1994

The interplanetary current sheet at 0.72 AU

Carl Fredrik Quist
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Quist, Carl Fredrik, "The interplanetary current sheet at 0.72 AU" (1994). *Master's Theses*. 947.
DOI: <https://doi.org/10.31979/etd.zhte-anc6>
https://scholarworks.sjsu.edu/etd_theses/947

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

The Interplanetary Current Sheet at 0.72 AU

A Thesis

Presented to

The Faculty of the Department of Physics
San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Carl Fredrik Quist

October 1994

UMI Number: 1361206

Copyright 1994 by
Quist, Carl Fredrik
All rights reserved.

UMI Microform Edition 1361206
Copyright 1995, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
copying under Title 17, United States Code.

UMI

300 North Zeeb Road
Ann Arbor, MI 48103

© 1994

Carl Fredrik Quist

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF PHYSICS

A handwritten signature in cursive script, reading "Patrick Hamill", written over a horizontal line.

Dr. Patrick Hamill

A handwritten signature in cursive script, reading "Paul Gazis", written over a horizontal line.

Dr. Paul Gazis

A handwritten signature in cursive script, reading "Leslie J. Tomley", written over a horizontal line.

Dr. Leslie Tomley

APPROVED FOR THE UNIVERSITY

A handwritten signature in cursive script, reading "Serena K. Stanford", written over a horizontal line.

ABSTRACT

The Interplanetary Current Sheet at 0.72 AU

by Carl Fredrik Quist

The interplanetary current sheet is a discontinuity in the interplanetary solar magnetic field. The discontinuity results from the solar wind pulling the Sun's magnetic field lines outward in a radial direction. This phenomena is known as "Frozen-in magnetic lines of force." The interplanetary current sheet has a complex structure which is discussed in some detail in Sec. I. By examining data collected by the Pioneer Venus Orbiter, it is possible to determine at what time and where in space the satellite crossed the current sheet. It is also possible, using a technique developed by Sonnerup and Cahill, to determine the orientation of the current sheet in space. Analyzed data from 2 six month periods led to finding 6 well defined current sheet crossings whose normal vectors are presented in Sec. IV.

ACKNOWLEDGMENTS

The pursuit of a Ph.D. is a long and difficult process. Up to this point, I have completed my bachelors degree at UCSB and my masters degree at SJSU. The final phase of my education will be done at Lund University in Lund, Sweden. The prospect of attending school in Europe is both exciting at frightening at the same time. I look forward to the adventure.

There are so many people who have made my educational quest possible and enjoyable. First, I want to thank my parents Agneta and Dave for their emotional and financial support over these (way too many) years. I could not have made it without you. I am forever in your debt and you are forever in my heart. My sister Maria and brother John have also been of great emotional support to me.

Dr. Phil Lubin of UCSB, my undergraduate advisor, was the first to introduce me to research while designing and building an automated telescope system. Dr. Patrick Hamill is my current advisor at SJSU. Thank you PJ for your support, your friendship, and for introducing me to NASA Ames Research Center. Thank you Paul Gazis and Aaron Barnes for the opportunity to work with you at NASA.

To Mary Ann: Thank you for opening up my heart and my mind.

The following list of people should be thanked for their friendship, their help, and their ability to make my life a brighter place.

UCSB friends: James², Joe, Vinny, Anh, and Michael.

SJSU friends: Don, Chuck, and Selwyn.

Important others: Carol, Steve², Kim, Baylee (woof-woof), Pierce, Mark, Tina, and Cassy.

TABLE OF CONTENTS

I.	FORMATION OF THE CURRENT SHEET	1
A.	Introduction	1
B.	Solar wind facts	2
C.	Dynamics of the solar wind	3
	1. <i>Static equilibrium</i>	4
	2. <i>Stationary expansion</i>	8
D.	Frozen-in concept	11
	1. <i>Infinite conductivity</i>	12
	2. <i>Finite conductivity</i>	16
E.	Interplanetary current sheet	21
	1. <i>Non-rotating Sun</i>	21
	2. <i>Rotating Sun</i>	24
II.	PIONEER VENUS DATA AND HARDWARE	29
A.	The data set	29
B.	PVO hardware	32
	1. <i>The magnetometer</i>	33
	2. <i>The plasma analyzer</i>	35
III.	DETERMINATION OF THE CURRENT SHEET	38
A.	A multi-step process	38
B.	Locating approximate crossing times	38
C.	Using the high resolution data	43
D.	Determining the current sheet normal	46
E.	Graphical results	51
IV.	THE RESULTS	56
A.	Dates and times to be determined	56
B.	Locating approximate times	56
C.	Locating actual times	56
D.	Results and conclusions	58

REFERENCES	69
APPENDIX A	70
APPENDIX B	73
APPENDIX C	79
CONVERTOUT.FOR	80
PARK.FOR	83
BxxxPLOT.FOR	86
MATRIX.FOR	103
APPENDIX D	117
APPENDIX E	121

LIST OF TABLES

1	Dates of Potential Current Sheet Crossings	57
2	Dates of Well Defined Current Sheet Crossings	58
3	The Calculated Current Sheet Normals	58

LIST OF FIGURES

1	Volume of Gas in Equilibrium	5
2	Plot of Solar Wind Velocity-vs-Distance	11
3	Surface Moving Through a Magnetic Field	13
4	Effect of a Plasma on Magnetic Field Lines	17
5	Dipole View of the Sun	22
6	Radial Magnetic Field Lines from the Sun	23
7	Polar View of Magnetic Field Lines	25
8	Computerized Current Sheet	27
9	Artist's Conception of the Current Sheet	27
10	Example of a Current Sheet Crossing	28
11	Coordinate System of Received Data	31
12	Coordinate System as Used	33
13	Magnetometer	34
14	Plasma Analyzer	36
15	Spiral Form of Magnetic Field Lines	40
16	Output from PARK.FOR	43
17	Output from BxxxPLOT.FOR	45
18	Gaussian Pillbox for a Single Boundary	47
19	Gaussian Pillbox for a Multiple Boundary	48
20	Tangential Discontinuity	52
21	Rotational Discontinuity	53
22	Rotational Nature of the Discontinuity	55
23	Crossing of Date 84096	60
24	Crossing of Date 84111	61
25	Crossing of Date 86256	62
26	Crossing of Date 86338	63
27	Crossing of Date 87011	64
28	Crossing of Date 87066	65
29	Magnetic Field Structure of Crossing 84096	66
30	Magnetic Field Structure of Crossing 84111	66
31	Magnetic Field Structure of Crossing 86256	67
32	Magnetic Field Structure of Crossing 86338	67
33	Magnetic Field Structure of Crossing 87011	68
34	Magnetic Field Structure of Crossing 87066	68

I. FORMATION OF THE CURRENT SHEET

A. Introduction

The Pioneer Venus Orbiter (PVO) Satellite orbited Venus from December 1978 until December 1992. This satellite mission is considered one of NASA's most successful missions, in part, because of the satellite's longevity. Designed only to operate until the early to mid 1980's, the PVO successfully orbited Venus for 8 years beyond the expected lifetime of the satellite. The mission was the first to extensively explore another planet in our solar system. Using technology similar to that found on earthbound weather satellites, Venus' surface, weather, and cloud patterns were examined in detail. In addition to the instruments dedicated to the study of Venus, a solar plasma analyzer and a magnetometer were used to study the solar wind and the solar magnetic field. These two instruments also allow us to study the structure of the interplanetary current sheet.

The interplanetary current sheet is a discontinuity in the Sun's interplanetary magnetic field. This discontinuity, described later in detail, is a dividing plain between the interplanetary northern and southern magnetic field regions which are generated by the Sun. The division of these two hemispheres is near the plane of the ecliptic. These two distinct magnetic regions are formed from the interaction between the solar wind, a plasma, and the Sun's magnetic field. The goal of Sec. I is to give the background physics needed to understand the formation of the current sheet.

Understanding the dynamics of the solar wind, the concept of “frozen-in magnetic lines of force,” and the interaction between the solar wind and the solar magnetic field are all important in understanding the formation of the interplanetary current sheet. E. N. Parker was the first to correctly predict the dynamics of the solar wind by examining the equations of motion for the Sun’s coronal particles. Parker’s results [1] will be given in detail in Sec. I.C. The concept of “frozen-in magnetic lines of force” is a well known phenomenon in plasma physics. The term "frozen-in" comes from the way a moving plasma pulls or drags magnetic field lines in the direction of plasma motion. This concept will be presented in Sec. I.D. Section I.E will describe the actual formation of the current sheet in interplanetary space.

B. Solar wind facts

Before describing the dynamics of the solar wind, a brief list of facts may prove helpful. The solar wind is an ionized gas ejected radially from the Sun. The composition of the solar wind is about 95% protons and electrons and about 5% doubly ionized helium nuclei, though trace amounts of other elements can also be found. The average plasma density is 5 particles/cm³ at a distance of 0.72 astronomical units, the distance from the Sun to Venus. The mean velocity of the solar wind is between 400 km/sec and 500 km/sec. The wind has been observed to be as slow as 200 km/sec and as fast as 900 km/sec. The plasma temperature is between 100,000 K and 150,000 K for protons and electrons and about 4 to 5 times that for the helium nuclei.

C. Dynamics of the solar wind

In 1962, the Venus probe, Mariner 2, confirmed the existence of the solar wind that E. N. Parker had predicted four years earlier. Prior to the Mariner 2 satellite, there was considerable speculation concerning whether or not a gas could be continuously ejected from the Sun. Another point of concern was whether or not enough energy could be supplied to the particles to achieve escape velocity from the Sun's gravitational pull. The escape velocity from the Sun is 500 km/sec.

Parker's work was not the first to address the issue of a solar wind. Parker described previous work done by L. Biermann who studied the motions of several comet tails, as well as the motions of structures within the comet tails [2]. After these observations, he concluded that there must be a gas flowing radially outward from the Sun. Furthermore, Biermann saw no evidence for an inward flow of particles, which led to the conclusion that the solar wind particles must reach escape velocity. Biermann suggested that the solar wind velocity in interstellar space was between 500 km/sec and 1500 km/sec. Parker's goal was to give a mathematical explanation of Biermann's observations.

Parker had two approaches for demonstrating the existence of the solar wind. First, he showed that it was physically impossible for there to be *no* solar wind. This model, which Parker referred to as the “static equilibrium” model, will be covered in Sec. I.C.1 of this paper. Second, after concluding that the solar wind must exist, he derived the

basic physics needed to understand the solar wind and what the expected solar wind velocities might be. Parker referred to this model as “stationary expansion” and it will be covered in Sec. I.C.2 of this paper. Though this model was simple, it correctly predicted the average speeds of the solar wind as a function of distance from the Sun. Subsequent models based on more complicated assumptions [3] will not be covered in this paper.

1. *Static equilibrium*

If a small cylinder of solar ionized gas is in equilibrium, as shown in Fig. 1, the forces on the volume of gas must be balanced. Parker assumed just such a stationary state for the coronal gas in the static equilibrium model. The only upward force on the disk is PA , where P is the pressure on the bottom of the cylinder, and A is the area on the top and bottom of the cylinder. The downward forces on the disk are dW and $(P + dP)A$, where dW is the weight of the gas, and $P + dP$ is the pressure on the top of the cylinder. In equilibrium, the net upward force must balance the net downward force. The net horizontal force is zero by symmetry. The balanced forces are,

$$PA = (P + dP)A + dW . \quad (1.1)$$

The gas has a mass density ρ and the local value of the gravitational acceleration is g . If $(\rho g A)dr$ is substituted in for the weight of the disk and if we cancel the PA terms, we get

$$0 = dP + (\rho g)dr . \quad (1.2)$$

The mass density ρ can be expressed as Nm , where N is the hydrogen particle density, and m is the mass of a single hydrogen atom. Also, the gravitational constant g can be expressed as MG/r^2 , where M is the mass of the Sun, G is the gravitational constant, and r is the radial distance from the center of the Sun. Dividing equation 1.2 through by dr and substituting in the values for g and ρ , one gets

$$0 = \frac{dP}{dr} + \frac{GMmN}{r^2} . \quad (1.3)$$

FIG. 1. A small volume of solar plasma in equilibrium. The forces on the element of solar plasma are shown. The net horizontal force is zero by symmetry.

Using the ideal gas law for the pressure, the resulting equation is

$$0 = \frac{d}{dr}(2NkT) + \frac{GMmN}{r^2}, \quad (1.4)$$

where T is the temperature, and k is the Boltzmann constant. The factor of 2 in the $(2NkT)$ term of equation 1.4 can be explained by the fact that the hydrogen gas is ionized, giving twice as many particles as neutral hydrogen and, therefore, twice the pressure.

In order to solve equation 1.4 for N , the particle density, as a function of distance, a general form of the temperature must be found. It is reasonable to assume the only particles capable of becoming part of the solar wind must be near the edge of the Sun. At this radial distance, defined as $r = a$, local heating stops because there are no more local heat sources. Therefore, the heat-flow coming to the particles is from the Sun's interior and is assumed to be symmetric and steady-state. The steady-state heat-flow equation [1], which is similar to the steady-state diffusion equation [4], is

$$0 = \nabla \cdot [\kappa(T) \nabla T], \quad (1.5)$$

where $\kappa(T)$ is the thermal conductivity. For ionized hydrogen of the densities we are considering, the thermal conductivity is approximately equal to $(5 \times 10^{-7}) \cdot T^n$ [1]. The differential equation (1.5) can be solved by assuming a solution of the form $T(r) = cr^d$,

where c and d are constants to be determined. Substituting the forms of $T(r)$ and $\kappa(T)$ into equation 1.5, the temperature is found to be

$$T(r) = T_o \left(\frac{a}{r} \right)^{1/(n+1)}, \quad (1.6)$$

where a is the radius where local heating stops, T_o is the temperature at $r = a$, and n is an experimentally determined plasma constant equal to 2.5 for ionized hydrogen [1].

Using the form of the temperature in 1.6, the solution to equation 1.4, (see Appendix A), is found by direct integration. The resulting particle density as a function of distance is

$$N(r) = N_o \left(\frac{r}{a} \right)^{1/(n+1)} \exp \left\{ \left[\frac{\lambda(n+1)}{n} \right] \left[\left(\frac{a}{r} \right)^{n/(n+1)} - 1 \right] \right\}, \quad (1.7)$$

where λ is the dimensionless parameter $GMm/2kaT_o$.

It is convenient to rewrite equation 1.7 in terms of pressure by substituting in the temperature ratio $T/T_o = (a/r)^{1/(n+1)}$ and the ideal gas law $P = 2NkT$, where the factor of 2 was previously explained. The subsequent equation for the pressure as a function of distance is

$$P(r) = P_0 \exp \left\{ \left[\frac{\lambda(n+1)}{n} \right] \left[\left(\frac{a}{r} \right)^{n/(n+1)} - 1 \right] \right\}. \quad (1.8)$$

The pressure as r approaches infinity becomes

$$P(\infty) = P_0 \exp \left[\frac{-\lambda(n+1)}{n} \right]. \quad (1.9)$$

Using the values 1.5×10^6 K for T_0 and 10^{11} cm for a . The value of λ is 5.35. The calculated value for the pressure at $r = \infty$ is 0.6×10^{-5} dynes/cm².

It seems obvious, however, that if the Sun were the only source of particles, as assumed in this model, the pressure would vanish at large distances. In addition, Parker knew that the value of gas pressure in interstellar space was approximately 1.4×10^{-13} dynes/cm², seven orders of magnitude less than that calculated at infinity. These points contradict the value at infinity calculated above. The conclusion was that the static equilibrium model gave non-physical results and, therefore, the solar wind must exist.

2. Stationary expansion

Having concluded that the solar wind must exist, Parker then predicted what solar wind velocities might be expected. Assuming the net force on the solar particles was non-

zero, a term accounting for particle acceleration was included in equation 1.1. Our new equation, ($F = ma$), is

$$(NmAdr)v \frac{dv}{dr} = PA - (P + dP)A - dW, \quad (1.10)$$

where the acceleration is $dv/dt = (dv/dr)(dr/dt) = v(dv/dr)$ and the mass of the disk is $NAm(dr)$. The definitions for N , A , and m are the same as in Sec. I.C.1. Subtracting the two PA terms and substituting in $(\rho g A)dr$ for the weight, we get

$$(NmAdr)v \frac{dv}{dr} = -dPA - (\rho g A)dr. \quad (1.11)$$

Dividing through by Adr , equation 1.11 becomes

$$Nm v \frac{dv}{dr} = -\frac{dP}{dr} - \rho g. \quad (1.12)$$

Putting in the same definitions for P , ρ , and g as in the previous section, we get the equation of motion

$$Nm v \frac{dv}{dr} = -\frac{d}{dr}(2NkT) - \frac{GMmN}{r^2}. \quad (1.13)$$

Again, we will assume a steady-state heat flow and that all heating occurs in a region of radius less than a .

Parker found it useful to make equation 1.13 dimensionless, giving a dimensionless solution. The solution to equation 1.13 can be shown to be, (see Appendix B),

$$\psi - \ln \psi = -3 - 4 \ln \frac{\lambda}{2} + 4 \ln \xi + \frac{2\lambda}{\xi}. \quad (1.14)$$

Where ξ is the reduced distance r/a , ψ is the dimensionless kinetic energy $mv^2/2kT_o$, and λ is the dimensionless potential energy defined earlier. Figure 2 shows Parker's calculated outward gas velocities $v(r)$ as a function of ξ for various temperatures T_o at $r = a$. At temperatures on the order of one million degrees, the predicted solar wind velocities were between 200 km/sec and 800 km/sec. A temperature of one million degrees was sufficient to give the particles a velocity of 500 km/sec at 1 AU, which is equal to the solar escape velocity.

As previously stated, the existence of the solar wind was experimentally verified in 1962, and the observed solar wind velocities were within the range of Parker's predictions. Parker's model is still a basic starting point for many solar wind calculations. Having shown a solar wind does exist, we next consider how it might affect the Sun's magnetic field.

FIG. 2. A plot of the solar wind velocities versus radial distance form the Sun. Since the velocity is a function of initial ($r=a$) temperatures, several possible temperatures are shown.

D. Frozen-in concept

The solar wind consists of ionized particles moving approximately radially outward from the Sun. The study of the magnetohydrodynamic interaction between moving ionized particles and magnetic field lines yields a concept known as "frozen-in magnetic lines of force." This section (I.D) gives the mathematics needed to understand this concept. The first derivation assumes that the conductivity in a plasma is infinite, (Sec. I.D.1), while the second derivation assumes a finite conductivity (Sec. I.D.2).

1. Infinite conductivity

Because the solar wind is an ionized gas, it is obvious that the solar wind has a large conductivity. For ease in initial calculations, we assume the conductivity is infinite. This will allow us to understand the physics of “frozen-in magnetic lines of force” without having to resort to solving the diffusion equation.

In order to understand the frozen-in concept, it is necessary to understand how the time rate of change of magnetic flux behaves on a moving surface [5]. Figure 3 shows a surface moving with velocity $\mathbf{v}$. Recalling that magnetic flux is defined by $\Phi = \int \mathbf{B} \cdot d\mathbf{s}$. The change in flux through the surface in time dt is

$$\frac{d}{dt} \int \mathbf{B} \cdot d\mathbf{s} = \frac{1}{dt} \left[\int (\mathbf{B}_{t+dt} \cdot d\mathbf{s}_{t+dt}) - \mathbf{B}_t \cdot d\mathbf{s}_t \right] = \frac{1}{dt} \left[\int (\mathbf{B}_{t+dt} \cdot d\mathbf{s}_2 - \mathbf{B}_t \cdot d\mathbf{s}_1) \right], \quad (1.15)$$

where s_1 is the surface at time t , s_2 is the surface at time $t + dt$, and $\mathbf{B}_t$ and $\mathbf{B}_{t+dt}$ are the magnetic fields at times t and $t + dt$, respectively.

The magnetic field $\mathbf{B}$ at time $t + dt$ can be expressed in terms of time t by using Taylor's theorem,

$$\mathbf{B}_{t+dt} = \mathbf{B}_t + \frac{\partial \mathbf{B}}{\partial t} dt + \dots \quad (1.16)$$

Substituting in the above expansion into 1.15 for $\mathbf{B}_{t+dt}$ and rearranging, we get

$$\frac{d}{dt} \int \mathbf{B} \cdot d\mathbf{s} = \frac{1}{dt} \left[\int (\mathbf{B}_t \cdot d\mathbf{s}_2 - \mathbf{B}_t \cdot d\mathbf{s}_1) \right] + \int \frac{\partial \mathbf{B}_t}{\partial t} \cdot d\mathbf{s}_2. \quad (1.17)$$

FIG. 3. An imaginary surface moving with velocity $\mathbf{v}$. The magnetic field $\mathbf{B}$ is shown crossing through the surface at time t and time $t + dt$.

It is also possible to find the flux through the whole cylinder shown Fig. 3. Assuming the magnetic field is constant and equal to the value at time t , the total flux through the cylinder is

$$\int (\nabla \cdot \mathbf{B}_t) d\tau = \int \mathbf{B}_t \cdot d\mathbf{s}_2 - \int \mathbf{B}_t \cdot d\mathbf{s}_1 - \int \mathbf{B}_t \cdot (\mathbf{v}dt \times d\mathbf{l}), \quad (1.18)$$

where $\mathbf{v}$ is the velocity of the surface, $d\tau$ is a volume element, and $d\mathbf{l}$ is a path element about the surface s . The three terms on the right hand side represent the flux through the bottom, top, and wall of the cylinder.

Rearranging the terms in equation 1.18, we get

$$\int \mathbf{B}_t \cdot d\mathbf{s}_2 - \int \mathbf{B}_t \cdot d\mathbf{s}_1 = \int (\nabla \cdot \mathbf{B}_t) d\tau + \int \mathbf{B}_t \cdot (\mathbf{v} d\mathbf{l} \times d\mathbf{l}). \quad (1.19)$$

Note that $\mathbf{B}_t \cdot (\mathbf{v} d\mathbf{l} \times d\mathbf{l})$ is equal to $(\mathbf{B}_t \times \mathbf{v} d\mathbf{l}) \cdot d\mathbf{l}$. Using this in equation 1.19 and then applying Stokes' law equation 1.19 becomes

$$\int \mathbf{B}_t \cdot d\mathbf{s}_2 - \int \mathbf{B}_t \cdot d\mathbf{s}_1 = \int (\nabla \cdot \mathbf{B}_t) d\tau + \int (\nabla \times \mathbf{B}_t \times \mathbf{v}) \cdot d\mathbf{s}_2. \quad (1.20)$$

Notice that the left hand side of 1.20 can be substituted directly into equation 1.17. This substitution leads to the equation

$$\frac{d}{dt} \int \mathbf{B} \cdot d\mathbf{s} = \frac{d}{dt} \int (\nabla \cdot \mathbf{B}) d\tau + \int (\nabla \times \mathbf{B} \times \mathbf{v}) \cdot d\mathbf{s}_2 + \int \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{s}_2. \quad (1.21)$$

The first term on the right hand side represents a change due to an inhomogeneous field in which flux lines are generated, which is zero in our solar wind problem. The second term

of this expression represents the flux loss across the boundary of the moving surface. The last term is due to the time rate of change of the magnetic field.

