a fairly straightforward exercise.

GHOSTpy was designed with expandability in mind. The system is based on a plug-in like architecture, whereby data and field-tracing classes can be sub-classed to provide new sources of input. So long as the interface requirements are met, any magnetic field model or field line integrator can be used. This allows for an opportunity to explore other models, and to test different methods of field line tracing. This architecture also separates the data interface, allowing for more robust and parallel access to data if needed.

The system also has room for improvement. As a research tool, it should forever be updated to improve its functionality. There is room for additional strong parallelization within the invariant
algorithms themselves, with the potential to reduce computation time even further. The best scaling attempted during this research utilized 3,072 cores, and processed half a million points to a 4-point convergence in just under 5 minutes. This was not replicated in the course of research due to system problems with the new Summit supercomputer, but it does illustrate the potential for future research possibilities. These tools also open the opportunity for new visualizations of the magnetosphere, offering the ability to process entire 3-dimensional portions of a grid; video imagery showing the evolution of L-shells is a real possibility.

The advent of this system opens the opportunity to explore a multitude of open science questions. Some of these questions might include: How does the choice of model for invariant calculations affect the overall distribution of particles in phase space density? Can the use of a more realistic model, such as the LFM, change the way we understand radiation belt dynamics during magnetic storms, or is the current understanding relatively insensitive to the model choice? How does the ring current affect phase space density when dealing with the LFM model as a base? Why does the 4-point convergence method seem to correlate better to the LFM/RCM storm presented in this thesis, when compared to the LFM without RCM? Is this a coincidence for this selected storm, or does it suggest an underlying physical process that could simplify the computation of $L^*$ in the future? GOSHTpy is ideally suited for exploring these, and many more, science questions. The extensibility of the system allows for the addition of new models to explore even more questions, allowing for growth as model sophistication increases.


