Magnetosphere Particle Interaction

New Mexico Supercomputing Challenge

Final Report

April 1, 2009

Team 68

Team Members

Brandon Ramirez

Dennis Trujillo

Francisco Vigil

Teachers

John Paul Lorenzen

Irina Cislaru

Project Mentors

Michelle Thomsen

Robert Robey

Table of Contents

Executive Summary

- Charged particles damage spacecraft
- FORTRAN model generates magnetosphere and particle position
- NetLogo model visualizes both

We decided to approach this project through the use of a computer generated model as well as a physical model in the hope that they will complement each other. In these models we used many different aspects of the disciplines of physics and mathematics to help construct these interpretations of the magnetic field produced by our planet.

The model produced by this simulation is described by a set of integral equations which combine aspects of the magnetic field and its strength as well as the movement and later repulsion by the magnetosphere. These equations are primarily related to the natural structure of the magnetosphere and can be seen as computing an average simulation which recreates the magnetic field and particle density at differentiating quantities. In an attempt to make our observations as accurate as possible we have coupled our work with a series of existing models created by N.A. Tsyganenko, which have proven accurate to known values for magnetosphere strength and average particle density in a given area.

We successfully wrote a model in FORTRAN which simulates particle movement and magnetosphere generation and have imported this data into Net logo for visualization. Our current models tend to agree with known models and data collected regarding the average particle density and movement of these materials around the magnetosphere. Also we have been able to view the movement tendencies of individual particles as they progress vertically and horizontally around the earth. Through our work we intend to add to our program and make it as precise as possible in order to apply our simulations to the space industry, which suffers as a result of spacecraft destruction due to interaction between these craft and masses of highly charged particles.

As a visual representation of magnetosphere generation we have also created and designed a physical model which uses electromagnetic properties to generate a magnetic field similar to the means in which the earth generates the magnetosphere. Through this model we have been able to collect data which further validates the information produced by our models.

Introduction

The interaction between charged particles and the magnetosphere has been a topic closely followed by physicists and those involved in the space industry for a number of years and have proven an area of importance when considering spacecraft shielding. Due to this interaction and the inherit damage which occurs as a result of particle-spacecraft interaction, we feel that the mapping of particles as they encircle the magnetosphere is extremely important and required in order to minimize the expense associated with spacecraft shielding. Through our models we have been able to follow the probable movement of these particles in order to map areas of relative less particle density.

When considering the interaction of charged particles the source of this radiation should be considered in order to determine areas of mass particle density as determined by the number of particles in a given area. In terms of our simulation the main source of proton emission is the sun which emits these particles as well as other atomic radiation in a fairly consistent area of the earth, although lateral rotation assures that different areas of the surface and magnetosphere are exposed to this radiation. As these particles then shift about the magnetosphere they move in patterns of horizontal and vertical movement. This movement applies to minimal particle movement and cannot be applied to mass particle density due to the fluid movement of these particles as they form groups which move vertically between the two poles and rotate about the earth.

As a result of this concentration about certain points of the magnetosphere these particles then spread outward from this point and move around the earth, whereas some are trapped in orbit about the earth, others permeate through the magnetosphere and the majority simply fall away from the field lines and are pushed past the earth creating a tail of protons and other atomic particles and radiation. In terms of our model this phenomenon is simulated through incorporating magnetosphere permeability, solar wind velocity, particle acceleration, and magnetosphere generation.

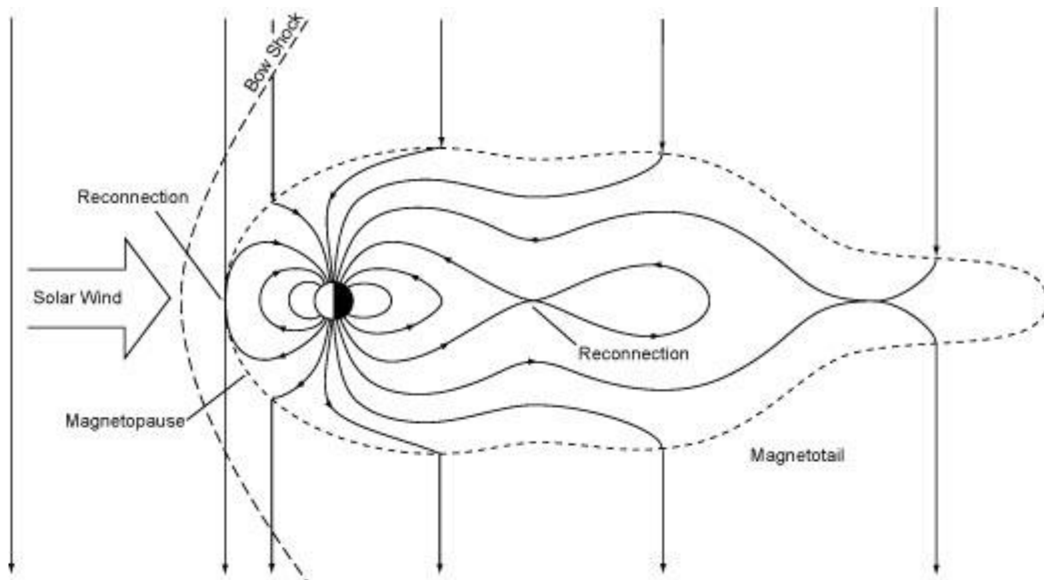


Figure 1.1 Magnetosphere Diagram (A Brief Introduction to Geomagnetism, 2008)

Because these particles have a high rate of acceleration and an inherit charge, when they encounter electronic devices including those located within spacecraft, damage occurs which renders the entire operation of launching this spacecraft useless and occurs a large waste of money due to the destruction of electronic components aboard spacecraft as they encircle the earth. In attempt to prevent this damage spacecraft are often provided with layers of dense shielding which block these particles from interfering with electronic components and destroying these probes or satellites, which are somewhat successful although with the added weight of this shielding in then becomes much more expensive to launch these devices. With our simulation we intend to map areas of relative less particle density in order to designate these areas as safer for spacecraft entry.

Mathematical Model

The model described below is based on the Newtonian equations of motion and the geodynamo equations relating to the geomagnetic generation as well as several equations which represent the interaction of them. The RECALC_08, IGRF_GSW_08, and T89C subroutines all calculate the magnetic field **B** based on the scalar potentials **A**.

$$\vec{\nabla} \times A_{(x,y,z)} = B_{(x,y,z)}$$

curl of **A** scalar potential field
B is the magnetic field vetor

The geomagnetic field vector, **B**, is best described by the orthogonal components X (northerly intensity), Y (easterly intensity) and Z (vertical intensity, positive downwards); total intensity F; horizontal intensity H; inclination (or dip) I (the angle between the horizontal plane and the field vector, measured positive downwards) and declination (or magnetic variation) D (the horizontal angle between true north and the field vector, measured positive eastwards). Declination, inclination and total intensity and almost every variable can be computed from the orthogonal components using the equations:

$$X = H\cos(D) \qquad X = F\cos(I)\cos(D) \qquad Z = F\sin(I)$$

$$Y = H\sin(D) \qquad Y = F\cos(I)\sin(D) \qquad F = \sqrt{H^2 + Z^2}$$

$$Z = F^2 - H^2 \qquad F = \sqrt{X^2 + Y^2 + Z^2} \qquad H = X^2 + Y^2$$

$$H = F\cos(I) \qquad I = \tan^{-1}\left({}^{Z}/_{H}\right) \qquad D = \tan^{-1}\left({}^{Y}/_{X}\right)$$

However both are utilized in the subroutines to calculate the appropriate values with respect to the time and the scalar potentials at their positions to get the magnetic field components at their positions along with the orthogonal calculations of such values.

$$M \frac{\partial^2 \vec{x}}{\partial t^2} = q \left(\frac{\partial \vec{x}}{\partial t} \times B \right)$$

M is Mass
Q is the charge of the particle
B is the magnetic field vector

When we have the magnetic field components then we use the particle motion equation to calculate the interaction in between the field and the charged particle. We use this equation to calculate the interaction of a certain proton to simplify the computation.

$$\frac{\partial^2 x}{\partial t^2} = (9.58e - 2) \cdot \left(\frac{\partial x}{\partial t} \times B \right)$$

The first derivative in the velocity of the particle that is known or set to comparative to real values.

$$\frac{\partial x}{\partial t} = [V_x, V_y, V_z]$$

The cross product at the positions x, y, and z is found implicitly by:

$$\left(\frac{\partial x}{\partial t} \times B \right)_x = (V_Y B_Z - V_Z B_Y)$$

$$\left(\frac{\partial x}{\partial t} \times B \right)_y = -(V_X B_Z - V_Z B_X)$$

$$\left(\frac{\partial x}{\partial t} \times B \right)_z = (V_X B_Y - V_Y B_X)$$

After we receive the values for this particle interaction we the use them to determine the change in the position of the particle with respect to time. Each time data is receive whether it is positions or components they are then stored in their particular array.

Background

In our project there are various different concepts that are necessary to comprehend the program and what the math model evaluates. The sun produces solar wind which is plasma particles that are charged. We focus on the positively charged particles. This bombardment of solar wind constantly being produced is what creates the shape of the magnetosphere. The magnetosphere is represented by the magnetic field components at locations at the shells at certain distances in radii.

Computational Model

Introduction

- Almost all of our data is calculated in and stored in arrays of one two or three dimensions.

In the FORTRAN portion of our computational model we are initializing the points of the magnetosphere and calculating the x, y, z coordinates of the particle interacting with the magnetosphere. Our FORTRAN model uses a Tsyganenko magnetospheric magnetic field model. We chose to use the T89c source code along with GEOPACK-2008. One of our mentors suggested we use the T89c source code because it best suits the purpose of our project. GEOPACK-2008 is a collection of subroutines that are a result of several upgrades of the original package written by N. A. Tsyganenko in 1978-1979. We use the T89c source code to call GEOPACK-2008 subroutines as well as other subroutines from a module. The data from the FORTRAN model is then modeled in NetLogo. Our NetLogo model visualizes the coordinate and component data from the FORTRAN model.

FORTRAN Model

The FORTRAN code calculates every value visualized by our Netlogo model. These values are calculated through our written code as well as the subroutines that we received as a source code. The set values that are initialized have been taken from realistic situations provided through research as well as credible references relaying information based on their contact with such information. The difficulties in the calculations are writing code to utilize equations based on variables and yet ones that can still be utilized by the computer.

The values we set at the beginning of our program are either options, in detail, for the programmer to choose based on his computing power and yet still be a good representation of what he is trying to calculate or technical initialization for many of the subroutines. Also we set some scenario values in order to display interaction between the particles and the magnetosphere in a varying range of actual situations. The variables that manage time are created for precision. So depending on number of shells, the distance between them, the total distance in radii, the total number of points at the positions found on the surface of our shells, and the number of cases or scenarios we create for the interaction, and the modified magnetosphere situations we allocate the size of the array at which the data will be stored.

The first subroutine called CHG_DAY takes the numbered month and day that, with the help of the year, calculates the universal or international day which is needed for the RECALC_08 subroutine.

The RECALC_08 subroutine:

- Prepares the value used to help calculate the convertion of the space physics coordinate systems.
- Prepares the coefficients for the internal (main) magnetic field calculation with the IGRF_GSW_08 subroutine.

These variables are calculated with a relation to time and are all put into the common blocks[1] /GEOPACK1/ and /GEOPACK2/. /GEOPACK1/ is dedicated to the storage of the coordinate system elements. The interval 1965-2010 is the intervals at which the magnetic field coefficients are known. After they are multiplied by the Schmidt normalization factors[2]. The components of a dipole field are calculated relating to GEO[3] coordinates because it is parallel to the geodipole

---

[1] In FORTRAN common blocks provide global data.
[2] These coefficients are determined from vector components of the field and they are used to represent the vector field and do not recreate the magnetic scalar potential.
[3] Geographic Coordinate system is defined so that its *X*-axis is in the Earth's equatorial plane but is fixed with the rotation of the Earth. Its *Z*-axis is parallel to the rotation axis of the Earth, and its *Y*-axis ($\boldsymbol{Y} = \boldsymbol{Z} \times \boldsymbol{X}$)

axis. Then the GEI[4] components are calculated with respect to the sun at the universal time which includes a call to the SUN_08 subroutine which calculates four quantities necessary for transformations between the coordinate systems. Then they are used to create the scalar products of the field in the GSW, which in our model is set to equal the GSM system. In this main system we use the *X*-axis is from the Earth to the Sun. The *Y*-axis is defined to be perpendicular to the Earth's magnetic dipole so that the *X*-*Z* plane contains the dipole axis. The positive *Z*-axis is chosen to be in the same sense as the northern magnetic pole.

Once all of the main field coefficients have been calculated the values for the positions of each point on each shell is calculate based on the distance in radii which is gradually changed within the do-loop to allow them to progress in distance from the earth. We do this with equations that allow the positions to be in the same spot in spherical coordinates and still receive those changes.

- A(:) = the parameters, depending on the chosen set-up of the solar wind, used to modify the magnetosphere

With the do-loop that has created the positions where the components will be calculated and has steadily spaced the shells at a set distance, we convert then from spherical coordinates to Cartesian coordinates. After, we calculate the vector components at the points for the field including the internal and external field that varies from a regular di-pole field to extreme and conservative solar-wind magnetosphere malformation.

The T89C subroutine calls the T89 subroutine to calculate the GSM components for the magnetosphere at set solarwind velocities to transform it. The parameters that are set based on the disturbance level are created from merged A, C, D, E, F, G, H, I, J (1966-1974), HEOS-1

[4] Geocentric Equatorial Inertial System (GEI) has its *X*-axis pointing from the Earth towards the first point of Aries (i.e. the position of the sun at the vernal equinox). This direction is the intersection of the Earth's equatorial plane and the ecliptic plane. The *Z*-axis is parallel to the rotation axis of the Earth and *Y* completes the right-handed orthogonal set (***Y*** = ***Z*** x ***X***).

AND -2 (1969-1974), and ISEE-1 and -2 spacecraft data sets. Then it computes the external magnetic field components based on:

" Model formulas for the magnetic field components contain in total
30 free parameters (17 linear and 13 nonlinear parameters).
First 2 independent linear parameters A(1)-A(2) correspond to contribu-
tion from the tail current system, then follow A(3) and A(4) which are the
amplitudes of symmetric and antisymmetric terms in the contribution from
the closure currents; A(5) is the ring current amplitude. Then follow the
coefficients A(6)-A(15) which define Chapman-Ferraro+Birkeland current field.
The coefficients c16-c19 (see Formula 20 in the original paper),
due to DivB=0 condition, are expressed through A(6)-A(15) and hence are not
independent ones.
A(16) AND A(17) CORRESPOND TO THE TERMS WHICH YIELD THE TILT ANGLE DEPEN-
DENCE OF THE TAIL CURRENT INTENSITY (ADDED ON APRIL 9, 1992)
Nonlinear parameters:
A(18) : DX - Characteristic scale of the Chapman-Ferraro field along the X-axis
A(19) : ADR (aRC) - Characteristic radius of the ring current
A(20) : D0 - Basic half-thickness of the tail current sheet
A(21) : DD (GamRC)- defines rate of thickening of the ring current, as
we go from night- to dayside
A(22) : Rc - an analog of "hinging distance" entering formula (11)
A(23) : G - amplitude of tail current warping in the Y-direction
A(24) : aT - Characteristic radius of the tail current
A(25) : Dy - characteristic scale distance in the Y direction entering
in W(x,y) in (13)
A(26) : Delta - defines the rate of thickening of the tail current sheet
in the Y-direction (in T89 it was fixed at 0.01)
A(27) : Q - this parameter was fixed at 0 in the final version of T89;
initially it was introduced for making Dy to depend on X
A(28) : Sx (Xo) - enters in W(x,y) ; see (13)
A(29) : Gam (GamT) - enters in DT in (13) and defines rate of tail sheet
thickening on going from night to dayside; in T89 fixed at 4.0
A(30) : Dyc - the Dy parameter for closure current system; in T89 fixed at 20.0 "

As soon as the components for the every point of every shell has been calculate for each solarwind scenario the particle interaction has to be calculated. There are a set number of cases at which the particles interact with the different scenarios. In each the particle is declared a proton beforehand. The nearest magnetic field point position value is found within the magnetosphere to calculate the interaction with. The particle position is then calculated relative to the time step and calculated interaction. Its positions, as they changed depending on the time, are stored in the array.

Limits

The computing done is simplified yet still accurate based on the components and realistic data. However there is a lot of variables that we could not take into consideration when dealing with computing power. The earth's magnetic field itself is produce in a way that has been proven to be unstable yet our calculations occur at a time when the coefficients are known and so reversal is not taken into account. The number of points and shells at which the components are calculated are restricted as well even though there is an ever extending and present magnetic field. The number of particles and all of the varying aspects to the particles that we were able to model and map was also limited.

- Models coordinate points for magnetosphere
- Models coordinate points for charged particle
- Visualizes and verifies FORTRAN code data

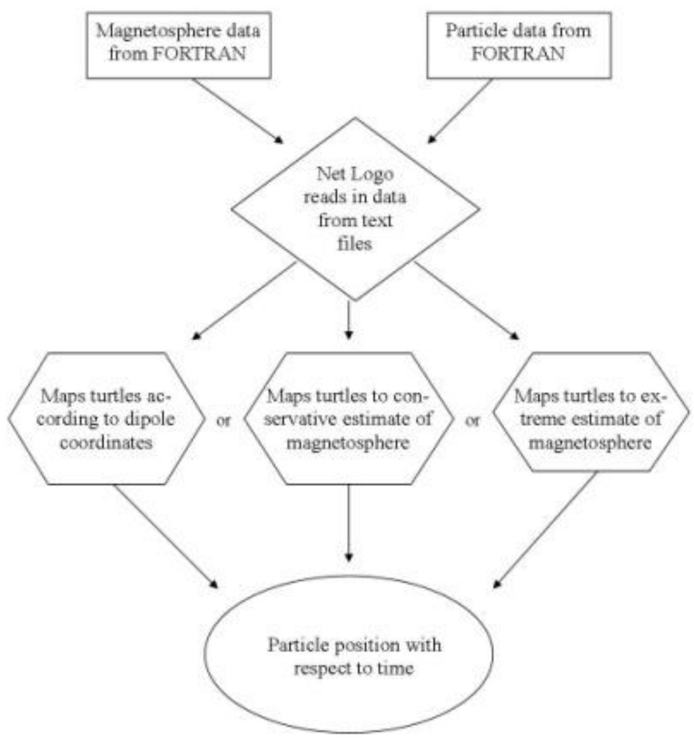
NetLogo Model

Our FORTRAN model computes coordinate sets along the magnetosphere as well as coordinates for the particle. We use a NetLogo model to both visualize and verify the data output from our FORTRAN code. The NetLogo model reads the coordinate values from the text files written by our FORTRAN code. NetLogo then models the turtles on the x, y, z coordinate plane according to their coordinate values. The result is an agent based model that shows the various shells of the magnetosphere, the dipole, and the positions of the particle.

The Netlogo model uses turtles to display the positions and vector components at those positions for the Magnetosphere as well as the positions of a particle as it moves as time progress.

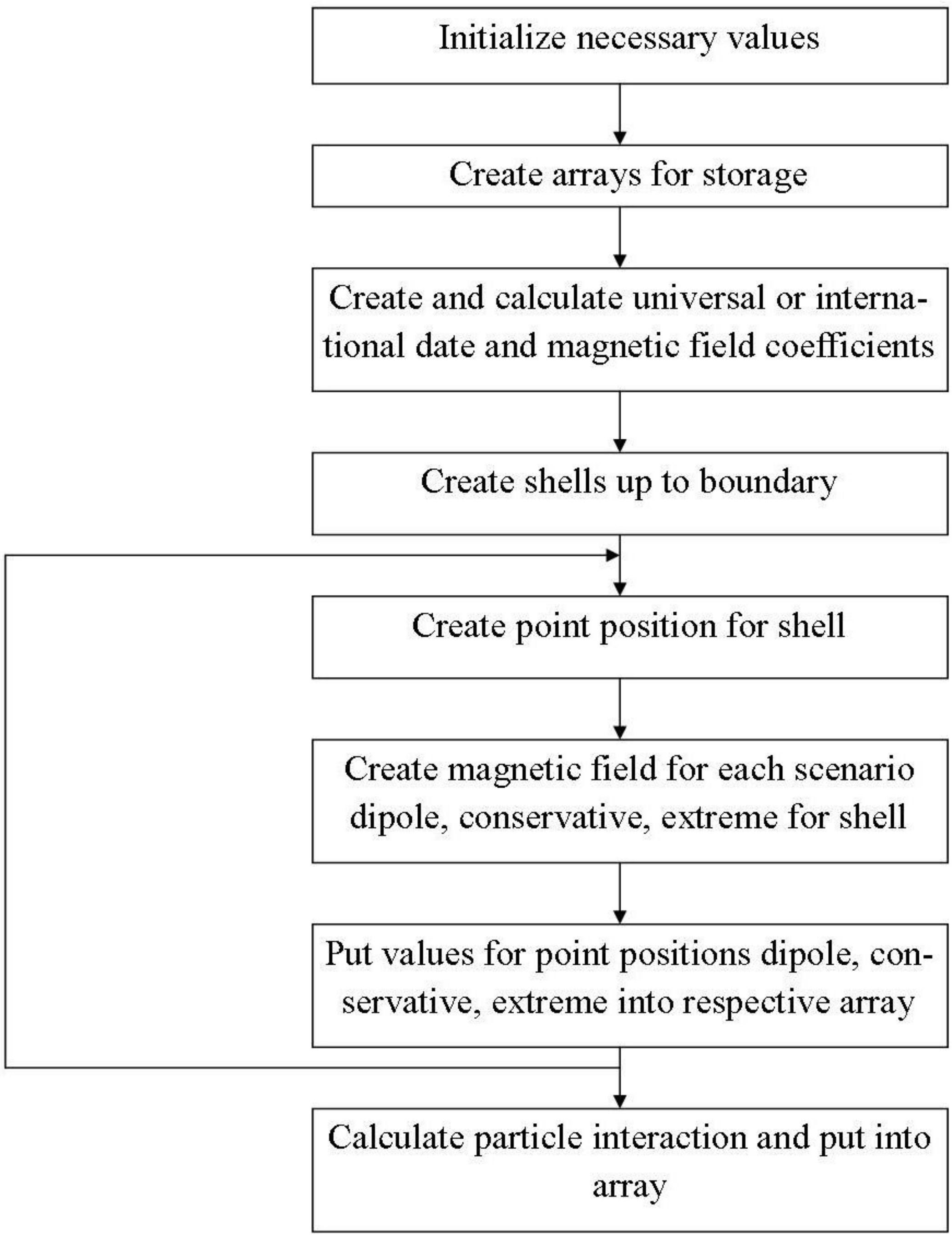
Architecture of Software

Our code was written on a computer we built and installed the Ubuntu operating system on. To program in FORTRAN we installed all the required gFortran software and installed medit to edit our code on. FORTRAN is best suited for computations of the physics variety. All of our computations are done in FORTRAN because that is what the source code's language is, as well as the fact that it is the only language we know. We chose to use NetLogo3D to visualize our project with the help of Stephen Guerin. NetLogo reads the coordinate positions from the array FORTRAN writes and maps out the turtles. The user will soon be able to control the number of magnetosphere shells shown through the use of a slider. In addition to the slider, the user will also be able introduce the particle into the magnetosphere with a button on the interface. NetLogo 3D allows the user to rotate the perspective so that the magnetosphere particle interaction from different angles. Architecture of NetLogo Model:



Architecture of FORTRAN

Model:



Physical Model

Purpose

- To gain a deeper understanding of particle interactions with magnetic fields.
- To verify the data from our computational model.