Therefore, in our application, the time rate of change of the flux through a moving surface is given by,

$$\frac{d}{dt} \int \mathbf{B} \cdot d\mathbf{s} = \frac{d\Phi}{dt} = \int \left[\frac{\partial \mathbf{B}}{\partial t} + \nabla \times (\mathbf{B} \times \mathbf{v}) \right] \cdot d\mathbf{s}. \quad (1.22)$$

If the only force on our solar particles was due to an electric field, the expression for the current density would be $\mathbf{j} = \sigma \mathbf{E}$. However, the solar particles are near the Sun's magnetic field which produces secondary particle motion due to the Lorentz force. Therefore we need to add a term to the above current density equation. Applying this Lorentz force, our new expression for the current density becomes $\mathbf{j} = \sigma(\mathbf{E} + \mathbf{v} \times \mathbf{B})$ [5]. If one solves for the electric field $\mathbf{E}$ in the new expression for the current density and substitutes it into Faraday's Law, $\nabla \times \mathbf{E} = -\partial \mathbf{B} / \partial t$, one obtains

$$\nabla \times \left(\frac{\mathbf{j}}{\sigma} - \mathbf{v} \times \mathbf{B} \right) = -\frac{\partial \mathbf{B}}{\partial t}. \quad (1.23)$$

If the conductivity is taken to be infinite, equation 1.23 simplifies to

$$-\frac{\partial \mathbf{B}}{\partial t} + (\nabla \times \mathbf{v} \times \mathbf{B}) = \frac{\partial \mathbf{B}}{\partial t} + (\nabla \times \mathbf{B} \times \mathbf{v}) = 0. \quad (1.24)$$

Note that equation 1.24 is the integrand in equation 1.22. The time rate of change of the flux through the moving surface moving with velocity $\mathbf{v}$ is,

$$\frac{d\Phi}{dt} = 0. \quad (1.25)$$

Therefore, a surface, moving in the direction and speed of the solar wind, must have all the magnetic field lines remain constant through that surface, (see Fig. 4). The conductivity of the solar wind is extremely high, and to a good approximation, can be taken to be infinite. Therefore, the magnetic field lines of the Sun would be pulled out radially.

2. Finite conductivity

The conductivity of the plasma, though high, is not infinite. As a starting point, let us use the equation for the current density, $\mathbf{j} = \sigma(\mathbf{E} + \mathbf{v} \times \mathbf{B})$. Solving for $\mathbf{E}$ and taking the curl of both sides, one obtains,

$$\nabla \times \mathbf{E} = \left(\nabla \times \frac{\mathbf{j}}{\sigma} - \nabla \times \mathbf{v} \times \mathbf{B} \right). \quad (1.26)$$

Plugging in Faraday's equation, $\left(\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \right)$, into equation 1.26, we get

$$\nabla \times \frac{\mathbf{j}}{\sigma} - \nabla \times \mathbf{v} \times \mathbf{B} = -\frac{\partial \mathbf{B}}{\partial t}. \quad (1.27)$$

FIG. 4. The imaginary surface moving with the same speed and direction as the solar wind forces the magnetic field lines to be parallel to the solar wind direction. This is the Frozen-in Concept.

It is advantageous to rewrite equation 1.27 in terms of the magnetic field $\mathbf{B}$ only by substituting in Ampere's equation for the current density $\mathbf{j}$. Ampere's equation is,

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{d\mathbf{E}}{dt}. \quad (1.28)$$

Assuming the displacement current dE/dt is negligible [6], equation 1.28 simplifies to $\mathbf{j} = (\nabla \times \mathbf{B})/\mu_0$. Plugging into equation 1.27, we get

$$-\frac{\partial \mathbf{B}}{\partial t} = \frac{\nabla \times \nabla \times \mathbf{B}}{\sigma \mu_0} - \nabla \times \mathbf{v} \times \mathbf{B}. \quad (1.29)$$

Note that $\nabla \times \nabla \times \mathbf{B} = -\nabla^2 \mathbf{B}$ and that the last term of equation 1.29 can be taken as zero if one considers the magnetic field to be moving instead of the plasma ($\mathbf{v} = 0$) [6]. The resulting equation is the diffusion equation.

It is convenient to examine this diffusion equation in order to get a time dependent picture of the solar magnetic field-solar wind interaction. The use of the diffusion equation seems appropriate if we compare it to what we know about heat flow. In the same way heat diffuses through a bar with some thermal conductivity, the solar magnetic field diffuses through a plasma with some electrical conductivity. The magnetic field diffusion equation is

$$\nabla^2 \mathbf{B}(\mathbf{r}, t) = \sigma \mu_0 \frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t}. \quad (1.30)$$

It is convenient to use separation of variables in order to solve this diffusion equation [7]. The total solution is split into spatial and temporal parts, $\mathbf{B}(\mathbf{r},t) = \mathbf{X}(\mathbf{r})T(t)$. Therefore, we can rewrite the magnetic diffusion equation as

$$\nabla^2 \mathbf{X}(\mathbf{r})T(t) = \sigma\mu_o \frac{d}{dt} \mathbf{X}(\mathbf{r})T(t). \quad (1.31)$$

Dividing through by $\mathbf{X}(\mathbf{r})T(t)$ and simplifying the notation, we get

$$\frac{1}{\mathbf{X}} \nabla^2 \mathbf{X} = \frac{\sigma\mu_o}{T} \frac{dT}{dt}. \quad (1.32)$$

The right hand side of equation 1.32 is a function only of position while the left hand side is a function only of time. For this to be true, both sides must equal a constant. Therefore, we get

$$\frac{1}{\mathbf{X}} \nabla^2 \mathbf{X} = \frac{\sigma\mu_o}{T} \frac{dT}{dt} = -\frac{1}{\ell^2}, \quad (1.33)$$

where ℓ is a characteristic length [3,5,6]. Solving the time dependent part, we get

$$\int \frac{dT}{T} = -\int \left(\frac{1}{\sigma \mu_o \ell^2} \right) dt. \quad (1.34)$$

The solution to the time dependent part of the diffusion equation can be found by simple integration. The solution is

$$T(t) = \exp\left(\frac{-t}{\sigma \mu_o \ell^2}\right), \quad (1.35)$$

0

where t is the time. Note that the time dependent part of the diffusion equation equals unity if the conductivity is infinite, giving a time independent solution where the magnetic field lines would be totally ‘frozen-in.’

If the conductivity is not taken to be infinite, the time dependent part would not disappear and the solution for the magnetic field would be,

$$\mathbf{B}(r, t) = \mathbf{X}(r) \exp\left(\frac{-t}{T_m}\right), \quad (1.36)$$

where $T_m = \sigma \mu_o \ell^2$. The time needed for the magnetic field to diffuse a distance ℓ through the surface is T_m .

If we take the ratio of the time needed for the magnetic field to diffuse some distance ℓ to the time needed for the solar wind to travel the same distance ℓ , we get a

ratio that represents how “frozen-in” the magnetic field lines are. Using the values of $\ell = 10^{13}$ cm, $\sigma = 2 \times 10^{-14}$ T^{3/2} (c.g.s. units) [3], $T = 10^5$ K, and $v = 5 \times 10^7$ cm/sec, the ratio is on the order of 10^{16} . This large of a ratio implies the Sun’s magnetic field is almost completely “frozen-in.”

E. Interplanetary current sheet

With an understanding of the how the solar wind is formed and how a plasma affects magnetic field lines, we now examine how the solar wind influences the Sun’s own magnetic field. In Sec. I.E.1, we ignore the rotation of the Sun, while in Sec. I.D.2 we include the rotation of the Sun. This will lead to an accurate description of the interplanetary current sheet.

1. Non-rotating Sun

The Sun possesses a magnetic field. Because there are no magnetic monopoles, the magnetic field lines leaving the Sun’s magnetic north pole must return to the Sun’s magnetic south pole. The magnetic field structure of the Sun is, however, very dynamic and complicated. For the ease of explaining the concepts of this thesis, the Sun will be idealized as magnetically static and as being similar to a magnetic dipole. If the solar wind had no influence on this static solar magnetic field, the magnetic field lines would take on the typical dipole shape, as shown in Fig. 5.

The frozen-in concept demonstrated that a plasma moving with velocity v pulls magnetic field lines with it. In the case of the Sun, the solar wind is a plasma which pulls the Sun's magnetic field lines out radially (see Fig. 6). After examining Fig. 6, it becomes obvious that there is a large change in the direction of the magnetic field near the magnetic equator. Two separate hemispheric regions of magnetic field orientation are formed, where the upper half consists of magnetic field lines pointing radially outward and the lower half consists of magnetic field lines pointing radially inward. The magnetic field lines of the solar dipole (shown in Fig. 5) also changed from outward to inward at the equator, but the change was less abrupt.

FIG. 5. If the solar wind had no effect on our idealized Sun, the magnetic field lines would take on a typical dipole shape.

Near the magnetic equator (see dark dashed line in Fig. 6) there lies a discontinuity in the magnetic field. If we assume the magnetic equator is a straight line, the discontinuity between the two magnetic regions would be a two dimensional plane near the ecliptic.

FIG. 6. The effect of the solar wind pulling the magnetic field lines in a radial direction, assuming a non-rotating Sun. The dark dashed line is the ideal "flat" magnetic equator has a flat two dimensional current sheet associated with it. The solid dark line is the "real" magnetic equator forms a three dimensional current sheet.

However, the solar magnetic field and the magnetic equator are both dynamic systems. The solar magnetic equator is not a straight line circling the Sun, but is a curved

line resembling a wave oscillating about the “straight” equator, as shown in Fig. 6. Therefore, the true discontinuity (the solid dark line in Fig. 6) would lie not in a flat plane but would lie in a three dimensional plane. Before showing sketches of the current sheet, we must consider how the rotation of the Sun affects the magnetic field lines.

2. Rotating Sun

Up to this point, the rotation of the Sun has not been considered. The Sun completes one revolution about every twenty seven days. How does this rotation affect the Sun’s magnetic field at interplanetary distances?

The magnetic field lines are continuous. As the solar wind drags the magnetic field lines radially outward, the magnetic lines beneath the Sun's surface are continuously moved horizontally due to the Sun's rotation. This continuous process of radial and horizontal motion of the magnetic field lines results in the magnetic field lines taking on the form of an Archimedes' spiral. The formation of the magnetic field lines into this spiral shape is the same as the spiral formed by a water sprinkler (see Fig. 7).

Near the Sun's surface, the lines are nearly radial in direction but become more nearly tangential to the surface at greater distances. As will be shown in Sec. III, the Sun's magnetic field lines, at the distance of Venus, are 35° from the radial direction. At the distance of the Earth and Jupiter, the lines are 45° and 75° from radial, respectively.

The oscillations in the current sheet will also take on this same spiral form. Figures 8 and Fig. 9 give a computer generated model and an artist’s illustration of the

current sheet using our idealized static and dipolar magnetic field. Note in Fig. 9 that the planetary orbits out to Jupiter are plotted and that the vertical extent of the current sheet can be enormous [8].

As will be discussed later, we are looking for the times when and locations where the PVO, or other satellites, crossed the current sheet. Actually, the current sheet rotates every twenty seven days with the Sun and, therefore, the current sheet sweeps over the satellite. By looking for changes azimuthal magnetic field direction changes of about 180° , as in Fig. 10, it is possible to find these crossing times and locations. Note the large change in the BAZM plot between the 4th and 5th hours.

FIG. 7. A polar view of the northern region of interplanetary magnetic field. The field lines are curved from the solar rotation. The magnetic field lines above and below the rotational equator would spiral outward in helical fashion, while the magnetic field lines at the rotational equator would lie in a plane.

In Sec. III.D, a vector minimization technique developed by Sonnerup and Cahill [9] will be discussed. Using this technique, one can determine the orientation of the current sheet at the location of the crossing. The results given in this thesis using the vector minimization technique will be compared to the results found by P. Gazis [10].

It should be noted that this current sheet is now a well accepted phenomenon. Interplanetary magnetic measurements from several satellites, including Pioneers 10 and 11, and Pioneer Venus, have confirmed the existence of the current sheet from a closest distance of .5 AU to beyond the orbit of Jupiter (5.2 AU). Up to this point in time, all magnetic field and solar wind measurements have been near the plane of the ecliptic. However, a future NASA satellite will travel about the Sun in a polar orbit to check for the existence of the solar wind and the structure of the magnetic field far above the plane of the ecliptic.

FIG. 8. A computer model of a current sheet in interstellar space. Note the spirals formed by the rotation of the Sun.

FIG. 9. An illustration of the current sheet. Several planet orbits have been shown in order to give the vertical size of the ripples in the current sheet. Again, the spiral nature of the field lines is apparent.

FIG. 10. A three hour plot of the PVO high resolution magnetic field data. Note the change from about 350° to 140° in the azimuthal magnetic field direction (BAZM) between the 4th and 5th hour. The data is from the 96th day of 1984. The vertical dashed line is when the crossing took place.

II. PIONEER VENUS DATA AND HARDWARE

A. The data set

Section II.A describes the Pioneer Venus Orbiter data sets. Included are descriptions of what data sets are used, where the data are stored, and what coordinate systems are used. Section II.B describes some of the PVO hardware needed for current sheet studies.

The PVO satellite collected an enormous amount of data during its fourteen years of operation. Images of cloud patterns, upper atmosphere wind velocities, and vertical pressure profiles are but a few of the data sets available for study and interpretation. The current sheet crossings, being formed by the interaction between the solar wind and the Sun's magnetic field, are found and studied by analyzing the data collected by the PVO plasma analyzer and PVO magnetometer.

The question arises, "How does the magnetometer read the solar magnetic field and not the magnetic field of Venus?" It is the answer to that question that makes Venus a perfect location for solar magnetic field and current sheet study. Venus has no, or a very small, magnetic field and, therefore, what is registered by the magnetometer is almost entirely due to the Sun.

Dr. Chris Russell, of the Institute of Geophysics and Planetary Physics, University of California, Los Angeles (UCLA), has been put in charge of storing and distributing the solar wind and solar magnetic field data collected by the PVO satellite.

Two coordinate systems are available when ordering data sets, the Spacecraft coordinate (SC) system and the Venus-Solar-Orbital (VSO) coordinate system. The orientation of the VSO coordinate system, the most common and easily used system, has the X component positive in the direction of the Sun measured along the Venus-Sun line, the Y component positive in the direction opposing Venus's orbital motion and lying in the orbital plane of Venus, and the Z component positive in the direction parallel to the pole of the Venus orbital plane and positive in the northward direction (upward normal), as in Fig. 11.

Depending on the user's needs, the data can have various time resolutions and averages. The highest resolution available comes in approximately half second intervals, and all other data sets are created using these data. Twelve second, one hour, and one hour on a two hour window are three commonly prepared data sets which are available from UCLA. The twelve second data best fit our storage needs and the needs of the techniques used for finding current sheet crossings and calculating the orientation of the current sheet in space. Hourly averaged data are also used in finding current sheet crossings

The information on the magnetic tapes we ordered includes dates, times, and magnetic field magnitudes and directions. The twelve second data dates and times are

represented by year, month, day, hour, second, and millisecond. The magnetic magnitudes are given in the three directions X, Y, and Z, as previously defined by the VSO coordinate system.

FIG. 11. The axis orientations of the VSO coordinate system as received by UCLA. The X-axis is pointing radially inward from the Venus-Sun line. The Y-axis is pointing in the direction opposing Venus's orbital motion. The Z-axis is upward normal.

Once the data are loaded into the computer, the data are converted and stored in a different format. There are three reasons for the conversion: the reduction of required storage space, the needs of the graphics package DISSPLA, and conformity to a previously used coordinate system. Because the data contain only high resolution magnetic field data and times, the data are converted into the High Resolution

Interplanetary Magnetic Field (PxxIMF) data format. PxxIMF is only one of several possible file structures used for Pioneer satellite data storage. Each field of data has 8 eight-bit words per line: 1) spacecraft identification, 2) date of record (yyddd), 3) time of day (seconds), 4) milliseconds, 5) scalar magnitude of the magnetic field, 6) vector magnitude of the magnetic field, 7) azimuthal angle of the magnetic field, and 8) polar angle of the magnetic field.

As previously mentioned, the orientation of the coordinate system was also converted to match previously used data sets. The new directions have the X component positive in the radially outward direction from the Venus-Sun line, the Y component positive in the direction of orbital motion, and the Z component is unchanged (see Fig. 12).

B. PVO hardware

Two instruments are needed to determine current sheet crossings. These two instruments are the magnetometer and the plasma analyzer. The magnetometer is an instrument that measures the magnetic field at the satellite. The plasma analyzer measures the direction and velocities of the solar wind. These instruments are discussed in Sec. II.B.1 and Sec. II.B.2, respectively

1. *The magnetometer*

Venus has been studied by numerous spacecraft, including Mariners 2, 5, and 10, and Veneras 4, 9, and 10, carrying magnetometers. The PVO magnetometer was new and improved compared to those carried on previous satellites, as were other PVO instrumentation.

FIG. 12. The axis orientation as stored at NASA Ames Research Center. Current sheet calculations are performed using this coordinate system. The X-axis is pointing radially outward from the Sun. The Y-axis is positive in the direction of Venus's orbital motion. The Z-axis is positive in the upward normal direction.

The PVO carried three fluxgate magnetometers [11]. Each magnetometer used a ring core type sensor [11] and each was mounted on a three meter boom, see Fig 13. The

first sensor (P) was mounted at the end of the boom parallel to the spacecraft spin axis. The second sensor (T) was also mounted at the end of the boom, but transverse to the spin axis. The third sensor (G) was mounted 1 meter from the outboard end of the boom and laid in the plane containing the boom and the spacecraft spin axis, but was tilted at a 45° angle. The G sensor is redundant in that its output contains field information both parallel and transverse to the spacecraft spin axis.

Fig. 13. A simple schematic of the Pioneer Venus Orbiter showing the orientation of the three magnetometer sensors G, P, and T. The satellite spin axis also shown. The P sensor was parallel to the spin axis. The T sensor was transverse to the spin axis. The G sensor was in the plane of the boom and the spin axis, but was tilted 45° .

Each magnetometer can be sampled as rapidly as 12 times per second or as slowly as one sample each 64 seconds. For the faster rates, the magnetometer output could be

taken directly with an accurate spatial orientation of the data, while the slower rates must be demodulated by a spin data analyzer because of the large satellite rotation during the magnetic field measurement.

The dynamic range of each magnetometer was $\pm 128\gamma$, where $1\gamma = 1 \times 10^{-5}$ gauss. The resolution was between 1γ and $.125\gamma$ depending on the sampling rate.

2. The plasma analyzer

The Pioneer Venus Orbiter uses a type “B”, medium resolution, quadrispherical plasma analyzer to determine the solar wind velocities. The type “B” plasma analyzer has an energy resolution of $\pm 4.2\%$ and an azimuthal resolution between 15° and 47.5° . The high resolution type “A” plasma analyzer has an energy resolution of $\pm 2.8\%$ and an angular resolution between 6.6° and 8.3° .

The quadrispherical plasma analyzer is made from two concentric conducting shells separated by a vacuum gap. These shells are not complete spheres but are $1/4$ spheres, (see Fig. 14). At the “equator,” there is a set of ion collectors. The number of collectors determines the azimuthal resolution. The front edge of the spheres is closed off by conducting material. The particles enter the analyzer through a small aperture near the north pole.

The two conducting surfaces are maintained at different constant potentials. An ionized particle traveling in such an electric field will be deflected into a circular path due to electrostatic forces. Because of the close proximity of the conducting spheres, an

incoming particle will either hit one of the conducting shells, or it will find its way to one of the ion collectors at the equator. Clearly, only solar wind particles of the correct energies will hit the ion collectors while the rest of the particles will hit the conducting walls and be ignored.

FIG. 14. The quadrispherical plasma analyzer. The ion particles enter the aperture and travel between the two conducting spheres. If the ion particles are of the correct energy, the particles reach the ion collectors where the particles get recorded.

For each plate-to-plate potential difference, only a small range of solar wind velocities (energies) will be detected for each species. By varying the potential difference applied between the two spherical shells, different solar wind velocities (energies) can be registered. A complete energy spectrum can be constructed after all the necessary potential differences have been applied between the shells.

The azimuthal angles are determined by registering which of the ionic collectors the solar wind particle hit and by knowing the orientation of the plasma analyzer in space.

Knowing the azimuthal angle, the energy, and the orientation of the aperture in space, a three-dimensional velocity distribution of the plasma ions can be produced to within the resolution of the instrument.

III. DETERMINATION OF THE CURRENT SHEET

A. A multi-step process

Finding when a current sheet crossing took place and determining the orientation of the current sheet is a multi-step process. Section III.B will discuss how an approximate day of a crossing is found using hourly averaged data. Once the approximate date of a crossing is determined, high resolution data are used to find the actual time of the crossing (see Sec. III.C). Section III.D discusses the mathematical technique used to determine the orientation of the current sheet at the location of the PVO satellite crossing. Section III.E describes some the graphical results.

As previously mentioned, the current sheet sweeps over the PVO satellite and not vice-versa. It is only at this crossing time and location that the current sheet can be determined.

This Section (III) mentions the programs used in determining the current sheet. These programs are in Appendix C.

B. Locating approximate crossing times

The first step in locating a current sheet crossing is to narrow the search to some highly probable days. This section (III.B) discusses how that is accomplished using the program PARK.FOR.

As discussed in Sec. I.E.2, the interplanetary magnetic field lines leave the Sun nearly perpendicular to the surface. The rotation of the Sun causes the magnetic field lines to spiral outward, similar to a water sprinkler (shown in Fig. 7). Knowledge of the spiral shape of the solar magnetic field lines can help us locate approximate current sheet crossings. Let us find what shape the interplanetary magnetic field takes on as a function of radial distance from the Sun. Note that we will only be concerned with angles near the plane of the ecliptic.

The concept of “frozen-in” magnetic field lines demonstrated that the solar wind pulls the Sun’s magnetic field lines in the direction of the moving plasma. Therefore, it seems obvious that the velocity of the solar wind must play an important role in the curvature of the magnetic field lines. If the solar wind moved quickly, the Sun would not have a chance to rotate in the time the plasma pulled the magnetic field line outward. In this case, the magnetic lines would be nearly radial. Conversely, if the solar wind was moving slowly, the Sun might rotate several times before the plasma reached the same distance, and the magnetic field lines would coil up and be nearly horizontal to the Sun’s surface at that distance. Using known information about the solar wind velocities and solar rotation, it is possible to determine the curvature of the magnetic field as a function of distance.

The Sun rotates at an average angular velocity ω of 2.7×10^{-6} rad/sec, giving a rotation about every 27 days, and the average solar wind velocity v is 430 km/sec. Figure 15 shows the angle ψ to be measured, where ψ is the angle between the radial direction

from the Sun, (the x-axis), and the tangent to the magnetic field line at a given distance r .

The angular velocity ω and solar wind velocity v are assumed constant for this analysis.

FIG. 15. A schematic of the spiral form of the interplanetary magnetic field and the quantities used to describe it. The values are defined as follows: ω is the angular velocity, r is the distance from the center of the Sun, v is the solar wind velocity, a is the distance to the surface, and ψ is the angle to be determined.

At an angular velocity of ω , the Sun can rotate through an angle $\phi - \phi_0 = \omega t$ in time t , where ϕ_0 is the initial angle of the Sun, and ϕ is the final angle of the Sun.

However, the time t is equal to $(r - a)/v$, where $r - a$ is the change in radial distance of the magnetic field line. In the limit that $t \rightarrow 0$, the relation $(\phi - \phi_0) = \omega(a - r)/v$ becomes

$$-\frac{d\phi}{dr} = \frac{\omega}{v}. \quad (3.1)$$

Referring to Fig. 15 again, we see the spiral angle ψ can be defined as

$$\tan \psi = -(r - a) \frac{d\phi}{dr}. \quad (3.2)$$

The $-a$ term is necessary because ψ must be zero on the surface. Substituting equation 3.1 into equation 3.2, the final result relating the angle of the magnetic field versus distance is

$$\tan \psi = (r - a) \frac{\omega}{v}. \quad (3.3)$$

Using our coordinate system with X pointing radially outward and Y along the orbital motion of Venus, the angle ψ , at the distance of Venus, is approximately -34° from the $+x$ -axis, assuming a distance of 108×10^6 km. It should be noted that the angle of $+136^\circ$ could also be a solution to equation 3.3. This will become important later when we

compare magnetic field data to these two solutions in order to determine if the satellite was in the northern or southern magnetic region. This spiral structure of the interplanetary magnetic field is consistent with my PVO observations and it has also been verified by B. Thomas and E. Smith [12] between 1 and 8.5 AU.

At any given moment, the angle ψ can vary by tens of degrees from the expected values [12]. However, by looking at averaged data, much of this variability is removed.

The first goal of the program PARK.FOR is to find a daily average of the magnetic field data from the PVOMHOUR.DAT data file. The user inputs start and stop dates. For each day of interest, the given daily average of the magnetic field direction and magnitude is compared with the expected values of -34° and 136° . If the averaged data has a value $\pm 60^\circ$ from -34° , the magnetic field is assumed to be from the north magnetic pole. Likewise, if the averaged data has a value $\pm 60^\circ$ from 136° , the magnetic field is assumed to be toward the south magnetic pole. Most days can be defined as being either in the magnetic north or magnetic south region [13].

Figures 8 and 9 suggested that only a few crossings are expected for every solar revolution, and the output of PARK.FOR confirms this, see Fig. 16. The output from PARK.FOR between the dates of 84090 and 84110 shows only one major current sheet crossing, where 84090 is day 90 of year 1984 and 84110 is day 110 of year 1984. Figure 16 shows there were 6 consecutive days where the satellite was in the magnetic north region and then 7 days where the satellite was in the magnetic south region.

When there were several consecutive days in one magnetic region and then the satellite changed to the other magnetic region, one concludes the satellite had crossed the current sheet. In the case of Fig. 16, the current sheet crossing occurred on either 84095 or 84096. Further examination of the dates 84095 and 84096 is now carried out.

The magnetic field was “ + “ on 84090.
The magnetic field was “ + “ on 84091.
The magnetic field was “ + “ on 84092.
The magnetic field was “ + “ on 84093.
The magnetic field was “ + “ on 84094.
The magnetic field was “ + “ on 84095.
The magnetic field was “ - “ on 84096.
The magnetic field was “ - “ on 84097.
The magnetic field was “ - “ on 84098.
The magnetic field was “ - “ on 84099.
The magnetic field was “ - “ on 84100.
The magnetic field was “ - “ on 84101.
The magnetic field was “ - “ on 84102.

FIG. 16. The output from PARK.FOR showing a possible current sheet crossing. The first several days shows the satellite was in the magnetic north region. The satellite then changes to the magnetic south region.

C. Using the high resolution data

The low resolution data in the file PVOMHOUR.DAT have narrowed a possible current sheet crossing between the 2 days of 84095 and 84096. The high resolution data are examined next in order to better define the crossing time.

Because the high resolution data contain so many data points for a given day, only potential crossing dates are ordered. This helps in minimizing the storage space needed. By including an additional day on each side of the expected dates, the crossing time is sure

to be included. In the example of Fig. 16, the only data ordered were between the beginning of day 84094 and the end of day 84097.

A data file for each prospective crossing is created. The file names are PVOxxxxxC.DAT, where PVO stands for the Pioneer Venus Orbiter, C stands for a file that has had its coordinate system converted, as discussed in Sec. II, DAT stands for a data file, and xxxxx is the starting date of the file. In our example, the data file name is PVO84094C.DAT.

Each day of high resolution data is plotted in spherical coordinates by the program BxxxPLOT.FOR. There is a user option for plotting the results in rectangular or spherical coordinates. The outputs from BxxxPLOT.FOR are the magnetic field strength, the polar angle, the azimuthal angle, and the ratio of maximum eigenvalue to the minimum eigenvalue, see Fig. 17. The eigenvalue significance will be discussed in Sec. III.D.

Looking for a large change in the azimuthal magnetic field direction is the easiest way to locate a current sheet crossing. The BxxxPLOT.FOR output enables us to look for these changes. Ideally, a 180° change in the azimuthal angle would appear on the output.

After examining the high resolution data between 84094 and 84097, it was concluded that the current sheet crossing occurred early on the day of 84096. Figure 17 shows the data between hour 3 and hour 6 on that day. The magnetic field was fairly constant between hour 3 and hour 4 when the magnetic field had an average azimuthal angle (BAZM) of approximately $+345^\circ$. After hour 5, the azimuthal angle of the magnetic

FIG 17. A plot of the high resolution data between the hours of 3 and 6 for the date 84096. The dashed vertical line is where a current sheet crossing took place. The polar angle and azimuthal angle both show large changes in field direction at that time.

field was approximately $+150^\circ$. Note that both of the average values were fairly close to the expected values.

Between the hours of 4 and 5, a current sheet crossing took place. There is a particularly large change in both the azimuthal and polar magnetic field directions at about 4 hours and 40 minutes on 84096. The large bump in the MAX/MIN plot on the bottom of Fig. 17 helps confirm the crossing. As previously mentioned, the change in magnetic field direction is not a true discontinuity, but is something that happens over a short interval of time and space. Later plots show how the magnetic field actually rotates around from one side of the current sheet to the other side.

D. Determining the current sheet normal

Figure 17 showed when a crossing took place. The goal is now to determine the orientation of the current sheet using a mathematical technique developed by B. Sonnerup and L. Cahill [9]. This technique, which will be described in detail here, gives the necessary tools for calculating the normal to the current sheet. Having the normal to the current sheet is, of course, equivalent to finding the plane of the current sheet.

Figure 18 shows a pillbox dividing a boundary surface between two sides of different media. The net flux through the surface is

$$\oint (\mathbf{B} \cdot \mathbf{n}) dA = 0, \quad (3.4)$$

where $\mathbf{n}$ is the normal to the boundary as shown. If the sides of the pillbox are infinitesimally close together, the sides of the pillbox do not contribute to the above integral, and only the top and bottom components of the integral remain, which becomes

$$\int [(\mathbf{B}^2 \cdot \mathbf{n}) - (\mathbf{B}^1 \cdot \mathbf{n})] dA = 0. \quad (3.5)$$

Equation 3.5 shows that the normal component to the surface is continuous by noting that $\mathbf{B}^1 \cdot \mathbf{n} - \mathbf{B}^2 \cdot \mathbf{n} = 0$ and, therefore,

$$\mathbf{B}^1 \cdot \mathbf{n} = \mathbf{B}^2 \cdot \mathbf{n}. \quad (3.6)$$

FIG 18. The Gaussian pillbox straddles the boundary surface separating the two magnetic fields. Applying Gauss's Law, one sees that the normal component across the boundary is continuous.

The current sheet is a boundary similar to that shown in Fig. 18, but it has a finite thickness. Two Gaussian pillboxes may still be applied, one to each boundary, providing that the current sheet radius of curvature and lateral changes to its structure are much larger than the thickness of the current sheet, see Fig. 19.

Applying the divergence technique used above, one sees that normal component is continuous across the entire current sheet boundary,

$$\mathbf{B}^1 \cdot \mathbf{n} = \mathbf{B}^c \cdot \mathbf{n} = \mathbf{B}^2 \cdot \mathbf{n} . \quad (3.7)$$

We have assumed the changes in the magnetic field within the finite boundary are negligible.

FIG. 19. A Gaussian pillbox is applied to each side of the current sheet boundary. If the curvature and lateral changes of the current sheet are small, one can approximate the normal components from each side to be continuous.

Because the normal component is of the same magnitude and direction on each side of the current sheet, if we subtract any two of these magnetic field vectors, we obtain a vector that is tangential to the current sheet. Taking the cross product between the two tangential vectors, $(\mathbf{B}^2 - \mathbf{B}^1)$ and $(\mathbf{B}^2 - \mathbf{B}^c)$, we get a vector perpendicular to the plane containing the two vectors. If we normalize this cross product, we get

$$\mathbf{n} = \pm \frac{(\mathbf{B}^2 - \mathbf{B}^1) \times (\mathbf{B}^2 - \mathbf{B}^c)}{\|(\mathbf{B}^2 - \mathbf{B}^1) \times (\mathbf{B}^2 - \mathbf{B}^c)\|}. \quad (3.8)$$

This result is the normal to the current sheet.

This method for calculating the normal always works well if we have three simultaneous measurements above, below, and within the current sheet boundaries. In our case, there were not three satellites, only the PVO satellite was the taking measurements during these crossings. There is a problem if the current sheet structure changes rapidly compared to the penetration time. If the changes in the current sheet structure are slow compared to the satellite penetration time, we could use this method. We are not guaranteed the changes are slow; therefore, a better method should be found.

The PVO satellite did not make three field measurements but rather 100 or so relatively accurate measurements before, during, and after the crossing. It would be possible to average all of the measurements into the three groups of $\mathbf{B}^1$, $\mathbf{B}^2$, and $\mathbf{B}^c$ and

proceed with equation 3.8. However, a more convenient method is to choose the normal vector which minimizes the standard deviation [9] between $\mathbf{B}^i \cdot \mathbf{n}$ and $\mathbf{B} \cdot \mathbf{n}$, where $\mathbf{B}^i$ is an individual magnetic field measurement, and $\bar{\mathbf{B}}$ is the average of all N field measurements. The normal vector is obtained by minimizing the standard deviation

$$\sigma^2 = \frac{1}{N} \sum_i^N (\mathbf{B}^i \cdot \mathbf{n} - \bar{\mathbf{B}} \cdot \mathbf{n})^2. \quad (3.9)$$

The advantage of using this second method is that it takes into account the various non-stationary effects by using the average magnetic field $\bar{\mathbf{B}}$. In the least-mean-square sense, the continuity of this component across the layer is satisfied in the best possible way.

Appendix D shows the minimization of equation 3.9. The solution is a set of three equations and three unknowns. The coefficients are, in matrix form,

$$M_{\alpha\beta} = (\overline{B_\alpha B_\beta} - \bar{B}_\alpha \bar{B}_\beta) \quad \alpha, \beta = x, y, z, \quad (3.10)$$

where α and β are indices. Finding the eigenvalues and eigenvectors of the 3×3 matrix, composed of the elements in 3.10, is equivalent to determining the directions that define the plane of the current sheet and its normal. In fact, the eigenvector corresponding to the smallest eigenvalue is the normal to the current sheet while the other two eigenvectors lie in the plane of the current sheet. The 3×3 matrix is real and symmetric so the

corresponding eigenvectors are orthogonal. Once the matrix elements above have been determined, the eigenvalues and eigenvectors can easily be found.

This method breaks down if all the vectors $\mathbf{B}^1$, $\mathbf{B}^2$, and $\mathbf{B}^c$ are colinear or nearly colinear. Therefore, we don't actually want all the magnetic vectors of the discontinuity to be exactly 180° apart. This becomes obvious if we think about the cross product between two parallel lines. Of course, this cross product is zero. But if we think about trying to define a direction using the right hand rule, this cross product can not define a direction. That is because there are an infinite number of directions that are perpendicular to two parallel lines.

A small cross product is equivalent to having the ratio of the intermediate eigenvalue to the smallest eigenvalue being close to one. To resolve this problem, we insist that the ratio of these two eigenvalues be greater than 1.5 [9].

The program MATRIX.FOR and its subroutines calculates the aforementioned matrix elements, eigenvalues and eigenvectors. As Sec. III.E discusses, MATRIX.FOR also plots all N magnetic field components that lie in the plane of the current sheet.

E. Graphical results

As previously mentioned, the discontinuity is not an instantaneous one, but changes on the order of minutes. Using the high resolution data, it should be possible to examine the magnetic field during the time of the current sheet crossing. By examining these discontinuities, we hope to define how the magnetic field behaves in the boundary.

There are two distinct types of discontinuities, the tangential discontinuity and the rotational discontinuity.

The tangential discontinuity (TD) has all the magnetic field lines rotating about the normal direction while each of the magnetic field vectors is perpendicular to the normal vector [14]. In other words, all the magnetic field vectors which make up the current sheet lie parallel to the plane of the current sheet, see Fig. 20.

The second type of discontinuity is the rotational discontinuity (RD). The magnetic field vectors of the RD also rotate about the normal vector. However, each magnetic field vector also contains a component parallel to the normal vector [14]. When a RD rotates about the normal, the magnetic field vectors sweep out a cone, see Fig. 21.

FIG. 20. A schematic of a tangential discontinuity. All of the magnetic field vectors lie in the plane of the current sheet.

In this thesis, we are not interested in which type of discontinuity the satellite has crossed. We are only concerned with how the magnetic field components parallel to the current sheet change as the PVO passed through the crossing. What we are essentially doing is making the rotational discontinuities look like tangential discontinuities by subtracting off any normal component from each vector. The results are projections of the original vectors onto the plane of the current sheet.

FIG. 21. The schematic of the rotational discontinuity. The magnetic field vectors have a component in the direction of the normal vector and a component lying in the current sheet. The projection of each magnetic field vector onto the current sheet is shown as well as the normal component $\hat{n}$. The figure also shows the individual magnetic field vectors $\mathbf{B}^i$ sweeping out a cone.

The MATRIX.FOR program never assumes one type of discontinuity or the other, it always attempts to find the projection of each vector onto the plane of the current sheet, even if there is no normal component to begin with. These remaining components are stored in the data file SHEET.DIP for later plotting.

One of our eventual goals is to graphically examine these tangential components of the rotation to see how the magnetic field behaves before, during, and after the rotation and to compare the accuracy between the Sonnerup and Cahill method to the Gazis synoptic map method.

The data were plotted for our sample date of 84096 for 101 data points, or approximately 22.7 minutes. The actual crossing on 84096 was graphically determined to be near the 16762th second of that day. The 22.7 minutes were split 11.35 minutes per side. Figure 22 shows the 101 components that are tangential to the current sheet. Note the data points are connected. We see that the vectors started in the upper right hand side until the crossing at which time the magnetic field vectors rotated counter clock-wise to the upper left hand side. The axes (heavy lines in Fig. 22) are the eigenvectors corresponding to the two larger eigenvalues. As previously mentioned, these vectors are orthogonal and lie in the plane of the current sheet. They are being used as the coordinate system because of their reproducibility.

The eigenvectors are similar to the solutions found by solving the moment of inertia tensor. Therefore, if we rotated about the eigenvector associated with the largest eigenvalue, we would expect the “largest moment of inertia,” while the rotation about the

eigenvector associated with the second largest eigenvector would be less. We are, of course, dealing with magnetic fields not mass, but we can still imagine performing these rotations as a test to make sure the eigenvectors point in the correct directions.

FIG. 22. A plot of the magnetic field vectors that were used to calculate the current sheet normal on 84096. The plot shows the vectors starting in the upper right quadrant and rotating counter clock-wise to the upper left quadrant.

With all the tools needed to find each current sheet crossing, its normals, and the resulting plots, we are ready to examine the 25 potential current sheet crossings received from UCLA. These results are given in the next section.

IV. THE RESULTS

A. Dates and times to be determined

Dr. P. Gazis has requested dates for which we compare his synoptic map current sheet orientations [10] to the ones found here using the Sonnerup and Cahill technique [9] (discussed in Sec. III). There are 2 six-month periods in which we will attempt to compare results. The requested 2 six-month time spans were between the dates of 84064-84247 and 86245-87067.

B. Locating approximate times

As was demonstrated in Fig. 16, the hourly data file PVOMHOUR.DAT was used to find approximate crossing dates. Instead of listing several pages of output from PARK.FOR (see Appendix E) the 25 potential crossing dates that were found are listed in Table I. We found 8 potential crossings for the 84064-84247 dates and 17 potential crossings for the 86245-87067 dates.

C. Locating actual times

Each potential current sheet crossing was plotted and examined using the high resolution data in an attempt to determine the actual crossing times. However, many of the crossings could not be determined for any number of reasons. First, the magnetometer

data are incomplete. The down times were a result of the many onboard systems competing for CPU time; some operations had to be shut down so others could run. It was possible for a current sheet crossing to occur during one of these down times, as was the case for the dates: 84206-84207, 84236-84237, 86271-86272, 86344-86346, 86355-86356, and 87004-87005. Second, while the satellite was in Venus' bow shock, the solar magnetic field could be altered and, therefore, these data were not trustworthy. The satellite was in the bow shock for the dates: 84193-84194, 87044-87045, and 87046-87047. Third, it was possible for a current sheet crossing to contain multiple crossings [15]. The data showed multiple crossings for the dates: 84082-84083, 84222-84223, 86327-86328, 87015-87016, and 87020-87022. Of the remaining dates, only the ones with clear, well defined crossing times were considered.

TABLE I. A listing of potential crossing dates for the selected times in 1984, 1986, and 1987.

Day	Year		
	1984	1986	1987
	68-69	256-257	4-5
	82-83	271-272	9-11
	95-96	273-274	15-16
	111-112	327-328	20-22
	193-194	338-339	44-45
	206-207	344-345	46-47
	222-223	348-349	50-51
	236-237	355-356	51-52
			65-66

Of the initial 25 potential crossings, only 6 dates were well defined and examined in further detail (see Table II). The high resolution plots for each current sheet crossing

are shown in Figs. 23-28. Each plot is marked with a dashed vertical line showing where the crossing took place. Note how each crossing is accompanied by a large MAX/MIN ratio (described in Sec. III.D).

TABLE II. Dates of the well defined current sheet crossings.

	Year		
Day/Sec	1984	1986	1987
	96/16760	256/74130	11/15180
	111/39710	338/67610	66/17040

D. Results and conclusions

The normal was calculated for each date listed in Table II using the MATRIX.FOR program and subroutines. Each crossing was calculated using 22.7 minutes per side, or approximately 50 data points per side. The resulting normals are given in Table III.

TABLE III. The calculated normals.

	Normals		
Date	X-comp	Y-comp	Z-comp
84096	-0.7758	-0.1749	0.6062
84111	0.4781	-0.3861	0.7889
86256	-0.6082	0.4654	0.6296
86338	-0.4631	-0.8551	0.2332
87011	0.8969	0.0769	0.4355
87066	0.5957	0.7997	0.0753

The plots of the projections of the magnetic field lines onto the currents sheet are given in Fig. 29 through Fig. 34. The magnetic field structure in Figs. 29, 30, 31, and 33

appears normal as they rotate about the origin with minimal deviations. However, Figs. 32 and 34 both have large fluctuations occurring during the rotation.

All plots, if rotated about the each eigenvector axis, appear to have the correct orientation. Each plot has the “largest magnetic moment of inertia” about the eigenvector corresponding to the largest eigenvalue.

FIG 23. A plot of the high resolution data between the hours of 3 and 6 for the date 84096. The dashed vertical line is where a current sheet crossing took place.

FIG 24. A plot of the high resolution data between the hours of 10 and 12 for the date 84111. The dashed vertical line is where a current sheet crossing took place.

FIG 25. A plot of the high resolution data between the hours of 19 and 22 for the date 86256. The dashed vertical line is where a current sheet crossing took place.

FIG 26. A plot of the high resolution data between the hours of 18 and 20 for the date 86338. The dashed vertical line is where a current sheet crossing took place.

FIG 27. A plot of the high resolution data between the hours of 3 and 5 for the date 87011. The dashed vertical line is where a current sheet crossing took place.

FIG 28. A plot of the high resolution data between the hours of 3 and 6 for the date 87066. The dashed vertical line is where a current sheet crossing took place.

Fig. 29. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 84096.

Fig. 30. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 84111.

Fig. 31. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 86256.

Fig. 32. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 86338.

Fig. 33. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 87011.

Fig. 34. The plot of structure of the magnetic field before, during and after the current sheet crossing for the date of 87066.

REFERENCES

- [1] E. N. Parker, *Astrophys. J.*, **128**, 664 (1958).
- [2] L. Biermann, *Z. Naturforsch.*, **7a**, 161 (1948).
- [3] J. C. Brandt, *Introduction to the Solar Wind* (W. H. Freeman and Company, 1970), Chap. 1-3.
- [4] R. Haberman, *Elementary Applied Partial Differential Equations* (Prentice-Hall, Inc., Englewood Cliffs, 1983), Chap. 1.
- [5] W. K. H. Panofsky and M. Phillips, *Classical Electricity and Magnetism* (Addison-Wesley Publishing Company, Inc., London, England, 1962), Chap. 9 and 12.
- [6] N. A. Krall and A. W. Trivelpiece, *Principles of Plasma Physics* (San Francisco Press, Inc., San Francisco, 1986), Chap. 3.
- [7] M. Boas, *Mathematical Methods in the Physical Sciences* (John Wiley and Sons, Inc., 1983), Chap. 10 and 13.
- [8] P. V. Foukal, *Solar Astrophysics* (John Wiley and Sons, New York, 1990), Chap. 10-12.
- [9] B. U. Ö. Sonnerup and L. J. Cahill Jr., *J. Geophys. Res.*, **72**, 171 (1967).
- [10] P. R. Gazis, A. Barnes, J. D. Mihalov, and A. J. Lazarus, in *Proceedings of the Solar Wind 7 conference, 1992*, edited by E. Marsch and R. Schwenn, p. 183.
- [11] R. C. Snare and J. D. Means, *IEEE Tran. Magn.*, **Mag-13**, 1107 (1977).
- [12] B. T. Thomas and E. J. Smith, *J. Geophys. Res.*, **85**, 6861 (1980).
- [13] P. R. Gazis, *J. Geophys. Res.*, **98**, 9391 (1993).
- [14] L. F. Burlaga, J. F. Lemaire, and J. M. Turner, *J. Geophys. Res.*, **82**, 3191 (1977).
- [15] N. U. Crooker *et al.*, *J. Geophys. Res.*, **98**, 9371 (1993).

APPENDIX A

Section I.C.1 showed that an element of gas in static equilibrium could be represented by

$$0 = \frac{d}{dr}(2NkT) + \frac{GMmN}{r^2}, \quad (A1)$$

where M was the mass of the Sun, m was the mass of a hydrogen atom, r was the distance from the center of the Sun, T was the temperature, and G was the gravitational constant. The particle density N was the unknown to be determined.

Expanding out the first term on the right hand side of equation A1, we get

$$\frac{d}{dr}(2NkT) = 2kT \frac{dN}{dr} + 2kN \frac{dT}{dr}. \quad (A2)$$

Recall that the temperature was given by $T(r) = T_0(a/r)^{1/(n+1)}$, where a was the radial distance at which local heating stops. Substituting $T(r)$ into A2 and taking the derivative dT/dr , we obtain

$$\frac{d}{dr}(2NkT) = 2kT_o \left(\frac{a}{r}\right)^{1/(n+1)} \left(\frac{dN}{dr}\right) - \left(\frac{2NkT_o a^{1/(n+1)}}{n+1}\right) \frac{1}{r^{(n+2)/(n+1)}}. \quad (A3)$$

Replacing A3 into A1, we get

$$0 = 2kT_o \left(\frac{a}{r}\right)^{1/(n+1)} \left(\frac{dN}{dr}\right) - \left(\frac{2NkT_o a^{1/(n+1)}}{n+1}\right) \frac{1}{r^{(n+2)/(n+1)}} + \frac{GMmN}{r^2}. \quad (A4)$$

Dividing through by $2NkT_o a^{1/(n+1)}$, equation A4 becomes

$$0 = \left(\frac{1}{Nr^{1/(n+1)}}\right) \left(\frac{dN}{dr}\right) - \left(\frac{1}{n+1}\right) \frac{1}{r^{(n+2)/(n+1)}} + \frac{GMm}{2kT_o a^{1/(n+1)} r^2}. \quad (A5)$$

Multiplying A5 through by $r^{1/(n+1)}$, we get

$$0 = \left(\frac{1}{N}\right) \left(\frac{dN}{dr}\right) - \left(\frac{1}{n+1}\right) \frac{1}{r} + \frac{GMm}{2kT_o a^{1/(n+1)} r^{(2n+1)/(n+1)}}. \quad (A6)$$

Separating the variables, defining $\lambda = GMm/2kT_o a$, and integrating from the initial values of a and N_o to the final values of r and N , equation A6 becomes

$$\int_{N_o}^N \frac{dN}{N} = \int_a^r \left(\frac{1}{(n+1)} r^{-1} - \lambda a^{n/(n+1)} r^{-(2n+1)/(n+1)} \right) dr. \quad (A7)$$

Direct integration of A7 gives

$$\ln\left(\frac{N}{N_o}\right) = \frac{1}{n+1} \ln\left(\frac{r}{a}\right) + \lambda \frac{n+1}{n} \left(\left(\frac{a}{r}\right)^{n/(n+1)} - 1 \right). \quad (A8)$$

Using the laws of logs, equation A8 becomes

$$\ln\left(\frac{N}{N_o}\right) = \ln\left(\frac{r}{a}\right)^{1/(n+1)} + \lambda \frac{n+1}{n} \left(\left(\frac{a}{r}\right)^{n/(n+1)} - 1 \right). \quad (A9)$$

Finally, solving for N gives

$$N = N_o \left(\frac{r}{a}\right)^{1/(n+1)} \exp\left(\frac{\lambda(n+1)}{n} \left(\left(\frac{a}{r}\right)^{n/(n+1)} - 1 \right)\right), \quad (A10)$$

which is the desired result, equation 1.7.