We felt that by creating a physical model we would be able to simulate a particle-magnetic field interaction similar to that which takes place between the Magnetosphere and charged particles from the sun. To create a magnetic field similar to that of the earth's we built a dynamo. Although the earth's core is a geo dynamo, we believed that by building a dynamo in which magnets move relative to the conductor a sufficient magnetic field would be created.

Electromagnetic Properties of the Physical Model

Our physical model is a dynamo and creates a magnetic field through electromagnetic induction. When the magnets move relative to the conductor an electromotive force is created. The created field is called an induced electromagnetic field because the magnets rotate and the conductor is stationary. The electromagnetically induced electric field is determined from the geometry of the conductor as well as the rotating magnets and the rate of change of the magnetic flux through the coil. The motional electromotive force can be found with Faraday's Law of Induction:

$$\varepsilon = -\frac{d\Phi_B}{dt}$$

- ε is the electromotive force in volts.
- Φ_B is the magnetic flux through the coil (in Webers).

Application of our Physical Model

Our physical model applies to our computational model because we plan to introduce charged particles into the electromagnetic field created by our dynamo. To introduce the charged particles we will use a plasma lamp. When the dynamo is running and the plasma lamp is within the electromagnetic field, the charged particles should be repelled. Although we are not able to track charged particles we are able to view physically how charged particles interact with a magnetic field.

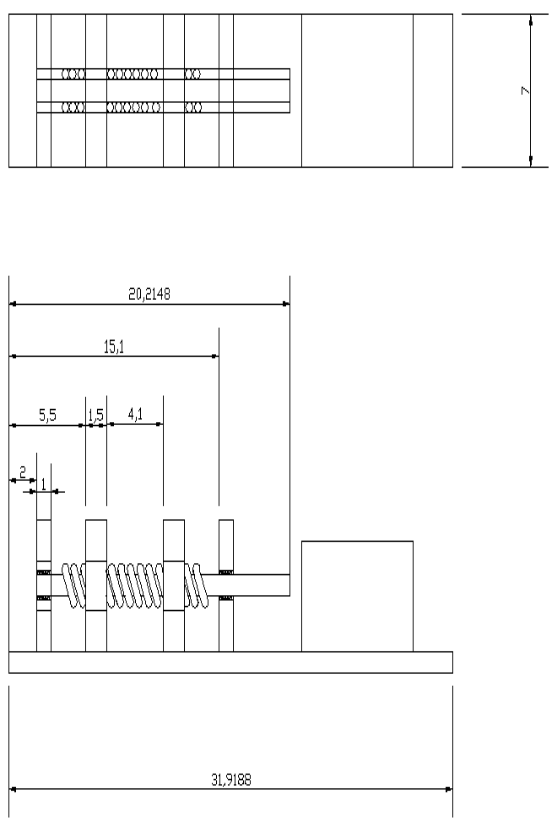
Problems Faced

Although we consulted several engineers when building our physical model, we still faced many problems. The first issue we dealt with was in designing a dynamo that would satisfy our needs. We consulted an electrical engineer who helped us design our dynamo. Our next issue was that we wrapped our coil so that it is parallel to the magnetic field and it produced no current. To fix this we need a coil wrapped so that it is wrapped perpendicular to the magnetic field to create a motional electromotive force. Another problem is that the north and south poles of the magnets are on the horizontal faces; as a result we did not align the magnets in the required north-south orientation for inductance the first time. The dynamo we created creates an alternating electromagnetic field and the earth's magnetosphere is constant. An electrical engineer that we consulted informed us that putting a diode in the coil will produce a constant electromagnetic field.

Future Plans

Our physical model is currently not working due to various reasons. To improve our physical model we plan to:

- wrap a new coil
- replace the magnets in a north-south orientation
- put a diode in the circuit
- test the charged particle interaction with a plasma lamp



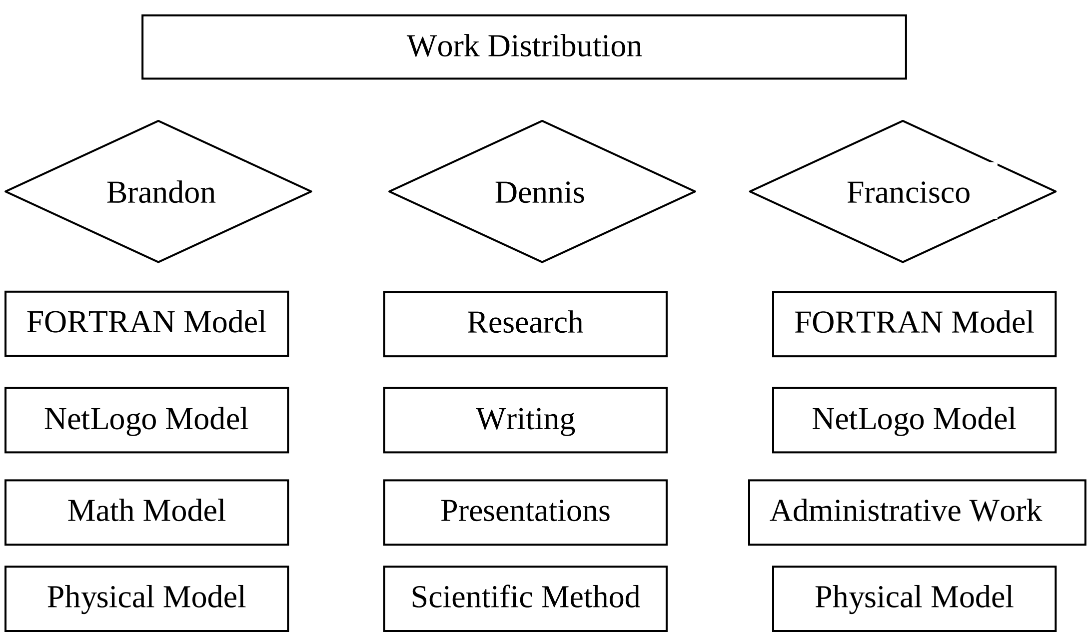
Physical Model Design (above).

Results and Discussion

Using the T89c source code along with GEOPACK-2008 we were able to successfully compute x, y, z coordinates for the dipole and sixty shells of the magnetosphere as well as write a program to track the charged particle motion. These shells represent the distortion of the magnetosphere with a conservative amount of solar wind as well as with an extreme amount of solar wind. Our NetLogo model reads in the coordinates from the text file and shows the dipole. After completing our models we have collected our results and evaluated our data. Based on our findings we have been able to compare our readings to real world data and have confirmed our results with real world numbers. We have also been able to make a number of other inferences regarding our simulations which are related to the movement of particles around the magnetosphere and their overall behavior.

Teamwork

Our team consists of three members, Brandon Ramirez, Francisco Vigil, and Dennis Trujillo. Each of us brings a different skill set or perspective as to how things should be approached. We had three main parts of our projects: the computational/math model, the written portion (reports and board), and the physical model/administrative work. Although each of us focused on one of these parts of the project, each of us still maintained an understanding of the other portions of the project and helped out. When our team would meet with mentors all of us would attend the meetings if possible so that each of us might have an understanding of the various parts of the project. Brandon has previous experience with StarLogo and FORTRAN so he does the majority of programming for our project and he is also the lead team member for the mathematical model. Francisco helps Brandon with parts of the NetLogo and FORTRAN models and takes care of the administrative work such as contacting mentors for help as well as constructing the physical model. Dennis does is in charge of most of the research such as looking up values for variables as well as the writing portion of the project.



Conclusion

The interaction between particles and the magnetosphere is a topic in which there is much potential in terms of the space industry and other endeavors which require traveling outside the protection of the magnetosphere. As seen through our simulations the movement of these particles around or past the magnetosphere is dependent on the particle mass, particle acceleration, and points of introduction, all factors which mandate the travel around the magnetosphere and there orbit around the earth. Another factor which we have observed as relative to our simulations includes the curl of the magnetic field and the equations of spherical shell generation, which have been integral to our project and simulations.

In terms of our project we have been able to write our program in FORTRAN based on a source code written by N.A. Tsyganenko which describes the interaction between the magnetosphere and charged particles. Based on data received from this model we have been able to confirm our results with real world data collected by satellite and ground based magnetosphere readings. In terms of our other models created for the purpose of visualizing our code and physically producing a model of the magnetosphere we have been able to import our data into Net logo and have produced a visualization which produces the magnetosphere and introduces a charged particle with varying mass and acceleration, effectively modeling the natural phenomenon of magnetosphere-particle interaction. After designing and building our physical model we have been able to run this model and collect data, which has currently been inconclusive.

Based on the work we have done so far we intend to further our project through introducing an internal magnetic field to the model we have created, and possibly also adding other variables which could be used to model the changing structure of the magnetosphere, as opposed to creating an average simulation of the magnetosphere.

Works Cited

SphericalShell in a magnetic field from The Wolfram Demonstrations Project http://demonstrations.wolfram.com/SphericalShellInAMagneticField/

Electromagnetic Spectrum. In *Wikipedia* [Web]. Retrieved August 10, 2008, from http://en.wikipedia.org/wiki/Electromagnetic_spectrum

Electromagnetic radiation . In *Wikipedia* [Web]. Retrieved September 21, 2008, from http://en.wikipedia.org/wiki/Electromagnetic_radiation

Electromagnetic wave equation. In Wikipedia [Web]. Retrieved November 12, 2008, from http://en.wikipedia.org/wiki/Gauss%27s_law_for_magnetism

Maxwell's Equations. In Wikipedia [Web]. Retrieved September 13, 2008, from http://en.wikipedia.org/wiki/Maxwell%27s_equations

A Brief Introduction to Geomagnetism. (2008, January 13). Retrieved February 10, 2009, from United States Geological Survey: http://geomag.usgs.gov/intro.php

Cosmic Rays. (2008, August 12). Retrieved August 12, 2008, from Wikipedia: http://wikipedia.org/wiki/Cosmic_ray

Earth's Magnetic Field. (2008, November 12). Retrieved November 13, 2008, from Wikipedia: http://en.wikipedia.org/wiki/Earth%magneticfield

Geomagnetic Storm. (2009, March 15). Retrieved March 15, 2009, from Wikipedia: http://en.wikipedia.org/wiki/Geomagnetic_storm

Geomagnetism. (2008, September 20). Retrieved November 10, 2008, from http://gsc.nrcan.gc.ca/geomag/field/field2_e.php

Geomagnetism. (2009, January 15). Retrieved February 16, 2009, from http://.geomag.bgs.ac.uk Glatzmaier, G. (2008, November 10).

When North Goes South. Retrieved March 1, 2008, from http://www.psc.edu/science/glatzmaier.html

Lathrop, D. (2009, February 1). *The Study of the Earth's Magnetic Field*. Retrieved March 2, 2009, from http://blackandwhiteprogram.com/interview/dr-dan-lathrop-the-study-of-the-earths-magnetic-field/2

List of satellites which have provided data on Earth's magnetosphere. (2009, January 19). Retrieved January 19, 2009, from Wikipedia: http://en.wikipedia.org/wiki/List_of_satellites_which_have_provided_data_on_Earth%27smagnetosphere

Macmillan, S. and J. M. Quinn, 200. The Derivation of World Magnetic Model 2000. *British Geological Survey Technical Report* WM/00/17r

Magnetic Reversal. (2008, September 12). Retrieved September 30, 2008, from National Geographic: http://news.nationalgeographic.com/news/2004/09/040927field_flip.html

Magnetosphere. (2008, November 13). Retrieved November 13, 2008, from Wikipedia: http://en.wikipedia.org/wiki/Magnetosphere

Magnetosphere Particle Motion. (2009, January 10). Retrieved January 10, 2009, from Wikipedia: http://en.wikipedia.org/wiki/Magnetosphere_particle_motion

NASA. (2009, February 12). Retrieved February 27, 2009, from Reversal of the Earth's Magnetic Field: http://absabs.harvard.edu/science/glatzmaier.html

National Geomagnetism Program. (2009, February 15). Retrieved February 15, 2009, from USGS.com: http://geomag.usgs.gov/operations.php

Nova: Magnetic Storm. (2009, January 18). Retrieved February 12, 2009, from Nova.com: http://www.pbs.org/wgbh/nova/magnetic/reversals.html

Problem with the Magnetic Pole Location . (2008, December 12). Retrieved December 20, 2008, from http://www.agu.org/sci_soc/campbell.html

Solar Flare. (2008, November 18). Retrieved November 18, 2008, from Wikipedia: http://en.wikipedia.org/wiki/Solar_flare

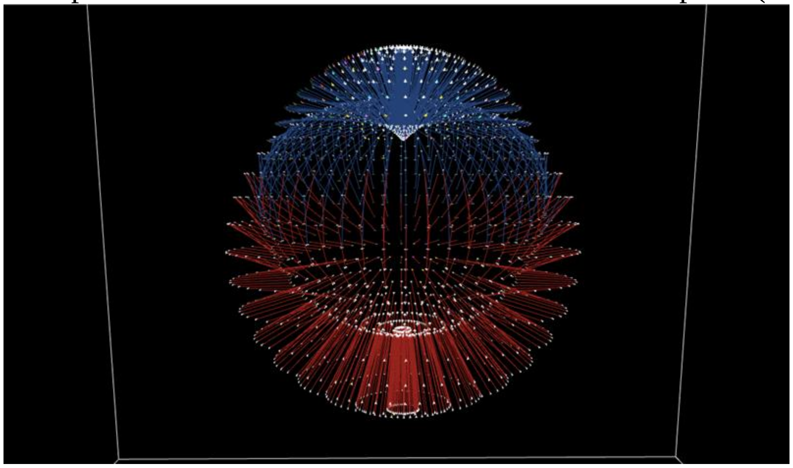
Tridiagonal Matrix Algorithm. (2009, February 16). Retrieved February 16, 2009, from Wikipedia: http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm

Acknowledgements

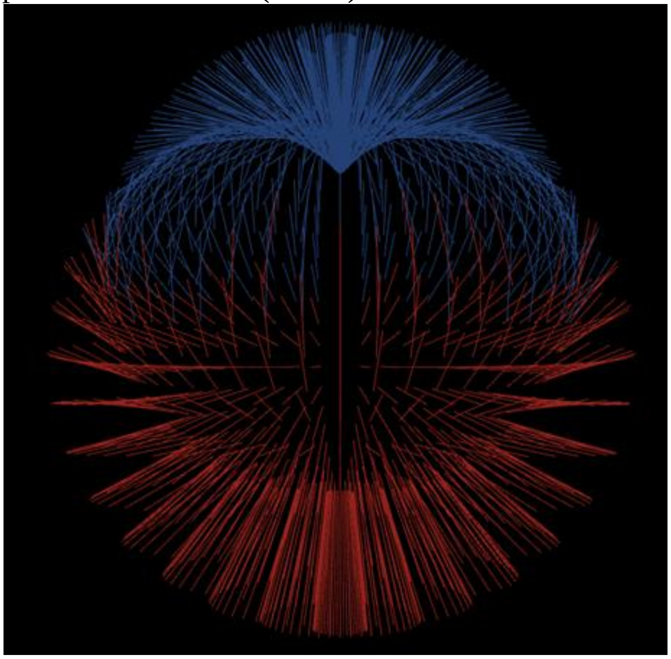
In consideration of support received from external sources and people we would like to accredit the success of this project to Michelle Thompson, who has mentored us, and provided great explanation in terms of the physical behavior of the magnetosphere. Philip Sanchez, Dennis Trujillo, and Duane Vigil have provided guidance in the building of our physical model. Robert Robey helped us fix our computer when it was not working and helped us with our FORTRAN problems. We would like to give a very special thank you to Stephen Guerin and everybody at the Santa Fe Complex who has helped us with our visualization. Many people have helped us along the way, although they may not be listed they have helped us immensely and we owe them great thanks.

Appendix A: Screenshots of our 3D Visualization

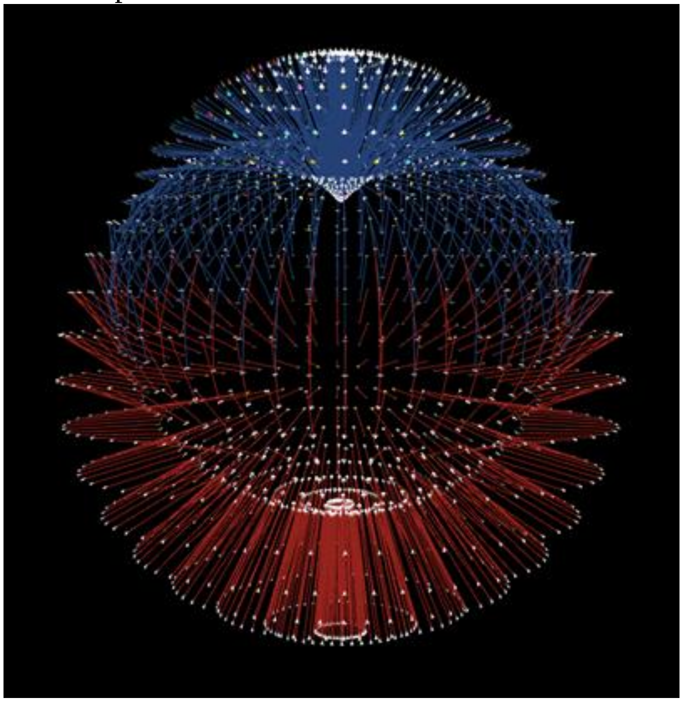
(A) 3D model of dipole with blue and red links to differentiate the poles (below).



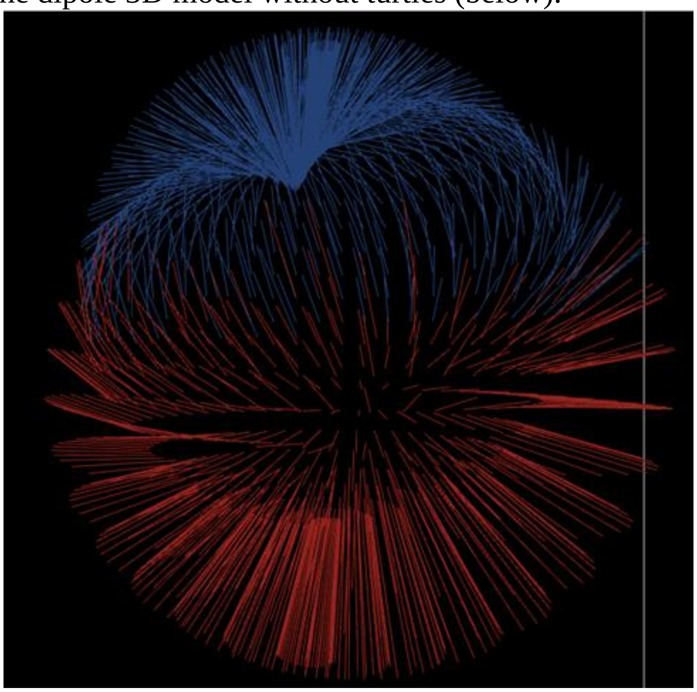
(B) 3D model of dipole without turtles (below).



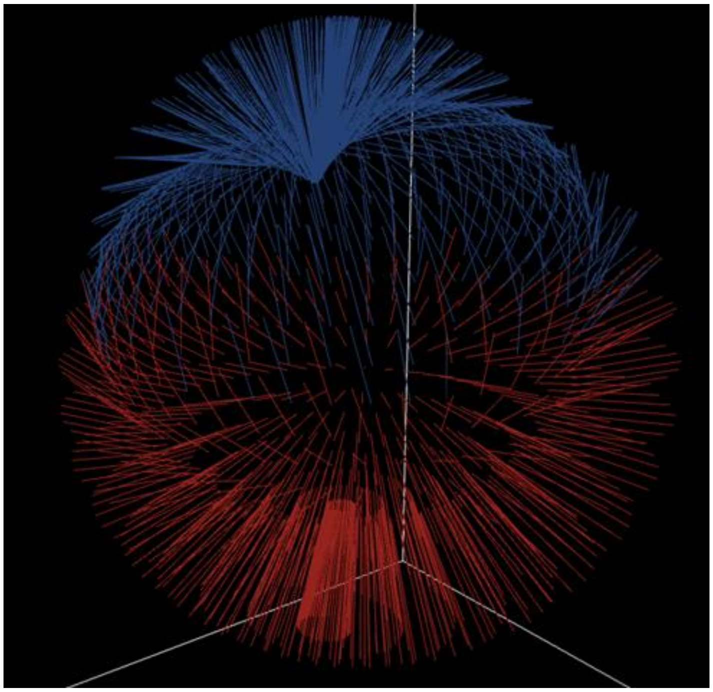
(C) Front view of the 3D dipole model with white turtles and red and blue links (below).



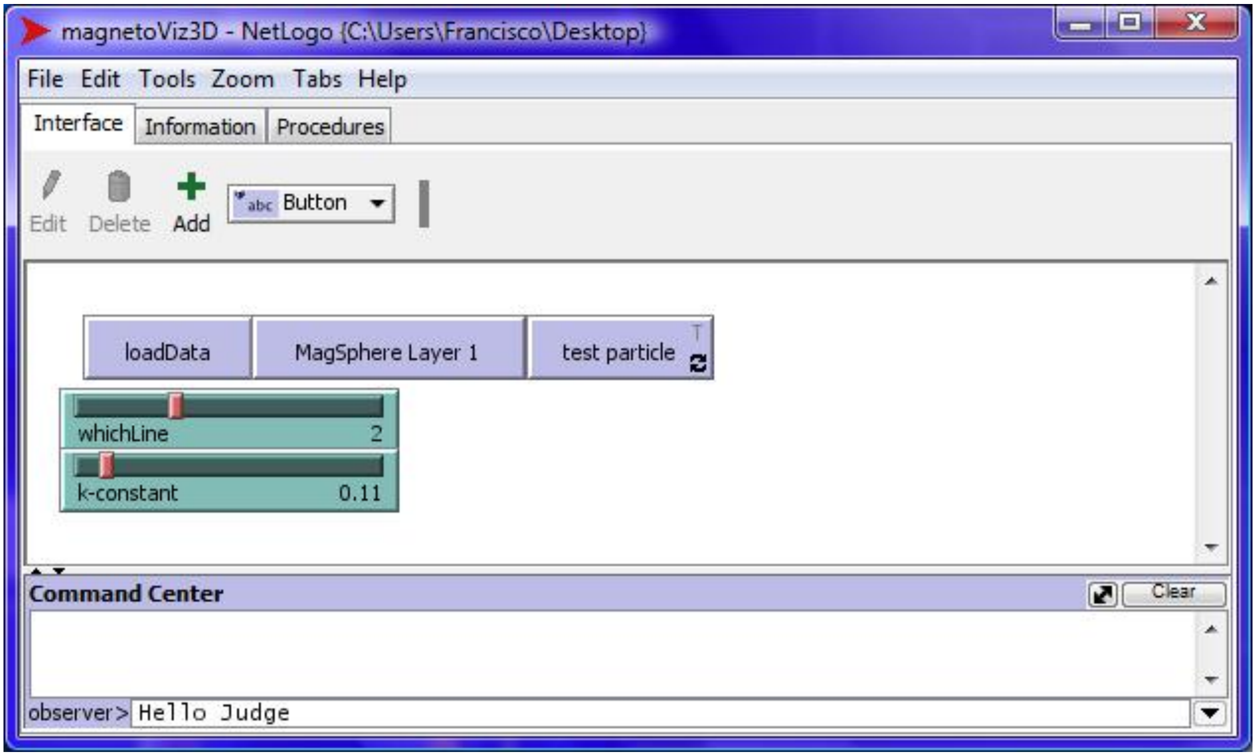
(D) Side view of the dipole 3D model without turtles (below).