APPENDIX B

The stationary expansion of the solar wind was given in Sec. I.C.2 as

$$Nmv\left(\frac{dv}{dr}\right) = -\frac{d}{dr}(2NkT) - \frac{GNMm}{r^2}, \quad (\text{B1})$$

where N was the particle density, m was the mass of a hydrogen atom, T was the temperature, M was the mass of the Sun, G was the gravitational constant, v was the velocity of the solar wind, and r was the radial distance from the Sun.

Expanding the derivative, as in Appendix A, equation B1 becomes

$$Nmv\left(\frac{dv}{dr}\right) = -2Nk\left(\frac{dT}{dr}\right) - 2kT\left(\frac{dN}{dr}\right) - \frac{GNMm}{r^2}. \quad (\text{B2})$$

Our goals will be to eliminate $N(r)$ from B2 and to make B2 dimensionless. The definitions of the dimensionless variables used in the rest of this derivation are $\tau = T/T_0$, $\lambda = GMm/2akT_0$, $\psi = mv^2/2kT_0$, and $\xi = r/a$. Also, the derivative of r is $dr = a d\xi$. We can eliminate $N(r)$ from B2 by using the continuity equation,

$$\frac{d}{dr}(r^2 N v) = 0. \quad (B3)$$

If the derivative in equation B3 is always zero, $r^2 N v$ must equal a constant. Therefore, the values on the surface, (a , N_o , and v_o), must equal the values at some point r in space, (r , N , v). Equating these values and for $N(r)$, we get

$$N(r) = \frac{N_o v_o}{v} \left(\frac{a}{r} \right)^2. \quad (B4)$$

Plugging in B4 into B2 and eliminating the common terms N_o , and v_o , equation B2 becomes

$$\frac{m a^2}{r^2} \left(\frac{dv}{dr} \right) = - \frac{2k}{v} \left(\frac{a}{r} \right)^2 \left(\frac{dT}{dr} \right) - (2k T a^2) \frac{d}{dr} \left(\frac{1}{v r^2} \right) - \frac{G M m a^2}{v r^4}. \quad (B5)$$

Substituting in the definitions for ξ and dr , we get

$$\frac{m}{a \xi^2} \left(\frac{dv}{d\xi} \right) = - \frac{2k}{v a \xi^2} \left(\frac{dT}{d\xi} \right) - \frac{2k T}{a} \frac{d}{d\xi} \left(\frac{1}{v \xi^2} \right) - \frac{G M m}{a^2 v \xi^4}. \quad (B6)$$

Multiplying through by $a \xi^2 / k T_o$, and using the definition for λ , we obtain

$$\frac{m}{kT_0} \left(\frac{dv}{d\xi} \right) = -\frac{2}{vT_0} \left(\frac{dT}{d\xi} \right) - \frac{2T\xi^2}{T_0} \frac{d}{d\xi} \left(\frac{1}{v\xi^2} \right) - \frac{2\lambda}{v\xi^2}. \quad (\text{B7})$$

Multiplying through by v , and recalling the definition of $\tau = T/T_0$, equation B7 becomes

$$\frac{mv}{kT_0} \left(\frac{dv}{d\xi} \right) = -2 \left(\frac{d\tau}{d\xi} \right) - 2\tau v \xi^2 \frac{d}{d\xi} \left(\frac{1}{v\xi^2} \right) - \frac{2\lambda}{\xi^2}. \quad (\text{B8})$$

The left hand side of equation B8 can be rewritten by using the chain rule,

$$v \left(\frac{dv}{d\xi} \right) = \frac{1}{2} \left(\frac{dv^2}{d\xi} \right). \quad (\text{B9})$$

Substituting B9 along with the above definition for ψ , B8 becomes

$$\left(\frac{d\psi}{d\xi} \right) = -2 \left(\frac{d\tau}{d\xi} \right) - 2\tau v \xi^2 \frac{d}{d\xi} \left(\frac{1}{v\xi^2} \right) - \frac{2\lambda}{\xi^2}. \quad (\text{B10})$$

The second term on the right hand side of equation B10 can be expanded out by the chain rule into

$$2\tau v \xi^2 \frac{d}{d\xi} \left(\frac{1}{v \xi^2} \right) = 2\tau v \frac{d}{d\xi} \left(\frac{1}{v} \right) + 2\tau \xi^2 \frac{d}{d\xi} \left(\frac{1}{\xi^2} \right). \quad (\text{B11})$$

Substituting B11 into B10, we get

$$\frac{d\psi}{d\xi} = -2 \left(\frac{d\tau}{d\xi} \right) - 2\tau v \frac{d}{d\xi} \left(\frac{1}{v} \right) - 2\tau \xi^2 \frac{d}{d\xi} \left(\frac{1}{\xi^2} \right) - \frac{2\lambda}{\xi^2}. \quad (\text{B12})$$

The first and third terms on the right hand side of B12 can be combined by the chain rule and the second term can be rewritten to get,

$$\frac{d\psi}{d\xi} = -2\xi^2 \frac{d}{d\xi} \left(\frac{\tau}{\xi^2} \right) + \frac{\tau}{v^2} \left(\frac{dv^2}{d\xi} \right) - \frac{2\lambda}{\xi^2}. \quad (\text{B13})$$

Recalling the definition of $\psi = mv^2/2kT_0$, the second term on the right hand side can be rewritten and brought to the left hand side,

$$\left(1 - \frac{\tau}{\psi} \right) \frac{d\psi}{d\xi} = -2\xi^2 \frac{d}{d\xi} \left(\frac{\tau}{\xi^2} \right) - \frac{2\lambda}{\xi^2}. \quad (\text{B14})$$

We are looking for a solution of ψ versus ξ , which is essentially velocity versus distance. In order to integrate B14, let us assume the temperature is maintained at the

temperature $T(a < r < b) = T_0$ between $r = a$ out to some radial point $r = b$. The heating vanishes and the temperature $T(r > b) = 0$ beyond $r = b$. Because the outward coronal expansion consumes 100 times more energy than the thermal conduction possibly could, it is reasonable to take $T(r > b)$ to be negligible beyond $r = b$ [1].

For $r < b$, equation B14 can be integrated by separating the variables. Note that the term $-2\xi^2 d/d\xi(\tau/\xi^2)$ now equals $4/\xi$ giving,

$$\int_{\psi_0}^{\psi'} \left(1 - \frac{1}{\psi}\right) d\psi = \int_1^{\xi'} \left(\frac{4}{\xi} - \frac{2\lambda}{\xi^2}\right) d\xi. \quad (\text{B15})$$

Direct integration of B15 gives

$$\psi' - \ln \psi' = \psi_0 - \ln \psi_0 + 4 \ln \xi' - 2\lambda \left(1 - \frac{1}{\xi'}\right). \quad (\text{B16})$$

For $r > b$ and $\tau \approx 0$, equation B14 becomes

$$\psi(\xi) = \psi(b/a) - \lambda \left(\frac{a}{b} - \frac{1}{\xi}\right). \quad (\text{B17})$$

Because b can be large [1], B16 is the desired solution. The constant $\psi_0 - \ln \psi_0$ still needs to be determined. The terms on the left hand side of B16 are a function of velocity,

call them $Z(v)$, while the last two terms on the right hand side are a function of distance, call them $Y(r)$. We can rewrite B16 as

$$Z(v) = \psi_o - \ln \psi_o + Y(r). \quad (B18)$$

The functions Y and Z both decrease from there initial values at $r = a$ and both functions pass through a minimum. If B17 is to remain constant, the minimums must occur simultaneously. Taking the derivatives of Z and Y , we find the minimums occurs at $\psi = 1$ and at $\xi = \lambda / 2$ when. Substituting these values into B16, we find the constant to be

$$\psi_o - \ln \psi_o = 2\lambda - 3 - 4 \ln \frac{\lambda}{2}. \quad (B19)$$

Thus, the final expression for the solar wind in these dimensionless units is

$$\psi - \ln \psi = -3 - 4 \ln \left(\frac{\lambda}{2} \right) + 4 \ln \xi - \frac{2\lambda}{\xi}, \quad (B20)$$

which is equation 1.14 of Sec. I. Note we have dropped the prime superscript.

APPENDIX C

Appendix C includes copies of the programs used in determining a current sheet crossing.

1) CONVERTOUT.FOR	80
a) Purpose: Converts the ordered UCLA data into PxxIMF format.	
b) Subroutines: none	
2) PARK.FOR	83
a) Purpose: To find approximate day of a current sheet crossing.	
b) Subroutines: none	
3) BxxxPLOT.FOR	86
a) Purpose: To find the actual current sheet crossing time.	
b) Subroutines: POINTS.FOR, MKARRAY.FOR, HPOINTS.FOR, HMKARRAY.FOR, LAMBDA.FOR, and the DISSPLA library functions.	
4) MATRIX.FOR	103
a) Purpose: To calculate the normal vector to the current sheet.	
b) Subroutines: EIGENVEC.FOR, PLANE.FOR, PLANEPLOT.FOR, EULER.FOR, and the DISSPLA library functions.	

```

C*****
C FILE NAME: CONVERTOUT.FOR
C
C PURPOSE: THIS PROGRAM IS USED TO CONVERT THE ASCII DATA
C FILES
C RECEIVED FROM UCLA TO REAL VALUES IN THE FORM OF
C PXXIMF HIGH RESOLUTION DATA. ALSO IT WILL CONVERT
C THE Bx, By, Bz COORDINATES TO POLAR FORM.
C
C COORDINATE SYSTEM: THE INCOMING DATA HAS THE FOLLOWING
C AXIS.
C THE +X-DIRECTION IS POSITIVE TOWARD THE SUN.
C THE +Y-DIRECTION IS POSITIVE IN THE DIRECTION
C OPPOSING ORBITAL MOTION.
C THE +Z-DIRECTION IS IN THE NORTHWARD DIRECTION.
C IE. PARALLEL TO THE POLE OF THE VENUS ORBITAL
C PLANE.
C
C AUTHOR: FRED QUIST
C
C DATE: 7-14-93 MODIFIED: 12-20-93
C*****

PROGRAM CONVERTOUT

C DEFINE THE OUTPUT VARIABLES TO BE USED IN THIS PROGRAM
C
C SCID - THE SPACECRAFT IDENTIFICATION
C YRDY - THE YEAR AND DAY OF THE DATA. IE (78228) IS 1978, 228TH DAY
C SEC - THE NUMBER OF SECONDS IN THE DAY WHEN THE DATA WAS
C TAKEN
C MSEC - THE NUMBER OF MILLISECONDS WHEN THE DATA WAS TAKEN
C BT - THE SCALAR AVERAGE
C BMAG - THE VECTOR MAGNITUDE OF THE DATA
C BAZM - THE AZMUTHAL ANGLE OF THE DATA
C BPOL - THE POLAR ANGLE OF THE DATA

INTEGER YR, MON, DY, HR, MN, SC, MS, CNTR2, DAYS
INTEGER LARRY, CURLEY, MOE, RES
REAL*4 BX, BY, BZ
REAL*4 SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL
REAL*4 REMDR2, ABSBY, TEST

C*****
C INITIALIZE THESE VARIABLES
C

```

```

LARRY = 11
CURLEY = 12
MOE = 13

WRITE (6,222)
222 FORMAT(1X,'WHAT ORIENTATION? X-AXIS TOWARD SUN(1) OR
AWAY(2)')
READ (5,223) RES
223 FORMAT(1I)

C*****
C OPEN AND READ THE DATA FILE THAT HAS THE PVO HDR IN THE
CORRECT FORM

OPEN(UNIT = CURLEY,
+ FILE = 'SCISDISK:[QUIST.PVO.PVODATA]PVOHDR.DAT',
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')

READ (CURLEY, END = 300) SCID

C*****
C OPEN THE FILE THAT THE PROCESSED DATA WILL BE PUT INTO

OPEN (UNIT = LARRY,
+ FILE = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84106C.DAT',
+ FORM = 'UNFORMATTED',
+ STATUS = 'NEW')

C*****
C OPEN THE DATA FILE TO BE READ. FILES FROM UCLA

OPEN (UNIT = MOE,
+ FILE = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84106.DAT',
+ FORM = 'FORMATTED',
+ READONLY,
+ STATUS = 'OLD')

DO WHILE (.TRUE.)

DAYS = 0
CNTR2 = 1
READ (MOE, 100, END = 300) YR, MON, DY, HR, MN, SC, MS, BX, BY, BZ, BT

```

```

100  FORMAT(6I3.2, I4.3, 4G13.5)

C*****
C  CONVERT THE HOURS, MINUTES, AND SECONDS INTO JUST SECONDS
      SEC = REAL((HR * 3600) + (MIN * 60) + SC)
C*****
C  CONVERTS THE MILLISECONDS INTO A REAL NUMBER
      MSEC = REAL(MS)
C*****
C  CONVERT MONTHS AND DAYS TO JUST DAYS, ADD ON YEARS TO
      BEGINING.
C  THIS LEADS TO THE FOLLOWING FORM "YYDDD"
      DO WHILE (CNTR2.LT. MON)
C  IF THE YEAR / 4 = 0 THEN IT IS A LEAP YEAR
      REMDR2 = MOD(REAL(YR),4.0)
      IF ((CNTR2.EQ. 4).OR.(CNTR2.EQ. 6).OR.(CNTR2.EQ. 9).OR.
      - (CNTR2.EQ. 11)) THEN
        DAYS = DAYS + 30
      ELSE IF (CNTR2.EQ. 2) THEN
        IF (REMDR2.EQ. 0.0) THEN
          DAYS = DAYS + 29
        ELSE
          DAYS = DAYS + 28
        END IF
      ELSE
        DAYS = DAYS + 31
      END IF
      CNTR2 = CNTR2 + 1
    END DO
      YR = YR * 1000
      DAYS = YR + DAYS + DY
      YRDY = REAL(DAYS)
C*****
C  CONVERT FROM CARTESIAN COORDINATES TO POLAR COORDINATES
      C*****
      IF ((BX.EQ. 10000E-33).OR.(BY.EQ. 10000E+33).OR.
      + (BZ.EQ. 10000E-33)) THEN
        BMAG = 0.0
        BPOL = 0.0
        BAZM = 0.0
        BT = 0.0
      ELSE
        C  FINDING THE MAGNITUDE OF BX, BY, AND BZ
        BMAG = SQRT((BX**2 + BY**2 + BZ**2))
        C  FINDING THE POLAR AND AZMUTHAL ANGELS
        ABSBY = ABS(BY)
        BPOL = ASIND(BZ/BMAG)
        TEST = ABSBY/(BMAG*COSD(BPOL))
        WRITE(6,*) TEST
        IF (TEST.GE. 1.0) THEN
          BAZM = 90.0
        ELSE IF (TEST.LE. -1.0) THEN
          BAZM = -90.0
        ELSE
          BAZM = ASIND(TEST)
        END IF
        IF ((BX.LE. 0.0).AND.(BY.GE. 0.0)) THEN
          BAZM = 180.0 - BAZM
        ELSE IF ((BX.LE. 0.0).AND.(BY.LE. 0.0)) THEN
          BAZM = 180.0 + BAZM
        ELSE IF ((BX.GE. 0.0).AND.(BY.LE. 0.0)) THEN
          BAZM = 360.0 - BAZM
        END IF
      C*****
      C  IF RES = 2, ROTATE THE X-Y PLANE 180 DEGREES. IE. X IS OUTWARD
      FROM SUN.
      IF (RES.EQ. 2) THEN
        BAZM = BAZM + 180.0
      IF (BAZM.GE. 360) THEN
        BAZM = BAZM - 360.0
      C*****

```

```

        END IF
        END IF

C*****WRITE THE CONVERTED TO THE FILE:
[QUIST.PVO.PVODATA\|PVO\DATE\C.DAT*****

        WRITE (LARRY)SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL
        END IF

        END DO

300  CLOSE(UNIT = 11)
      CLOSE(UNIT = 12)
      CLOSE(UNIT = 13)

      STOP
      END

```

```

C*****
C File name: PARK_FOR
C
C Name: PARK
C
C Purpose: This program will retrieve selected data from the
C pvinthour file in the pxxdata.pdata.pvo directory
C for calculating the Parker spiral angle.
C
C Author: Fred Quist 4/15/93
C
C Modified: 3/31/93
C*****
PROGRAM PARK
C*****
C PI AND DEG ARE SELF EVIDENT
C*****
PARAMETER (PI = 3.141593)
PARAMETER (DEG = 180.0)
C*****
C RAD IS THE AVERAGE DISTANCE FROM VENUS TO THE SUN
C OMEGA IS THE ROTATIONAL VELOCITY OF THE SUN
C*****
PARAMETER (RAD = 108000000)
PARAMETER (OMEGA = .0000027)
INTEGER LUN, RES
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
PVINTHOUR.DAT FILE
C*****
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 10
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 11
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 12
C YRDY IS THE YEAR AND DAY OF THE DATA
C HOUR IS THE HOUR THE DATA WAS TAKEN
C*****
REAL*4 HDR, YRDY, HR, DUMM1(3), NBREC, NSWREC, BS, BMAG
REAL*4 BAZM, BPOL, D2(6), VEL
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO CALCULATE THE P.S.
ANGLE
C*****
C AND THE AVERAGE VECTOR VALUES OF THE B-FIELD.
C*****
REAL*4 MEANX, MEANY, MEANZ, CNTR1, ZERO, EYRDY, SYRDY
REAL*4 MEANB, AVGVEL, ANGLE, Epx, Epy, THETA, OLDYRDY
C*****
C INITIALIZE THESE VARIABLES
C*****
AVGVEL = 0.0
MEANX = 0.0
MEANY = 0.0
MEANZ = 0.0
MEANB = 0.0
CNTR1 = 0
ZERO = 0.0
LUN = 11
SUN = 12
C*****
OPEN (UNIT = SUN,
+ FILE = 'SCISDISK:[QUIST,PVO,PVODATA]OUTPUT.DAT',
+ FORM = 'FORMATTED',
+ STATUS = 'NEW')
C*****
C READS THE FIRST RECORD TO FIND THE STARTING YYDDD AND HOUR.
C AND THEN CLOSSES THE FILE
C*****
OPEN (UNIT = LUN,
+ FILE = 'SCISDISK:[PXXDATA,PDATA,PVO]PVOIMHOUR.DAT',
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
READ (LUN, END = 3500) HDR, YRDY, HR
WRITE(6,2100) HERE IS THE STARTING DATE, 'YRDY
2100 FORMAT(1X, A28, F8.1)
CLOSE (UNIT = LUN)
C*****
C THIS IS A SIMPLE LOOP TO GET THE INPUTS FOR STARTING AND
STOPPING TIMES
C*****

```



```

ELSE IF (THETA.GT. (DEG/1.8)) THEN
  WRITE(6,100) 'THE MAGNETIC FIELD WAS "- " ON 'OLDYRDY
  FORMAT (1X, A33, F8.1)
  WRITE(SUN,112) 'THE MAGNETIC FIELD WAS "- " ON 'OLDYRDY
112  FORMAT (A33,F6.0)
  END IF
  END IF
  END IF
  MEANB = 0
  LOGIC2 = .TRUE.
  MEANX = 0.0
  MEANY = 0.0
  MEANZ = 0.0
  CNTR1 = 0
  IF (YRDY.GT. EYRDY) THEN
    LOGIC3 = .FALSE.
  END IF
  END DO

C*****
C  CLOSE UP SHOP AND GO HOME.

3500 CLOSE( UNIT = LLN )
  CLOSE( UNIT = SUN )
  STOP
  END

```

```

C*****
C
C FILE NAME: BXXXPLOT.FOR
C
C NAME: BXXXPLOT
C
C PURPOSE: THIS WILL LINK ALL OF THE PROGRAMS FOR THE
C GRAPHING
C ROUTINES THAT I NEED.
C
C AUTHOR: FRED QUIST DATE: 11-4-93
C
C MODIFIED: 6/17/94
C
C LINKED PROGRAMS: POINTS.FOR, MK-ARRAY.FOR, DISSPLA PROGRAMS,
C HPOINTS.FOR, HMK-ARRAY.FOR, LAMBDA.FOR
C
C*****
C
C PROGRAM BXXXPLOT
C*****
C VARIABLE DEFINITIONS
C*****
PARAMETER (SIZE = 30000)
PARAMETER (HSIZE = 30)
PARAMETER (SIZ2 = 10000)
INTEGER PNTS,HPNTS,I,J,RES,RES2,RES3,CNTR3
CHARACTER*50 FNAME, HFNAME
CHARACTER*5 QST1,QST2
REAL*4 TIME(SIZE),ONE(SIZE),TWO(SIZE),THREE(SIZE),FOUR(SIZE)
REAL*4 HTIME(HSIZE),HONE(HSIZE),HTWO(HSIZE),HTHREE(HSIZE)
REAL*4 HFOUR(HSIZE),RATIO(SIZ2),TIME2(SIZ2)
REAL*4 SYRDY,SSEC,EYRDY,FSEC

REAL*4 CENTER(2) /0.0, 0.0/
REAL*4 CENTER2(2) /180.0, 180.0/
REAL*4 TIME2(2)

LOGIC1 = .TRUE.
LOGIC2 = .TRUE.
LOGIC3 = .TRUE.

C*****
C ASK IF THE USER WANTS HIGH RESOLUTION AND/OR HOURLY DATA
C PLOTTED.

C*****
DO WHILE (LOGIC2)
WRITE (6,800)
800 FORMAT('DO YOU WANT HIGH RESOLUTION DATA PLOTTED [Y/N]?')
READ(5,810)QST1
810 FORMAT(A1)
IF ((QST1.EQ.'Y') OR. (QST1.EQ.'N')) THEN
LOGIC2 = .FALSE.
END IF
END DO

DO WHILE (LOGIC3)
WRITE (6,900)
900 FORMAT ('DO YOU WANT HOURLY DATA PLOTTED [Y/N]?')
READ(5,910)QST2
910 FORMAT(A1)
IF ((QST2.EQ.'Y') OR. (QST2.EQ.'N')) THEN
LOGIC3 = .FALSE.
END IF
END DO

C*****
C FIND OUT WHAT YEAR THE USER WANTS TO LOOK AT

IF (QST1.EQ.'Y') THEN
WRITE(6,300)
300 FORMAT(1X,'WHAT IS THE NUMBER OF THE YEAR YOU WANT?')
WRITE(6,310)
310 FORMAT(1X,'1 = 1984,1X,'2 = 1986,1X,'3 = 1987')
READ(5,320)RES3
320 FORMAT(I2)

C*****
C GET THE NAMES OF THE FILES THAT WILL BE USED IN THESE
ROUTINES

WRITE(6,1000)
1000 FORMAT(1X,'WHAT IS THE NUMBER OF DATA FILE YOU WANT?')
IF (RES3.EQ.1) THEN
WRITE(6,1020)
1020 FORMAT(1X,'1 = PVO84066C.DAT')
WRITE(6,1030)
1030 FORMAT(1X,'2 = PVO84081C.DAT')
WRITE(6,1040)
1040 FORMAT(1X,'3 = PVO84094C.DAT')
WRITE(6,1050)
1050 FORMAT(1X,'4 = PVO84106C.DAT')