(E) Alternate side view of the dipole 3D model without turtles (below).

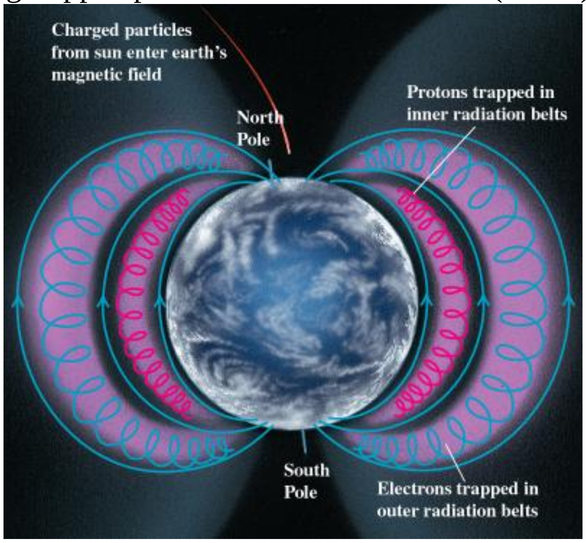


(F) NetLogo interface (below).

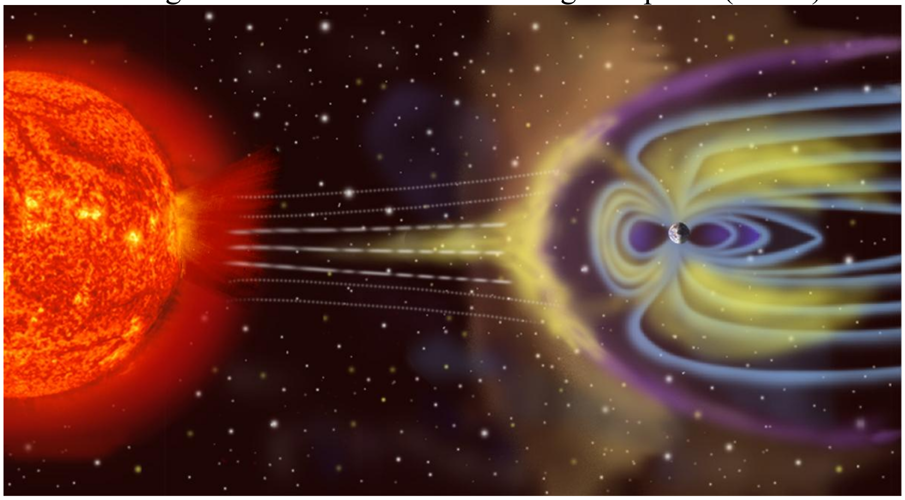


Appendix B: Illustrations

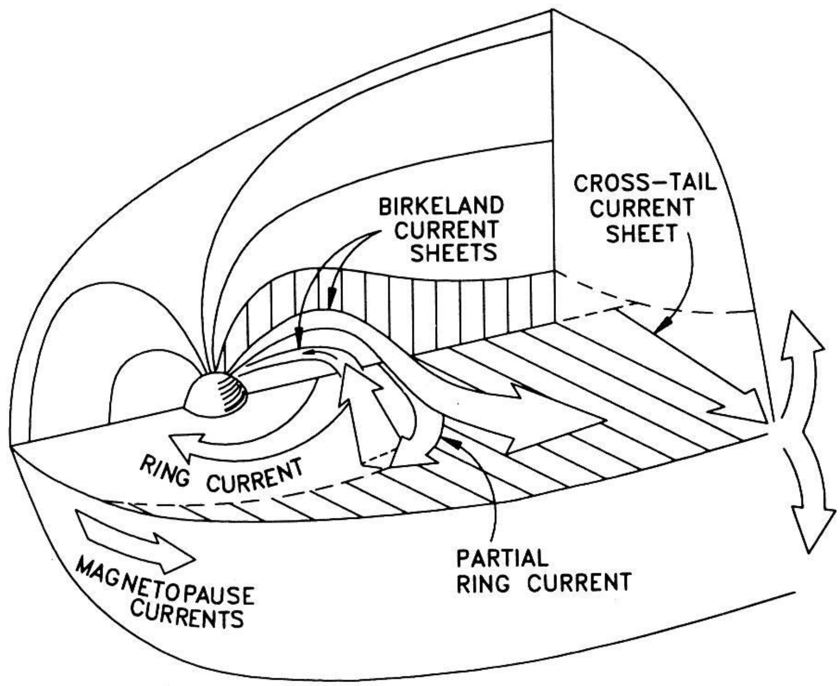
(A) Illustration showing trapped particles in Van Allen belts (below).



(B) Illustration showing solar wind's influence on magnetosphere (below).

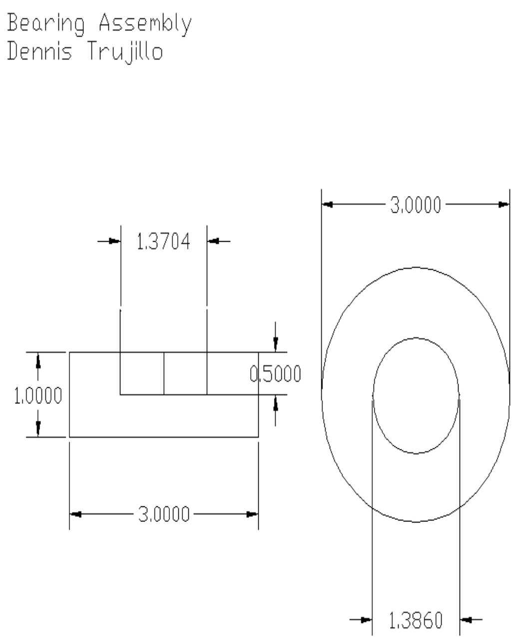


(C) Illustration showing magnetosphere currents (below).

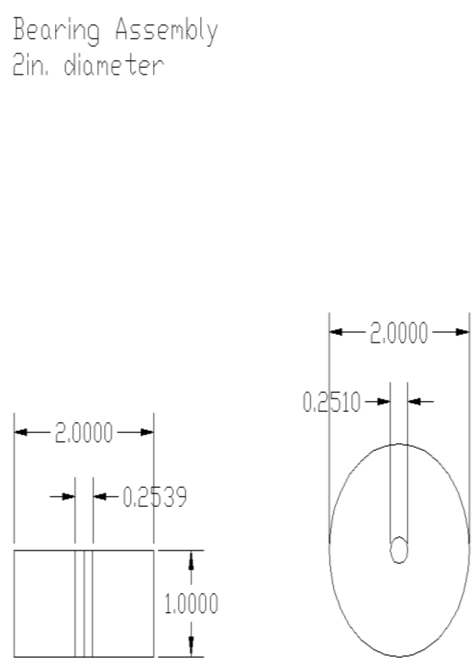


Appendix C: Physical Model Schematics

(A) Bearing assembly (below).

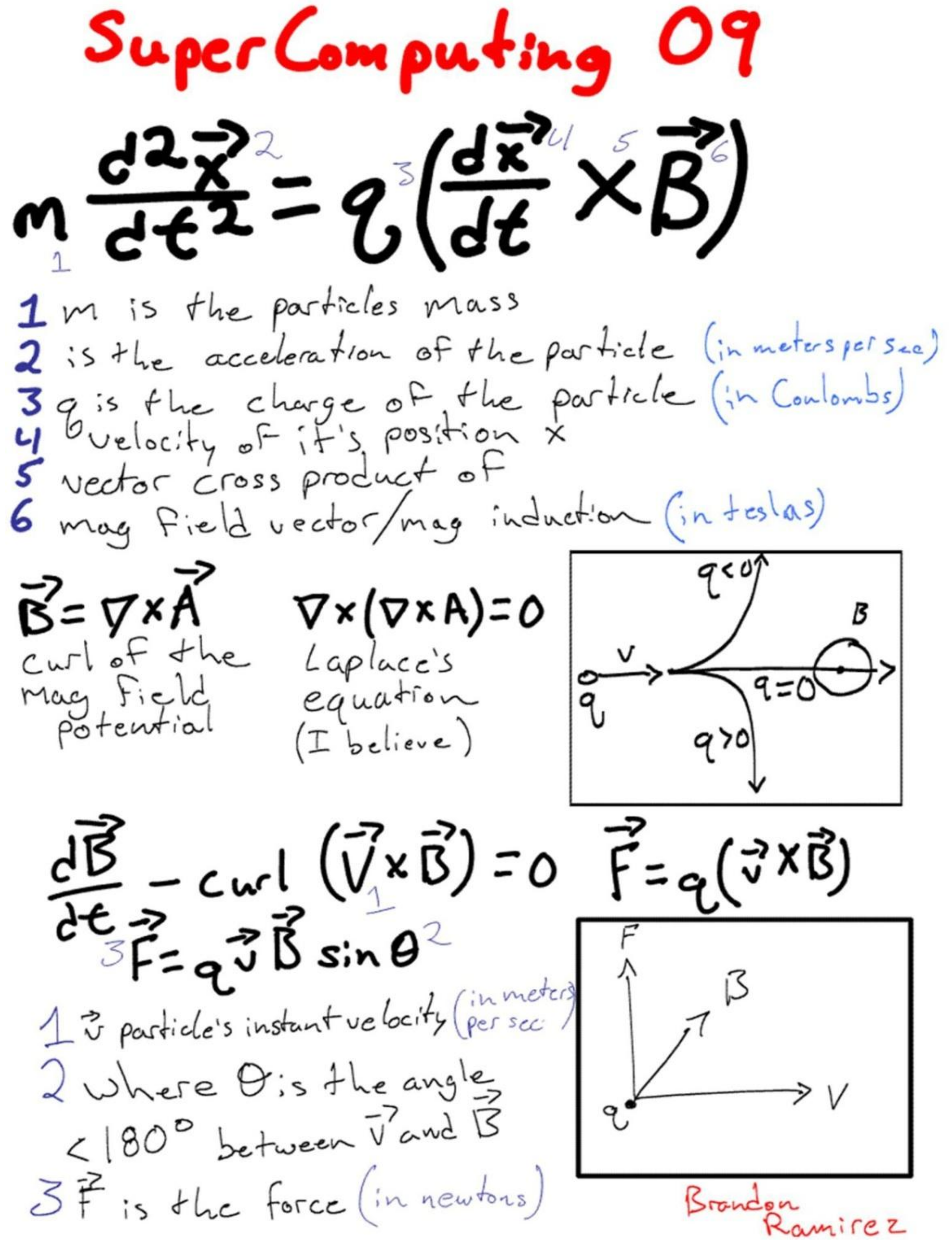


(B) Alternate bearing assembly (below).



Appendix D: Mathematical Model Notes

(A) Mathematical model notes (below).



SuperComputing 09

$$m \frac{d^2\vec{x}}{dt^2} = q\left(\frac{d\vec{x}}{dt} \times \vec{B}\right)$$

1 m is the particles mass
2 is the acceleration of the particle (in meters per sec)
3 q is the charge of the particle (in Coulombs)
4 velocity of it's position x
5 vector cross product of
6 mag field vector/mag induction (in teslas)

$\vec{B} = \nabla \times \vec{A}$ curl of the mag field potential

$\nabla \times (\nabla \times A) = 0$ Laplace's equation (I believe)

$\frac{d\vec{B}}{dt} - \text{curl}(\vec{V} \times \vec{B}) = 0$

$\vec{F} = q(\vec{v} \times \vec{B})$

$\vec{F} = q\vec{v}\vec{B}\sin\theta$

1 $\vec{v}$ particle's instant velocity (in meters per sec)
2 where θ is the angle < 180° between $\vec{v}$ and $\vec{B}$
3 $\vec{F}$ is the force (in newtons)

Brandon Ramirez

(B) Mathematical model notes continued

the vector ("cross") product of $\vec{\nabla}$ with $\vec{A}(x,y,z)$ is
is the curl of $\vec{A}$

$$\vec{\nabla}\times\vec{A} = \vec{i}\left(\frac{dA_z}{dy} - \frac{dA_y}{dz}\right) + \vec{j}\left(\frac{dA_x}{dz} - \frac{dA_z}{dx}\right) + \vec{k}\left(\frac{dA_y}{dx} - \frac{dA_x}{dy}\right)$$

$$\vec{r}_g = \frac{m\vec{v}_\perp}{|q|\vec{B}}$$

1 r_g is the gyroradius. The motion of a charged particle in a uniform mag field

2 velocity perpendicular to the direction of the mag field

$$\frac{m\vec{v}_\perp^2}{\vec{r}_g} = q\vec{v}_\perp\vec{B}$$

Notice that the direction of the force $\vec{F}$ is given by the cross product of the velocity $\vec{v}$ and magnetic field $\vec{B}$. Thus, the Lorentz force will always act perpendicular to the direction of motion, causing the particle to move in a circle (gyrate). The radius of this circle r_g can be determined by equating the magnitude of the Lorentz force to the centripetal force

$\vec{F}$

$$m \cdot a = q(v \times (\nabla \times A)) = \frac{mv_\perp}{|q|B}$$

$\vec{V}$, $\vec{B}$

} r_g if mag field is constant

$$a = \frac{d^2x}{dt^2}$$

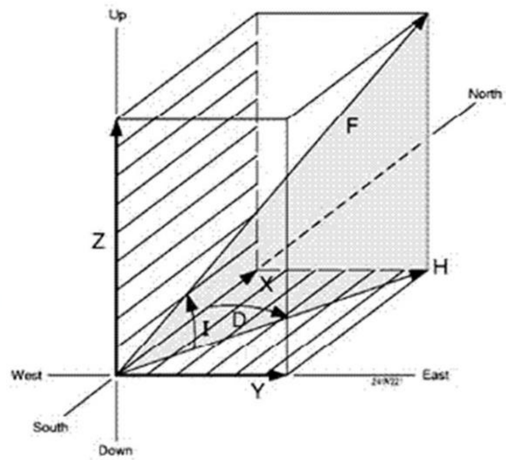
$$v = \frac{dx}{dt}$$

need $\vec{A}(x,y,z)$ and $\vec{x}$, m, q

(below).

(C) Mathematical model notes continued (below).

X = projection of F on the horizontal plane and directed to geographic North;
Y = projection of F on the horizontal plane and directed to geographic East;
Z = vertical component of F, assumed positive if directed downward;
H = total intensity of the field;
F = total intensity of the field;
D = magnetic declination, angle between the direction of H and the geographic meridian passing through the observation point, D is taken positive when H points eastward the geographic meridian;
I = magnetic inclination, angle between F and H, I is assumed positive when F points downwards.



$H = X^2 + Y^2 \quad X = H\cos(D) \quad Y = H\sin(D) \quad Z = (F^2 - H^2)$

$H = F\cos(I) \quad X = F\cos(I)\cos(D) \quad Y = F\cos(I)\sin(D) \quad Z = F\sin(I)$

$F = \sqrt{X^2 + Y^2 + Z^2} \quad D = \arctan(Y/X) \quad I = \arctan(Z/H)$

$F = \sqrt{H^2 + Z^2} \quad H = \sqrt{X^2 + Y^2}$

$$\overset{1}{B(r,t)} = \overset{2}{B_m(r,t)} + \overset{3}{B_c(r)} + \overset{4}{B_d(r,t)}$$

1 $\vec{B}$ is dependent on r position and t time
2 B_m is the main field produced by the core
3 B_c is the field from rocks
4 B_d the field created by the electric currents in the ground and the sea

(D) Mathematical model notes continued (below).

$$\vec{B}(r,t) = -\vec{\nabla}\vec{V}(r,\theta,\lambda,t) \qquad \vec{\nabla}^2\vec{V} = \vec{\nabla}^2 \times \vec{A} = 0$$

1 negative gradient of $\vec{V}$
2 r is the radial distance from the centre of the Earth
3 θ the geocentric colatitude (90°-latitude)
4 λ the longitude east of Greenwich

$$V(r,\theta,\lambda,t) = a\sum_{n=1}^{ni_{\max}}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}\left(g_n^m(t)\cdot\cos m\lambda + h_n^m(t)\cdot\sin m\lambda\right)\cdot P_n^m(\theta)$$

$$\Sigma \qquad + a\sum_{n=1}^{ne_{\max}}\left(\frac{r}{a}\right)^{n}\sum_{m=0}^{n}\left(q_n^m(t)\cdot\cos m\lambda + s_n^m(t)\cdot\sin m\lambda\right)\cdot P_n^m(\theta)$$

1 a is the mean radius of the earth
2 g_n^m h_n^m are internal Gauss coefficients
3 q_n^m s_n^m are external Gauss coefficients
4 the Schmidt-normalized associate Legendre functions to the degree of n and order m
commonly n=10 or n=12

This last piece is intense I don't really understand it

Explaination of sumation:

$$\sum_{k=2}^{6} k^2 = 2^2 + 3^2 + 4^2 + 5^2 + 6^2 = 90$$

1 This was previously explained

$$\frac{d\vec{B}}{dt} = \vec{\nabla} \times (\vec{v} \times \vec{B}) + \frac{1}{\sigma\mu}\nabla^2\vec{B}$$

Appendix E: Equations Used in Program

$$M\frac{\partial^2\vec{x}}{\partial t^2} = q\left(\frac{\partial\vec{x}}{\partial t}\times B\right)$$

M is Mass
Q is the charge of the particle
B is the magnetic field vector

$$\vec{\nabla}\times\vec{A} = \vec{B}$$

A curl variable of B

$$r_g = \frac{m\vec{v}_\perp}{|q|\vec{B}}$$

$\mathbf{r_g}$ is the gyro radius

$$X = H\cos(D) \qquad X = F\cos(I)\cos(D) \qquad Z = F\sin(I)$$

$$Y = H\sin(D) \qquad Y = F\cos(I)\sin(D) \qquad F = \sqrt{H^2+Z^2}$$

$$Z = F^2 - H^2 \qquad F = \sqrt{X^2+Y^2+Z^2} \qquad H = \sqrt{X^2+Y^2}$$

$$H = F\cos(I) \qquad I = \tan^{-1}\left({}^{Z}\!/_{H}\right) \qquad D = \tan^{-1}\left({}^{Y}\!/_{X}\right)$$

X = projection of F on the horizontal plane and directed to geographic North
Y = projection of F on the horizontal plane and directed to geographic East
Z = vertical component of F, assumed positive if directed downward
D = magnetic declination, angle between the direction of H and the geographic meridian passing through the observation point, D is taken positive when H points eastward the geographic meridian
H = total intensity of the field
I = magnetic inclination, angle between F and H, I is assumed positive when F points downwards
F = total intensity of the field

$$\vec{B}_{(R,t)} = -\vec{\nabla}\,\vec{V}(r, \theta, \lambda, t)$$

R is the position of B
T is time
r is the radial distance from the core of the earth
θ is the colatitude
λ is the longitude
∇ is the gradient

$$\vec{\nabla} \times \vec{A}_{(x,y,z)} = \vec{i}\left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right) + \vec{j}\left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right) + \vec{k}\left(\frac{\partial A_x}{\partial y} - \frac{\partial A_y}{\partial x}\right)$$

A is the vector field variable

(Lines 2675-2676)

$$\theta = 90 - \left(ptsd \times \left[\left(\frac{i+1}{3}\right) - j\right] \times \frac{\pi}{180}\right)$$

R=Re

$$\Phi = \left[(ptsd + k) \times \left(\frac{\pi}{180}\right)\right]$$

Pi=3.14159265358979323846264338327950
A=(size(mag)/3.0)
Ptsd=(360/$\sqrt{(a)}$)
J=0
K=ptsd

Appendix F: Program

(A) FORTRAN code with T89c, GEOPACK-2008, and particle interaction.

```
program charged_particle_interaction
use charged_particle_interaction_module
implicit none
    Real*8, Allocatable, Dimension(:, :, :) :: Particle, Magnetosphere
    Real*8, Allocatable, Dimension(:,:) ::
Magnetopos,Magnetocon,Magnetoex,Magnetodi,Particle_con,Particle_ex,Particle_di
    Real*8, Allocatable, Dimension(:) :: Mag_pos,Mag_con,Mag_ex,Mag_di          !XX,YY,ZZ
    Real, Dimension(10) ::  Parmod
    Real*4 :: PS,FX,FY,FZ,FBX,FBY,FBZ
    Real*8 :: R,Theta,Phi,BR,BTheta,BPhi,X,Y,Z,HXgsw,HYgsw,HZgsw,VgseX,VgseY,VgseZ
    Real*8 ::
XGSW,YGSW,ZGSW,BXGSW,BYGSW,BZGSW,XN_PD,VEL,BZIMF,PXGSW,PYGSW,PZ
GSW
    Real*8 :: DIST,XMGNP,YMGNP,ZMGNP,XI,YI,ZI,DIR,DSMAX,ERR,RLIM,R0,DeltaTB
    Real*8 :: EXNAME,INNAME,XF,YF,ZF,DeltaT,BX,BY,BZ,dshls,num_shls,shel
    Real*8, Allocatable, Dimension(:) ::C,CEnergy,CVel,CR,Cangle
    Integer :: num_points,i,j,k,IYear,IDay,THour,Month,Day,ID,L,LMAX,NTsteps,Cnum,num_r

    num_shls=60.0d0
    dshls=1.0/3.0
    num_r= int(num_shls*dshls)
    shel=1.0d0
    num_points=500

    VgseX=-400.0d0
    VgseY=0.0d0
    VgseZ=VgseY
!Trace variables
!     XI=1.0d0
!     YI=1.0d0
!     ZI=1.0d0
!     DSMAX=20.0d0
!     RLIM=10.0d0
!     LMAX=200
!     PS=0.0d0

! Other variables used for some options
!     Bzimf=1.0d0
!     Xn_Pd=8.7d0
!     Vel=468.0d0

! Situation Variables
```

```
      Allocate(C(1:18))
      Allocate(CEnergy(1:4))
      Allocate(CVel(1:4))
      Allocate(CR(1:4))
      Allocate(Cangle(1:4))
      C = (/1.0d0,0.0d0,0.0d0,0.0d0,1.0d0,0.0d0,0.2588d0,0.0d0,0.9659d0,0.5d0, &
           0.0d0,0.866d0,0.7071d0,0.0d0,0.7071d0,0.866d0,0.0d0,0.5d0/)
      CEnergy = (/ 10000.0d0,100000.0d0,1000000.0d0,10000000.0d0/) !in eV
      CVel = (/1.39e3,4.38e3,1.39e4,4.38e4/) ! in km/s w/ C array 4 vels
      CR = (/2.0d0,4.0d0,6.0d0,8.0d0 /) !in Radii
      Cangle = (/0.0d0,90.0d0,180.0d0,270.0d0/) !in degrees
      Cnum=(size(C)/3)*size(CEnergy)*size(CR)*size(Cangle)

! Time variables
      DeltaT= 1.0e-3
      NTsteps=100     !1000
      DeltaTB=NTsteps/(100.0d0*DeltaT)
      IYear=2008
      THour=12
      Month=1
      Day=15

      Allocate(Magnetosphere(1:4 , 1:(num_shls) , 1:(num_points*3) ))
      Allocate(Particle(1:3 , 1:Cnum , 1:(NTsteps*3) ))

      Allocate(Magnetopos(1:(num_shls) , 1:(num_points*3) ) )
      Allocate(Magnetocon(1:(num_shls) , 1:(num_points*3) ) )
      Allocate(Magnetoex(1:(num_shls) , 1:(num_points*3) ) )
      Allocate(Magnetodi(1:(num_shls) , 1:(num_points*3) ) )

      Allocate(Particle_con(1:Cnum , 1:(NTsteps*3)))
      Allocate(Particle_ex(1:Cnum , 1:(NTsteps*3)))
      Allocate(Particle_di(1:Cnum , 1:(NTsteps*3)))

      Allocate(Mag_pos(1:(num_points*3)))
      Allocate(Mag_con(1:(num_points*3)))
      Allocate(Mag_ex(1:(num_points*3)))
      Allocate(Mag_di(1:(num_points*3)))

      !do !!!!!!!!!!!!!!!

      Call CHG_DAY (Month,Day,IYear,IDay)                    ! #21
!                                                          Call RECALC_08
(IYEAR,IDAY,IHOUR,MIN,ISEC,VGSEX,VGSEY,VGSEZ)
!     if  VGSEX=-400.0 AND VGSEY=VGSEZ=0.  GSW=GSM
      Call RECALC_08 (IYear,IDay,Thour,00,00,VGSEX,VGSEY,VGSEZ)   ! #8
```

```
     do k= 1,num_shls   ! magnetosphere shell array set
      Call fill_mag(Mag_pos,shel)
      do i= 2,((num_points*3)-1),3 ! shell values
        R = Mag_pos(i-1)
        Theta = Mag_pos(i)
        Phi = Mag_pos(i+1)
        call SPHCAR_08 (R,THETA,PHI,X,Y,Z,1)                ! #5
        Mag_pos(i-1)= X
        Mag_pos( i )= Y
        Mag_pos(i+1)= Z
        call IGRF_GSW_08 (X,Y,Z,HXGSW,HYGSW,HZGSW)          ! #1
        Mag_con(i-1)= HXGSW
        Mag_con( i )= HYGSW
        Mag_con(i+1)= HZGSW
        Call DIP_08 (X,Y,Z,BXGSW,BYGSW,BZGSW)      ! #3
        Mag_di(i-1)= BXGSW
        Mag_di( i )= BYGSW
        Mag_di(i+1)= BZGSW

           FX = x !real * 4 pos
           FY = y !
           FZ = z !
           Call T89C(6,PARMOD,PS,FX,FY,FZ,FBX,FBY,FBZ) !more extreme ... real*4
subroutine
           Mag_ex(i-1)=Mag_con(i-1)+FBX
           Mag_ex( i )=Mag_con( i )+FBY
           Mag_ex(i+1)=Mag_con(i+1)+FBZ
           Call T89C(1,PARMOD,PS,FX,FY,FZ,FBX,FBY,FBZ) !conservative ...
        Mag_con(i-1)= FBX + Mag_con(i-1)
        Mag_con( i )= FBY + Mag_con( i )
        Mag_con(i+1)= FBZ + Mag_con(i+1)
      enddo
      Magnetopos( k , 1:(num_points*3) )= Mag_pos
      Magnetodi( k , 1:(num_points*3) )= Mag_di
      Magnetoex( k , 1:(num_points*3) )= Mag_ex
      Magnetocon( k , 1:(num_points*3) )= Mag_con
      shel = shel + dshls

     enddo

     Call
PARTICLAC(Particle_di,deltat,NTsteps,CVel,CR,Cangle,CEnergy,C,Magnetopos,Magnetodi,2)
)!
```

```
     Call
PARTICLAC(Particle_ex,deltat,NTsteps,CVel,CR,Cangle,CEnergy,C,Magnetopos,Magnetoex,2
)
     Call
PARTICLAC(Particle_con,deltat,NTsteps,CVel,CR,Cangle,CEnergy,C,Magnetopos,Magnetoco
n,2)
print*, "here"
     Magnetosphere(1 , 1:(num_shls) , 1:(num_points*3) ) =Magnetopos
     Magnetosphere(2 , 1:(num_shls) , 1:(num_points*3) ) =Magnetodi
     Magnetosphere(3 , 1:(num_shls) , 1:(num_points*3) ) =Magnetoex
     Magnetosphere(4 , 1:(num_shls) , 1:(num_points*3) ) =Magnetocon
     Particle(1 , 1:Cnum , 1:(NTsteps*3) ) =Particle_di
     Particle(2 , 1:Cnum , 1:(NTsteps*3) ) =Particle_ex
     Particle(3 , 1:Cnum , 1:(NTsteps*3) ) =Particle_con

     Open(1,file='magdata.txt',status='old')
!        write(*,fmt= "(10f12.5)") Magnetosphere
!        write(*,*) Magnetosphere
       write(1,fmt= "(10f12.5)") Magnetosphere

     Close(1)

     Open(2,file='particledata.txt',status='old')
       write(*,*) Particle
       write(2,*) Particle
     Close(2)
end program charged_particle_interaction
!^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
          !write(unit= , fmt= , rec= , advance= )

!        write(*,fmt= '(5f27.20)') Particle
!        write(*,fmt= '(x2)') Particle

!        write(*,FMT="15",REC=12,ADVANCE="yes)")
Mag_pos,"\n\n",Mag_di,"\n\n",Mag_con,"\n\n",Mag_ex
!          write(*,FMT="(15.12)",ADVANCE="yes")
"\n\n",Mag_Shl2pos,"\n\n",Mag_Shl2di,"\n\n",Mag_Shl2con,"\n\n",Mag_Shl2ex
          !write(1,*,advance = "no") Mag_1,"\n\n",Mag_di,"\n\n",Mag_2,"\n\n",Mag_3
!        write(*,fmt= "(10f12.5)") Magnetosphere
!        write(*,*) Magnetosphere
```

```
Module charged_particle_interaction_module
contains
```

! 1.CALCULATES COMPONENTS OF THE MAIN (INTERNAL) GEOMAGNETIC FIELD IN THE GEOCENTRIC SOLAR-WIND
! (GSW) COORDINATE SYSTEM
! 2.CALCULATES COMPONENTS OF THE MAIN (INTERNAL) GEOMAGNETIC FIELD IN THE SPHERICAL GEOGRAPHIC
! (GEOCENTRIC) COORDINATE SYSTEM
! 3.CALCULATES GSW (GEOCENTRIC SOLAR-WIND) COMPONENTS OF GEODIPOLE FIELD WITH THE DIPOLE MOMENT
! CORRESPONDING TO THE EPOCH
! 4.CALCULATES FOUR QUANTITIES NECESSARY FOR COORDINATE TRANSFORMATIONS
! WHICH DEPEND ON SUN POSITION (AND, HENCE, ON UNIVERSAL TIME AND SEASON
! 5.CONVERTS SPHERICAL COORDS INTO CARTESIAN ONES AND VICE VERSA
! 6.CALCULATES CARTESIAN FIELD COMPONENTS FROM LOCAL SPHERICAL ONES
! 7.CALCULATES LOCAL SPHERICAL FIELD COMPONENTS FROM THOSE IN CARTESIAN SYSTEM
! 8.A. PREPARES ELEMENTS OF ROTATION MATRICES FOR TRANSFORMATIONS OF VECTORS BETWEEN
! SEVERAL COORDINATE SYSTEMS, MOST FREQUENTLY USED IN SPACE PHYSICS.
! B. PREPARES COEFFICIENTS USED IN THE CALCULATION OF THE MAIN GEOMAGNETIC FIELD
! (IGRF MODEL)
! 9.THIS SUBROUTINE TRANSFORMS COMPONENTS OF ANY VECTOR BETWEEN THE STANDARD GSE
! COORDINATE SYSTEM AND THE GEOCENTRIC SOLAR-WIND (GSW, aka GSWM)
! 10.CONVERTS GEOGRAPHIC (GEO) TO DIPOLE (MAG) COORDINATES OR VICE VERSA.
! 11.CONVERTS EQUATORIAL INERTIAL (GEI) TO GEOGRAPHICAL (GEO) COORDS
! OR VICE VERSA.
! 12.CONVERTS DIPOLE (MAG) TO SOLAR MAGNETIC (SM) COORDINATES OR VICE VERSA
! 13.CONVERTS SOLAR MAGNETIC (SM) TO GEOCENTRIC SOLAR-WIND (GSW) COORDINATES OR VICE VERSA.
! 14.CONVERTS GEOGRAPHIC (GEO) TO GEOCENTRIC SOLAR-WIND (GSW) COORDINATES OR VICE VERSA.
! 15.THIS SUBROUTINE (A) CONVERTS VERTICAL LOCAL HEIGHT (ALTITUDE) H AND GEODETIC
! LATITUDE XMU INTO GEOCENTRIC COORDINATES R AND THETA (GEOCENTRIC RADIAL
! DISTANCE AND COLATITUDE, RESPECTIVELY; ALSO KNOWN AS ECEF COORDINATES),
! AS WELL AS (B) PERFORMS THE INVERSE TRANSFORMATION FROM {R,THETA} TO {H,XMU}

```
!   16.CALCULATES THE COMPONENTS OF THE RIGHT HAND SIDE VECTOR IN THE
GEOMAGNETIC FIELD
!    LINE EQUATION
!   17.RE-CALCULATES THE INPUT VALUES {X,Y,Z} (IN GSW COORDINATES) FOR
ANY POINT ON A FIELD LINE
!   18.TRACES A FIELD LINE FROM AN ARBITRARY POINT OF SPACE TO THE
EARTH'S
!    SURFACE OR TO A MODEL LIMITING BOUNDARY.
!   19.FOR ANY POINT OF SPACE WITH COORDINATES (XGSW,YGSW,ZGSW) AND
SPECIFIED CONDITIONS
!    IN THE INCOMING SOLAR WIND, THIS SUBROUTINE:
!    A. DETERMINES IF THE POINT (XGSW,YGSW,ZGSW) LIES INSIDE OR OUTSIDE
THE
!       MODEL MAGNETOPAUSE OF SHUE ET AL. (JGR-A, V.103, P. 17691, 1998).
!    B. CALCULATES THE GSW POSITION OF A POINT {XMGNP,YMGNP,ZMGNP},
LYING AT THE MODEL
!       MAGNETOPAUSE AND ASYMPTOTICALLY TENDING TO THE NEAREST
BOUNDARY POINT WITH
!       RESPECT TO THE OBSERVATION POINT {XGSW,YGSW,ZGSW}, AS IT
APPROACHES THE MAGNETO-
!       PAUSE.
!   20.FOR ANY POINT OF SPACE WITH GIVEN COORDINATES (XGSW,YGSW,ZGSW),
THIS SUBROUTINE DEFINES
!    THE POSITION OF A POINT (XMGNP,YMGNP,ZMGNP) AT THE T96 MODEL
MAGNETOPAUSE WITH THE
!    SAME VALUE OF THE ELLIPSOIDAL TAU-COORDINATE, AND THE DISTANCE
BETWEEN THEM.
!   21.
!   22.
!   23.
!   24.CONVERTS NUMERICAL NOTATION DATE INTO UNIVERSAL
!   25.FILL IN SUBROUTINE FOR THE MAGNETIC FIELD VECTOR ARRAY
!   26.
!   27.PRINTS MAGNETIC FIELD VECTOR ARRAY DISPLAY
!
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@
!
!
##########################################################################
!         #                                                  #
!         #                  GEOPACK-2008_dp                 #
!         #             (MAIN SET OF FORTRAN CODES)              #
!         #         (double-precision version of 03/21/08)      #
!         #         (with a minor correction made on 04/21/08)     #
```

```
!         #                                                                            #
!
################################################################################
!
!
!
!  This collection of subroutines is a result of several upgrades of the original package
!  written by N. A. Tsyganenko in 1978-1979.
!
!  PREFATORY NOTE TO THE VERSION OF FEBRUARY 4, 2008:
!
!  To avoid inappropriate use of obsolete subroutines from earlier versions, a suffix 08 was
!  added to the name of each subroutine in this release.
!
!  A possibility has been added in this version to calculate vector components in the
!  "Geocentric Solar Wind" (GSW) coordinate system, which, to our knowledge, was first
!  introduced by Hones et al., Planet. Space Sci., v.34, p.889, 1986 (aka GSWM, see Appendix,
!  Tsyganenko et al., JGRA, v.103(A4), p.6827, 1998). The GSW system is analogous to the
!  standard GSM, except that its X-axis is antiparallel to the currently observed solar wind
!  flow vector, rather than aligned with the Earth-Sun line. The orientation of axes in the
!  GSW system can be uniquely defined by specifying three components
(VGSEX,VGSEY,VGSEZ) of
!  the solar wind velocity, and in the case of a strictly radial anti-sunward flow (VGSEY=
!  VGSEZ=0) the GSW system becomes identical to the standard GSM, which fact was used here
!  to minimize the number of subroutines in the package. To that end, instead of the special
!  case of the GSM coordinates, this version uses a more general GSW system, and three more
!  input parameters are added in the subroutine RECALC_08, the observed values
(VGSEX,VGSEY,
!  VGSEZ) of the solar wind velocity. Invoking RECALC_08 with VGSEY=VGSEZ=0 restores
the
!  standard (sunward) orientation of the X axis, which allows one to easily convert vectors
!  between GSW and GSM, as well as to/from other standard and commonly used systems. For
more
!  details, see the documentation file GEOPACK-2008.DOC.
!
!  Another modification allows users to have more control over the procedure of field line
!  mapping using the subroutine TRACE_08. To that end, three new input parameters were added
!  in that subroutine, allowing one to set (i) an upper limit, DSMAX, on the automatically
!  adjusted step size, (ii) a permissible step error, ERR, and (iii) maximal length, LMAX,
!  of arrays where field line point coordinates are stored. Minor changes were also made in
!  the tracing subroutine, to make it more compact and easier for understanding, and to
!  prevent the algorithm from making uncontrollable large number of multiple loops in some
!  cases with plasmoid-like field structures.
!
!  One more subroutine, named GEODGEO_08, was added to the package, allowing one to
convert
```

```
!  geodetic coordinates of a point in space (altitude above the Earth's WGS84 ellipsoid and
!  geodetic latitude) to geocentric radial distance and colatitude, and vice versa.
!
!  For a complete list of modifications made earlier in previous versions, see the
!  documentation file GEOPACK-2008.DOC.
!
!----------------------------------------------------------------------------------
!
      SUBROUTINE IGRF_GSW_08 (XGSW,YGSW,ZGSW,HXGSW,HYGSW,HZGSW)
!
!  CALCULATES COMPONENTS OF THE MAIN (INTERNAL) GEOMAGNETIC FIELD IN
THE GEOCENTRIC SOLAR-WIND     1.1.1.1.1.1.1.1
!  (GSW) COORDINATE SYSTEM, USING IAGA INTERNATIONAL GEOMAGNETIC
REFERENCE MODEL COEFFICIENTS
!  (e.g., http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html, revised 22 March, 2005)
!
!  THE GSW SYSTEM IS ESSENTIALLY SIMILAR TO THE STANDARD GSM (THE TWO
SYSTEMS BECOME IDENTICAL
!  TO EACH OTHER IN THE CASE OF STRICTLY ANTI-SUNWARD SOLAR WIND
FLOW). FOR A DETAILED
!  DEFINITION, SEE INTRODUCTORY COMMENTS FOR THE SUBROUTINE
GSWGSE_08 .
!
!  BEFORE THE FIRST CALL OF THIS SUBROUTINE, OR, IF THE DATE/TIME
(IYEAR,IDAY,IHOUR,MIN,ISEC),
!  OR THE SOLAR WIND VELOCITY COMPONENTS (VGSEX,VGSEY,VGSEZ) HAVE
CHANGED, THE MODEL COEFFICIENTS
!  AND GEO-GSW ROTATION MATRIX ELEMENTS SHOULD BE UPDATED BY
CALLING THE SUBROUTINE RECALC_08.
!
!-----INPUT PARAMETERS:
!
!     XGSW,YGSW,ZGSW - CARTESIAN GEOCENTRIC SOLAR-WIND COORDINATES
(IN UNITS RE=6371.2 KM)
!
!-----OUTPUT PARAMETERS:
!
!     HXGSW,HYGSW,HZGSW - CARTESIAN GEOCENTRIC SOLAR-WIND
COMPONENTS OF THE MAIN GEOMAGNETIC
!                         FIELD IN NANOTESLA
!
!     LAST MODIFICATION:  MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!     THIS VERSION OF THE CODE ACCEPTS DATES FROM 1965 THROUGH 2010.
!
!     AUTHOR: N. A. TSYGANENKO
!
```

```
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK2/ G(105),H(105),REC(105)

      DIMENSION A(14),B(14)

      CALL GEOGSW_08 (XGEO,YGEO,ZGEO,XGSW,YGSW,ZGSW,-1)
      RHO2=XGEO**2+YGEO**2
      R=DSQRT(RHO2+ZGEO**2)
      C=ZGEO/R
      RHO=DSQRT(RHO2)
      S=RHO/R
      IF (S.LT.1.D-10) THEN
        CF=1.D0
        SF=0.D0
      ELSE
        CF=XGEO/RHO
        SF=YGEO/RHO
      ENDIF
!
      PP=1.D0/R
      P=PP
!
!  IN THIS VERSION, THE OPTIMAL VALUE OF THE PARAMETER NM (MAXIMAL
ORDER OF THE SPHERICAL
!    HARMONIC EXPANSION) IS NOT USER-PRESCRIBED, BUT CALCULATED INSIDE
THE SUBROUTINE, BASED
!      ON THE VALUE OF THE RADIAL DISTANCE R:
!
      IRP3=R+2
      NM=3+30/IRP3
      IF (NM.GT.13) NM=13

      K=NM+1
      DO 150 N=1,K
         P=P*PP
         A(N)=P
150      B(N)=P*N

      P=1.D0
      D=0.D0
      BBR=0.D0
      BBT=0.D0
      BBF=0.D0

      DO 200 M=1,K
```

```
      IF(M.EQ.1) GOTO 160
      MM=M-1
      W=X
      X=W*CF+Y*SF
      Y=Y*CF-W*SF
      GOTO 170
160     X=0.D0
      Y=1.D0
170     Q=P
      Z=D
      BI=0.D0
      P2=0.D0
      D2=0.D0
      DO 190 N=M,K
        AN=A(N)
        MN=N*(N-1)/2+M
        E=G(MN)
        HH=H(MN)
        W=E*Y+HH*X
        BBR=BBR+B(N)*W*Q
        BBT=BBT-AN*W*Z
        IF(M.EQ.1) GOTO 180
        QQ=Q
        IF(S.LT.1.D-10) QQ=Z
        BI=BI+AN*(E*X-HH*Y)*QQ
180       XK=REC(MN)
        DP=C*Z-S*Q-XK*D2
        PM=C*Q-XK*P2
        D2=Z
        P2=Q
        Z=DP
190       Q=PM
      D=S*D+C*P
      P=S*P
      IF(M.EQ.1) GOTO 200
      BI=BI*MM
      BBF=BBF+BI
200   CONTINUE
!
    BR=BBR
    BT=BBT
    IF(S.LT.1.D-10) GOTO 210
    BF=BBF/S
    GOTO 211
210   IF(C.LT.0.) BBF=-BBF
    BF=BBF
```

```
211   HE=BR*S+BT*C
      HXGEO=HE*CF-BF*SF
      HYGEO=HE*SF+BF*CF
      HZGEO=BR*C-BT*S
!
      CALL GEOGSW_08 (HXGEO,HYGEO,HZGEO,HXGSW,HYGSW,HZGSW,1)
!
      RETURN
      END SUBROUTINE IGRF_GSW_08
!
!==================================================================
=====================
!
!
      SUBROUTINE IGRF_GEO_08 (R,THETA,PHI,BR,BTHETA,BPHI)
!
!  CALCULATES COMPONENTS OF THE MAIN (INTERNAL) GEOMAGNETIC FIELD IN
THE SPHERICAL GEOGRAPHIC      2.2.2.2.2.2.2.2
!  (GEOCENTRIC) COORDINATE SYSTEM, USING IAGA INTERNATIONAL
GEOMAGNETIC REFERENCE MODEL
!  COEFFICIENTS  (e.g., http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html, revised: 22 March,
2005)
!
!  BEFORE THE FIRST CALL OF THIS SUBROUTINE, OR IF THE DATE (IYEAR AND
IDAY) WAS CHANGED,
!  THE MODEL COEFFICIENTS SHOULD BE UPDATED BY CALLING THE
SUBROUTINE RECALC_08
!
!-----INPUT PARAMETERS:
!
!   R, THETA, PHI - SPHERICAL GEOGRAPHIC (GEOCENTRIC) COORDINATES:
!   RADIAL DISTANCE R IN UNITS RE=6371.2 KM, COLATITUDE THETA AND
LONGITUDE PHI IN RADIANS
!
!-----OUTPUT PARAMETERS:
!
!     BR, BTHETA, BPHI - SPHERICAL COMPONENTS OF THE MAIN GEOMAGNETIC
FIELD IN NANOTESLA
!      (POSITIVE BR OUTWARD, BTHETA SOUTHWARD, BPHI EASTWARD)
!
!     LAST MODIFICATION:  MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!     THIS VERSION OF THE  CODE ACCEPTS DATES FROM 1965 THROUGH 2010.
!
!     AUTHOR: N. A. TSYGANENKO
!
```

```
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK2/ G(105),H(105),REC(105)

      DIMENSION A(14),B(14)

      C=DCOS(THETA)
      S=DSIN(THETA)
      CF=DCOS(PHI)
      SF=DSIN(PHI)
!
      PP=1.D0/R
      P=PP
!
!  IN THIS NEW VERSION, THE OPTIMAL VALUE OF THE PARAMETER NM
(MAXIMAL ORDER OF THE SPHERICAL
!    HARMONIC EXPANSION) IS NOT USER-PRESCRIBED, BUT CALCULATED INSIDE
THE SUBROUTINE, BASED
!      ON THE VALUE OF THE RADIAL DISTANCE R:
!
      IRP3=R+2
      NM=3+30/IRP3
      IF (NM.GT.13) NM=13
      K=NM+1
      DO 150 N=1,K
        P=P*PP
        A(N)=P
150     B(N)=P*N

      P=1.D0
      D=0.D0
      BBR=0.D0
      BBT=0.D0
      BBF=0.D0

      DO 200 M=1,K
        IF(M.EQ.1) GOTO 160
        MM=M-1
        W=X
        X=W*CF+Y*SF
        Y=Y*CF-W*SF
        GOTO 170
160     X=0.D0
        Y=1.D0
170     Q=P
        Z=D
```

```
        BI=0.D0
        P2=0.D0
        D2=0.D0
        DO 190 N=M,K
          AN=A(N)
          MN=N*(N-1)/2+M
          E=G(MN)
          HH=H(MN)
          W=E*Y+HH*X
          BBR=BBR+B(N)*W*Q
          BBT=BBT-AN*W*Z
          IF(M.EQ.1) GOTO 180
          QQ=Q
          IF(S.LT.1.E-5) QQ=Z
          BI=BI+AN*(E*X-HH*Y)*QQ
180         XK=REC(MN)
          DP=C*Z-S*Q-XK*D2
          PM=C*Q-XK*P2
          D2=Z
          P2=Q
          Z=DP
190        Q=PM
        D=S*D+C*P
        P=S*P
        IF(M.EQ.1) GOTO 200
        BI=BI*MM
        BBF=BBF+BI
200   CONTINUE
!
      BR=BBR
      BTHETA=BBT
      IF(S.LT.1.D-10) GOTO 210
      BPHI=BBF/S
      RETURN
210   IF(C.LT.0.D0) BBF=-BBF
      BPHI=BBF

      RETURN
      END SUBROUTINE IGRF_GEO_08
!
!==========================================================================
======================
!
       SUBROUTINE DIP_08 (XGSW,YGSW,ZGSW,BXGSW,BYGSW,BZGSW)
!
```

```
!  CALCULATES GSW (GEOCENTRIC SOLAR-WIND) COMPONENTS OF GEODIPOLE
FIELD WITH THE DIPOLE MOMENT     3.3.3.3.3.3.3.3.3
!  CORRESPONDING TO THE EPOCH, SPECIFIED BY CALLING SUBROUTINE
RECALC_08 (SHOULD BE
!  INVOKED BEFORE THE FIRST USE OF THIS ONE, OR IF THE DATE/TIME, AND/OR
THE OBSERVED
!  SOLAR WIND DIRECTION, HAVE CHANGED.
!
!  THE GSW COORDINATE SYSTEM IS ESSENTIALLY SIMILAR TO THE STANDARD
GSM (THE TWO SYSTEMS BECOME
!  IDENTICAL TO EACH OTHER IN THE CASE OF STRICTLY RADIAL ANTI-
SUNWARD SOLAR WIND FLOW). ITS
!  DETAILED DEFINITION IS GIVEN IN INTRODUCTORY COMMENTS FOR THE
SUBROUTINE GSWGSE_08 .

!--INPUT PARAMETERS: XGSW,YGSW,ZGSW - GSW COORDINATES IN RE (1 RE =
6371.2 km)
!
!--OUTPUT PARAMETERS: BXGSW,BYGSW,BZGSW - FIELD COMPONENTS IN GSW
SYSTEM, IN NANOTESLA.
!
!  LAST MODIFICATION:   MARCH 21, 2008 (DOUBLE-PRECISION VERSION).
!
!  AUTHOR: N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ AA(10),SPS,CPS,BB(22)
      COMMON /GEOPACK2/ G(105),H(105),REC(105)
!
      DIPMOM=DSQRT(G(2)**2+G(3)**2+H(3)**2)
!
      P=XGSW**2
      U=ZGSW**2
      V=3.D0*ZGSW*XGSW
      T=YGSW**2
      Q=DIPMOM/DSQRT(P+T+U)**5
      BXGSW=Q*((T+U-2.D0*P)*SPS-V*CPS)
      BYGSW=-3.D0*YGSW*Q*(XGSW*SPS+ZGSW*CPS)
      BZGSW=Q*((P+T-2.D0*U)*CPS-V*SPS)
      RETURN
      END SUBROUTINE DIP_08

!*****************************************************************
!
      SUBROUTINE SUN_08 (IYEAR,IDAY,IHOUR,MIN,ISEC,GST,SLONG,SRASN,SDEC)
!
```

```
!  CALCULATES FOUR QUANTITIES NECESSARY FOR COORDINATE
TRANSFORMATIONS                               4.4.4.4.4.4.4.4
!  WHICH DEPEND ON SUN POSITION (AND, HENCE, ON UNIVERSAL TIME AND
SEASON)
!
!-------  INPUT PARAMETERS:
!  IYR,IDAY,IHOUR,MIN,ISEC -  YEAR, DAY, AND UNIVERSAL TIME IN HOURS,
MINUTES,
!    AND SECONDS  (IDAY=1 CORRESPONDS TO JANUARY 1).
!
!-------  OUTPUT PARAMETERS:
!  GST - GREENWICH MEAN SIDEREAL TIME, SLONG - LONGITUDE ALONG
ECLIPTIC
!  SRASN - RIGHT ASCENSION,  SDEC - DECLINATION  OF THE SUN (RADIANS)
!  ORIGINAL VERSION OF THIS SUBROUTINE HAS BEEN COMPILED FROM:
!  RUSSELL, C.T., COSMIC ELECTRODYNAMICS, 1971, V.2, PP.184-196.
!
!  LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!     ORIGINAL VERSION WRITTEN BY:    Gilbert D. Mead
!
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA RAD/57.295779513D0/
!
      IF(IYEAR.LT.1901.OR.IYEAR.GT.2099) RETURN
      FDAY=DFLOAT(IHOUR*3600+MIN*60+ISEC)/86400.D0
      DJ=365*(IYEAR-1900)+(IYEAR-1901)/4+IDAY-0.5D0+FDAY
      T=DJ/36525.D0
      VL=DMOD(279.696678D0+0.9856473354D0*DJ,360.D0)
      GST=DMOD(279.690983D0+.9856473354D0*DJ+360.D0*FDAY+180.D0,360.D0)/RAD
      G=DMOD(358.475845D0+0.985600267D0*DJ,360.D0)/RAD
      SLONG=(VL+(1.91946D0-0.004789D0*T)*DSIN(G)+0.020094D0*DSIN(2.D0*G))/RAD
      IF(SLONG.GT.6.2831853D0) SLONG=SLONG-6.283185307D0
      IF (SLONG.LT.0.D0) SLONG=SLONG+6.283185307D0
      OBLIQ=(23.45229D0-0.0130125D0*T)/RAD
      SOB=DSIN(OBLIQ)
      SLP=SLONG-9.924D-5
!
!   THE LAST CONSTANT IS A CORRECTION FOR THE ANGULAR ABERRATION DUE
TO
!   EARTH'S ORBITAL MOTION
!
      SIND=SOB*DSIN(SLP)
      COSD=DSQRT(1.D0-SIND**2)
      SC=SIND/COSD
      SDEC=DATAN(SC)
```

```
      SRASN=3.141592654D0-DATAN2(DCOS(OBLIQ)/SOB*SC,-DCOS(SLP)/COSD)
      RETURN
      END SUBROUTINE SUN_08
!
!==================================================================
============
!
      SUBROUTINE SPHCAR_08 (R,THETA,PHI,X,Y,Z,J)
!
!   CONVERTS SPHERICAL COORDS INTO CARTESIAN ONES AND VICE VERSA
5.5.5.5.5.5.5.5
!    (THETA AND PHI IN RADIANS).
!
!                  J>0            J<0
!-----INPUT:   J,R,THETA,PHI     J,X,Y,Z
!----OUTPUT:      X,Y,Z        R,THETA,PHI
!
!  NOTE: AT THE POLES (X=0 AND Y=0) WE ASSUME PHI=0 WHEN CONVERTING
!        FROM CARTESIAN TO SPHERICAL COORDS (I.E., FOR J<0)
!
!   LAST MOFIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!   AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)

      IF(J.GT.0) GOTO 3
      SQ=X**2+Y**2
      R=DSQRT(SQ+Z**2)
      IF (SQ.NE.0.D0) GOTO 2
      PHI=0.D0
      IF (Z.LT.0.D0) GOTO 1
      THETA=0.D0
      RETURN
  1   THETA=3.141592654D0
      RETURN
  2   SQ=DSQRT(SQ)
      PHI=DATAN2(Y,X)
      THETA=DATAN2(SQ,Z)
      IF (PHI.LT.0.D0) PHI=PHI+6.283185307D0
      RETURN
  3   SQ=R*DSIN(THETA)
      X=SQ*DCOS(PHI)
      Y=SQ*DSIN(PHI)
      Z=R*DCOS(THETA)
      RETURN
```

```
      END SUBROUTINE SPHCAR_08
!
!=====================================================================
=======
!
      SUBROUTINE BSPCAR_08 (THETA,PHI,BR,BTHETA,BPHI,BX,BY,BZ)
!
!   CALCULATES CARTESIAN FIELD COMPONENTS FROM LOCAL SPHERICAL ONES
6.6.6.6.6.6.6.6
!
!-----INPUT:   THETA,PHI - SPHERICAL ANGLES OF THE POINT IN RADIANS
!              BR,BTHETA,BPHI -  LOCAL SPHERICAL COMPONENTS OF THE FIELD
!-----OUTPUT:  BX,BY,BZ - CARTESIAN COMPONENTS OF THE FIELD
!
!   LAST MOFIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!   WRITTEN BY:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)

      S=DSIN(THETA)
      C=DCOS(THETA)
      SF=DSIN(PHI)
      CF=DCOS(PHI)
      BE=BR*S+BTHETA*C
      BX=BE*CF-BPHI*SF
      BY=BE*SF+BPHI*CF
      BZ=BR*C-BTHETA*S
      RETURN
      END SUBROUTINE BSPCAR_08
!
!=====================================================================
==========
!
      SUBROUTINE BCARSP_08 (X,Y,Z,BX,BY,BZ,BR,BTHETA,BPHI)
!
!CALCULATES LOCAL SPHERICAL FIELD COMPONENTS FROM THOSE IN
CARTESIAN SYSTEM                  7.7.7.7.7.7.7.7
!
!-----INPUT:   X,Y,Z  - CARTESIAN COMPONENTS OF THE POSITION VECTOR
!              BX,BY,BZ - CARTESIAN COMPONENTS OF THE FIELD VECTOR
!-----OUTPUT:  BR,BTHETA,BPHI - LOCAL SPHERICAL COMPONENTS OF THE FIELD
VECTOR
!
!  NOTE: AT THE POLES (THETA=0 OR THETA=PI) WE ASSUME PHI=0,
!        AND HENCE BTHETA=BX, BPHI=BY
```

```
!
!   WRITTEN AND ADDED TO THIS PACKAGE:  APRIL 1, 2003
!   LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!   AUTHOR:   N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)

      RHO2=X**2+Y**2
      R=DSQRT(RHO2+Z**2)
      RHO=DSQRT(RHO2)

      IF (RHO.NE.0.D0) THEN
        CPHI=X/RHO
        SPHI=Y/RHO
       ELSE
        CPHI=1.D0
        SPHI=0.D0
      ENDIF

      CT=Z/R
      ST=RHO/R

      BR=(X*BX+Y*BY+Z*BZ)/R
      BTHETA=(BX*CPHI+BY*SPHI)*CT-BZ*ST
      BPHI=BY*CPHI-BX*SPHI

      RETURN
      END SUBROUTINE BCARSP_08
!
!==================================================================
=================
!
      SUBROUTINE RECALC_08 (IYEAR,IDAY,IHOUR,MIN,ISEC,VGSEX,VGSEY,VGSEZ)
!
!  1. PREPARES ELEMENTS OF ROTATION MATRICES FOR TRANSFORMATIONS OF
VECTORS BETWEEN                8.8.8.8.8.8.8.8
!     SEVERAL COORDINATE SYSTEMS, MOST FREQUENTLY USED IN SPACE
PHYSICS.
!
!  2. PREPARES COEFFICIENTS USED IN THE CALCULATION OF THE MAIN
GEOMAGNETIC FIELD
!      (IGRF MODEL)
!
!  THIS SUBROUTINE SHOULD BE INVOKED BEFORE USING THE FOLLOWING
SUBROUTINES:
```