```

```

WRITE(6,1060)
1060  FORMAT(1X, '5 = PVO84192C.DAT')
WRITE(6,1070)
1070  FORMAT(1X, '6 = PVO84205C.DAT')
WRITE(6,1080)
1080  FORMAT(1X, '7 = PVO84221C.DAT')
WRITE(6,1090)
1090  FORMAT(1X, '8 = PVO84235C.DAT')
ELSE IF (RES2.EQ.2) THEN
WRITE(6,1100)
1110  FORMAT(1X, '9 = PVO86255C.DAT')
WRITE(6,1120)
1120  FORMAT(1X, '10 = PVO86272C.DAT')
WRITE(6,1130)
1130  FORMAT(1X, '11 = PVO86326C.DAT')
WRITE(6,1140)
1140  FORMAT(1X, '12 = PVO86337C.DAT')
WRITE(6,1150)
1150  FORMAT(1X, '13 = PVO86343C.DAT')
WRITE(6,1160)
1160  FORMAT(1X, '14 = PVO86354C.DAT')
ELSE IF (RES3.EQ.3) THEN
WRITE(6,1210)
1210  FORMAT(1X, '15 = PVO87003C.DAT')
WRITE(6,1220)
1220  FORMAT(1X, '16 = PVO87009C.DAT')
WRITE(6,1230)
1230  FORMAT(1X, '17 = PVO87014C.DAT')
WRITE(6,1240)
1240  FORMAT(1X, '18 = PVO87019C.DAT')
WRITE(6,1250)
1250  FORMAT(1X, '19 = PVO87043C.DAT')
WRITE(6,1260)
1260  FORMAT(1X, '20 = PVO87064C.DAT')
END IF
READ(5,1010)RES2
1010  FORMAT(I2)
IF (RES2.EQ.1) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84066C.DAT'
ELSE IF (RES2.EQ.2) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84081C.DAT'
ELSE IF (RES2.EQ.3) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84094C.DAT'
ELSE IF (RES2.EQ.4) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84106C.DAT'
ELSE IF (RES2.EQ.5) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84192C.DAT'
ELSE IF (RES2.EQ.6) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84205C.DAT'
ELSE IF (RES2.EQ.7) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84221C.DAT'
ELSE IF (RES2.EQ.8) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO84235C.DAT'
ELSE IF (RES2.EQ.9) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86255C.DAT'
ELSE IF (RES2.EQ.10) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86272C.DAT'
ELSE IF (RES2.EQ.11) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86326C.DAT'
ELSE IF (RES2.EQ.12) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86337C.DAT'
ELSE IF (RES2.EQ.13) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86343C.DAT'
ELSE IF (RES2.EQ.14) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO86354C.DAT'
ELSE IF (RES2.EQ.15) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87003C.DAT'
ELSE IF (RES2.EQ.16) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87009C.DAT'
ELSE IF (RES2.EQ.17) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87014C.DAT'
ELSE IF (RES2.EQ.18) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87019C.DAT'
ELSE IF (RES2.EQ.19) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87043C.DAT'
ELSE IF (RES2.EQ.20) THEN
FNAME = 'SCISDISK:[QUIST.PVO.PVODATA]PVO87064C.DAT'
END IF
END IF
IF (QST2.EQ.'Y') THEN
WRITE(6,4000)
4000  FORMAT('WHAT IS THE FILENAME OF THE HOURLY DATA (A50)?')
READ(5,4010)HFNAME
4010  FORMAT(A50)
HFNAME = 'SCISDISK:[PXXDATA.PDATA.PVO]PVO8HOUR.DAT'
END IF
C*****
C ASK IF THE USER WANTS THE GRAPHS IN RECTANGULAR OR
SPHERICAL COORD.
DO WHILE (LOGIC1)

```

```

C*****
C  CREATE A TWO POINT ARRAY FOR DRAWING THE CENTER LINES
C*****
DO I = 1,2
  IF (I.EQ. 1) THEN
    TIME2(I) = SSEC/3600.0
  ELSE
    TIME2(I) = ESEC/3600.0
  END IF
END DO
C*****
C  THE PLOTTING SUBROUTINES AS PER DISPLA
C*****
CALL DIP(5, 'BMAG.DIP', 100)
CALL PAGE(8.5, 11.0)
CALL NOBRDR
CALL PHYSOR(1.0,0.5)
DO J = 1,4
  IF (J.EQ. 1) THEN
    CALL AREA2D(6.5,2.25)
    CALL XNAME('TIME(HOURS)', 100)
  IF (RES.EQ. 1) THEN
    CALL YTICKS(4)
    CALL XTICKS(4)
    CALL GRAF((SSEC/3600),900, (ESEC/3600),0,500,1000.)
    CALL YGRANS(0.0, 500, 1000, 2.25, 'MAX/MINS',100,0.0.)
    CALL CURVE(TIME2,RATIO,(CNTR3-1),0)
  ELSE
    CALL YTICKS(4)
    CALL XTICKS(4)
    CALL GRAF((SSEC/3600), 1, (ESEC/3600), 0.0, 10.0, 40.0)
    CALL YGRANS(0.0, 10.0, 40.0, 2.25, 'BMAG(NT)',
+
    100,0.0,0.0)
    IF (QST1.EQ. 'Y') THEN
      CALL CURVE(TIME, ONE, PNTS,0)
    END IF
    IF (QST2.EQ. 'Y') THEN
      CALL DOT
      CALL CURVE(HTIME,HONE,HPNTS,0)
      CALL RESET ('DOT')
    END IF
  END IF
END IF
C*****
C  THE PLOTTING SUBROUTINES AS PER DISPLA
C*****
WRITE(6,210)
210 FORMAT(1X, 'DO YOU WANT SPHERICAL(1) OR RECTANGULAR(2)
COORD?')
READ(5,*)RES
IF ((RES.EQ. 1) OR (RES.EQ. 2)) THEN
  LOGIC1 = .FALSE.
ELSE
  WRITE(6,230)
230 FORMAT(1X, 'WRONG ANSWER DUDE...TRY AGAIN.')
```



```

CALL RESET ('DOT')
ENDIF
ELSE
CALL GRAF((SSEC/3600), 1., (ESEC/3600), -40.0, 20.0, 40.0)
CALL YGRAXS(-40.0, 20.0, 40.0, 2.25, 'BZ AXIS(NT)S',
100.0, 0.0, 0.0)
IF (QST1.EQ.'Y') THEN
CALL CURVE(TME, FOUR, PNTS, 0)
ENDIF
IF (QST2.EQ.'Y') THEN
CALL DOT
CALL CURVE(HTME, HFOUR, HPNTS, 0)
CALL RESET('DOT')
ENDIF
CALL DASH
CALL CURVE(TME2, CENTER, 2, 0)
CALL RESET('DASH')
ENDIF
ENDIF
CALL ENDGR(0)
END DO

CALL ENDPL(0)
CALL DONEPL(0)

100 STOP
END

```

```

C*****
C FILE NAME: POINTS.FOR
C
C NAME: POINTS
C
C PURPOSE: THIS PROGRAM WILL RETRIEVE SELECTED DATA FROM
THE
C PVO84066C.DAT FILE IN THE QUIST.PVO.PVODATA DIRECTORY
C TO FIND THE NUMBER OF POINTS IN THE ARRAYS TO BE USED
C WITH DISSPLA
C
C AUTHOR: FRED QUIST 12/27/93
C
C MODIFIED: NEVER
C*****
SUBROUTINE POINTS(NPNTS,FILENAME,SYRDY,SSEC,EYRDY,ESEC)
C*****
C DEFINE INTEGERS TO BE USED
C
INTEGER LUN, NPNTS
C*****
C DEFINE THE CHARACTER STRING FOR THE FILENAME OF THE DATA
SET
C*****
CHARACTER*50 FILENAME
C*****
C DEFINE THE VARIOUS WORDS TO BE READ FORM PVO84066C.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD3
C MSEC IS THE MILLISECOND OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD8
C
REAL*4 SCID, YRDY, SEC
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE NUMBER
OF POINTS
C OF THE TIME OF INTEREST
C*****
C*****
C FILE NAME: POINTS.FOR
C
C NAME: POINTS
C
C PURPOSE: THIS PROGRAM WILL RETRIEVE SELECTED DATA FROM
THE
C PVO84066C.DAT FILE IN THE QUIST.PVO.PVODATA DIRECTORY
C TO FIND THE NUMBER OF POINTS IN THE ARRAYS TO BE USED
C WITH DISSPLA
C
C AUTHOR: FRED QUIST 12/27/93
C
C MODIFIED: NEVER
C*****
SUBROUTINE POINTS(NPNTS,FILENAME,SYRDY,SSEC,EYRDY,ESEC)
C*****
C DEFINE INTEGERS TO BE USED
C
INTEGER LUN, NPNTS
C*****
C DEFINE THE CHARACTER STRING FOR THE FILENAME OF THE DATA
SET
C*****
CHARACTER*50 FILENAME
C*****
C DEFINE THE VARIOUS WORDS TO BE READ FORM PVO84066C.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD3
C MSEC IS THE MILLISECOND OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD8
C
REAL*4 SCID, YRDY, SEC
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE NUMBER
OF POINTS
C OF THE TIME OF INTEREST
C*****
C*****
REAL*4 EYRDY, SYRDY, ESEC, SSEC
C*****
C INITIALIZE THESE VARIABLES
C*****
LUN = 13
NPNTS = 1
LOGIC1 = .TRUE.
LOGIC2 = .TRUE.
LOGIC3 = .TRUE.
C*****
C READS THE FIRST RECORD TO FIND THE STARTING YRDAY AND SEC,
AND THEN
C CLOSES THE FILE.
C*****
OPEN(UNIT = LUN,
+ FILE = FILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
C*****
READ(LUN, END = 3000) SCID, YRDY, SEC
WRITE(6,2100)HERE IS THE STARTING DATE:, YRDY,SEC
2100 FORMAT(1X,A28.2F8.1)
CLOSE(UNIT = LUN)
C*****
C THIS IS A SIMPLE LOOP TO GET THE INPUTS FOR STARTING AND
STOPING TIMES
C*****
DO WHILE (LOGIC1)
WRITE(6,*)INPUT THE STARTING YRDY(Y'YDDD) AND SECOND(SSSSS):'
READ(5,*) SYRDY, SSEC
WRITE(6,*)INPUT THE ENDING YRDY(Y'YDDD) AND SECOND(SSSSS):'
READ(5,*) EYRDY, ESEC
IF ((SYRDY .GT. EYRDY) .OR. ((SYRDY .GE. EYRDY) .AND.
+ (SSEC .GT. ESEC))) THEN
WRITE(6,30)'STARTING TIME IS EARLIER THAN THE ENDING TIME.'
30 FORMAT(1X,A52)
ELSE
LOGIC1 = .FALSE.
END IF
C*****

```



```

END DO

C*****
C  OPEN INPUT FILE TO BE READ
C*****
      OPEN (UNIT = LUN,
+       FILE = FILENAME,
+       FORM = 'UNFORMATTED',
+       READONLY,
+       STATUS = 'OLD')

C*****
C  READS DATA FILE IN ORDER TO FIND THE NUMBER OF DATA POINTS
C*****
      DO WHILE (LOGIC3)
        READ(LUN, END = 3500)SCID, YRDY, SEC
        IF ((YRDY .GE. SYRDY) .AND. (SEC .GE. SSEC)) THEN
          DO WHILE (LOGIC2)
            READ(LUN, END = 3500)SCID, YRDY, SEC
            IF ((YRDY .GT. EYRDY) .OR. ((YRDY .GE. EYRDY) .AND.
+             (SEC .GE. ESEC))) THEN
              LOGIC2 = .FALSE.
              LOGIC3 = .FALSE.
            ELSE
              NPNTS = NPNTS + 1
            END IF
          END DO
        END IF
      END DO

C*****
C  THAT'S ALL SHE WROTE
C*****
      3500 CLOSE (UNIT = LUN )
      RETURN
      END

```

```

C*****
C File name: MKARRAY.FOR
C
C Name: MKARRAY
C
C Purpose: This program will retrieve selected data from the
C PVO12SEC.dat file in the quist.pvo.pvodata directory
C to create arrays to be used with DISSPLA
C
C Author: Fred Quist 11/24/93
C
C Modified: 12/27/93
C*****
SUBROUTINE MKARRAY(PNTS,FILENAME,SYRDY,SSEC,TME,ONE,TWO,
+ THREE,FOUR,SIZE,RES)
C*****
C DEFINE THE INTEGERS NEEDED IN THIS PROGRAM
C
INTEGER LUN, INDEX, PNTS, RES
C*****
C DEFINING THE CHARACTER STRING FOR THE FILENAME OF THE DATA
C
CHARACTER*50 FILENAME
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
PVMFHOUR.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD 1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD 2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD 3
C MSEC IS THE MILLISEC OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 8
C
REAL*4 SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE ARRAYS
USED
C IN THE DISSPLA ROUTINES
C*****
REAL*4 SYRDY, SSEC, BX, BY, BZ
C*****
C DEFINE THE ARRAYS THAT WILL BE USED WITH DISPLAY
C WHERE: ONE IS THE MAGNITUDE OF THE B-FIELD
C TWO IS THE Y-AXIS
C THREE IS THE X-AXIS OR THE AZMUTHAL ANGLE
C FOUR IS THE Z-AXIS OR THE POLAR ANGLE
C
REAL*4 TME(SIZE),ONE(SIZE),TWO(SIZE),THREE(SIZE),FOUR(SIZE)
C*****
C INITIALIZE THESE VARIABLES
C
LUN = 11
C*****
C OPEN INPUT FILE TO BE READ
C
OPEN (UNIT = LUN,
+ FILE = FILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
C*****
C READS DATA FILES AND CREATES ARRAY'S NEEDED
C
LOGIC3 = .TRUE.
DO WHILE (LOGIC3)
READ( LUN, END = 3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
IF (( YRDY .GE. SYRDY) .AND. (SEC .GE. SSEC)) THEN
DO INDEX = 1,PNTS
BX = BMAG*COSD(BPOL)*COSD(BAZM)
BY = BMAG*COSD(BPOL)*SIND(BAZM)
BZ = BMAG*SIND(BPOL)
IF (RES.EQ.1) THEN
TME(INDEX) = SEC/3600
ONE(INDEX) = BMAG
THREE(INDEX) = BAZM
FOUR(INDEX) = BPOL
ELSE
TME(INDEX) = SEC/3600
C*****
C FILE NAME: MKARRAY.FOR
C
C NAME: MKARRAY
C
C PURPOSE: This program will retrieve selected data from the
C PVO12SEC.dat file in the quist.pvo.pvodata directory
C to create arrays to be used with DISSPLA
C
C AUTHOR: Fred Quist 11/24/93
C
C MODIFIED: 12/27/93
C*****
SUBROUTINE MKARRAY(PNTS,FILENAME,SYRDY,SSEC,TME,ONE,TWO,
+ THREE,FOUR,SIZE,RES)
C*****
C DEFINE THE INTEGERS NEEDED IN THIS PROGRAM
C
INTEGER LUN, INDEX, PNTS, RES
C*****
C DEFINING THE CHARACTER STRING FOR THE FILENAME OF THE DATA
C
CHARACTER*50 FILENAME
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
PVMFHOUR.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD 1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD 2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD 3
C MSEC IS THE MILLISEC OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 8
C
REAL*4 SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE ARRAYS
USED
C IN THE DISSPLA ROUTINES
C*****
REAL*4 SYRDY, SSEC, BX, BY, BZ
C*****
C DEFINE THE ARRAYS THAT WILL BE USED WITH DISPLAY
C WHERE: ONE IS THE MAGNITUDE OF THE B-FIELD
C TWO IS THE Y-AXIS
C THREE IS THE X-AXIS OR THE AZMUTHAL ANGLE
C FOUR IS THE Z-AXIS OR THE POLAR ANGLE
C
REAL*4 TME(SIZE),ONE(SIZE),TWO(SIZE),THREE(SIZE),FOUR(SIZE)
C*****
C INITIALIZE THESE VARIABLES
C
LUN = 11
C*****
C OPEN INPUT FILE TO BE READ
C
OPEN (UNIT = LUN,
+ FILE = FILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
C*****
C READS DATA FILES AND CREATES ARRAY'S NEEDED
C
LOGIC3 = .TRUE.
DO WHILE (LOGIC3)
READ( LUN, END = 3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
IF (( YRDY .GE. SYRDY) .AND. (SEC .GE. SSEC)) THEN
DO INDEX = 1,PNTS
BX = BMAG*COSD(BPOL)*COSD(BAZM)
BY = BMAG*COSD(BPOL)*SIND(BAZM)
BZ = BMAG*SIND(BPOL)
IF (RES.EQ.1) THEN
TME(INDEX) = SEC/3600
ONE(INDEX) = BMAG
THREE(INDEX) = BAZM
FOUR(INDEX) = BPOL
ELSE
TME(INDEX) = SEC/3600

```

```

ONE(INDEX) = BMAG
TWO(INDEX) = BY
THREE(INDEX) = BX
FOUR(INDEX) = BZ
END IF
READ(LUN, END = 3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
END DO
LOGIC3 = .FALSE.
END IF
END DO

C*****
C  CLOSE UP SHOP AND GO HOME.

3500 CLOSE( UNIT = LUN)
RETURN
END

```

```

C*****
C FILE NAME: HPOINTS.FOR
C
C NAME: HPOINTS
C
C PURPOSE: THIS PROGRAM WILL RETRIEVE SELECTED DATA FROM
THE
C PVO84066C.DAT FILE IN THE QJST.PVO.PVODATA DIRECTORY
C TO FIND THE NUMBER OF POINTS IN THE ARRAYS TO BE USED
C WITH DISPLA
C
C AUTHOR: FRED QJST 1/24/93
C
C MODIFIED: NEVER
C*****
SUBROUTINE HPOINTS(HNPNTS, HFILENAME, SYRDY, SSEC, EYRDY,
+ ESEC, QST1)
C*****
C DEFINE INTEGERS TO BE USED
C
INTEGER LUN, HNPNTS
C*****
C DEFINE THE CHARACTER STRING FOR THE FILENAME OF THE DATA
SET
CHARACTER*50 HFILENAME
CHARACTER*5 QST1
C*****
C DEFINE THE VARIOUS WORDS TO BE READ FORM PVO84066C.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD3
C MSEC IS THE MILLISECOND OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD8
C
REAL*4 SCID, YRDY, HR
C*****
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE NUMBER
OF POINTS
C OF THE TIME OF INTEREST
C
REAL*4 EYRDY, SYRDY, ESEC, SSEC, SHR, EHR
C*****
C INITIALIZE THESE VARIABLES
C*****
LUN = 13
HNPNTS = 1
LOGIC1 = .TRUE.
LOGIC2 = .TRUE.
LOGIC3 = .TRUE.
C*****
C READS THE FIRST RECORD TO FIND THE STARTING YRDAY AND SEC.
AND THEN
C CLOSES THE FILE ONLY IF I DIDN'T GET THEM IN POINTS FOR (IE QST1 =
N')
C*****
IF (QST1.EQ. N') THEN
OPEN (UNIT = LUN,
+ FILE = HFILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
READ(LUN, END = 3500) SCID, YRDY, HR
WRITE(6,2100)'HERE IS THE STARTING DATE:', YRDY,HR
2100 FORMAT(1X,A28.2F8.1)
CLOSE (UNIT = LUN)
C*****
C THIS IS A SIMPLE LOOP TO GET THE INPUTS FOR STARTING AND
STOPPING TIMES
C
DO WHILE (LOGIC1)
WRITE(6,*)'INPUT THE STARTING YRDY(YYYDDD) AND HR(SSSS):'
READ (5,*) SYRDY, SHR
WRITE(6,*)'INPUT THE ENDING YRDY(YYYDDD) AND HR(SSSS):'
READ (5,*) EYRDY, EHR

```

```

C   THAT'S ALL SHE WROTE
3500 CLOSE ( UNIT = LUN )
RETURN
END

```

```

IF ((SYRDY .GT. EYRDY) .OR. ((SYRDY .GE. EYRDY) .AND.
+ (SHR .GT. EHR))) THEN
  WRITE(6,30)
30  FORMAT(1X,'STARTING TIME MUST BE BEFORE THE ENDING TIME.')
  ELSE
    LOGIC1 = .FALSE.
  END IF
  END DO
  SSEC = SHR*3600.0
  ESEC = EHR*3600.0
  ELSE
    SHR = SSEC/3600.0
    EHR = ESEC/3600.0
  END IF

```

```

C*****
C  OPEN INPUT FILE TO BE READ

```

```

OPEN (UNIT = LUN,
+ FILE = HFILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')

```

```

C*****
C  READS DATA FILE IN ORDER TO FIND THE NUMBER OF DATA POINTS

```

```

DO WHILE (LOGIC3)

  READ( LUN, END = 3500)SCID, YRDY, HR

  IF ((YRDY .GE. SYRDY) .AND. (HR .GE. SHR)) THEN
    DO WHILE (LOGIC2)
      READ(LUN, END = 3500)SCID, YRDY, HR
      IF ((YRDY .GT. EYRDY) .OR. ((YRDY .GE. EYRDY) .AND.
+ (HR .GT. EHR))) THEN
        LOGIC2 = .FALSE.
        LOGIC3 = .FALSE.
      ELSE
        HNPNTS = HNPNTS + 1
      END IF
    END DO
  END IF
  END DO

```

```

C*****

```

```

C*****
C File name: HMKARRAY.FOR
C
C Name: HMKARRAY
C
C Purpose: This program will retrieve selected data from the
C PVO12SEC.dat file in the quist.pvo.pvodata directory
C to create arrays to be used with DISSPLA
C
C Author: Fred Quist 11/24/93
C
C Modified: 12/27/93
C*****
SUBROUTINE
HMKARRAY(HFILENAME,SYRDY,SSEC,HTME,HONE,HTWO,
+ HTHREE,HFOUR,HSIZE,RES)
C*****
C DEFINE THE INTEGERS NEEDED IN THIS PROGRAM
C
INTEGER LUN, INDEX, HPNTS, RES
C*****
C DEFINGING THE CHARACTER STRING FOR THE FILENAME OF THE DATA
C
CHARACTER*50 HFILENAME
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
PVMFHOUR.DAT FILE
C
C SCID IS THE SPACECRAFT IDENTIFICATION, WORD 1
C YRDY IS THE YEAR AND DAY OF THE DATA, WORD 2
C SEC IS THE SECONDS OF THE DAY THE DATA WAS TAKEN, WORD 3
C MSEC IS THE MILLISEC OF THE DATA, WORD 4
C BT IS THE SCALAR AVERAGE OF THE B-FIELD, WORD 5
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 8
C
REAL*4 SCID, YRDY, HR, DUMMY(6), BMAG, BAZM, BPOL
C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO FIND THE ARRAYS
USED
C
C IN THE DISSPLA ROUTINES
REAL*4 SYRDY, SSEC, BX, BY, BZ, SHR
C*****
C DEFINE THE ARRAYS THAT WILL BE USED WITH DISPLAY
C WHERE: ONE IS THE MAGNITUDE OF THE B-FIELD
C TWO IS THE Y-AXIS
C THREE IS THE X-AXIS OR THE AZMUTHAL ANGLE
C FOUR IS THE Z-AXIS OR THE POLAR ANGLE
C
REAL*4 HTME(HSIZE),HONE(HSIZE),HTWO(HSIZE),HTHREE(HSIZE)
REAL*4 HFOUR(HSIZE)
C*****
C INITIALIZE THESE VARIABLES
C
LUN = 11
SHR = SSEC/3600.0
C*****
C OPEN INPUT FILE TO BE READ
C
OPEN (UNIT = LUN,
+ FILE = HFILENAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')
C*****
C READS DATA FILES AND CREATES ARRAYS NEEDED
C
LOGIC3 = .TRUE.
DO WHILE (LOGIC3)
READ( LUN, END = 3500) SCID,YRDY,HR,DUMMY,BMAG,BAZM,BPOL
IF (( YRDY .GE. SYRDY) .AND. (HR .GE. SHR)) THEN
DO INDEX = 1,HPNTS
BX = BMAG*COSD(BPOL)*COSD(BAZM)
BY = BMAG*COSD(BPOL)*SIND(BAZM)
BZ = BMAG*SIND(BPOL)
IF (RES.EQ.1) THEN
HTME(INDEX) = HR
HONE(INDEX) = BMAG