! IGRF_GEO_08, IGRF_GSW_08, DIP_08, GEOMAG_08, GEOGSW_08, MAGSW_08, SMGSW_08, GSWGSE_08,
! GEIGEO_08, TRACE_08, STEP_08, RHAND_08.
!
! THERE IS NO NEED TO REPEATEDLY INVOKE RECALC_08, IF MULTIPLE CALCULATIONS ARE MADE
! FOR THE SAME DATE/TIME AND SOLAR WIND FLOW DIRECTION.
!
!-----INPUT PARAMETERS:
!
! IYEAR - YEAR NUMBER (FOUR DIGITS)
! IDAY - DAY OF YEAR (DAY 1 = JAN 1)
! IHOUR - HOUR OF DAY (00 TO 23)
! MIN - MINUTE OF HOUR (00 TO 59)
! ISEC - SECONDS OF MINUTE (00 TO 59)
! VGSEX,VGSEY,VGSEZ - GSE (GEOCENTRIC SOLAR-ECLIPTIC) COMPONENTS OF THE OBSERVED
! SOLAR WIND FLOW VELOCITY (IN KM/S)
!
! IMPORTANT: IF ONLY QUESTIONABLE INFORMATION (OR NO INFORMATION AT ALL) IS AVAILABLE
! ON THE SOLAR WIND SPEED, OR, IF THE STANDARD GSM AND/OR SM COORDINATES ARE
! INTENDED TO BE USED, THEN SET VGSEX=-400.0 AND VGSEY=VGSEZ=0. IN THIS CASE,
! THE GSW COORDINATE SYSTEM BECOMES IDENTICAL TO THE STANDARD GSM.
!
! IF ONLY SCALAR SPEED V OF THE SOLAR WIND IS KNOWN, THEN SETTING
! VGSEX=-V, VGSEY=29.78, VGSEZ=0.0 WILL TAKE INTO ACCOUNT THE ~4 degs
! ABERRATION OF THE MAGNETOSPHERE DUE TO EARTH'S ORBITAL MOTION
!
! IF ALL THREE GSE COMPONENTS OF THE SOLAR WIND VELOCITY ARE AVAILABLE,
! PLEASE NOTE THAT IN SOME SOLAR WIND DATABASES THE ABERRATION EFFECT
! HAS ALREADY BEEN TAKEN INTO ACCOUNT BY SUBTRACTING 29.78 KM/S FROM VYGSE;
! IN THAT CASE, THE UNABERRATED (OBSERVED) VYGSE VALUES SHOULD BE RESTORED
! BY ADDING BACK THE 29.78 KM/S CORRECTION. WHETHER OR NOT TO DO THAT, MUST
! BE EITHER VERIFIED WITH THE DATA ORIGINATOR OR DETERMINED BY AVERAGING

```
!           VGSEY OVER A SUFFICIENTLY LONG TIME INTERVAL.
!
!-----OUTPUT PARAMETERS:  NONE (ALL OUTPUT QUANTITIES ARE PLACED
!                     INTO THE COMMON BLOCKS /GEOPACK1/ AND /GEOPACK2/)
!
!   OTHER SUBROUTINES CALLED BY THIS ONE: SUN_08
!
!   AUTHOR:  N.A. TSYGANENKO
!   DATE:    DEC.1, 1991
!
!   REVISION OF NOVEMBER 15, 2007: ADDED THE POSSIBILITY TO TAKE INTO
ACCOUNT THE OBSERVED
!    DEFLECTION OF THE SOLAR WIND FLOW FROM STRICTLY RADIAL
DIRECTION. TO THAT END, THREE
!    GSE COMPONENTS OF THE SOLAR WIND VELOCITY WERE ADDED TO THE
INPUT PARAMETERS.
!
!   CORRECTION OF MAY 9, 2006:  INTERPOLATION OF THE COEFFICIENTS
(BETWEEN
!    LABELS 50 AND 105) IS NOW MADE THROUGH THE LAST ELEMENT OF THE
ARRAYS
!    G(105)  AND H(105) (PREVIOUSLY MADE ONLY THROUGH N=66, WHICH IN
SOME
!    CASES CAUSED RUNTIME ERRORS)
!
!   REVISION OF MAY 3, 2005:
!    The table of IGRF coefficients was extended to include those for the epoch 2005
!      the maximal order of spherical harmonics was also increased up to 13
!        (for details, see http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html)
!
!   REVISION OF APRIL 3, 2003:
!   The code now includes preparation of the model coefficients for the subroutines
!   IGRF_08 and GEOMAG_08. This eliminates the need for the SAVE statements, used
!   in the old versions, making the codes easier and more compiler-independent.
!
!   LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
      IMPLICIT REAL*8 (A-H,O-Z)
      SAVE ISW
!
      COMMON /GEOPACK1/ ST0,CT0,SL0,CL0,CTCL,STCL,CTSL,STSL,SFI,CFI, &
       SPS,CPS,DS3,CGST,SGST,PSI,A11,A21,A31,A12,A22,A32,A13,A23,A33, &
       E11,E21,E31,E12,E22,E32,E13,E23,E33
!
!  THE COMMON BLOCK /GEOPACK1/ CONTAINS ELEMENTS OF THE ROTATION
MATRICES AND OTHER
```

```
!   PARAMETERS RELATED TO THE COORDINATE TRANSFORMATIONS
PERFORMED BY THIS PACKAGE
!
      COMMON /GEOPACK2/ G(105),H(105),REC(105)
!
!  THE COMMON BLOCK /GEOPACK2/ CONTAINS COEFFICIENTS OF THE IGRF
FIELD MODEL, CALCULATED
!    FOR A GIVEN YEAR AND DAY FROM THEIR STANDARD EPOCH VALUES. THE
ARRAY REC CONTAINS
!    COEFFICIENTS USED IN THE RECURSION RELATIONS FOR LEGENDRE
ASSOCIATE POLYNOMIALS.
!
      DIMENSION G65(105),H65(105),G70(105),H70(105),G75(105),H75(105), &
       G80(105),H80(105),G85(105),H85(105),G90(105),H90(105),G95(105), &       ! took out +
at beginning
       H95(105),G00(105),H00(105),G05(105),H05(105),DG05(45),DH05(45)          ! same
!
      DATA ISW /0/
!
      DATA G65/0.D0,-30334.D0,-2119.D0,-1662.D0,2997.D0,1594.D0,1297.D0, &
     -2038.D0,1292.D0,856.D0,957.D0,804.D0,479.D0,-390.D0,252.D0, &
     -219.D0,358.D0,254.D0,-31.D0,-157.D0,-62.D0,45.D0,61.D0,8.D0, &
     -228.D0,4.D0,1.D0,-111.D0,75.D0,-57.D0,4.D0,13.D0,-26.D0,-6.D0, &
     13.D0,1.D0,13.D0,5.D0,-4.D0,-14.D0,0.D0,8.D0,-1.D0,11.D0,4.D0, &
     8.D0,10.D0,2.D0,-13.D0,10.D0,-1.D0,-1.D0,5.D0,1.D0,-2.D0,-2.D0, &
     -3.D0,2.D0,-5.D0,-2.D0,4.D0,4.D0,0.D0,2.D0,2.D0,0.D0,39*0.D0/
      DATA H65/0.D0,0.D0,5776.D0,0.D0,-2016.D0,114.D0,0.D0,-404.D0, &
     240.D0,-165.D0,0.D0,148.D0,-269.D0,13.D0,-269.D0,0.D0,19.D0, &
     128.D0,-126.D0,-97.D0,81.D0,0.D0,-11.D0,100.D0,68.D0,-32.D0,-8.D0, &
     -7.D0,0.D0,-61.D0,-27.D0,-2.D0,6.D0,26.D0,-23.D0,-12.D0,0.D0,7.D0, &
     -12.D0,9.D0,-16.D0,4.D0,24.D0,-3.D0,-17.D0,0.D0,-22.D0,15.D0,7.D0, &
     -4.D0,-5.D0,10.D0,10.D0,-4.D0,1.D0,0.D0,2.D0,1.D0,2.D0,6.D0,-4.D0, &
     0.D0,-2.D0,3.D0,0.D0,-6.D0,39*0.D0/
!
      DATA G70/0.D0,-30220.D0,-2068.D0,-1781.D0,3000.D0,1611.D0,1287.D0, &
      -2091.D0,1278.D0,838.D0,952.D0,800.D0,461.D0,-395.D0,234.D0, &
      -216.D0,359.D0,262.D0,-42.D0,-160.D0,-56.D0,43.D0,64.D0,15.D0, &
      -212.D0,2.D0,3.D0,-112.D0,72.D0,-57.D0,1.D0,14.D0,-22.D0,-2.D0, &
      13.D0,-2.D0,14.D0,6.D0,-2.D0,-13.D0,-3.D0,5.D0,0.D0,11.D0,3.D0, &
      8.D0,10.D0,2.D0,-12.D0,10.D0,-1.D0,0.D0,3.D0,1.D0,-1.D0,-3.D0, &
      -3.D0,2.D0,-5.D0,-1.D0,6.D0,4.D0,1.D0,0.D0,3.D0,-1.D0,39*0.D0/
      DATA H70/0.D0,0.D0,5737.D0,0.D0,-2047.D0,25.D0,0.D0,-366.D0, &
      251.D0,-196.D0,0.D0,167.D0,-266.D0,26.D0,-279.D0,0.D0,26.D0, &
      139.D0,-139.D0,-91.D0,83.D0,0.D0,-12.D0,100.D0,72.D0,-37.D0,-6.D0, &
      1.D0,0.D0,-70.D0,-27.D0,-4.D0,8.D0,23.D0,-23.D0,-11.D0,0.D0,7.D0, &
      -15.D0,6.D0,-17.D0,6.D0,21.D0,-6.D0,-16.D0,0.D0,-21.D0,16.D0,6.D0, &
```

```
     -4.D0,-5.D0,10.D0,11.D0,-2.D0,1.D0,0.D0,1.D0,1.D0,3.D0,4.D0,-4.D0, &
     0.D0,-1.D0,3.D0,1.D0,-4.D0,39*0.D0/
!
     DATA G75/0.D0,-30100.D0,-2013.D0,-1902.D0,3010.D0,1632.D0,1276.D0, &
     -2144.D0,1260.D0,830.D0,946.D0,791.D0,438.D0,-405.D0,216.D0, &
     -218.D0,356.D0,264.D0,-59.D0,-159.D0,-49.D0,45.D0,66.D0,28.D0, &
     -198.D0,1.D0,6.D0,-111.D0,71.D0,-56.D0,1.D0,16.D0,-14.D0,0.D0, &
     12.D0,-5.D0,14.D0,6.D0,-1.D0,-12.D0,-8.D0,4.D0,0.D0,10.D0,1.D0, &
     7.D0,10.D0,2.D0,-12.D0,10.D0,-1.D0,-1.D0,4.D0,1.D0,-2.D0,-3.D0, &
     -3.D0,2.D0,-5.D0,-2.D0,5.D0,4.D0,1.D0,0.D0,3.D0,-1.D0,39*0.D0/
!
     DATA H75/0.D0,0.D0,5675.D0,0.D0,-2067.D0,-68.D0,0.D0,-333.D0, &
     262.D0,-223.D0,0.D0,191.D0,-265.D0,39.D0,-288.D0,0.D0,31.D0, &
     148.D0,-152.D0,-83.D0,88.D0,0.D0,-13.D0,99.D0,75.D0,-41.D0,-4.D0, &
     11.D0,0.D0,-77.D0,-26.D0,-5.D0,10.D0,22.D0,-23.D0,-12.D0,0.D0, &
     6.D0,-16.D0,4.D0,-19.D0,6.D0,18.D0,-10.D0,-17.D0,0.D0,-21.D0, &
     16.D0,7.D0,-4.D0,-5.D0,10.D0,11.D0,-3.D0,1.D0,0.D0,1.D0,1.D0,3.D0, &
     4.D0,-4.D0,-1.D0,-1.D0,3.D0,1.D0,-5.D0,39*0.D0/
!
     DATA G80/0.D0,-29992.D0,-1956.D0,-1997.D0,3027.D0,1663.D0,1281.D0, &
     -2180.D0,1251.D0,833.D0,938.D0,782.D0,398.D0,-419.D0,199.D0, &
     -218.D0,357.D0,261.D0,-74.D0,-162.D0,-48.D0,48.D0,66.D0,42.D0, &
     -192.D0,4.D0,14.D0,-108.D0,72.D0,-59.D0,2.D0,21.D0,-12.D0,1.D0, &
     11.D0,-2.D0,18.D0,6.D0,0.D0,-11.D0,-7.D0,4.D0,3.D0,6.D0,-1.D0, &
     5.D0,10.D0,1.D0,-12.D0,9.D0,-3.D0,-1.D0,7.D0,2.D0,-5.D0,-4.D0, &
     -4.D0,2.D0,-5.D0,-2.D0,5.D0,3.D0,1.D0,2.D0,3.D0,0.D0,39*0.D0/
!
     DATA H80/0.D0,0.D0,5604.D0,0.D0,-2129.D0,-200.D0,0.D0,-336.D0, &
     271.D0,-252.D0,0.D0,212.D0,-257.D0,53.D0,-297.D0,0.D0,46.D0, &
     150.D0,-151.D0,-78.D0,92.D0,0.D0,-15.D0,93.D0,71.D0,-43.D0,-2.D0, &
     17.D0,0.D0,-82.D0,-27.D0,-5.D0,16.D0,18.D0,-23.D0,-10.D0,0.D0, &
     7.D0,-18.D0,4.D0,-22.D0,9.D0,16.D0,-13.D0,-15.D0,0.D0,-21.D0, &
     16.D0,9.D0,-5.D0,-6.D0,9.D0,10.D0,-6.D0,2.D0,0.D0,1.D0,0.D0,3.D0, &
     6.D0,-4.D0,0.D0,-1.D0,4.D0,0.D0,-6.D0,39*0.D0/
!
     DATA G85/0.D0,-29873.D0,-1905.D0,-2072.D0,3044.D0,1687.D0,1296.D0, &
     -2208.D0,1247.D0,829.D0,936.D0,780.D0,361.D0,-424.D0,170.D0, &
     -214.D0,355.D0,253.D0,-93.D0,-164.D0,-46.D0,53.D0,65.D0,51.D0, &
     -185.D0,4.D0,16.D0,-102.D0,74.D0,-62.D0,3.D0,24.D0,-6.D0,4.D0, &
     10.D0,0.D0,21.D0,6.D0,0.D0,-11.D0,-9.D0,4.D0,4.D0,4.D0,-4.D0,5.D0, &
     10.D0,1.D0,-12.D0,9.D0,-3.D0,-1.D0,7.D0,1.D0,-5.D0,-4.D0,-4.D0, &
     3.D0,-5.D0,-2.D0,5.D0,3.D0,1.D0,2.D0,3.D0,0.D0,39*0.D0/
!
     DATA H85/0.D0,0.D0,5500.D0,0.D0,-2197.D0,-306.D0,0.D0,-310.D0, &
     284.D0,-297.D0,0.D0,232.D0,-249.D0,69.D0,-297.D0,0.D0,47.D0, &
     150.D0,-154.D0,-75.D0,95.D0,0.D0,-16.D0,88.D0,69.D0,-48.D0,-1.D0, &
```

```
     21.D0,0.D0,-83.D0,-27.D0,-2.D0,20.D0,17.D0,-23.D0,-7.D0,0.D0,8.D0, &
     -19.D0,5.D0,-23.D0,11.D0,14.D0,-15.D0,-11.D0,0.D0,-21.D0,15.D0, &
     9.D0,-6.D0,-6.D0,9.D0,9.D0,-7.D0,2.D0,0.D0,1.D0,0.D0,3.D0,6.D0, &
     -4.D0,0.D0,-1.D0,4.D0,0.D0,-6.D0,39*0.D0/
!
     DATA G90/0.D0,-29775.D0,-1848.D0,-2131.D0,3059.D0,1686.D0,1314.D0, &
        -2239.D0,  1248.D0,  802.D0,  939.D0, 780.D0, 325.D0,-423.D0, &
          141.D0,  -214.D0,  353.D0,  245.D0,-109.D0,-165.D0, -36.D0, &
           61.D0,    65.D0,   59.D0, -178.D0,    3.D0,  18.D0, -96.D0, &
           77.D0,   -64.D0,    2.D0,   26.D0,  -1.D0,    5.D0,   9.D0, &
            0.D0,    23.D0,    5.D0,   -1.D0, -10.D0, -12.D0,   3.D0, &
            4.D0,     2.D0,   -6.D0,    4.D0,   9.D0,    1.D0, -12.D0, &
            9.D0,    -4.D0,   -2.D0,    7.D0,   1.D0,  -6.D0,  -3.D0, &
           -4.D0,     2.D0,   -5.D0,   -2.D0,   4.D0,    3.D0,   1.D0, &
            3.D0,     3.D0,    0.D0,39*0.D0/

     DATA H90/0.D0,  0.D0,5406.D0,   0.D0,-2279.D0,-373.D0,  0.D0, &
        -284.D0,293.D0,-352.D0,   0.D0,  247.D0,-240.D0, 84.D0, &
        -299.D0,  0.D0,  46.D0, 154.D0, -153.D0, -69.D0, 97.D0, &
            0.D0,-16.D0,  82.D0,  69.D0,  -52.D0,   1.D0, 24.D0, &
            0.D0,-80.D0, -26.D0,   0.D0,   21.D0,  17.D0,-23.D0, &
           -4.D0,  0.D0,  10.D0, -19.D0,    6.D0, -22.D0, 12.D0, &
           12.D0,-16.D0, -10.D0,   0.D0,  -20.D0,  15.D0, 11.D0, &
           -7.D0, -7.D0,   9.D0,   8.D0,   -7.D0,   2.D0,  0.D0, &
            2.D0,  1.D0,   3.D0,   6.D0,   -4.D0,   0.D0, -2.D0, &
            3.D0, -1.D0,  -6.D0,39*0.D0/

     DATA G95/0.D0,-29692.D0,-1784.D0,-2200.D0,3070.D0,1681.D0,1335.D0, &
        -2267.D0,  1249.D0,  759.D0,  940.D0, 780.D0, 290.D0,-418.D0, &
          122.D0,  -214.D0,  352.D0,  235.D0,-118.D0,-166.D0, -17.D0, &
           68.D0,    67.D0,   68.D0, -170.D0,  -1.D0,  19.D0, -93.D0, &
           77.D0,   -72.D0,    1.D0,   28.D0,   5.D0,    4.D0,   8.D0, &
           -2.D0,    25.D0,    6.D0,   -6.D0,  -9.D0, -14.D0,   9.D0, &
            6.D0,    -5.D0,   -7.D0,    4.D0,   9.D0,    3.D0, -10.D0, &
            8.D0,    -8.D0,   -1.D0,   10.D0,  -2.D0,  -8.D0,  -3.D0, &
           -6.D0,     2.D0,   -4.D0,   -1.D0,   4.D0,    2.D0,   2.D0, &
            5.D0,     1.D0,    0.D0,  39*0.D0/

     DATA H95/0.D0,  0.D0,5306.D0,  0.D0,-2366.D0,-413.D0,  0.D0, &
        -262.D0,302.D0,-427.D0,  0.D0,  262.D0,-236.D0, 97.D0, &
        -306.D0,  0.D0,  46.D0,165.D0, -143.D0, -55.D0,107.D0, &
            0.D0,-17.D0,  72.D0, 67.D0,  -58.D0,   1.D0, 36.D0, &
            0.D0,-69.D0, -25.D0,  4.D0,   24.D0,  17.D0,-24.D0, &
           -6.D0,  0.D0,  11.D0,-21.D0,    8.D0, -23.D0, 15.D0, &
           11.D0,-16.D0,  -4.D0,  0.D0,  -20.D0,  15.D0, 12.D0, &
           -6.D0, -8.D0,   8.D0,  5.D0,   -8.D0,   3.D0,  0.D0, &
```

```
        1.D0,  0.D0,   4.D0,  5.D0,   -5.D0,  -1.D0, -2.D0, &
        1.D0, -2.D0,  -7.D0,39*0.D0/

  DATA G00/0.D0,-29619.4D0,-1728.2D0,-2267.7D0,3068.4D0,1670.9D0, &
      1339.6D0,  -2288.D0, 1252.1D0,  714.5D0, 932.3D0, 786.8D0, &
       250.D0,   -403.D0,  111.3D0, -218.8D0, 351.4D0, 222.3D0, &
     -130.4D0,  -168.6D0,  -12.9D0,   72.3D0,  68.2D0,  74.2D0, &
     -160.9D0,    -5.9D0,   16.9D0,  -90.4D0,  79.0D0, -74.0D0, &
        0.D0,    33.3D0,    9.1D0,    6.9D0,   7.3D0,  -1.2D0, &
       24.4D0,     6.6D0,   -9.2D0,   -7.9D0, -16.6D0,   9.1D0, &
        7.0D0,    -7.9D0,    -7.D0,     5.D0,   9.4D0,    3.D0, &
      - 8.4D0,     6.3D0,   -8.9D0,   -1.5D0,   9.3D0,  -4.3D0, &
       -8.2D0,    -2.6D0,    -6.D0,    1.7D0,  -3.1D0,  -0.5D0, &
        3.7D0,      1.D0,     2.D0,    4.2D0,   0.3D0,  -1.1D0, &
        2.7D0,    -1.7D0,   -1.9D0,    1.5D0,  -0.1D0,   0.1D0, &
       -0.7D0,     0.7D0,    1.7D0,    0.1D0,   1.2D0,   4.0D0, &
       -2.2D0,    -0.3D0,    0.2D0,    0.9D0,  -0.2D0,   0.9D0, &
       -0.5D0,     0.3D0,   -0.3D0,   -0.4D0,  -0.1D0,  -0.2D0, &
       -0.4D0,    -0.2D0,   -0.9D0,    0.3D0,   0.1D0,  -0.4D0, &
        1.3D0,    -0.4D0,    0.7D0,   -0.4D0,   0.3D0,  -0.1D0, &
        0.4D0,      0.D0,    0.1D0/


  DATA H00/0.D0,   0.D0,5186.1D0,   0.D0,-2481.6D0,-458.0D0,   0.D0, &
     -227.6D0,293.4D0,-491.1D0,   0.D0,  272.6D0,-231.9D0,119.8D0, &
     -303.8D0,   0.D0,  43.8D0,171.9D0, -133.1D0, -39.3D0,106.3D0, &
        0.D0,-17.4D0,  63.7D0, 65.1D0,  -61.2D0,   0.7D0, 43.8D0, &
        0.D0,-64.6D0, -24.2D0,  6.2D0,    24.D0,  14.8D0,-25.4D0, &
       -5.8D0,  0.0D0,  11.9D0,-21.5D0,    8.5D0, -21.5D0, 15.5D0, &
        8.9D0,-14.9D0,  -2.1D0,  0.0D0,  -19.7D0,  13.4D0, 12.5D0, &
       -6.2D0, -8.4D0,   8.4D0,  3.8D0,   -8.2D0,   4.8D0,  0.0D0, &
        1.7D0,  0.0D0,   4.0D0,  4.9D0,   -5.9D0,  -1.2D0, -2.9D0, &
        0.2D0, -2.2D0,  -7.4D0,  0.0D0,    0.1D0,   1.3D0, -0.9D0, &
       -2.6D0,  0.9D0,  -0.7D0, -2.8D0,   -0.9D0,  -1.2D0, -1.9D0, &
       -0.9D0,  0.0D0,  -0.4D0,  0.3D0,    2.5D0,  -2.6D0,  0.7D0, &
        0.3D0,  0.0D0,   0.0D0,  0.3D0,   -0.9D0,  -0.4D0,  0.8D0, &
        0.0D0, -0.9D0,   0.2D0,  1.8D0,   -0.4D0,  -1.0D0, -0.1D0, &
        0.7D0,  0.3D0,   0.6D0,  0.3D0,   -0.2D0,  -0.5D0, -0.9D0/


  DATA G05/0.D0,-29556.8D0,-1671.8D0,-2340.5D0, 3047.D0,1656.9D0, &
     1335.7D0, -2305.3D0, 1246.8D0,  674.4D0, 919.8D0, 798.2D0, &
      211.5D0,  -379.5D0,  100.2D0, -227.6D0, 354.4D0, 208.8D0, &
     -136.6D0,  -168.3D0,  -14.1D0,   72.9D0,  69.6D0,  76.6D0, &
     -151.1D0,   -15.0D0,   14.7D0,  -86.4D0,  79.8D0, -74.4D0, &
       -1.4D0,    38.6D0,   12.3D0,    9.4D0,   5.5D0,   2.0D0, &
```

```
          24.8D0,     7.7D0,  -11.4D0,   -6.8D0, -18.0D0,  10.0D0, &
           9.4D0,   -11.4D0,   -5.0D0,    5.6D0,   9.8D0,   3.6D0, &
          -7.0D0,     5.0D0,  -10.8D0,   -1.3D0,   8.7D0,  -6.7D0, &
          -9.2D0,    -2.2D0,   -6.3D0,    1.6D0,  -2.5D0,  -0.1D0, &
           3.0D0,     0.3D0,    2.1D0,    3.9D0,  -0.1D0,  -2.2D0, &
           2.9D0,    -1.6D0,   -1.7D0,    1.5D0,  -0.2D0,   0.2D0, &
          -0.7D0,     0.5D0,    1.8D0,    0.1D0,   1.0D0,   4.1D0, &
          -2.2D0,    -0.3D0,    0.3D0,    0.9D0,  -0.4D0,   1.0D0, &
          -0.4D0,     0.5D0,   -0.3D0,   -0.4D0,   0.0D0,  -0.4D0, &
           0.0D0,    -0.2D0,   -0.9D0,    0.3D0,   0.3D0,  -0.4D0, &
           1.2D0,    -0.4D0,    0.7D0,   -0.3D0,   0.4D0,  -0.1D0, &
           0.4D0,  -0.1D0,  -0.3D0/

      DATA H05/0.D0,  0.0D0,5080.0D0,  0.0D0,-2594.9D0,-516.7D0,  0.0D0, &
         -200.4D0,269.3D0,-524.5D0,  0.0D0,  281.4D0,-225.8D0,145.7D0, &
         -304.7D0,  0.0D0,  42.7D0,179.8D0, -123.0D0, -19.5D0,103.6D0, &
           0.0D0,-20.2D0,  54.7D0, 63.7D0,  -63.4D0,   0.0D0, 50.3D0, &
           0.0D0,-61.4D0, -22.5D0,  6.9D0,   25.4D0,  10.9D0,-26.4D0, &
          -4.8D0,  0.0D0,  11.2D0,-21.0D0,    9.7D0, -19.8D0, 16.1D0, &
           7.7D0,-12.8D0,  -0.1D0,  0.0D0,  -20.1D0,  12.9D0, 12.7D0, &
          -6.7D0, -8.1D0,   8.1D0,  2.9D0,   -7.9D0,   5.9D0,  0.0D0, &
           2.4D0,  0.2D0,   4.4D0,  4.7D0,   -6.5D0,  -1.0D0, -3.4D0, &
          -0.9D0, -2.3D0,  -8.0D0,  0.0D0,    0.3D0,   1.4D0, -0.7D0, &
          -2.4D0,  0.9D0,  -0.6D0, -2.7D0,   -1.0D0,  -1.5D0, -2.0D0, &
          -1.4D0,  0.0D0,  -0.5D0,  0.3D0,    2.3D0,  -2.7D0,  0.6D0, &
           0.4D0,  0.0D0,   0.0D0,  0.3D0,   -0.8D0,  -0.4D0,  1.0D0, &
           0.0D0, -0.7D0,   0.3D0,  1.7D0,   -0.5D0,  -1.0D0,  0.0D0, &
           0.7D0,  0.2D0,   0.6D0,  0.4D0,   -0.2D0,  -0.5D0, -1.0D0/

      DATA DG05/0.0D0, 8.8D0,10.8D0,-15.0D0,-6.9D0,-1.0D0,-0.3D0, &
           -3.1D0,-0.9D0,-6.8D0, -2.5D0, 2.8D0,-7.1D0, 5.9D0, &
           -3.2D0,-2.6D0, 0.4D0, -3.0D0,-1.2D0, 0.2D0,-0.6D0, &
           -0.8D0, 0.2D0,-0.2D0,  2.1D0,-2.1D0,-0.4D0, 1.3D0, &
           -0.4D0, 0.0D0,-0.2D0,  1.1D0, 0.6D0, 0.4D0,-0.5D0, &
            0.9D0,-0.2D0, 0.2D0, -0.2D0, 0.2D0,-0.2D0, 0.2D0, &
            0.5D0,-0.7D0, 0.5D0/

      DATA DH05/0.0D0, 0.0D0,-21.3D0, 0.0D0,-23.3D0,-14.0D0, 0.0D0, &
            5.4D0,-6.5D0, -2.0D0, 0.0D0,  2.0D0,  1.8D0, 5.6D0, &
            0.0D0, 0.0D0,  0.1D0, 1.8D0,  2.0D0,  4.5D0,-1.0D0, &
            0.0D0,-0.4D0, -1.9D0,-0.4D0, -0.4D0, -0.2D0, 0.9D0, &
            0.0D0, 0.8D0,  0.4D0, 0.1D0,  0.2D0, -0.9D0,-0.3D0, &
            0.3D0, 0.0D0, -0.2D0, 0.2D0,  0.2D0,  0.4D0, 0.2D0, &
           -0.3D0, 0.5D0,  0.4D0/
!
      IF (VGSEY.EQ.0..AND.VGSEZ.EQ.0..AND.ISW.NE.1) THEN
```

```
      PRINT *, ''
      PRINT *, ' RECALC_08: RADIAL SOLAR WIND --> GSW SYSTEM IDENTICAL
HERE'
      PRINT *, ' TO STANDARD GSM (I.E., XGSW AXIS COINCIDES WITH EARTH-SUN
LINE)'
      PRINT *, ''
      ISW=1
      ENDIF

      IF (VGSEY.NE.0.D0.OR.VGSEZ.NE.0.D0.AND.ISW.NE.2) THEN
      PRINT *, ''
      PRINT *, ' WARNING: NON-RADIAL SOLAR WIND FLOW SPECIFIED IN
RECALC_08;'
      PRINT *, ' HENCE XGSW AXIS IS ASSUMED ORIENTED ANTIPARALLEL TO V_SW
VECTOR'
      PRINT *, ''
      ISW=2
      ENDIF
!
      IY=IYEAR
!
!  WE ARE RESTRICTED BY THE INTERVAL 1965-2010, FOR WHICH THE IGRF
COEFFICIENTS
!    ARE KNOWN; IF IYEAR IS OUTSIDE THIS INTERVAL, THEN THE SUBROUTINE
USES THE
!      NEAREST LIMITING VALUE AND PRINTS A WARNING:
!
      IF(IY.LT.1965) THEN
       IY=1965
       WRITE (*,10) IYEAR,IY
      ENDIF

      IF(IY.GT.2010) THEN
       IY=2010
       WRITE (*,10) IYEAR,IY
      ENDIF

!
!  CALCULATE THE ARRAY REC, CONTAINING COEFFICIENTS FOR THE
RECURSION RELATIONS,
!  USED IN THE IGRF SUBROUTINE FOR CALCULATING THE ASSOCIATE
LEGENDRE POLYNOMIALS
!  AND THEIR DERIVATIVES:
!
      DO 20 N=1,14
        N2=2*N-1
```

```
      N2=N2*(N2-2)
      DO 20 M=1,N
        MN=N*(N-1)/2+M
20    REC(MN)=DFLOAT((N-M)*(N+M-2))/DFLOAT(N2)
!
      IF (IY.LT.1970) GOTO 50        !INTERPOLATE BETWEEN 1965 - 1970
      IF (IY.LT.1975) GOTO 60        !INTERPOLATE BETWEEN 1970 - 1975
      IF (IY.LT.1980) GOTO 70        !INTERPOLATE BETWEEN 1975 - 1980
      IF (IY.LT.1985) GOTO 80        !INTERPOLATE BETWEEN 1980 - 1985
      IF (IY.LT.1990) GOTO 90        !INTERPOLATE BETWEEN 1985 - 1990
      IF (IY.LT.1995) GOTO 100       !INTERPOLATE BETWEEN 1990 - 1995
      IF (IY.LT.2000) GOTO 110       !INTERPOLATE BETWEEN 1995 - 2000
      IF (IY.LT.2005) GOTO 120       !INTERPOLATE BETWEEN 2000 - 2005
!
!     EXTRAPOLATE BEYOND 2005:
!
      DT=DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-2005.D0
      DO 40 N=1,105
        G(N)=G05(N)
        H(N)=H05(N)
        IF (N.GT.45) GOTO 40
        G(N)=G(N)+DG05(N)*DT
        H(N)=H(N)+DH05(N)*DT
40    CONTINUE
      GOTO 300
!
!     INTERPOLATE BETWEEEN 1965 - 1970:
!
50    F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1965)/5.D0
      F1=1.D0-F2
      DO 55 N=1,105
        G(N)=G65(N)*F1+G70(N)*F2
55      H(N)=H65(N)*F1+H70(N)*F2
      GOTO 300
!
!     INTERPOLATE BETWEEN 1970 - 1975:
!
60    F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1970)/5.D0
      F1=1.D0-F2
      DO 65 N=1,105
        G(N)=G70(N)*F1+G75(N)*F2
65      H(N)=H70(N)*F1+H75(N)*F2
      GOTO 300
!
!     INTERPOLATE BETWEEN 1975 - 1980:
!
```

```
70   F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1975)/5.D0
     F1=1.D0-F2
     DO 75 N=1,105
        G(N)=G75(N)*F1+G80(N)*F2
75      H(N)=H75(N)*F1+H80(N)*F2
     GOTO 300
!
!      INTERPOLATE BETWEEN 1980 - 1985:
!
80   F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1980)/5.D0
     F1=1.D0-F2
     DO 85 N=1,105
        G(N)=G80(N)*F1+G85(N)*F2
85      H(N)=H80(N)*F1+H85(N)*F2
     GOTO 300
!
!      INTERPOLATE BETWEEN 1985 - 1990:
!
90   F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1985)/5.D0
     F1=1.D0-F2
     DO 95 N=1,105
        G(N)=G85(N)*F1+G90(N)*F2
95      H(N)=H85(N)*F1+H90(N)*F2
     GOTO 300
!
!      INTERPOLATE BETWEEN 1990 - 1995:
!
100  F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1990)/5.D0
     F1=1.D0-F2
     DO 105 N=1,105
        G(N)=G90(N)*F1+G95(N)*F2
105     H(N)=H90(N)*F1+H95(N)*F2
     GOTO 300
!
!      INTERPOLATE BETWEEN 1995 - 2000:
!
110  F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-1995)/5.D0
     F1=1.D0-F2
     DO 115 N=1,105   !  THE 2000 COEFFICIENTS (G00) GO THROUGH THE ORDER 13,
NOT 10
        G(N)=G95(N)*F1+G00(N)*F2
115     H(N)=H95(N)*F1+H00(N)*F2
     GOTO 300
!
!      INTERPOLATE BETWEEN 2000 - 2005:
!
```

```
120   F2=(DFLOAT(IY)+DFLOAT(IDAY-1)/365.25D0-2000)/5.D0
      F1=1.D0-F2
      DO 125 N=1,105
        G(N)=G00(N)*F1+G05(N)*F2
125       H(N)=H00(N)*F1+H05(N)*F2
      GOTO 300
!
!   COEFFICIENTS FOR A GIVEN YEAR HAVE BEEN CALCULATED; NOW MULTIPLY
!   THEM BY SCHMIDT NORMALIZATION FACTORS:
!
300   S=1.D0
      DO 130 N=2,14
        MN=N*(N-1)/2+1
        S=S*DFLOAT(2*N-3)/DFLOAT(N-1)
        G(MN)=G(MN)*S
        H(MN)=H(MN)*S
        P=S
        DO 130 M=2,N
          AA=1.D0
          IF (M.EQ.2) AA=2.D0
          P=P*DSQRT(AA*DFLOAT(N-M+1)/DFLOAT(N+M-2))
          MNN=MN+M-1
          G(MNN)=G(MNN)*P
130         H(MNN)=H(MNN)*P

         G10=-G(2)
         G11= G(3)
         H11= H(3)
!
!  NOW CALCULATE GEO COMPONENTS OF THE UNIT VECTOR EzMAG, PARALLEL
TO GEODIPOLE AXIS:
!   SIN(TETA0)*COS(LAMBDA0), SIN(TETA0)*SIN(LAMBDA0), AND COS(TETA0)
!        ST0 * CL0                ST0 * SL0                CT0
!
      SQ=G11**2+H11**2
      SQQ=DSQRT(SQ)
      SQR=DSQRT(G10**2+SQ)
      SL0=-H11/SQQ
      CL0=-G11/SQQ
      ST0=SQQ/SQR
      CT0=G10/SQR
      STCL=ST0*CL0
      STSL=ST0*SL0
      CTSL=CT0*SL0
      CTCL=CT0*CL0
!
```

```
!  NOW CALCULATE GEI COMPONENTS (S1,S2,S3) OF THE UNIT VECTOR S =
EX_GSE
!    POINTING FROM THE EARTH'S CENTER TO SUN
!
      CALL SUN_08 (IY,IDAY,IHOUR,MIN,ISEC,GST,SLONG,SRASN,SDEC)
!
      S1=DCOS(SRASN)*DCOS(SDEC)
      S2=DSIN(SRASN)*DCOS(SDEC)
      S3=DSIN(SDEC)
!
!  NOW CALCULATE GEI COMPONENTS (DZ1,DZ2,DZ3) OF THE UNIT VECTOR
EZGSE
!  POINTING NORTHWARD AND ORTHOGONAL TO THE ECLIPTIC PLANE, AS
!  (0,-SIN(OBLIQ),COS(OBLIQ)). FOR THE EPOCH 1978, OBLIQ = 23.44214 DEGS.
!  HERE WE USE A MORE ACCURATE TIME-DEPENDENT VALUE, DETERMINED AS:
!
      DJ=DFLOAT(365*(IY-1900)+(IY-1901)/4 +IDAY) &
        -0.5+DFLOAT(IHOUR*3600+MIN*60+ISEC)/86400.D0
      T=DJ/36525.D0
      OBLIQ=(23.45229D0-0.0130125D0*T)/57.2957795D0
      DZ1=0.D0
      DZ2=-DSIN(OBLIQ)
      DZ3=DCOS(OBLIQ)
!