```

```

      HTHREE(INDEX) = BAZM
      HFOUR(INDEX) = BPOL
    ELSE
      HTME(INDEX) = HR
      HONE(INDEX) = BMAG
      HTWO(INDEX) = BY
      HTHREE(INDEX) = BX
      HFOUR(INDEX) = BZ
    END IF
  C   WRITE(6,*) HPNTS,BMAG,BAZM,BPOL
  READ(LUN, END =3500) SCID,YRDY,HR,DUMMY,BMAG,BAZM,BPOL
  END DO
  LOGIC3 = .FALSE.
  END IF
  END DO

C*****
C   CLOSE UP SHOP AND GO HOME.

3500 CLOSE(UNIT = LUN)
  RETURN
  END

```

```

C*****
C File name: LAMBDA FOR
C
C Name: LAMBDA
C
C Purpose: This program will retrieve selected data from the
C PVO8xxxxC.dat file in the [quist.pvo.pvodata] directory
C for calculating the matrix elements and e-values
C for a plot of the ratio of lambda(max) to lambda(min).
C
C Author: Fred Quist 6/15/94
C
C Modified: NEVER
C
C*****
SUBROUTINE LAMBDA(FNAME, SYRDY, SSEC, EYRDY, ESEC, RATIO,
+ TIME2, CNTR3, SIZE2)
C*****
C PI IS SELF EVIDENT
PARAMETER (PI = 3.141593)
C*****
C SIZES OF THE
PARAMETER (SIZE = 1000)
INTEGER LUN, CNTR2, I, J, K, CNTR3, CNTR4, CNTR5
C*****
C DEFINE THE CHARACTER STRING TO BE USED FOR THE FILENAME
CHARACTER*60 FNAME
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
PVMFHOURL.DAT FILE
C
C BMAG IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6
C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 8
C SCID IS THE NAME OF THE SATELLITE WORD 1
C YRDY IS THE YEAR AND DAY OF THE DATA WORD 2
C SEC IS THE SEC OF THE DAY THE DATA WAS TAKEN WORD 3

REAL*4 SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL

C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO CALCULATE THE
MATRIX
C ELEMENTS AND THE EIGENVECTOR.

REAL*4 MEANX, MEANY, MEANZ, CNTR1, ZERO, EYRDY, SYRDY
REAL*4 MEANZZ, MEANXX, MEANYX, MEANXZ, MEANYZ, MEANYZ,
THETA
REAL*4 Mxx, Mxy, Mxz, Myx, Myz, Mzy, Mzz, A, B, C, Q, R
REAL*4 Bx, By, Bz, Bxy, Bxz, SOLN1, SOLN2, SOLN3
REAL*4 ESEC, SSEC, RATIO(SIZE2), TIME(SIZE), AVGYZ(SIZE)
REAL*4 AVGX(SIZE), AVGY(SIZE), AVGZ(SIZE), AVGXX(SIZE), TIME2(SIZE2)
REAL*4 AVGY(SIZE), AVGZZ(SIZE), AVGXY(SIZE), AVGYZ(SIZE)
REAL*4 LARGE, SMALL

C*****
C INITIALIZE THESE VARIABLES

MEANX = 0.0
MEANY = 0.0
MEANZ = 0.0
CNTR1 = 0.0
CNTR3 = 1
ZERO = 0.0
LUN = 11
LOGIC4 = .TRUE.
LOGIC5 = .TRUE.

C*****
C GET THE NUMBER OF TERMS NEEDED IN A WINDOW

WRITE(6,666)
666 FORMAT('X',ENTER THE NUMBER OF TERMS (ODD NUMBER < 1000) '-')
READ(5,667)CNTR5
667 FORMAT(I4)

C*****
C OPEN INPUT FILE ON UNIT = LUN = 11

OPEN (UNIT = LUN,
+ FILE = FNAME,
+ FORM = 'UNFORMATTED',
+ READONLY,
+ STATUS = 'OLD')

```



```

C*****
C  READS AND WRITES DATA FILES
C*****
LOGIC3 = .TRUE.
LOGIC2 = .TRUE.

DO WHILE (LOGIC3)

  READ( LUN, END = 3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL

  IF ((BMAG.NE.ZERO).AND.(BPOL.NE.ZERO).AND.
      (BAZM.NE.ZERO))) THEN
    MEANX = BMAG*(COSD(BPOL) * COSD(BAZM))
    MEANY = BMAG*(COSD(BPOL) * SIND(BAZM))
    MEANZ = BMAG*(SIND(BPOL))

    MEANXX = (BMAG*(COSD(BPOL) * COSD(BAZM)))**2
    MEANYY = (BMAG*(COSD(BPOL) * SIND(BAZM)))**2
    MEANZZ = (BMAG*(SIND(BPOL)))**2

    MEANXY = (BMAG**2)*((COSD(BPOL)**2)*COSD(BAZM)*SIND(BAZM))
    MEANXZ = (BMAG**2)*(COSD(BPOL) * COSD(BAZM) * SIND(BPOL))
    MEANYZ = (BMAG**2)*(COSD(BPOL) * SIND(BAZM) * SIND(BPOL))
    CNTR1 = 1
  ELSE
    CNTR1 = 0
    MEANX = 0
    MEANY = 0
    MEANZ = 0

    MEANXX = 0
    MEANYY = 0
    MEANZZ = 0

    MEANXY = 0
    MEANXZ = 0
    MEANYZ = 0
  END IF

  IF ((YRDY.GT.SYRDY).OR.((YRDY.GE.SYRDY).AND.
      (SEC.GE.SSEC))) THEN
    +
    AVGX(1) = MEANX
    AVGXZ(1) = MEANXZ
    AVGYZ(1) = MEANYZ
    TIME(1) = SEC
    DO CNTR2 = 2, CNTR5
      READ( LUN, END = 3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL.

      IF ((BMAG.NE.ZERO).AND.(BPOL.NE.ZERO).AND.
          (BAZM.NE.ZERO))) THEN
        +
        C-----CALCULATION OF THE B-FIELD IN X,Y, AND Z COORDINATES.-----
        AVGX(CNTR2) = BMAG*(COSD(BPOL) * COSD(BAZM))
        AVGY(CNTR2) = BMAG*(COSD(BPOL) * SIND(BAZM))
        AVGZ(CNTR2) = BMAG*(SIND(BPOL))

        AVGNX(CNTR2) = (BMAG*(COSD(BPOL) * COSD(BAZM)))**2
        AVGYY(CNTR2) = (BMAG*(COSD(BPOL) * SIND(BAZM)))**2
        AVGZZ(CNTR2) = (BMAG*(SIND(BPOL)))**2

        AVGXY(CNTR2)=(BMAG**2)*((COSD(BPOL)**2)*COSD(BAZM)*SIND(BAZM))
        AVGXZ(CNTR2)=(BMAG**2) * (COSD(BPOL) *COSD(BAZM) * SIND(BPOL))
        AVGYZ(CNTR2)=(BMAG**2) * (COSD(BPOL) *SIND(BAZM) * SIND(BPOL))
        TIME(CNTR2) = SEC/3600
        CNTR1 = CNTR1 + 1
      END IF
      IF ((YRDY.GT.EYRDY).OR.((YRDY.GE.EYRDY).AND.
          (SEC.GE.ESEC))) THEN
        +
        LOGIC3 = .FALSE.
        CNTR4 = 102
      END IF
      END DO
      IF ((CNTR4.GE.101).OR.(CNTR1.EQ.REAL(CNTR5))) THEN
        LOGIC3 = .FALSE.
      END IF
      END IF
      END DO

      IF (CNTR1.EQ.REAL(CNTR5)) THEN
        LOGIC3 = .TRUE.
      ELSE
        WRITE(6,*)THERE ARE LESS THAN',CNTR5,'TERMS'
      END IF
      DO WHILE (LOGIC3)

```

```

MEANY = 0.0
MEANY = 0.0
MEANZ = 0.0
MEANXX = 0.0
MEANY = 0.0
MEANZZ = 0.0
MEANY = 0.0
MEANXZ = 0.0
MEANYZ = 0.0

DO J = 1, INT(CNTR1)
  MEANX = MEANX + AVGX(J)
  MEANY = MEANY + AVGY(J)
  MEANZ = MEANZ + AVGZ(J)
  MEANXX = MEANXX + AVGXX(J)
  MEANY = MEANY + AVGY(J)
  MEANZZ = MEANZZ + AVGZZ(J)
  MEANY = MEANY + AVGY(J)
  MEANXZ = MEANXZ + AVGXZ(J)
  MEANYZ = MEANYZ + AVGYZ(J)
END DO

C*****CALCULATE AVERAGE
VALUES*****
Bx = MEANX / CNTR1
By = MEANY / CNTR1
Bz = MEANZ / CNTR1

Bxx = (MEANXX) / CNTR1
Byy = (MEANY) / CNTR1
Bzz = (MEANZZ) / CNTR1

Bxy = (MEANXY) / CNTR1
Bxz = (MEANXZ) / CNTR1
Byz = (MEANYZ) / CNTR1

C*****CALCULATE MATRIX
ELEMENTS*****
Mxx = (Bxx - (Bx * Bx))
Mxy = (Bxy - (Bx * By))
Mxz = (Bxz - (Bx * Bz))
Myx = Mxy
Myy = (Byy - (By * By))
Myz = (Byz - (By * Bz))
Mzx = Mxz

```

```

Mzy = Myz
Mzz = (Bzz - (Bz * Bz))

C*****FIND CHARACTERISTIC EQUATION*****
C      Y^3 + A*Y^2 + B*Y + C = 0

      A = (-1.0)*(Mxx + Myy + Mzz)
      B = (Mxx*Myy + Mxx*Mzz + Myy*Mzz - Mxy*Mxz - Mxz*Mxy - Mxz*Mxy - Mxz*Mxy)
      C = (Mxx*Myz + Myz*Mxx + Mzz*Myx - Myx*Mzz - Mxz*Mxy - Mxy*Mxz - Mxz*Mxy - Mxy*Mxz)
      +   - Mxy*Mxz - Myz - Mxz*Myx - Mzy)

C*****SOLVE CUBIC POLYNOMIAL*****
C      FROM NUMERICAL RECIPES

      Q = (A**2 - 3.0*B)/9.0
      R = (2.0*(A**3) - 9.0*(A*B) + 27.0*C)/54.0

      IF ((Q**3 - R**2).GE. 0.0) THEN
        THETA = ACOS(R / (Q**1.5))
        SOLN1 = -2.0 * (Q**5) * COS(THETA / 3.0) - A/3
        SOLN2 = -2.0 * (Q**5) * COS((THETA + PI*2.0) / 3.0) - A/3
        SOLN3 = -2.0 * (Q**5) * COS((THETA + PI*4.0) / 3.0) - A/3

        LARGE = MAX(SOLN1,SOLN2,SOLN3)
        SMALL = MIN(SOLN1,SOLN2,SOLN3)
        RATIO(CNTR3) = LARGE/SMALL
        TIMEZ(CNTR3) = TIME((CNTR5 - 1)/2 + 1))
        CNTR3 = CNTR3 + 1
      ELSE
        WRITE (6,*) 'THE RESULTS MUST BE IMAGINARY'
      END IF

      READ(LUN,END=3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
      IF ((YRDY.GT.EYRDY).OR. ((YRDY.GE.EYRDY).AND.
      +   (SEC.GE.ESEC))) THEN
        LOGIC3 = .FALSE.
      ELSE
        DO1 = 1,INT((CNTR1 - 1))
          K = 1 + J
          AVGX(I) = AVGX(K)
          AVGY(I) = AVGY(K)
          AVGZ(I) = AVGZ(K)

          AVGXX(I) = AVGXX(K)
          AVGY(I) = AVGY(K)

```

```

AVGZZ(I) = AVGZZ(K)

AVGXY(I) = AVGXY(K)
AVGXZ(I) = AVGXZ(K)
AVGYZ(I) = AVGYZ(K)

TIME(I) = TIME(K)
END DO

AVGX(CNTR5) = BMAG*(COSD(BPOL) * COSD(BAZM))
AVGY(CNTR5) = BMAG*(COSD(BPOL) * SIND(BAZM))
AVGZ(CNTR5) = BMAG*(SIND(BPOL))

AVGX(CNTR5) = (BMAG*(COSD(BPOL) * COSD(BAZM)))**2
AVGY(CNTR5) = (BMAG*(COSD(BPOL) * SIND(BAZM)))**2
AVGZ(CNTR5) = (BMAG*(SIND(BPOL)))**2

AVGXY(CNTR5)=(BMAG**2)*((COSD(BPOL)**2)
+ *COSD(BAZM)*SIND(BAZM))
AVGXZ(CNTR5)=(BMAG**2)*(COSD(BPOL)*COSD(BAZM)*SIND(BPOL))
AVGYZ(CNTR5)=(BMAG**2)*(COSD(BPOL)*SIND(BAZM)*SIND(BPOL))
TIME(CNTR5) = SEC/3600
END IF
END DO

C*****
C CLOSE UP SHOP AND GO HOME.
3500 CLOSE(UNIT = LUN)
RETURN
END

```

```

C*****
C File name: MATRIX.FOR
C
C Name: MATRIX
C
C Purpose: This program will retrieve selected data from the
C PVO8xxxxC.dat file in the [quist.pvo.pvodata] directory
C for calculating the matrix elements and e-values
C for calculating the normal to the current sheet.
C
C Author: Fred Quist 5/14/93
C
C Modified: 6/15/94
C
C LINKED PROGRAMS: MATRIX.FOR, EIGENVEC.FOR, PLANE.FOR,
C PLANE.PLOT.FOR,
C EULER.FOR, AND THE DISPLA FUNCTIONS.
C*****

PROGRAM MATRIX
C*****
C PI AND DEG ARE SELF EVIDENT
C
PARAMETER (PI = 3.141593)
PARAMETER (DEG = 180.0)
C*****
C RAD IS THE AVERAGE DISTANCE FROM VENUS TO THE SUN
C OMEGA IS THE ROTATIONAL VELOCITY OF THE SUN
C
PARAMETER (RAD = 108000000)
PARAMETER (OMEGA = .0000027)
C
INTEGER LU'N, RES2, CNTR2, RES1
C*****
C DEFINE THE CHARACTER STRING TO BE USED FOR THE FILENAME
C
CHARACTER*60 FNAME
CHARACTER*1 RES3
C*****
C THE DEFINING OF THE VARIOUS WORDS TO BE READ FROM
C PVMIFHOUR.DAT FILE
C
BMAZ IS THE VECTOR AVERAGE OF THE B-FIELD, WORD 6

```

```

C BAZM IS THE AZMUTHAL ANGLE OF THE B-FIELD, WORD 7
C BPOL IS THE POLAR ANGLE OF THE B-FIELD, WORD 8
C SCID IS THE NAME OF THE SATELLITE WORD 1
C YRDY IS THE YEAR AND DAY OF THE DATA WORD 2
C SEC IS THE SEC OF THE DAY THE DATA WAS TAKEN WORD 3

REAL*4 SCID, YRDY, SEC, MSEC, BT, BMAG, BAZM, BPOL.

C*****
C DEFINE SOME VARIABLES THAT WILL BE USED TO CALCULATE THE
C MATRIX
C ELEMENTS AND THE EIGENVECTOR.

REAL*4 MEANX, MEANY, MEANZ, CNTR1, ZERO, EYRDY, SYRDY, MEANB
REAL*4 MEANZZ, MEANXX, MEANY, MEANXZ, MEANY, MEANYZ,
THETA
REAL*4 Mxx, Mxy, Mxz, Myx, Myy, Myz, Mzx, Mzy, Mzz, A, B, C, Q, R
REAL*4 Bx, By, Bz, Bxx, Bxy, Bxz, SOLN1, SOLN2, SOLN3
REAL*4 ESEC, SSEC, VECX, VECY, VECZ
REAL*4 VECN1X, VECN1Y, VECN1Z, VECN2X, VECN2Y, VECN2Z, VECN3X, VECN3Y, VECN3Z

C*****
C INITIALIZE THESE VARIABLES
C

MEANX = 0.0
MEANY = 0.0
MEANZ = 0.0
MEANB = 0.0
CNTR1 = 0
ZERO = 0.0
LU'N = 11
RES3 = 'O'
LOGIC4 = .TRUE.
LOGIC5 = .TRUE.

C*****
C GET THE FILENAME OF THE DATA FILE TO BE USED
C

WRITE(6,120)
120 FORMAT(1X,ENTER THE NUMBER OF THE YEAR YOU WANT?)
121 FORMAT(1X,1 = 1984,1X,2 = 1986,1X,3 = 1987)
READ(5,122)RES1
122 FORMAT(I2)

WRITE(6,75)
75 FORMAT(1X,ENTER THE NUMBER OF THE FILE TO BE READ)

```

```

IF (RES1.EQ.1) THEN
  WRITE(6,76)
  FORMAT(1X,'(1) = PVO84066C.DAT')
76  WRITE(6,77)
  FORMAT(1X,'(2) = PVO84081C.DAT')
77  WRITE(6,78)
  FORMAT(1X,'(3) = PVO84094C.DAT')
78  WRITE(6,79)
  FORMAT(1X,'(4) = PVO84106C.DAT')
79  WRITE(6,80)
  FORMAT(1X,'(5) = PVO84192C.DAT')
80  WRITE(6,81)
  FORMAT(1X,'(6) = PVO84205C.DAT')
81  WRITE(6,82)
  FORMAT(1X,'(7) = PVO84221C.DAT')
82  WRITE(6,83)
  FORMAT(1X,'(8) = PVO84235C.DAT')
83  ELSE IF (RES1.EQ.2) THEN
  WRITE(6,84)
  FORMAT(1X,'(9) = PVO86255C.DAT')
84  WRITE(6,85)
  FORMAT(1X,'(10) = PVO86272C.DAT')
85  WRITE(6,86)
  FORMAT(1X,'(11) = PVO86326C.DAT')
86  WRITE(6,87)
  FORMAT(1X,'(12) = PVO86337C.DAT')
87  WRITE(6,88)
  FORMAT(1X,'(13) = PVO86343C.DAT')
88  WRITE(6,89)
  FORMAT(1X,'(14) = PVO86354C.DAT')
89  ELSE IF (RES1.EQ.3) THEN
  WRITE(6,90)
  FORMAT(1X,'(15) = PVO87003C.DAT')
90  WRITE(6,91)
  FORMAT(1X,'(16) = PVO87009C.DAT')
91  WRITE(6,92)
  FORMAT(1X,'(17) = PVO87014C.DAT')
92  WRITE(6,93)
  FORMAT(1X,'(18) = PVO87019C.DAT')
93  WRITE(6,94)
  FORMAT(1X,'(19) = PVO87043C.DAT')
94  WRITE(6,95)
  FORMAT(1X,'(20) = PVO87064C.DAT')
95  END IF

  READ(5,96) RES2
  FORMAT(12)

  IF (RES2.EQ.1) THEN
    FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84066C.DAT'
    ELSE IF (RES2.EQ.2) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84081C.DAT'
    ELSE IF (RES2.EQ.3) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84094C.DAT'
    ELSE IF (RES2.EQ.4) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84106C.DAT'
    ELSE IF (RES2.EQ.5) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84192C.DAT'
    ELSE IF (RES2.EQ.6) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84205C.DAT'
    ELSE IF (RES2.EQ.7) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84221C.DAT'
    ELSE IF (RES2.EQ.8) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo84235C.DAT'
    ELSE IF (RES2.EQ.9) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86255C.DAT'
    ELSE IF (RES2.EQ.10) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86272C.DAT'
    ELSE IF (RES2.EQ.11) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86326C.DAT'
    ELSE IF (RES2.EQ.12) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86337C.DAT'
    ELSE IF (RES2.EQ.13) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86343C.DAT'
    ELSE IF (RES2.EQ.14) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo86354C.DAT'
    ELSE IF (RES2.EQ.15) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87003C.DAT'
    ELSE IF (RES2.EQ.16) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87009C.DAT'
    ELSE IF (RES2.EQ.17) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87014C.DAT'
    ELSE IF (RES2.EQ.18) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87019C.DAT'
    ELSE IF (RES2.EQ.19) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87043C.DAT'
    ELSE IF (RES2.EQ.20) THEN
      FNAME = 'SCISDISK:QUIST.PVO.PVODATA\pvo87064C.DAT'
    END IF

    C*****
    C  ASK THE USER IF THEY WANT TO PLOT THE RESULTS.
    DO WHILE (LOGIC5)

C*****

```

```

97 WRITE(6,97)
   FORMAT(1X,'DO YOU WANT THE RESULTS PLOTTED? (Y/N)')
98 READ(5,98)RES3
   FORMAT(A)
   IF (RES3.EQ.'Y').OR.(RES3.EQ.'N') THEN
     LOGIC5 = .FALSE.
     END IF
     END DO

C*****
C READS THE FIRST RECORD TO FIND THE STARTING YYDDD AND HOUR.
C AND THEN CLOSSES THE FILE

   OPEN (UNIT = LUN,
+       FILE = FNAME,
+       FORM = 'UNFORMATTED',
+       READONLY,
+       STATUS = 'OLD')

2100 FORMAT(1X, A28, 2F8.1)
   CLOSE (UNIT = LUN)

C*****
C THIS IS A SIMPLE LOOP TO GET THE INPUTS FOR STARTING AND
STOPPING TIMES

   LOGIC1 = .TRUE.
   DO WHILE (LOGIC1)

     WRITE(6,*)'INPUT THE STARTING YEARDAY(yyddd) AND SEC(ssss):'
     READ (5,*)SYRDY,SSEC

     WRITE (6,*)'INPUT THE ENDING YEARDAY(yyddd) AND SECOND(ssss):'
     READ (5,*)EYRDY,ESEC

     IF (SYRDY.LT. YRDY) THEN
       WRITE (6,*)'PLEASE MAKE SURE YOU USE THE CORRECT TIMES.'
       WRITE (6,*)'YEARDAY IS GREATER THAN', YRDY, ','
       ELSE IF ((SYRDY.GT. EYRDY).OR. ((SYRDY.GE. EYRDY).AND.
+         (SSEC.GE. ESEC))) THEN
         WRITE(6,*)'STARTING TIME MUST BE BEFORE THE ENDING TIME.'
       ELSE
         LOGIC1 = .FALSE.
         END IF
         END DO

```

```

C*****
C OPEN INPUT FILE ON UNIT = LUN = 11

   OPEN (UNIT = LUN,
+       FILE = FNAME,
+       FORM = 'UNFORMATTED',
+       READONLY,
+       STATUS = 'OLD')

C*****
C READS AND WRITES DATA FILES

   LOGIC2 = .TRUE.
   LOGIC3 = .TRUE.

   DO WHILE (LOGIC3)

     READ( LUN, END =3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL

     IF (BMAG.NE. ZERO).AND. ((BPOL.NE. ZERO).AND.
+       (BAZM.NE. ZERO))) THEN
       MEANX = BMAG*( COSD(BPOL) * COSD(BAZM))
       MEANY = BMAG*( COSD(BPOL) * SIND(BAZM))
       MEANZ = BMAG*( SIND(BPOL))

       MEANXX = (BMAG*( COSD(BPOL) * COSD(BAZM)))**2
       MEANYY = (BMAG*( COSD(BPOL) * SIND(BAZM)))**2
       MEANZZ = (BMAG*( SIND(BPOL)))**2

       MEANXY = (BMAG**2)*((COSD(BPOL)**2) * COSD(BAZM)*SIND(BAZM))
       MEANXZ = (BMAG**2)*(COSD(BPOL) * COSD(BAZM) * SIND(BPOL))
       MEANYZ = (BMAG**2)*(COSD(BPOL) * SIND(BAZM) * SIND(BPOL))
       CNTRI = 1
       ELSE
         CNTRI = 0
         MEANX = 0
         MEANY = 0
         MEANZ = 0

         MEANXX = 0
         MEANYY = 0
         MEANZZ = 0

         MEANXY = 0
         MEANXZ = 0
         MEANYZ = 0