!  NOW WE OBTAIN GEI COMPONENTS OF THE UNIT VECTOR
EYGSE=(DY1,DY2,DY3),
!  COMPLETING THE RIGHT-HANDED SYSTEM. THEY CAN BE FOUND FROM THE
VECTOR
!  PRODUCT EZGSE x EXGSE = (DZ1,DZ2,DZ3) x (S1,S2,S3):
!
      DY1=DZ2*S3-DZ3*S2
      DY2=DZ3*S1-DZ1*S3
      DY3=DZ1*S2-DZ2*S1
!
!  NOW LET'S CALCULATE GEI COMPONENTS OF THE UNIT VECTOR X = EXGSW,
DIRECTED ANTIPARALLEL
!  TO THE OBSERVED SOLAR WIND FLOW. FIRST, CALCULATE ITS COMPONENTS
IN GSE:
!
      V=DSQRT(VGSEX**2+VGSEY**2+VGSEZ**2)
      DX1=-VGSEX/V
      DX2=-VGSEY/V
      DX3=-VGSEZ/V
!
!  THEN IN GEI:
!
```

```
      X1=DX1*S1+DX2*DY1+DX3*DZ1
      X2=DX1*S2+DX2*DY2+DX3*DZ2
      X3=DX1*S3+DX2*DY3+DX3*DZ3
!
!  NOW CALCULATE GEI COMPONENTS (DIP1,DIP2,DIP3) OF THE UNIT VECTOR DIP
= EZ_SM = EZ_MAG,
!   ALIGNED WITH THE GEODIPOLE AND POINTING NORTHWARD FROM ECLIPTIC
PLANE:
!
      CGST=DCOS(GST)
      SGST=DSIN(GST)
!
      DIP1=STCL*CGST-STSL*SGST
      DIP2=STCL*SGST+STSL*CGST
      DIP3=CT0
!
!  THIS ALLOWS US TO CALCULATE GEI COMPONENTS OF THE UNIT VECTOR Y =
EYGSW
!   BY TAKING THE VECTOR PRODUCT DIP x X AND NORMALIZING IT TO UNIT
LENGTH:
!
      Y1=DIP2*X3-DIP3*X2
      Y2=DIP3*X1-DIP1*X3
      Y3=DIP1*X2-DIP2*X1
      Y=DSQRT(Y1*Y1+Y2*Y2+Y3*Y3)
      Y1=Y1/Y
      Y2=Y2/Y
      Y3=Y3/Y
!
!   AND GEI COMPONENTS OF THE UNIT VECTOR Z = EZGSW = EXGSW x EYGSW =
X x Y:
!
      Z1=X2*Y3-X3*Y2
      Z2=X3*Y1-X1*Y3
      Z3=X1*Y2-X2*Y1
!
!   ELEMENTS OF THE MATRIX GSE TO GSW ARE THE SCALAR PRODUCTS:
!
!  E11=(EXGSE,EXGSW)  E12=(EXGSE,EYGSW)  E13=(EXGSE,EZGSW)
!  E21=(EYGSE,EXGSW)  E22=(EYGSE,EYGSW)  E23=(EYGSE,EZGSW)
!  E31=(EZGSE,EXGSW)  E32=(EZGSE,EYGSW)  E33=(EZGSE,EZGSW)
!
      E11= S1*X1 +S2*X2 +S3*X3
      E12= S1*Y1 +S2*Y2 +S3*Y3
      E13= S1*Z1 +S2*Z2 +S3*Z3
      E21=DY1*X1+DY2*X2+DY3*X3
```

```
      E22=DY1*Y1+DY2*Y2+DY3*Y3
      E23=DY1*Z1+DY2*Z2+DY3*Z3
      E31=DZ1*X1+DZ2*X2+DZ3*X3
      E32=DZ1*Y1+DZ2*Y2+DZ3*Y3
      E33=DZ1*Z1+DZ2*Z2+DZ3*Z3
!
!   GEODIPOLE TILT ANGLE IN THE GSW SYSTEM: PSI=ARCSIN(DIP,EXGSW)
!
      SPS=DIP1*X1+DIP2*X2+DIP3*X3
      CPS=DSQRT(1.D0-SPS**2)
      PSI=DASIN(SPS)
!
!   ELEMENTS OF THE MATRIX GEO TO GSW ARE THE SCALAR PRODUCTS:
!
!   A11=(EXGEO,EXGSW), A12=(EYGEO,EXGSW), A13=(EZGEO,EXGSW),
!   A21=(EXGEO,EYGSW), A22=(EYGEO,EYGSW), A23=(EZGEO,EYGSW),
!   A31=(EXGEO,EZGSW), A32=(EYGEO,EZGSW), A33=(EZGEO,EZGSW),
!
!   ALL THE UNIT VECTORS IN BRACKETS ARE ALREADY DEFINED IN GEI:
!
!  EXGEO=(CGST,SGST,0), EYGEO=(-SGST,CGST,0), EZGEO=(0,0,1)
!  EXGSW=(X1,X2,X3),  EYGSW=(Y1,Y2,Y3),   EZGSW=(Z1,Z2,Z3)
!                                                      AND  THEREFORE:
!
      A11=X1*CGST+X2*SGST
      A12=-X1*SGST+X2*CGST
      A13=X3
      A21=Y1*CGST+Y2*SGST
      A22=-Y1*SGST+Y2*CGST
      A23=Y3
      A31=Z1*CGST+Z2*SGST
      A32=-Z1*SGST+Z2*CGST
      A33=Z3
!
!  NOW CALCULATE ELEMENTS OF THE MATRIX MAG TO SM (ONE ROTATION
ABOUT THE GEODIPOLE AXIS);
!   THEY ARE FOUND AS THE SCALAR PRODUCTS:
CFI=GM22=(EYSM,EYMAG)=(EYGSW,EYMAG),
!                                   SFI=GM23=(EYSM,EXMAG)=(EYGSW,EXMAG),
!    DERIVED AS FOLLOWS:
!
! IN GEO, THE VECTORS EXMAG AND EYMAG HAVE THE COMPONENTS
(CT0*CL0,CT0*SL0,-ST0)
!  AND (-SL0,CL0,0), RESPECTIVELY.    HENCE, IN GEI THEIR COMPONENTS ARE:
!  EXMAG:    CT0*CL0*COS(GST)-CT0*SL0*SIN(GST)
!           CT0*CL0*SIN(GST)+CT0*SL0*COS(GST)
```

```
!           -ST0
!  EYMAG:    -SL0*COS(GST)-CL0*SIN(GST)
!           -SL0*SIN(GST)+CL0*COS(GST)
!            0
!  NOW, NOTE THAT GEI COMPONENTS OF EYSM=EYGSW WERE FOUND ABOVE AS
Y1, Y2, AND Y3,
!  AND WE ONLY HAVE TO COMBINE THESE QUANTITIES INTO SCALAR
PRODUCTS:
!
      EXMAGX=CT0*(CL0*CGST-SL0*SGST)
      EXMAGY=CT0*(CL0*SGST+SL0*CGST)
      EXMAGZ=-ST0
      EYMAGX=-(SL0*CGST+CL0*SGST)
      EYMAGY=-(SL0*SGST-CL0*CGST)
      CFI=Y1*EYMAGX+Y2*EYMAGY
      SFI=Y1*EXMAGX+Y2*EXMAGY+Y3*EXMAGZ
!
 10   FORMAT(//1X, &
      '**** RECALC_08 WARNS: YEAR IS OUT OF INTERVAL 1965-2010: IYEAR=', &
      I4,/,6X,'CALCULATIONS WILL BE DONE FOR IYEAR=',I4,/)
      RETURN
      END SUBROUTINE RECALC_08
!
!=====================================================================
==============

      SUBROUTINE GSWGSE_08 (XGSW,YGSW,ZGSW,XGSE,YGSE,ZGSE,J)
!
!  THIS SUBROUTINE TRANSFORMS COMPONENTS OF ANY VECTOR BETWEEN
THE STANDARD GSE                    9.9.9.9.9.9.9.9
!  COORDINATE SYSTEM AND THE GEOCENTRIC SOLAR-WIND (GSW, aka GSWM),
DEFINED AS FOLLOWS
!  (HONES ET AL., PLANET.SPACE SCI., V.34, P.889, 1986; TSYGANENKO ET AL.,
JGRA,
!  V.103(A4), P.6827, 1998):
!
!  IN THE GSW SYSTEM, X AXIS IS ANTIPARALLEL TO THE OBSERVED DIRECTION
OF THE SOLAR WIND FLOW.
!  TWO OTHER AXES, Y AND Z, ARE DEFINED IN THE SAME WAY AS FOR THE
STANDARD GSM, THAT IS,
!  Z AXIS ORTHOGONAL TO X AXIS, POINTS NORTHWARD, AND LIES IN THE
PLANE DEFINED BY THE X-
!  AND GEODIPOLE AXIS. THE Y AXIS COMPLETES THE RIGHT-HANDED SYSTEM.
!
!  THE GSW SYSTEM BECOMES IDENTICAL TO THE STANDARD GSM IN THE CASE
OF
```

```
!   A STRICTLY RADIAL SOLAR WIND FLOW.
!
!  AUTHOR:  N. A. TSYGANENKO
!  ADDED TO 2008 VERSION OF GEOPACK: JAN 27, 2008.
!  LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!                   J>0                         J<0
!-----INPUT:   J,XGSW,YGSW,ZGSW          J,XGSE,YGSE,ZGSE
!-----OUTPUT:    XGSE,YGSE,ZGSE            XGSW,YGSW,ZGSW
!
!  IMPORTANT THINGS TO REMEMBER:
!
!   (1) BEFORE CALLING GSWGSE_08, BE SURE TO INVOKE SUBROUTINE
RECALC_08, IN ORDER
!       TO DEFINE ALL NECESSARY ELEMENTS OF TRANSFORMATION MATRICES
!
!   (2) IN THE ABSENCE OF INFORMATION ON THE SOLAR WIND DIRECTION, E.G.,
WITH ONLY SCALAR
!       SPEED V KNOWN, THIS SUBROUTINE CAN BE USED TO CONVERT VECTORS
TO ABERRATED
!       COORDINATE SYSTEM, TAKING INTO ACCOUNT EARTH'S ORBITAL SPEED OF
29 KM/S.
!       TO DO THAT, SPECIFY THE LAST 3 PARAMETERS IN RECALC_08 AS FOLLOWS:
!       VGSEX=-V, VGSEY=29.0, VGSEZ=0.0.
!
!       IT SHOULD ALSO BE KEPT IN MIND THAT IN SOME SOLAR WIND DATABASES
THE ABERRATION
!       EFFECT HAS ALREADY BEEN TAKEN INTO ACCOUNT BY SUBTRACTING 29
KM/S FROM VYGSE;
!       IN THAT CASE, THE ORIGINAL VYGSE VALUES SHOULD BE RESTORED BY
ADDING BACK THE
!       29 KM/S CORRECTION. WHETHER OR NOT TO DO THAT, MUST BE VERIFIED
WITH THE DATA
!       ORIGINATOR (OR CAN BE DETERMINED BY CALCULATING THE AVERAGE
VGSEY OVER
!       A SUFFICIENTLY LONG TIME INTERVAL)
!
!   (3) IF NO INFORMATION IS AVAILABLE ON THE SOLAR WIND SPEED, THEN SET
VGSEX=-400.0
!       AND  VGSEY=VGSEZ=0. IN THAT CASE, THE GSW COORDINATE SYSTEM
BECOMES
!       IDENTICAL TO THE STANDARD ONE.
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ AAA(25),E11,E21,E31,E12,E22,E32,E13,E23,E33
!
!  DIRECT TRANSFORMATION:
```

```
!
     IF (J.GT.0) THEN
       XGSE=XGSW*E11+YGSW*E12+ZGSW*E13
       YGSE=XGSW*E21+YGSW*E22+ZGSW*E23
       ZGSE=XGSW*E31+YGSW*E32+ZGSW*E33
     ENDIF
!
!   INVERSE TRANSFORMATION: CARRIED OUT USING THE TRANSPOSED MATRIX:
!
     IF (J.LT.0) THEN
       XGSW=XGSE*E11+YGSE*E21+ZGSE*E31
       YGSW=XGSE*E12+YGSE*E22+ZGSE*E32
       ZGSW=XGSE*E13+YGSE*E23+ZGSE*E33
     ENDIF
!
     RETURN
     END SUBROUTINE GSWGSE_08
!
!==================================================================
====================
!
     SUBROUTINE GEOMAG_08 (XGEO,YGEO,ZGEO,XMAG,YMAG,ZMAG,J)
!
!    CONVERTS GEOGRAPHIC (GEO) TO DIPOLE (MAG) COORDINATES OR VICE
VERSA.                          10.10.10.10.10.
!
!                   J>0                       J<0
!-----INPUT:  J,XGEO,YGEO,ZGEO           J,XMAG,YMAG,ZMAG
!-----OUTPUT:    XMAG,YMAG,ZMAG           XGEO,YGEO,ZGEO
!
!  ATTENTION:  SUBROUTINE  RECALC_08  MUST BE INVOKED BEFORE
GEOMAG_08 IN TWO CASES:
!     /A/  BEFORE THE FIRST TRANSFORMATION OF COORDINATES
!     /B/  IF THE VALUES OF IYEAR AND/OR IDAY HAVE BEEN CHANGED
!
!  NO INFORMATION IS REQUIRED HERE ON THE SOLAR WIND VELOCITY, SO ONE
!  CAN SET VGSEX=-400.0, VGSEY=0.0, VGSEZ=0.0 IN RECALC_08.
!
!   LAST MOFIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!   AUTHOR:  N. A. TSYGANENKO
!
     IMPLICIT REAL*8 (A-H,O-Z)
     COMMON /GEOPACK1/ ST0,CT0,SL0,CL0,CTCL,STCL,CTSL,STSL,AB(26)

     IF(J.GT.0) THEN
```

```
       XMAG=XGEO*CTCL+YGEO*CTSL-ZGEO*ST0
       YMAG=YGEO*CL0-XGEO*SL0
       ZMAG=XGEO*STCL+YGEO*STSL+ZGEO*CT0
      ELSE
       XGEO=XMAG*CTCL-YMAG*SL0+ZMAG*STCL
       YGEO=XMAG*CTSL+YMAG*CL0+ZMAG*STSL
       ZGEO=ZMAG*CT0-XMAG*ST0
      ENDIF

      RETURN
      END SUBROUTINE GEOMAG_08
!
!==================================================================================
=====================
!
      SUBROUTINE GEIGEO_08 (XGEI,YGEI,ZGEI,XGEO,YGEO,ZGEO,J)
!
!   CONVERTS EQUATORIAL INERTIAL (GEI) TO GEOGRAPHICAL (GEO) COORDS
11.11.11.11.11.
!   OR VICE VERSA.
!                  J>0               J<0
!----INPUT:  J,XGEI,YGEI,ZGEI    J,XGEO,YGEO,ZGEO
!----OUTPUT:   XGEO,YGEO,ZGEO      XGEI,YGEI,ZGEI
!
!  ATTENTION:  SUBROUTINE  RECALC_08  MUST BE INVOKED BEFORE GEIGEO_08
IN TWO CASES:
!     /A/  BEFORE THE FIRST TRANSFORMATION OF COORDINATES
!     /B/  IF THE CURRENT VALUES OF IYEAR,IDAY,IHOUR,MIN,ISEC HAVE BEEN
CHANGED
!
!  NO INFORMATION IS REQUIRED HERE ON THE SOLAR WIND VELOCITY, SO ONE
!  CAN SET VGSEX=-400.0, VGSEY=0.0, VGSEZ=0.0 IN RECALC_08.
!
!     LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)

!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ A(13),CGST,SGST,B(19)
!
      IF(J.GT.0) THEN
       XGEO=XGEI*CGST+YGEI*SGST
       YGEO=YGEI*CGST-XGEI*SGST
       ZGEO=ZGEI
      ELSE
       XGEI=XGEO*CGST-YGEO*SGST
```

```
       YGEI=YGEO*CGST+XGEO*SGST
       ZGEI=ZGEO
      ENDIF

      RETURN
      END SUBROUTINE GEIGEO_08
!
!=====================================================================
===================
!
      SUBROUTINE MAGSM_08 (XMAG,YMAG,ZMAG,XSM,YSM,ZSM,J)
!
!  CONVERTS DIPOLE (MAG) TO SOLAR MAGNETIC (SM) COORDINATES OR VICE
VERSA                          12.12.12.12.12.
!
!                    J>0              J<0
!----INPUT: J,XMAG,YMAG,ZMAG     J,XSM,YSM,ZSM
!----OUTPUT:    XSM,YSM,ZSM       XMAG,YMAG,ZMAG
!
!  ATTENTION:  SUBROUTINE  RECALC_08  MUST BE INVOKED BEFORE MAGSM_08
IN THREE CASES:
!     /A/  BEFORE THE FIRST TRANSFORMATION OF COORDINATES, OR
!     /B/  IF THE VALUES OF IYEAR,IDAY,IHOUR,MIN,ISEC HAVE CHANGED, AND/OR
!     /C/  IF THE VALUES OF COMPONENTS OF THE SOLAR WIND FLOW VELOCITY
HAVE CHANGED
!
!    IMPORTANT NOTE:
!
!        A NON-STANDARD DEFINITION IS IMPLIED HERE FOR THE SOLAR MAGNETIC
COORDINATE
!        SYSTEM:  IT IS ASSUMED THAT THE XSM AXIS LIES IN THE PLANE DEFINED
BY THE
!        GEODIPOLE AXIS AND THE OBSERVED VECTOR OF THE SOLAR WIND FLOW
(RATHER THAN
!        THE EARTH-SUN LINE).  IN ORDER TO CONVERT MAG COORDINATES TO AND
FROM THE
!        STANDARD SM COORDINATES, INVOKE RECALC_08 WITH VGSEX=-400.0,
VGSEY=0.0, VGSEZ=0.0
!
!     LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ A(8),SFI,CFI,B(24)
!
```

```
      IF (J.GT.0) THEN
       XSM=XMAG*CFI-YMAG*SFI
       YSM=XMAG*SFI+YMAG*CFI
       ZSM=ZMAG
      ELSE
       XMAG=XSM*CFI+YSM*SFI
       YMAG=YSM*CFI-XSM*SFI
       ZMAG=ZSM
      ENDIF

      RETURN
      END SUBROUTINE MAGSM_08
!
!===================================================================
=================
!
       SUBROUTINE SMGSW_08 (XSM,YSM,ZSM,XGSW,YGSW,ZGSW,J)
!
!  CONVERTS SOLAR MAGNETIC (SM) TO GEOCENTRIC SOLAR-WIND (GSW)
COORDINATES OR VICE VERSA.          13.13.13.13.13.
!
!                   J>0                   J<0
!-----INPUT: J,XSM,YSM,ZSM        J,XGSW,YGSW,ZGSW
!----OUTPUT:  XGSW,YGSW,ZGSW       XSM,YSM,ZSM
!
!  ATTENTION:  SUBROUTINE RECALC_08 MUST BE INVOKED BEFORE SMGSW_08
IN THREE CASES:
!     /A/  BEFORE THE FIRST TRANSFORMATION OF COORDINATES
!     /B/  IF THE VALUES OF IYEAR,IDAY,IHOUR,MIN,ISEC HAVE BEEN CHANGED
!     /C/  IF THE VALUES OF COMPONENTS OF THE SOLAR WIND FLOW VELOCITY
HAVE CHANGED
!
!    IMPORTANT NOTE:
!
!        A NON-STANDARD DEFINITION IS IMPLIED HERE FOR THE SOLAR MAGNETIC
(SM) COORDINATE
!        SYSTEM:  IT IS ASSUMED THAT THE XSM AXIS LIES IN THE PLANE DEFINED
BY THE
!        GEODIPOLE AXIS AND THE OBSERVED VECTOR OF THE SOLAR WIND FLOW
(RATHER THAN
!        THE EARTH-SUN LINE).  IN ORDER TO CONVERT MAG COORDINATES TO AND
FROM THE
!        STANDARD SM COORDINATES, INVOKE RECALC_08 WITH VGSEX=-400.0,
VGSEY=0.0, VGSEZ=0.0
!
!     LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
```

```
!
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ A(10),SPS,CPS,B(22)

      IF (J.GT.0) THEN
       XGSW=XSM*CPS+ZSM*SPS
       YGSW=YSM
       ZGSW=ZSM*CPS-XSM*SPS
      ELSE
       XSM=XGSW*CPS-ZGSW*SPS
       YSM=YGSW
       ZSM=XGSW*SPS+ZGSW*CPS
      ENDIF

      RETURN
      END SUBROUTINE SMGSW_08
!
!===========================================================================
======================
!
      SUBROUTINE GEOGSW_08 (XGEO,YGEO,ZGEO,XGSW,YGSW,ZGSW,J)
!
! CONVERTS GEOGRAPHIC (GEO) TO GEOCENTRIC SOLAR-WIND (GSW)
COORDINATES OR VICE VERSA.              14.14.14.14.14.
!
!                   J>0                   J<0
!----- INPUT:  J,XGEO,YGEO,ZGEO    J,XGSW,YGSW,ZGSW
!---- OUTPUT:    XGSW,YGSW,ZGSW      XGEO,YGEO,ZGEO
!
!  ATTENTION:  SUBROUTINE  RECALC_08  MUST BE INVOKED BEFORE
GEOGSW_08 IN THREE CASES:
!     /A/  BEFORE THE FIRST TRANSFORMATION OF COORDINATES, OR
!     /B/  IF THE VALUES OF IYEAR,IDAY,IHOUR,MIN,ISEC  HAVE CHANGED, AND/OR
!     /C/  IF THE VALUES OF COMPONENTS OF THE SOLAR WIND FLOW VELOCITY
HAVE CHANGED
!
!  NOTE: THIS SUBROUTINE CONVERTS GEO VECTORS TO AND FROM THE SOLAR-
WIND GSW COORDINATE
!       SYSTEM, TAKING INTO ACCOUNT POSSIBLE DEFLECTIONS OF THE SOLAR
WIND DIRECTION FROM
!       STRICTLY RADIAL.  BEFORE CONVERTING TO/FROM STANDARD GSM
COORDINATES, INVOKE RECALC_08
!       WITH VGSEX=-400.0 and VGSEY=0.0, VGSEZ=0.0
!
```

```
!     LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /GEOPACK1/ AA(16),A11,A21,A31,A12,A22,A32,A13,A23,A33,B(9)
!
      IF (J.GT.0) THEN
       XGSW=A11*XGEO+A12*YGEO+A13*ZGEO
       YGSW=A21*XGEO+A22*YGEO+A23*ZGEO
       ZGSW=A31*XGEO+A32*YGEO+A33*ZGEO
      ELSE
       XGEO=A11*XGSW+A21*YGSW+A31*ZGSW
       YGEO=A12*XGSW+A22*YGSW+A32*ZGSW
       ZGEO=A13*XGSW+A23*YGSW+A33*ZGSW
      ENDIF

      RETURN
      END SUBROUTINE GEOGSW_08
!
!==================================================================
=================
!
      SUBROUTINE GEODGEO_08 (H,XMU,R,THETA,J)
!
!  THIS SUBROUTINE (1) CONVERTS VERTICAL LOCAL HEIGHT (ALTITUDE) H AND
GEODETIC                    15.15.15.15.15.
!  LATITUDE XMU INTO GEOCENTRIC COORDINATES R AND THETA (GEOCENTRIC
RADIAL
!  DISTANCE AND COLATITUDE, RESPECTIVELY; ALSO KNOWN AS ECEF
COORDINATES),
!  AS WELL AS (2) PERFORMS THE INVERSE TRANSFORMATION FROM {R,THETA}
TO {H,XMU}.
!
!  THE SUBROUTINE USES WORLD GEODETIC SYSTEM WGS84 PARAMETERS FOR
THE EARTH'S
!  ELLIPSOID. THE ANGULAR QUANTITIES (GEO COLATITUDE THETA AND
GEODETIC LATITUDE
!  XMU) ARE IN RADIANS, AND THE DISTANCES (GEOCENTRIC RADIUS R AND
ALTITUDE H
!  ABOVE THE EARTH'S ELLIPSOID) ARE IN KILOMETERS.
!
!  IF J>0, THE TRANSFORMATION IS MADE FROM GEODETIC TO GEOCENTRIC
COORDINATES
!   USING SIMPLE DIRECT EQUATIONS.
```

```
!  IF J<0, THE INVERSE TRANSFORMATION FROM GEOCENTRIC TO GEODETIC
COORDINATES
!   IS MADE BY MEANS OF A FAST ITERATIVE ALGORITHM.
!
!-------------------------------------------------------------------------------
!                 J>0                    |           J<0
!-----------------------------------------|------------------------------------
!--INPUT:   J       H          XMU       |    J         R          THETA
!         flag  altitude (km)  geodetic     |  flag   geocentric    spherical
!                              latitude     |        distance (km) colatitude
!                              (radians)    |                    (radians)
!-----------------------------------------|------------------------------------
!                                        |
!----OUTPUT:         R           THETA      |          H             XMU
!                geocentric    spherical    |     altitude (km)    geodetic
!                distance (km) colatitude   |                     latitude
!                              (radians)    |                   (radians)
!-------------------------------------------------------------------------------
!
!   AUTHOR:  N. A. TSYGANENKO
!   DATE:    DEC 5, 2007
!   LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA R_EQ, BETA /6378.137D0, 6.73949674228D-3/
!
!  R_EQ is the semi-major axis of the Earth's ellipsoid, and BETA is its
!  second eccentricity squared
!
      DATA TOL /1.D-6/
!
!   Direct transformation (GEOD=>GEO):
!
      IF (J.GT.0) THEN
       COSXMU=DCOS(XMU)
       SINXMU=DSIN(XMU)
       DEN=DSQRT(COSXMU**2+(SINXMU/(1.0D0+BETA))**2)
       COSLAM=COSXMU/DEN
       SINLAM=SINXMU/(DEN*(1.0D0+BETA))
       RS=R_EQ/DSQRT(1.0D0+BETA*SINLAM**2)
       X=RS*COSLAM+H*COSXMU
       Z=RS*SINLAM+H*SINXMU
       R=DSQRT(X**2+Z**2)
       THETA=DACOS(Z/R)
      ENDIF
```

```
!
!   Inverse transformation (GEO=>GEOD):
!
      IF (J.LT.0) THEN
       N=0
       PHI=1.570796327D0-THETA
       PHI1=PHI
  1    SP=DSIN(PHI1)
       ARG=SP*(1.0D0+BETA)/DSQRT(1.0D0+BETA*(2.0D0+BETA)*SP**2)
       XMUS=DASIN(ARG)
       RS=R_EQ/DSQRT(1.0D0+BETA*DSIN(PHI1)**2)
       COSFIMS=DCOS(PHI1-XMUS)
       H=DSQRT((RS*COSFIMS)**2+R**2-RS**2)-RS*COSFIMS
       Z=RS*DSIN(PHI1)+H*DSIN(XMUS)
       X=RS*DCOS(PHI1)+H*DCOS(XMUS)
       RR=DSQRT(X**2+Z**2)
       DPHI=DASIN(Z/RR)-PHI
       PHI1=PHI1-DPHI
       N=N+1
       IF (DABS(DPHI).GT.TOL.AND.N.LT.100) GOTO 1
       XMU=XMUS
      ENDIF

      RETURN
      END SUBROUTINE GEODGEO_08
!
!=====================================================================
=================
!
      SUBROUTINE RHAND_08 (X,Y,Z,R1,R2,R3,IOPT,PARMOD,EXNAME,INNAME)
!
!  CALCULATES THE COMPONENTS OF THE RIGHT HAND SIDE VECTOR IN THE
GEOMAGNETIC FIELD                16.16.16.16.16.
!    LINE EQUATION  (a subsidiary subroutine for the subroutine STEP_08)
!
!     LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PARMOD(10)
!
!     EXNAME AND INNAME ARE NAMES OF SUBROUTINES FOR THE EXTERNAL
AND INTERNAL
!     PARTS OF THE TOTAL FIELD, E.G., T96_01 AND IGRF_GSW_08
!
```

```
      COMMON /GEOPACK1/ A(12),DS3,BB(2),PSI,CC(18)

      CALL EXNAME (IOPT,PARMOD,PSI,X,Y,Z,BXGSW,BYGSW,BZGSW)
      CALL INNAME (X,Y,Z,HXGSW,HYGSW,HZGSW)

      BX=BXGSW+HXGSW
      BY=BYGSW+HYGSW
      BZ=BZGSW+HZGSW
      B=DS3/DSQRT(BX**2+BY**2+BZ**2)
      R1=BX*B
      R2=BY*B
      R3=BZ*B
      RETURN
      END SUBROUTINE RHAND_08
!
!============================================================================
===============
!
      SUBROUTINE
STEP_08(X,Y,Z,DS,DSMAX,ERRIN,IOPT,PARMOD,EXNAME,INNAME)
!
!   RE-CALCULATES THE INPUT VALUES {X,Y,Z} (IN GSW COORDINATES) FOR ANY
POINT ON A FIELD LINE,     17.17.17.17.17
!     BY MAKING A STEP ALONG THAT LINE USING RUNGE-KUTTA-MERSON
ALGORITHM (G.N. Lance, Numerical
!      methods for high-speed computers, Iliffe & Sons, London 1960.)
!   DS IS A PRESCRIBED VALUE OF THE CURRENT STEP SIZE, DSMAX IS ITS UPPER
LIMIT.
!   ERRIN IS A PERMISSIBLE ERROR (ITS OPTIMAL VALUE SPECIFIED IN THE S/R
TRACE_08)
!     IF THE ACTUAL ERROR (ERRCUR) AT THE CURRENT STEP IS LARGER THAN
ERRIN, THE STEP IS REJECTED,
!       AND THE CALCULATION IS REPEATED ANEW WITH HALVED STEPSIZE DS.
!     IF ERRCUR IS SMALLER THAN ERRIN, THE STEP IS ACCEPTED, AND THE
CURRENT VALUE OF DS IS RETAINED
!       FOR THE NEXT STEP.
!     IF ERRCUR IS SMALLER THAN 0.04*ERRIN, THE STEP IS ACCEPTED, AND THE
VALUE OF DS FOR THE NEXT STEP
!       IS INCREASED BY THE FACTOR 1.5, BUT NOT LARGER THAN DSMAX.
!   IOPT IS A FLAG, RESERVED FOR SPECIFYNG A VERSION OF THE EXTERNAL
FIELD MODEL EXNAME.
!   ARRAY PARMOD(10) CONTAINS INPUT PARAMETERS FOR THE MODEL
EXNAME.
!   EXNAME IS THE NAME OF THE SUBROUTINE FOR THE EXTERNAL FIELD
MODEL.
```

```
!   INNAME IS THE NAME OF THE SUBROUTINE FOR THE INTERNAL FIELD MODEL
(EITHER DIP_08 OR IGRF_GSW_08)
!
!   ALL THE ABOVE PARAMETERS ARE INPUT ONES; OUTPUT IS THE
RECALCULATED VALUES OF X,Y,Z
!
!     LAST MODIFICATION:    APRIL 21, 2008
!
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PARMOD(10)
      COMMON /GEOPACK1/ A(12),DS3,B(21)
      EXTERNAL EXNAME,INNAME

  1   DS3=-DS/3.D0
      CALL RHAND_08 (X,Y,Z,R11,R12,R13,IOPT,PARMOD,EXNAME,INNAME)
      CALL RHAND_08 (X+R11,Y+R12,Z+R13,R21,R22,R23,IOPT,PARMOD,EXNAME, &
        INNAME)
      CALL RHAND_08 (X+.5D0*(R11+R21),Y+.5D0*(R12+R22),Z+.5D0* &
       (R13+R23),R31,R32,R33,IOPT,PARMOD,EXNAME,INNAME)
      CALL RHAND_08 (X+.375D0*(R11+3.D0*R31),Y+.375D0*(R12+3.D0*R32 &
       ),Z+.375D0*(R13+3.D0*R33),R41,R42,R43,IOPT,PARMOD,EXNAME,INNAME)
      CALL RHAND_08 (X+1.5D0*(R11-3.D0*R31+4.D0*R41),Y+1.5D0*(R12- &
       3.D0*R32+4.D0*R42),Z+1.5D0*(R13-3.D0*R33+4.D0*R43), &
       R51,R52,R53,IOPT,PARMOD,EXNAME,INNAME)
      ERRCUR=DABS(R11-4.5D0*R31+4.D0*R41-.5D0*R51)+DABS(R12-4.5D0*R32 &
       +4.D0*R42-.5D0*R52)+DABS(R13-4.5D0*R33+4.D0*R43-.5D0*R53)
!
!  READY FOR MAKING THE STEP, BUT CHECK THE ACCURACY; IF INSUFFICIENT,
!   REPEAT THE STEP WITH HALVED STEPSIZE:
!
      IF (ERRCUR.GT.ERRIN) THEN
      DS=DS*.5D0
      GOTO 1
      ENDIF
!
!  ACCURACY IS ACCEPTABLE, BUT CHECK IF THE STEPSIZE IS NOT TOO LARGE;
!    OTHERWISE REPEAT THE STEP WITH DS=DSMAX
!
      IF (DABS(DS).GT.DSMAX) THEN
      DS=DSIGN(DSMAX,DS)
      GOTO 1
      ENDIF
!
!  MAKING THE STEP:
```

```
!
  2   X=X+.5D0*(R11+4.D0*R41+R51)
      Y=Y+.5D0*(R12+4.D0*R42+R52)
      Z=Z+.5D0*(R13+4.D0*R43+R53)
!
!  IF THE ACTUAL ERROR IS TOO SMALL (LESS THAN 4% OF ERRIN) AND DS
SMALLER
!   THAN DSMAX/1.5, THEN WE INCREASE THE STEPSIZE FOR THE NEXT STEP BY
50%
!
      IF(ERRCUR.LT.ERRIN*.04D0.AND.DS.LT.DSMAX/1.5D0) DS=DS*1.5D0
      RETURN
      END SUBROUTINE STEP_08
!
!=====================================================================
==========
!
      SUBROUTINE TRACE_08 (XI,YI,ZI,DIR,DSMAX,ERR,RLIM,R0,IOPT,PARMOD, &
           EXNAME,INNAME,XF,YF,ZF,XX,YY,ZZ,L,LMAX)
!
!  TRACES A FIELD LINE FROM AN ARBITRARY POINT OF SPACE TO THE EARTH'S
18.18.18.18.18.
!  SURFACE OR TO A MODEL LIMITING BOUNDARY.
!
!  THIS SUBROUTINE ALLOWS TWO OPTIONS:
!
!  (1) IF INNAME=IGRF_GSW_08, THEN THE IGRF MODEL WILL BE USED FOR
CALCULATING
!      CONTRIBUTION FROM EARTH'S INTERNAL SOURCES. IN THIS CASE,
SUBROUTINE
!      RECALC_08 MUST BE CALLED BEFORE USING TRACE_08, WITH PROPERLY
SPECIFIED DATE,
!      UNIVERSAL TIME, AND SOLAR WIND VELOCITY COMPONENTS, TO
CALCULATE IN ADVANCE
!      ALL QUANTITIES NEEDED FOR THE MAIN FIELD MODEL AND FOR
TRANSFORMATIONS
!      BETWEEN INVOLVED COORDINATE SYSTEMS.
!
!  (2) IF INNAME=DIP_08, THEN A PURE DIPOLE FIELD WILL BE USED INSTEAD OF
THE IGRF MODEL.
!      IN THIS CASE, THE SUBROUTINE RECALC_08 MUST ALSO BE CALLED BEFORE
TRACE_08.
!      HERE ONE CAN CHOOSE EITHER TO
!      (a) CALCULATE DIPOLE TILT ANGLE BASED ON DATE, TIME, AND SOLAR
WIND DIRECTION,
```