```

```

END IF

IF ((YRDY.GT.SYRDY).OR.((YRDY.GE.SYRDY).AND.
  (SEC.GE.SSEC))) THEN
  DO WHILE (LOGIC2)
    READ(LUN,END=3500) SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
  IF ((BMAG.NE.ZERO).AND.((BPOL.NE.ZERO).AND.
    (BAZM.NE.ZERO))) THEN
    C-----CALCULATION OF THE B-FIELD IN X,Y, AND Z COORDINATES.-----
    MEANX = BMAG*(COSD(BPOL) * COSD(BAZM)) + MEANX
    MEANY = BMAG*(COSD(BPOL) * SIND(BAZM)) + MEANY
    MEANZ = BMAG*(SIND(BPOL)) + MEANZ
    MEANXX = (BMAG*(COSD(BPOL) * COSD(BAZM)))**2 + MEANXX
    MEANYY = (BMAG*(COSD(BPOL) * SIND(BAZM)))**2 + MEANYY
    MEANZZ = (BMAG*(SIND(BPOL)))**2 + MEANZZ
    MEANYY = (BMAG**2)*((COSD(BPOL)**2) * COSD(BAZM)*SIND(BAZM))
      + MEANXX
    MEANZZ = (BMAG**2) * (COSD(BPOL) * COSD(BAZM) * SIND(BPOL))
      + MEANXZ
    MEANYZ = (BMAG**2) * (COSD(BPOL) * SIND(BAZM) * SIND(BPOL))
      + MEANYZ
    CNTRI = CNTRI + 1
  END IF
  IF ((YRDY.GT.EYRDY).OR.((YRDY.GE.EYRDY).AND.
    (SEC.GE.ESEC))) THEN
    LOGIC2 = .FALSE.
    LOGIC3 = .FALSE.
  END IF
END DO
IF (CNTRI.GT.1.0) THEN
  C-----CALCULATE AVERAGE
  VALUES-----
  MEANB = ((MEANX**2) - (MEANY**2) - (MEANZ**2))**.5
  Bx = MEANX / CNTRI

By = MEANY / CNTRI
Bz = MEANZ / CNTRI

Bxx = (MEANXX) / CNTRI
Byy = (MEANYY) / CNTRI
Bzz = (MEANZZ) / CNTRI

Bxy = (MEANXY) / CNTRI
Bxz = (MEANXZ) / CNTRI
Byz = (MEANYZ) / CNTRI

C-----CALCULATE MATRIX
ELEMENTS-----
Mxx = (Bxx - (Bx * Bx))
Mxy = (Bxy - (Bx * By))
Mxz = (Bxz - (Bx * Bz))
Myx = Mxy
Myz = (Byy - (By * By))
Mzx = Mxz
Mzy = Myz
Mzz = (Bzz - (Bz * Bz))

C-----FIND CHARACTERISTIC EQUATION-----
C Y**3 + A*Y**2 + B*Y + C = 0
A = (-1.0)*(Mxx + Myy + Mzz)
B = (Mxx*Myy + Mxx*Mzz + Myy*Mzz - Myz*Mxz - Mxz*Mzx - Mxz*Myz)
C = (Mxx*Myz*Mzy + Mzz*Myx + Myy*Mz - Mxz*Mzx - Mxz*Myz)
  + - Myx*Mz - Myz*Mzx - Mxz*Myx - Mzy*Mzx

C-----SOLVE CUBIC POLYNOMIAL-----
C FROM NUMERICAL RECIPES
Q = (A**2 - 3.0*B)/9.0
R = (2.0*(A**3) - 9.0*(A*B) + 27.0*C)/54.0
IF ((Q**3 - R**2).GE.0.0) THEN
  THETA = ACOS(R / (Q**1.5))
  SOLN1 = -2.0 * (Q**.5) * COS(THETA / 3.0) - A/3
  SOLN2 = -2.0 * (Q**.5) * COS(THETA + PI*2.0/3.0) - A/3
  SOLN3 = -2.0 * (Q**.5) * COS(THETA + PI*4.0/3.0) - A/3

```

```

ELSE
  WRITE (6,*) 'THE RESULTS MUST BE IMAGINARY'
  END IF
C*****SEND THE MATRIX ELEMENTS TO
EIGENVEC.FOR*****
      CALL EIGENVEC(SOLN1,SOLN2,SOLN3,Mxx,Mxy,Mxz,Myx,Myy,Mzz,
+      VECX,VECY,VE CZ,VE CXMX,VE CYMX,VE CZMX)
C*****SEND IMPORTANT INFORMATION TO BE
PLOTTED*****
      IF (RES3.EQ.'Y') THEN
        CNTR2 = INT(CNTR1)
        CALL
        PLANE(FNAME,SYRDY,SSEC,CNTR2,VECX,VECY,VE CZ,VE CXMX,
+      VECYMX,VE CZMX)
      END IF
    END IF
  END IF
  END DO
C*****
C  CLOSE UP SHOP AND GO HOME.
3500 CLOSE(UNIT = IUN)
      STOP
      END

```



```

C*****
C   File name: EIGENVEC.FOR
C
C   NAME:   EIGENVEC
C
C   PURPOSE: TO CALCULATE THE E-VECTORS THAT WILL GIVE THE
DIRECTION
C   OF THE NORMAL TO THE CURRENT SHEET. THE MATRIX
ELEMENTS
C   AND EIGENVALUES ARE CREATED BY THE MATRIX.FOR
PROGRAM.
C
C   AUTHOR:  FRED QUIST    6/18/93
C
C   MODIFIED: 9/1/94
C*****
C
SUBROUTINE EIGENVEC(EV1,EV2,EV3,M11,M12,M13,M22,M23,M33,
+   VECX,VECY,VECX,VECMX,VECYMX,VECMYMX)
C*****
C   DEFINE MATRIX VARIABLES
C
REAL*4 M11,M12,M13,M22,M23,M33,EV1,EV2,EV3,DOTP
REAL*8 Mxx,Myy,Mzz,Mxy,Mxz,Mzx,Mzy,Mzz
REAL*8 SOLN1,SOLN2,SOLN3,ONE,TWO,THREE,MAX,MID,MIN,RATIO
REAL*8 Mxx2,Mxz2,Mxy1,Mxz1,X1,X2,X3,Myy3,Myz3
REAL*8 Myzn,Mzxn,Mzzn,Myx2,Myz2,Mzy2,Mzz2
REAL*4 VECX,VECY,VECMX,VECMY,VECYMX,VECMYMX,NORM
REAL*4 TEMPX,TEMPY,TEMPZ
C*****
C   I NEED TO GO THROUGH THE LOOP THREE TIMES.
C   DO J = 1,3
C*****
C   INITIALIZE SOME VARIABLES
C*****
X1 = 0.0
X2 = 0.0
X3 = 1.0
RES = .FALSE.
ROW12 = .FALSE.
ROW13 = .FALSE.
C*****
C   CHECK TO SEE IF THE RATIO OF MID-MIN IS GREATER THAN 1.5
C   SEE SUNNERUP AND CAHILL PAPER FOR REASON
C*****

```

```

IF (MIN .GT. 0.0) THEN
  RATIO = ABS(MID / MIN)
ELSE
  RATIO = 2.0
END IF

IF (RATIO .GT. 1.5) THEN

C*****CALCULATION OF THE EIGEN-VECTOR FOR THE SMALLEST E-
VALUE*****
C*****AND THE LARGEST E-
VALUE*****
C*****SET UP THE MATRIX
ELEMENTS*****

  IF (J .EQ. 1) THEN
    Mxx = DBLE(M11)
    Mxx = Mxx - MIN
    Mxy = DBLE(M12)
    Mxz = DBLE(M13)
    Mxx = DBLE(M12)
    Mxy = DBLE(M22)
    Mxy = Myy - MIN
    Myz = DBLE(M23)
    Mxz = DBLE(M13)
    Mxz = DBLE(M23)
    Mzy = DBLE(M13)
    Mzz = DBLE(M33)
    Mzz = Mzz - MIN
  ELSE IF (J .EQ. 2) THEN
    Mxx = DBLE(M11)
    Mxx = Mxx - MID
    Mxy = DBLE(M12)
    Mxz = DBLE(M13)
    Mxy = DBLE(M12)
    Mxy = Myy - MID
    Myz = DBLE(M23)
    Mxz = DBLE(M13)
    Mxz = DBLE(M23)
    Mzy = DBLE(M13)
    Mzz = DBLE(M33)
    Mzz = Mzz - MID
  ELSE IF (J .EQ. 3) THEN
    Mxx = DBLE(M11)
    Mxx = Mxx - MAX
    Mxy = DBLE(M12)

Mxz = DBLE(M13)
Mxx = DBLE(M12)
Myy = DBLE(M22)
Myy = Myy - MAX
Mxz = DBLE(M23)
Mxz = DBLE(M13)
Mzy = DBLE(M23)
Mzz = DBLE(M33)
Mzz = Mzz - MAX
END IF

  IF (J .EQ. 1) THEN
    WRITE (6,100) 'THE SOLUTIONS ARE: MIN, MID, MAX'
100  FORMAT ('0', A18, 1X, F9.5, 1X, F9.5, 1X, F9.5)
    END IF

    WRITE (6,200) 'THE MATRIX ELEMENTS ARE:'
200  FORMAT (1X, A24)
    WRITE (6,300) Mxx, Mxy, Mxz
300  FORMAT (1X, F11.6, 1X, F11.6, 1X, F11.6)
    WRITE (6,400) Mxy, Myy, Myz
400  FORMAT (1X, F11.6, 1X, F11.6, 1X, F11.6)
    WRITE (6,500) Mxz, Mzy, Mzz
500  FORMAT (1X, F11.6, 1X, F11.6, 1X, F11.6)

C*****ROW REDUCE TO GET X1=X1(X3) AND
X2=X2(X3)*****
C*****THIS IS AN ANALYSIS OF ROW
ONE*****
  IF (Mxx .EQ. 0.0) THEN
    IF ((Mxy .EQ. 0.0) .AND. (Mxz .NE. 0.0)) THEN
      X3 = 0.0
      COND1 = .TRUE.
    ELSE IF ((Mxy .EQ. 0.0) .AND. (Mxz .EQ. 0.0)) THEN
      WRITE (6,550) 'THERE IS NO INFORMATION FROM ROW ONE.'
550  FORMAT (1X, A37)
      ROW23 = .TRUE.
    ELSE
      X2 = (-1.0) * Mxz / Mxy
      END IF

C*****SECOND COLUMN
YIELDS*****

```

```

C*****
C IF NEEDED, THIS PART RUNS IN ORDER TO FIND THE THIRD VALUE OF X1,
X2, OR X3
C THIS WOULD BE THE CASE IF ONE OF THE ROWS WERE ALL ZEROS.

IF (ROW12.EQ. .TRUE.) THEN
  Myx2 = Myx - (Myx * Mxx2)
  Myz2 = Myz - (Myx * Mxz2)
  IF (Myx2.EQ. 0.0) THEN
    X3 = 0.0
  ELSE
    X1 = (-1.0) * (Myz2 / Myx2)
  END IF
END IF

IF (ROW13.EQ. .TRUE.) THEN
  Mzy2 = Mzy - (Mzx * Mxy1)
  Mzz2 = Mzz - (Mzx * Mxz1)
  IF (Mzy2.EQ. 0.0) THEN
    X3 = 0.0
  ELSE
    X2 = (-1.0) * (Mzz2 / Mzy2)
  END IF
END IF

IF (ROW23.EQ. .TRUE.) THEN
  IF (Myx.EQ. 0.0) THEN
    IF (Myz.EQ. 0.0) THEN
      X3 = 0.0
    ELSE
      X2 = (-1.0) * Myz / Myx
    IF (Mzx.EQ. 0.0) THEN
      IF (Mzy.EQ. 0.0) THEN
        X3 = 0.0
      END IF
    ELSE
      X1 = (-1.0) * ((X2 * Mzy) + Mzz) / Mzx
    END IF
  END IF
ELSE
  IF (Myx.EQ. 0.0) THEN
    X1 = (-1.0) * Myz / Myx
  IF (Mzy.EQ. 0.0) THEN
    IF (Mzx.EQ. 0.0) THEN
      X3 = 0.0
    END IF
  ELSE
    IF (Myx.EQ. 0.0) THEN
      X1 = (-1.0) * Myz / Myx
    IF (Mzy.EQ. 0.0) THEN
      IF (Mzx.EQ. 0.0) THEN
        X3 = 0.0
      END IF
    END IF
  END IF
END IF

```

```

ELSE
  Mxy1 = Mxy / Mxx
  Mxz1 = Mxz / Mxx

  Myyn = Myy - (Myx * Mxy1)
  Myzn = Myz - (Myx * Mxz1)

  IF ((Myyn.EQ. 0.0) .AND. (Myzn.NE. 0.0)) THEN
    X3 = 0.0
  ELSE IF ((Myyn.EQ. 0.0) .AND. (Myzn.EQ. 0.0)) THEN
    WRITE (6,560) 'THERE IS NO INFORMATION FROM ROW TWO.'
    FORMAT (1X, A37)
    ROW13 = .TRUE.
  ELSE
    X2 = (-1.0) * Myzn / Myyn
  END IF
END IF

C*****AGAIN LOOKING AT ROW
ONE*****

IF (Mxy.EQ. 0.0) THEN
  IF (Mxx.NE. 0.0) THEN
    X1 = (-1.0) * Mxz / Mxx
  END IF
END IF

C*****THIRD COLUMN*****
YIELDS*****

ELSE
  Mxx2 = Mxx / Mxy
  Mxz2 = Mxz / Mxy

  Mxzn = Mzx - (Mzy * Mxx2)
  Mznn = Mzz - (Mzy * Mxz2)

  IF ((Mxzn.EQ. 0.0) .AND. (Mznn.NE. 0.0)) THEN
    X3 = 0.0
  ELSE IF ((Mxzn.EQ. 0.0) .AND. (Mznn.EQ. 0.0)) THEN
    WRITE (6,570) 'THERE IS NO INFORMATION FROM ROW THREE.'
    FORMAT (1X, A39)
    ROW12 = .TRUE.
  ELSE
    X1 = (-1.0) * Mznn / Mxzn
  END IF
END IF

```

```

ELSE
  X2 = (-1.0) * ((X1 * Mzx) + Mzz) / Mzy
END IF
ELSE
  Myy3 = Myy/Myx
  Myz3 = Myz/Myx
  X2 = (-1.0) * (Mzz - (Mzx * Myz3)) / (Mzy - (Mzx * Myy3))
  X1 = (-1.0) * ((Myy * X2) + Myz) / Myx
END IF
END IF
END IF

C*****
C IF X3 IS EQUAL TO ZERO, THEN I MUST FIND THE RELATION BETWEEN X1
AND X2

IF (X3.EQ.0.0) THEN
  IF (COND1.EQ.FALSE.) THEN
    IF (Mxx.EQ.0.0) THEN
      X2 = 0.0
      X1 = 1.0
    ELSE
      X1 = (-1.0) * Mxy/Mxx
      X2 = 1.0
    END IF
  ELSE
    IF (Myy.NE.0.0).OR.(Mzy.NE.0.0) THEN
      IF (Mxx.EQ.0.0) THEN
        X1 = 1.0
        X2 = 0.0
      ELSE
        X1 = (-1.0) * Myy/Myx
        X2 = 1.0
      END IF
    ELSE
      IF (Mzx.EQ.0.0) THEN
        X1 = 1.0
        X2 = 0.0
      ELSE
        X1 = (-1.0) * Mzy/Mzx
        X2 = 1.0
      END IF
    END IF
  END IF
END IF

NORM = SQRT(X1*X1 + X2*X2 - X3*X3)

X1 = X1/NORM
X2 = X2/NORM
X3 = X3/NORM

IF (J.EQ.1) THEN
  WRITE(6,600) THE NORMAL IS: X1, X2, X3
600 FORMAT (1X, A19, F9.5, 1X, F9.5, 1X, F9.5)
ELSE IF (J.EQ.2) THEN
  WRITE(6,601) THE MID EIGENVEC IS: X1, X2, X3
601 FORMAT (1X, A19, F9.5, 1X, F9.5, 1X, F9.5)
ELSE IF (J.EQ.3) THEN
  WRITE(6,602) THE MAX EIGENVEC IS: X1, X2, X3
602 FORMAT (1X, A19, F9.5, 1X, F9.5, 1X, F9.5)
END IF

ELSE
  WRITE(6,700) THE BOUNDARY MUST NOT BE WELL DEFINED.
700 FORMAT (1X, A27)
END IF

IF (J.EQ.1) THEN
  VECX = SNGL(X1)
  VECY = SNGL(X2)
  VECZ = SNGL(X3)
ELSE IF (J.EQ.2) THEN
  VECXMX = SNGL(X1)
  VECYMX = SNGL(X2)
  VECZMX = SNGL(X3)
  TEMPX = VECXMX
  TEMPY = VECYMX
  TEMPZ = VECZMX
ELSE IF (J.EQ.3) THEN
  VECXMX = SNGL(X1)
  VECYMX = SNGL(X2)
  VECZMX = SNGL(X3)
  DOTP = ACOSD(VECN*TEMPX + VECY*TEMPY + VECZ*TEMPZ)
  WRITE(6,*) THE NOR-MID DOT PRODUCT IS: DOTP

  DOTP = ACOSD(VECN*VECXMX + VECY*VECYMX + VECZ*VECZMX)
  WRITE(6,*) THE NOR-MAX DOT PRODUCT IS: DOTP

  DOTP = ACOSD(VECNMX*TEMPX + VECYMX*TEMPY +
  VECZMX*TEMPZ)
  WRITE(6,*) THE MID-MAX DOT PRODUCT IS: DOTP
END IF
END DO

```

```
C*****  
C      RETURN TO MATRN.FOR  
  
      RETURN  
      END
```

```

C*****
C   FILE NAME: PLANE.FOR
C   NAME:      PLANE
C
C   PURPOSE:  THIS PROGRAM WILL TAKE THE NORMAL TO THE
CURRENT SHEET.
C   THAT WAS FOUND USING EIGENVEC.FOR AND MATRIX.FOR.
AND
C   PLOT THE PROJECTION OF THE INDIVIDUAL B-FIELD VECTORS
C   ONTO THE CURRENT SHEET.
C
C   AUTHOR:   FRED QUIST   2/11/94
C   MODIFIED: NEVER
C*****
SUBROUTINE PLANE(FILENAME,SYRDY,SSEC,CNTRI,VECX,VECY,VE CZ,
+   VECXMX,VECYMX,VE CZMX)
PARAMETER (SIZE = 10000)
C*****
C   DEFINE THE VARIABLES
C
INTEGER LUN,CNTRI
REAL*4 SYRDY,SSEC,VECX,VECY,VE CZ,SSEC,EYRDY
REAL*4 SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
REAL*4
BX,BY,BZ,PROJ,PROJX,PROJY,PROJZ,ABSVECY,RPOL,RAZM,RAZM2
REAL*4 ONE(SIZE),TWO(SIZE),THREE(SIZE)
REAL*4 VECXMX,VECYMX,VE CZMX
REAL*4 TEMPX,TEMPY,TEMPZ,DOTP
CHARACTER*60 FILENAME
C*****
C   INITIALIZE THESE VALUES
LUN = 14
LOGIC1 = .TRUE.
C*****
C   FIND THE ANGLES BETWEEN THE CURRENT SHEET NORMAL AND THE
X,Y,Z
C   COORDINATES. THESE ARE NEEDED FOR THE ROTATION MATRIX
(EULER ANGLES)
ABSVECY = ABS(VECY)
RPOL = ASIND(VE CZ)
IF (RPOL.NE. 90.0).OR. (RPOL.NE. -90.0) THEN
RAZM = ASIND(ABSVECY/COSD(RPOL))
ELSE
RAZM = 0.0
END IF
IF (VECX.LE. 0.0).AND. (VECY.GT. 0.0) THEN
RAZM = 180 - RAZM
ELSE IF ((VECX.LT. 0.0).AND. (VECY.LE. 0.0)) THEN
RAZM = 180 + RAZM
ELSE IF ((VECX.GE. 0.0).AND. (VECY.LT. 0.0)) THEN
RAZM = 360 - RAZM
END IF
C***** THE FIRST TWO ROTATIONS ARE
*****
RPOL = 90 - RPOL
RAZM = RAZM + 90
C*****
C   FIND THE SECOND ROTATION THAT POINTS THE OLD X IN THE
DIRECTION
C   OF THE LAMBDA MAX EIGENVECTOR.
C*****
C   FIND THE ANGLE BETWEEN THE ROTATED X-AXIS AND THE LAMBDA
MAX VECTOR
C   BY USING THE DOT PRODUCT.
RAZM2 = ABS(ACOSD(VECXMX*COSD(RAZM) + VECYMX*SIND(RAZM)))
C*****
C   DETERMINE WHICH DIRECTION TO ROTATE: COUNTER CLOCKWISE OR
CLOCKWISE
C   THIS WILL BE THE THIRD ROTATION. IF CROSS PRODUCT IS PARALLEL
TO THE
C   NORMAL RAZM2 = RAZM2. IF CROSS PRODUCT IS INTL-PARALLEL
RAZM2 = -RAZM2

```

```

TEMPX = SIND(RAZM)*VECMX
TEMPY = -COSD(RAZM)*VECMX
TEMPZ = COSD(RAZM)*VECMX - SIND(RAZM)*VECMX

C*****DOT PRODUCT WITH
NORMAL*****
DOTP = TEMPX*VECX + TEMPY*VECY + TEMPZ*VE CZ
WRITE(6,*) RAZM2 TEST...DOTP IS ', DOTP

IF (DOTP .GT. 0) THEN
  RAZM2 = RAZM2
ELSE IF (DOTP .LT. 0) THEN
  RAZM2 = -RAZM2
END IF

WRITE(6,110)RAZM, RPOL, RAZM2
110  FORMAT(1X,THE ROTATION ANGLES ARE '3F8.3)

C*****
C  OPEN THE RAW DATA FILE:
OPEN(UNIT = LUN,
+   FILE = FILENAME,
+   FORM = 'UNFORMATTED',
+   READONLY,
+   STATUS = 'OLD')

C*****
C  THIS PART GETS THE PROJECTION OF EACH B-FIELD VECTOR ONTO
THE CURRENT
C  SHEET.

DO WHILE(LOGIC1)
  READ(LUN,END = 300)SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
  IF ((YRDY .GT. SYRDY) .OR. ((YRDY .GE. SYRDY) .AND.
+   (SEC .GE. SSEC))) THEN
    DO INDEX = 1,CNTRI
      BX = BMAG*COSD(BPOL)*COSD(BAZM)
      BY = BMAG*COSD(BPOL)*SIND(BAZM)
      BZ = BMAG*SIND(BPOL)
      IF (INDEX .EQ. 1) THEN
        WRITE (6,*) BX, BY, BZ
      END IF
      PROJ = (BX*VECX + BY*VECY + BZ*VE CZ)
    END DO
  END DO