!   OR (b) EXPLICITLY SPECIFY THAT ANGLE, WITHOUT ANY REFERENCE TO DATE/UT/SOLAR WIND.
!   IN THE LAST CASE (b), THE SINE (SPS) AND COSINE (CPS) OF THE DIPOLE TILT
!   ANGLE MUST BE SPECIFIED IN ADVANCE (BUT AFTER HAVING CALLED RECALC_08) AND FORWARDED
!   IN THE COMMON BLOCK /GEOPACK1/ (IN ITS 11th AND 12th ELEMENTS, RESPECTIVELY).
!   IN THIS CASE THE ROLE OF THE SUBROUTINE RECALC_08 IS REDUCED TO ONLY CALCULATING
!   THE COMPONENTS OF THE EARTH'S DIPOLE MOMENT.
!
!------------- INPUT PARAMETERS:
!
!   XI,YI,ZI - GSW COORDS OF THE FIELD LINE STARTING POINT (IN EARTH RADII, 1 RE = 6371.2 km),
!
!   DIR - SIGN OF THE TRACING DIRECTION: IF DIR=1.0 THEN THE TRACING IS MADE ANTIPARALLEL
!   TO THE TOTAL FIELD VECTOR (E.G., FROM NORTHERN TO SOUTHERN CONJUGATE POINT);
!   IF DIR=-1.0 THEN THE TRACING PROCEEDS IN THE OPPOSITE DIRECTION, THAT IS, PARALLEL TO
!   THE TOTAL FIELD VECTOR.
!
!   DSMAX - UPPER LIMIT ON THE STEPSIZE (SETS A DESIRED MAXIMAL SPACING BETWEEN
!   THE FIELD LINE POINTS)
!
!   ERR - PERMISSIBLE STEP ERROR. A REASONABLE ESTIMATE PROVIDING A SUFFICIENT ACCURACY FOR MOST
!   APPLICATIONS IS ERR=0.0001. SMALLER/LARGER VALUES WILL RESULT IN LARGER/SMALLER NUMBER
!   OF STEPS AND, HENCE, OF OUTPUT FIELD LINE POINTS. NOTE THAT USING MUCH SMALLER VALUES
!   OF ERR MAY REQUIRE USING A DOUBLE PRECISION VERSION OF THE ENTIRE PACKAGE.
!
!   R0 - RADIUS OF A SPHERE (IN RE), DEFINING THE INNER BOUNDARY OF THE TRACING REGION
!   (USUALLY, EARTH'S SURFACE OR THE IONOSPHERE, WHERE R0~1.0)
!   IF THE FIELD LINE REACHES THAT SPHERE FROM OUTSIDE, ITS INBOUND TRACING IS
!   TERMINATED AND THE CROSSING POINT COORDINATES XF,YF,ZF ARE CALCULATED.
!