```

```

PROJX = PROJ*VECX
PROJY = PROJ*VECY
PROJZ = PROJ*VE CZ
ONE(INDEX) = BX - PROJX
TWO(INDEX) = BY - PROJY
THREE(INDEX) = BZ - PROJZ
ESEC = SEC
EYRDY = YRDY
C  WRITE(6,*) YRDY,SEC,BX,BY,BZ
  READ(LUN,END = 300)SCID,YRDY,SEC,MSEC,BT,BMAG,BAZM,BPOL
  END DO

C*****CALLING SUBROUTINE EULER TO ROTATE THE COORDINATE
SYSTEM TO
C*****THE X AND Y LIE IN THE CURRENT SHEET.

CALL EULER(ONE,TWO,THREE,SIZE,CNTRI,RAZM,RPOL,RAZM2)
CALL PLANE PLOT(ONE,TWO,SIZE,CNTRI)

LOGIC1 = .FALSE.
END IF
END DO

300  CLOSE(UNIT = LUN)
RETURN
END

```

```

C*****
*
C   FILE NAME:  PLANEPL0T.F0R
C
C   NAME:      PLANEPL0T
C
C   PURPOSE:   THIS PROGRAM WILL TAKE THE PROJECTIONS OF THE
MAGNETIC
C              VECTORS ONTO THE CURRENT SHEET (FOUND IN PLANE.F0R)
C              AND PLOT THE RESULTS.
C
C   AUTHOR:    FRED QUIST   DATE: 2/24/94
C
C   MODIFIED:  NEVER
C*****

SUBROUTINE PLANEPL0T(XCOMP,YCOMP,SIZE,CNTR1)
C*****
C   DEFINE THE VARIABLES
      REAL*4  XCOMP(SIZE),YCOMP(SIZE)
      INTEGER  CNTR1
C*****
C   CALL THE DISPLA FUNCTIONS
      CALL DIP(5,'SHEET.DIP',100)
      CALL PAGE(8.5,11.0)
      CALL NOBRDR
      CALL PHYSOR(1.0,0.5)
      CALL AREA2D(6.5,6.5)
      CALL XNAME('LAMBDA MAX COMP OF SHEETS',100)
      CALL YTICKS(4)
      CALL XTICKS(4)
      CALL GRAF(-40.0,10.0,40.0,-40.0,10.0,40.0)
      CALL YGRXS(-40.0,10.0,40.0,40.0,6.5,'LAMBDA MID COMP OF SHEETS'
+      ,100.0,0.0,0)
      CALL GRID(2.2)
      CALL MARKER(15)
      CALL CURVE(NCOMP,YCOMP,CNTR1,0)
      CALL ENDGR(0)
      CALL ENDPL(0)
      CALL DONEPL(0)
      RETURN
      END

```



```

C*****
C   FILE NAME: EULER.FOR
C
C   NAME:      EULER
C
C   PURPOSE:  THIS SUBROUTINE TAKES THE ORIGINAL X,Y AND Z
COORD
C   SYSTEM AND ROTATES VIA THE EULER ANGLES INTO THE
C   NEW SYSTEM OF X',Y' AND Z'. THE USER MUST SUPPLY THE
C   THREE DESIRED ROTATIONS AND THE VECTORS TO BE
ROTATED.
C   ALSO THE ARRAY SIZE AND THE NUMBER OF ACTUAL POINTS
IN
C   THE ARRAYS.
C
C   AUTHOR:   FRED QUIST      DATE: 3/11/94
C
C   MODIFIED: 9/1/94
C*****
SUBROUTINE
EULER(XCOMP,YCOMP,ZCOMP,SIZE,CNTRI,ONE,TWO,THREE)
C*****
C   DEFINE THE VARIABLES TO BE USED
C   ONE, TWO, AND THREE ARE THE 3 ROTATIONS AS DEFINED IN
GOLDSTEIN (PG 109)
C
INTEGER CNTRI
REAL*4 XCOMP(SIZE), YCOMP(SIZE), ZCOMP(SIZE)
REAL*4 ONE, TWO, THREE
REAL*4 E11, E12, E13, E21, E22, E23, E31, E32, E33
REAL*4 XTEMP, YTEMP, ZTEMP
C*****
C   DEFINE THE ELEMENTS OF THE EULER MATRIX (3 X 3)...SEE GOLDSTEIN
C*****
E11 = COSD(THREE)*COSD(ONE) - COSD(TWO)*SIND(ONE)*SIND(THREE)
E12 = SIND(ONE)*COSD(THREE) + COSD(TWO)*COSD(ONE)*SIND(THREE)
E13 = SIND(THREE)*SIND(TWO)
E21 = -COSD(ONE)*SIND(THREE) - COSD(TWO)*SIND(ONE)*COSD(THREE)
E22 = -SIND(THREE)*SIND(ONE) + COSD(TWO)*COSD(ONE)*COSD(THREE)
E23 = SIND(TWO)*COSD(THREE)
E31 = SIND(TWO)*SIND(ONE)
E32 = -SIND(TWO)*COSD(ONE)
C*****
E33 = COSD(TWO)
WRITE(6,*) E11,E12,E13
WRITE(6,*) E21,E22,E23
WRITE(6,*) E31,E32,E33
C*****
C   THIS LOOP WILL DO THE MATRIX MULTIPLICATION TO FIND THE NEW
COORD SYSTEM
C*****
DO INDEX = 1,CNTRI
XTEMP = E11*XCOMP(INDEX) + E12*YCOMP(INDEX) + E13*ZCOMP(INDEX)
YTEMP = E21*XCOMP(INDEX) + E22*YCOMP(INDEX) + E23*ZCOMP(INDEX)
ZTEMP = E31*XCOMP(INDEX) + E32*YCOMP(INDEX) + E33*ZCOMP(INDEX)
XCOMP(INDEX) = XTEMP
YCOMP(INDEX) = YTEMP
ZCOMP(INDEX) = ZTEMP
C   WRITE(6,*)THE NEW CORRDS ARE 'XTEMP,YTEMP,ZTEMP
END DO
C
write(6,*) 'the number of points is: ',cntri1
WRITE(6,*) 'START POINT...',XCOMP(1),YCOMP(1),ZCOMP(1)
WRITE(6,*) 'MID POINT...',XCOMP(13),YCOMP(13),ZCOMP(13)
WRITE(6,*) 'END POINT...',XCOMP(CNTRI),YCOMP(CNTRI),ZCOMP(CNTRI)
RETURN
END

```

APPENDIX D

Appendix D shows the minimization of equation 3.9. Recall our method for determining the normal to the current sheet used this standard deviation because it best accounted for any time varying effects. The standard deviation was

$$\sigma^2 = \frac{1}{N} \sum_1^N [\mathbf{B}^i \cdot \mathbf{n} - \bar{\mathbf{B}} \cdot \mathbf{n}]^2, \quad (\text{D1})$$

where N was the number of terms in the summation, $\mathbf{B}^i$ was an individual magnetic field vector, $\bar{\mathbf{B}}$ was the average of the N magnetic field vectors, and $\mathbf{n}$ was the normal vector to be determined. In vector notation, these vectors are

$$\mathbf{n} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}}, \quad (\text{D2})$$

$$\mathbf{B}^i = B_x^i \hat{\mathbf{x}} + B_y^i \hat{\mathbf{y}} + B_z^i \hat{\mathbf{z}}, \text{ and} \quad (\text{D3})$$

$$\bar{\mathbf{B}} = \bar{B}_x \hat{\mathbf{x}} + \bar{B}_y \hat{\mathbf{y}} + \bar{B}_z \hat{\mathbf{z}}. \quad (\text{D4})$$

Taking the dot products of equation D1, we get

$$\mathbf{B}^i \cdot \mathbf{n} = xB_x^i + yB_y^i + zB_z^i \text{ and} \quad (\text{D5})$$

$$\bar{\mathbf{B}} \cdot \mathbf{n} = x\bar{B}_x + y\bar{B}_y + z\bar{B}_z. \quad (\text{D6})$$

Subtracting and squaring these two terms gives

$$\left(\mathbf{B}^i \cdot \mathbf{n} - \bar{\mathbf{B}} \cdot \mathbf{n}\right)^2 = \left(xB_x^i + yB_y^i + zB_z^i - x\bar{B}_x - y\bar{B}_y - z\bar{B}_z\right)^2. \quad (\text{D7})$$

When we expand out equation D7, we get 36 terms,

$$\begin{aligned} \left(\mathbf{B}^i \cdot \mathbf{n} - \bar{\mathbf{B}} \cdot \mathbf{n}\right)^2 = & x^2 B_x^{i^2} + 2xy B_x^i B_y^i + 2xz B_x^i B_z^i - 2x^2 B_x^i \bar{B}_x - 2xy B_x^i \bar{B}_y \\ & - 2xz B_x^i \bar{B}_z + y^2 B_y^{i^2} + 2yz B_y^i B_z^i - 2xy B_y^i \bar{B}_x - 2y^2 B_y^i \bar{B}_y \\ & - 2yz B_y^i \bar{B}_z + z^2 B_z^{i^2} - 2xz \bar{B}_x B_z^i - 2yz \bar{B}_y B_z^i - 2z^2 \bar{B}_z B_z^i \\ & + x^2 \bar{B}_x^2 + 2xy \bar{B}_x \bar{B}_y + 2xz \bar{B}_x \bar{B}_z + y^2 \bar{B}_y^2 + 2yz \bar{B}_y \bar{B}_z + z^2 \bar{B}_z^2. \end{aligned} \quad (\text{D8})$$

We bring in the summation sign by noting that

$$\frac{1}{N} \sum_1^N B_y^i \bar{B}_x = \frac{1}{N} \sum_1^N B_x^i \bar{B}_y = \bar{B}_x \bar{B}_y \quad \text{and} \quad (\text{D9})$$

$$\frac{1}{N} \sum_1^N B_x^i B_y^i = \overline{B_x B_y}, \quad (\text{D10})$$

with similar definitions for the other combinations of x, y, and z. Using these definitions, we get the complete, expanded expression for the standard deviation,

$$\begin{aligned} \frac{1}{N} \sum_1^N (\mathbf{B}^i \cdot \mathbf{n} - \bar{\mathbf{B}} \cdot \mathbf{n})^2 = & x^2 \overline{\mathbf{B}_x \mathbf{B}_x} + 2xy \overline{\mathbf{B}_x \mathbf{B}_y} + 2xz \overline{\mathbf{B}_x \mathbf{B}_z} - 2x^2 \bar{\mathbf{B}}_x \bar{\mathbf{B}}_x - 2xy \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y \\ & - 2xz \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z + y^2 \overline{\mathbf{B}_y \mathbf{B}_y} + 2yz \overline{\mathbf{B}_y \mathbf{B}_z} - 2xy \bar{\mathbf{B}}_y \bar{\mathbf{B}}_x - 2y^2 \bar{\mathbf{B}}_y \bar{\mathbf{B}}_y \quad (\text{D11}) \\ & - 2yz \bar{\mathbf{B}}_y \bar{\mathbf{B}}_z + z^2 \overline{\mathbf{B}_z \mathbf{B}_z} - 2xz \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z - 2yz \bar{\mathbf{B}}_y \bar{\mathbf{B}}_z - 2z^2 \bar{\mathbf{B}}_z \bar{\mathbf{B}}_z \\ & + x^2 \bar{\mathbf{B}}_x \bar{\mathbf{B}}_x + 2xy \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y + 2xz \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z + y^2 \bar{\mathbf{B}}_y \bar{\mathbf{B}}_y + 2yz \bar{\mathbf{B}}_y \bar{\mathbf{B}}_z + z^2 \bar{\mathbf{B}}_z \bar{\mathbf{B}}_z. \end{aligned}$$

We want to minimize σ^2 with respect to each of the three variables x, y, and z.

The minimization with respect to the x variable gives

$$\begin{aligned} \frac{\partial}{\partial x} \sigma^2 = 2 \frac{\partial}{\partial x} \sigma = 0 = & 2x \overline{\mathbf{B}_x \mathbf{B}_x} + 2y \overline{\mathbf{B}_x \mathbf{B}_y} + 2z \overline{\mathbf{B}_x \mathbf{B}_z} - 4x \bar{\mathbf{B}}_x \bar{\mathbf{B}}_x \\ & - 2y \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y - 2z \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z - 2y \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y - 2z \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z \quad (\text{D12}) \\ & + 2x \bar{\mathbf{B}}_x \bar{\mathbf{B}}_x + 2y \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y + 2z \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z. \end{aligned}$$

Canceling the factor of two and collecting the common factors with respect to each variable x, y, and z, we get

$$\frac{\partial}{\partial x} \sigma = 0 = (\overline{\mathbf{B}_x \mathbf{B}_x} - \bar{\mathbf{B}}_x \bar{\mathbf{B}}_x)x + (\overline{\mathbf{B}_x \mathbf{B}_y} - \bar{\mathbf{B}}_x \bar{\mathbf{B}}_y)y + (\overline{\mathbf{B}_x \mathbf{B}_z} - \bar{\mathbf{B}}_x \bar{\mathbf{B}}_z)z. \quad (\text{D13})$$

Using the same method of differentiation for y and z, we get the other two equations,

$$\frac{\partial}{\partial y} \sigma = 0 = (\overline{B_x B_y} - \overline{B_x} \overline{B_y})x + (\overline{B_y B_y} - \overline{B_y} \overline{B_y})y + (\overline{B_y B_z} - \overline{B_y} \overline{B_z})z \text{ and} \quad (D14)$$

$$\frac{\partial}{\partial z} \sigma = 0 = (\overline{B_x B_z} - \overline{B_x} \overline{B_z})x + (\overline{B_y B_z} - \overline{B_y} \overline{B_z})y + (\overline{B_z B_z} - \overline{B_z} \overline{B_z})z. \quad (D15)$$

Equations D13, D14, and D15 are a set of three equations and three unknowns. If we write these three equations in matrix form, we get

$$\begin{bmatrix} (\overline{B_x B_x} - \overline{B_x} \overline{B_x}) & (\overline{B_x B_y} - \overline{B_x} \overline{B_y}) & (\overline{B_x B_z} - \overline{B_x} \overline{B_z}) \\ (\overline{B_x B_y} - \overline{B_x} \overline{B_y}) & (\overline{B_y B_y} - \overline{B_y} \overline{B_y}) & (\overline{B_y B_z} - \overline{B_y} \overline{B_z}) \\ (\overline{B_x B_z} - \overline{B_x} \overline{B_z}) & (\overline{B_y B_z} - \overline{B_y} \overline{B_z}) & (\overline{B_z B_z} - \overline{B_z} \overline{B_z}) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0, \quad (D16)$$

which are the matrix elements as given in Sec. III.D.

APPENDIX E

Appendix E contains output from the program PARK.FOR. The output is between the dates 84065-84247 and 86256-87067.

84065-84247 output:

The magnetic field was "+" on 84065.
The magnetic field was "+" on 84066.
The magnetic field was "+" on 84067.
The magnetic field was "+" on 84068.
The magnetic field was "-" on 84069.
The magnetic field was "-" on 84070.
The magnetic field was "-" on 84071.
The magnetic field was "-" on 84072.
The magnetic field was "-" on 84073.
The magnetic field was "-" on 84074.
The magnetic field was "-" on 84075.
The magnetic field was "-" on 84076.
The magnetic field was "-" on 84077.
The magnetic field was "-" on 84078.
The magnetic field was "-" on 84079.
The magnetic field was "-" on 84080.
The magnetic field was "-" on 84081.
The magnetic field was "-" on 84082.
The magnetic field was "+" on 84083.
The magnetic field was "+" on 84084.
The magnetic field was "+" on 84085.
The magnetic field was "+" on 84087.
The magnetic field was "-" on 84088.
The magnetic field was "+" on 84089.
The magnetic field was "+" on 84090.
The magnetic field was "+" on 84091.
The magnetic field was "+" on 84092.

The magnetic field was "+" on 84093.
The magnetic field was "+" on 84094.
The magnetic field was "+" on 84095.
The magnetic field was "-" on 84096.
The magnetic field was "-" on 84097.
The magnetic field was "-" on 84098.
The magnetic field was "-" on 84099.
The magnetic field was "-" on 84100.
The magnetic field was "-" on 84101.
The magnetic field was "-" on 84102.
The magnetic field was "-" on 84103.
The magnetic field was "-" on 84104.
The magnetic field was "-" on 84105.
The magnetic field was "-" on 84106.
The magnetic field was "-" on 84107.
The magnetic field was "-" on 84108.
The magnetic field was "-" on 84109.
The magnetic field was "-" on 84110.
The magnetic field was "-" on 84111.
The magnetic field was "+" on 84112.
The magnetic field was "+" on 84113.
The magnetic field was "-" on 84181.
The magnetic field was "-" on 84182.
The magnetic field was "-" on 84183.
The magnetic field was "-" on 84184.
The magnetic field was "-" on 84185.
The magnetic field was "-" on 84186.
The magnetic field was "-" on 84187.
The magnetic field was "-" on 84188.
The magnetic field was "-" on 84189.
The magnetic field was "-" on 84190.
The magnetic field was "-" on 84191.
The magnetic field was "-" on 84192.
The magnetic field was "-" on 84193.
The magnetic field was "+" on 84194.
The magnetic field was "+" on 84195.
The magnetic field was "+" on 84196.
The magnetic field was "+" on 84197.
The magnetic field was "+" on 84198.
The magnetic field was "+" on 84199.
The magnetic field was "+" on 84200.
The magnetic field was "+" on 84201.
The magnetic field was "+" on 84202.

The magnetic field was "+" on 84203.
The magnetic field was "+" on 84204.
The magnetic field was "+" on 84205.
The magnetic field was "+" on 84206.
The magnetic field was "-" on 84207.
The magnetic field was "-" on 84208.
The magnetic field was "-" on 84209.
The magnetic field was "-" on 84210.
The magnetic field was "-" on 84211.
The magnetic field was "-" on 84212.
The magnetic field was "-" on 84213.
The magnetic field was "-" on 84214.
The magnetic field was "-" on 84215.
The magnetic field was "-" on 84216.
The magnetic field was "-" on 84217.
The magnetic field was "-" on 84218.
The magnetic field was "-" on 84219.
The magnetic field was "-" on 84220.
The magnetic field was "-" on 84221.
The magnetic field was "-" on 84222.
The magnetic field was "+" on 84223.
The magnetic field was "+" on 84224.
The magnetic field was "+" on 84225.
The magnetic field was "+" on 84226.
The magnetic field was "+" on 84227.
The magnetic field was "+" on 84228.
The magnetic field was "+" on 84229.
The magnetic field was "+" on 84231.
The magnetic field was "+" on 84232.
The magnetic field was "+" on 84233.
The magnetic field was "+" on 84234.
The magnetic field was "+" on 84235.
The magnetic field was "+" on 84236.
The magnetic field was "-" on 84237.
The magnetic field was "-" on 84238.
The magnetic field was "-" on 84239.
The magnetic field was "-" on 84240.
The magnetic field was "-" on 84241.
The magnetic field was "-" on 84242.
The magnetic field was "-" on 84243.
The magnetic field was "-" on 84244.
The magnetic field was "-" on 84245.
The magnetic field was "-" on 84246.

The magnetic field was “-” on 84247.

86245-87067 OUTPUT:

The magnetic field was “+” on 86245.
The magnetic field was “+” on 86246.
The magnetic field was “+” on 86247.
The magnetic field was “+” on 86248.
The magnetic field was “+” on 86249.
The magnetic field was “+” on 86251.
The magnetic field was “+” on 86252.
The magnetic field was “+” on 86253.
The magnetic field was “+” on 86254.
The magnetic field was “+” on 86255.
The magnetic field was “+” on 86256.
The magnetic field was “-” on 86257.
The magnetic field was “-” on 86258.
The magnetic field was “-” on 86259.
The magnetic field was “-” on 86260.
The magnetic field was “-” on 86261.
The magnetic field was “-” on 86262.
The magnetic field was “-” on 86263.
The magnetic field was “-” on 86264.
The magnetic field was “-” on 86265.
The magnetic field was “-” on 86266.
The magnetic field was “-” on 86267.
The magnetic field was “-” on 86268.
The magnetic field was “-” on 86269.
The magnetic field was “-” on 86270.
The magnetic field was “-” on 86271.
The magnetic field was “+” on 86272.
The magnetic field was “+” on 86273.
The magnetic field was “-” on 86274.
The magnetic field was “-” on 86275.
The magnetic field was “-” on 86276.
The magnetic field was “-” on 86278.
The magnetic field was “-” on 86325.
The magnetic field was “-” on 86326.
The magnetic field was “-” on 86327.
The magnetic field was “+” on 86328.

The magnetic field was "+" on 86329.
The magnetic field was "+" on 86330.
The magnetic field was "+" on 86331.
The magnetic field was "+" on 86332.
The magnetic field was "+" on 86333.
The magnetic field was "+" on 86334.
The magnetic field was "+" on 86335.
The magnetic field was "+" on 86336.
The magnetic field was "+" on 86337.
The magnetic field was "+" on 86338.
The magnetic field was "-" on 86339.
The magnetic field was "-" on 86340.
The magnetic field was "-" on 86341.
The magnetic field was "-" on 86342.
The magnetic field was "-" on 86343.
The magnetic field was "-" on 86344.
The magnetic field was "+" on 86345.
The magnetic field was "-" on 86346.
The magnetic field was "-" on 86347.
The magnetic field was "-" on 86348.
The magnetic field was "+" on 86349.
The magnetic field was "+" on 86350.
The magnetic field was "+" on 86351.
The magnetic field was "+" on 86352.
The magnetic field was "-" on 86353.
The magnetic field was "-" on 86354.
The magnetic field was "-" on 86355.
The magnetic field was "+" on 86356.
The magnetic field was "+" on 86357.
The magnetic field was "+" on 86358.
The magnetic field was "+" on 86359.
The magnetic field was "+" on 86360.
The magnetic field was "+" on 86361.
The magnetic field was "+" on 86362.
The magnetic field was "+" on 86363.
The magnetic field was "+" on 86364.
The magnetic field was "+" on 86365.
The magnetic field was "+" on 87001.
The magnetic field was "+" on 87002.
The magnetic field was "+" on 87003.
The magnetic field was "+" on 87004.
The magnetic field was "-" on 87005.
The magnetic field was "-" on 87006.

The magnetic field was “-” on 87008.
The magnetic field was “-” on 87009.
The magnetic field was “+” on 87011.
The magnetic field was “+” on 87012.
The magnetic field was “+” on 87013.
The magnetic field was “+” on 87014.
The magnetic field was “+” on 87015.
The magnetic field was “-” on 87016.
The magnetic field was “-” on 87019.
The magnetic field was “-” on 87020.
The magnetic field was “+” on 87022.
The magnetic field was “+” on 87023.
The magnetic field was “+” on 87024.
The magnetic field was “+” on 87025.
The magnetic field was “+” on 87026.
The magnetic field was “+” on 87027.
The magnetic field was “+” on 87028.
The magnetic field was “+” on 87029.
The magnetic field was “+” on 87030.
The magnetic field was “+” on 87031.
The magnetic field was “+” on 87032.
The magnetic field was “+” on 87033.
The magnetic field was “+” on 87034.
The magnetic field was “+” on 87035.
The magnetic field was “+” on 87036.
The magnetic field was “+” on 87037.
The magnetic field was “+” on 87038.
The magnetic field was “+” on 87039.
The magnetic field was “+” on 87041.
The magnetic field was “+” on 87042.
The magnetic field was “+” on 87043.
The magnetic field was “+” on 87044.
The magnetic field was “-” on 87045.
The magnetic field was “-” on 87046.
The magnetic field was “+” on 87047.
The magnetic field was “+” on 87048.
The magnetic field was “+” on 87049.
The magnetic field was “+” on 87050.
The magnetic field was “-” on 87051.
The magnetic field was “+” on 87052.
The magnetic field was “+” on 87053.
The magnetic field was “+” on 87054.
The magnetic field was “+” on 87055.

The magnetic field was "+" on 87056.
The magnetic field was "+" on 87057.
The magnetic field was "+" on 87058.
The magnetic field was "+" on 87059.
The magnetic field was "+" on 87060.
The magnetic field was "+" on 87061.
The magnetic field was "+" on 87062.
The magnetic field was "+" on 87063.
The magnetic field was "+" on 87064.
The magnetic field was "+" on 87065.
The magnetic field was "-" on 87066.
The magnetic field was "-" on 87067.