! RLIM - RADIUS OF A SPHERE (IN RE), DEFINING THE OUTER BOUNDARY OF THE TRACING REGION;
! IF THE FIELD LINE REACHES THAT BOUNDARY FROM INSIDE, ITS OUTBOUND TRACING IS
! TERMINATED AND THE CROSSING POINT COORDINATES XF,YF,ZF ARE CALCULATED.
!
! IOPT - A MODEL INDEX; CAN BE USED FOR SPECIFYING A VERSION OF THE EXTERNAL FIELD
! MODEL (E.G., A NUMBER OF THE KP-INDEX INTERVAL). ALTERNATIVELY, ONE CAN USE THE ARRAY
! PARMOD FOR THAT PURPOSE (SEE BELOW); IN THAT CASE IOPT IS JUST A DUMMY PARAMETER.
!
! PARMOD - A 10-ELEMENT ARRAY CONTAINING INPUT PARAMETERS NEEDED FOR A UNIQUE
! SPECIFICATION OF THE EXTERNAL FIELD MODEL. THE CONCRETE MEANING OF THE COMPONENTS
! OF PARMOD DEPENDS ON A SPECIFIC VERSION OF THAT MODEL.
!
! EXNAME - NAME OF A SUBROUTINE PROVIDING COMPONENTS OF THE EXTERNAL MAGNETIC FIELD
! (E.G., T89, OR T96_01, ETC.).
! INNAME - NAME OF A SUBROUTINE PROVIDING COMPONENTS OF THE INTERNAL MAGNETIC FIELD
! (EITHER DIP_08 OR IGRF_GSW_08).
!
! LMAX - MAXIMAL LENGTH OF THE ARRAYS XX,YY,ZZ, IN WHICH COORDINATES OF THE FIELD
! LINE POINTS ARE STORED. LMAX SHOULD BE SET EQUAL TO THE ACTUAL LENGTH OF
! THE ARRAYS, DEFINED IN THE MAIN PROGRAM AS ACTUAL ARGUMENTS OF THIS SUBROUTINE.
!
!-------------- OUTPUT PARAMETERS:
!
! XF,YF,ZF - GSW COORDINATES OF THE ENDPOINT OF THE TRACED FIELD LINE.
! XX,YY,ZZ - ARRAYS OF LENGTH LMAX, CONTAINING COORDINATES OF THE FIELD LINE POINTS.
! L - ACTUAL NUMBER OF FIELD LINE POINTS, GENERATED BY THIS SUBROUTINE.
!
! --
!
! LAST MODIFICATION: MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!

```
!     AUTHOR:  N. A. TSYGANENKO
!
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(LMAX),YY(LMAX),ZZ(LMAX), PARMOD(10)
      COMMON /GEOPACK1/ AA(12),DD,BB(21)
      EXTERNAL EXNAME,INNAME
!
      L=0
      NREV=0
      DD=DIR
!
!  INITIALIZE THE STEP SIZE AND STARTING PONT:
!
      DS=0.5D0*DIR
      X=XI
      Y=YI
      Z=ZI
!
!  here we call RHAND_08 just to find out the sign of the radial component of the field
!   vector, and to determine the initial direction of the tracing (i.e., either away
!   or towards Earth):
!
      CALL RHAND_08 (X,Y,Z,R1,R2,R3,IOPT,PARMOD,EXNAME,INNAME)
      AD=0.01D0
      IF (X*R1+Y*R2+Z*R3.LT.0.D0) AD=-0.01D0
!
!     |AD|=0.01 and its sign follows the rule:
! (1) if DIR=1 (tracing antiparallel to B vector) then the sign of AD is the same as of Br
! (2) if DIR=-1 (tracing parallel to B vector) then the sign of AD is opposite to that of Br
!     AD is defined in order to initialize the value of RR (radial distance at previous step):

      RR=DSQRT(X**2+Y**2+Z**2)+AD
!
  1   L=L+1
      IF(L.GT.LMAX) GOTO 7
      XX(L)=X
      YY(L)=Y
      ZZ(L)=Z
      RYZ=Y**2+Z**2
      R2=X**2+RYZ
      R=DSQRT(R2)
!
!  check if the line hit the outer tracing boundary; if yes, then terminate
!   the tracing (label 8). The outer boundary is assumed reached, when the line
!   crosses any of the 3 surfaces: (1) a sphere R=RLIM, (2) a cylinder of radius 40Re,
!   coaxial with the XGSW axis, (3) the plane X=20Re:
```

```
      IF (R.GT.RLIM.OR.RYZ.GT.1600.D0.OR.X.GT.20.D0) GOTO 8
!
!  check whether or not the inner tracing boundary was crossed from outside,
!  if yes, then calculate the footpoint position by interpolation (go to label 6):
!
      IF (R.LT.R0.AND.RR.GT.R) GOTO 6

!  check if we are moving outward, or R is still larger than 3Re; if yes, proceed further:
!
      IF (R.GE.RR.OR.R.GE.3.D0) GOTO 4
!
!  now we entered inside the sphere R=3: to avoid too large steps (and hence
!  inaccurate interpolated position of the footpoint), enforce the progressively
!  smaller stepsize values as we approach the inner boundary R=R0:
!
      FC=0.2D0
      IF(R-R0.LT.0.05D0) FC=0.05D0
      AL=FC*(R-R0+0.2D0)
      DS=DIR*AL
!
  4   XR=X
      YR=Y
      ZR=Z
!
      DRP=R-RR
      RR=R
!
      CALL STEP_08 (X,Y,Z,DS,DSMAX,ERR,IOPT,PARMOD,EXNAME,INNAME)
!
!  check the total number NREV of changes in the tracing radial direction; (NREV.GT.2) means
!   that the line started making multiple loops, in which case we stop the process:
!
      R=DSQRT(X**2+Y**2+Z**2)
      DR=R-RR
      IF (DRP*DR.LT.0.D0) NREV=NREV+1
      IF (NREV.GT.2) GOTO 8
!
      GOTO 1
!
!  find the footpoint position by interpolating between the current and previous
!   field line points:
!
  6   R1=(R0-R)/(RR-R)
      X=X-(X-XR)*R1
      Y=Y-(Y-YR)*R1
```

```
      Z=Z-(Z-ZR)*R1
      GOTO 8
  7   WRITE (*,10)
      L=LMAX
  8   XF=X
      YF=Y
      ZF=Z
      RETURN
 10   FORMAT(//,1X,'**** COMPUTATIONS IN THE SUBROUTINE TRACE_08 ARE', &
      ' TERMINATED: THE NUMBER OF POINTS EXCEEDED LMAX ****'//)
      END SUBROUTINE TRACE_08
!
!==================================================================
================
!
      SUBROUTINE SHUETAL_MGNP_08(XN_PD,VEL,BZIMF,XGSW,YGSW,ZGSW, &
        XMGNP,YMGNP,ZMGNP,DIST,ID)
!
!  FOR ANY POINT OF SPACE WITH COORDINATES (XGSW,YGSW,ZGSW) AND
SPECIFIED CONDITIONS              19.19.19.19.19.
!  IN THE INCOMING SOLAR WIND, THIS SUBROUTINE:
!
! (1) DETERMINES IF THE POINT (XGSW,YGSW,ZGSW) LIES INSIDE OR OUTSIDE
THE
!      MODEL MAGNETOPAUSE OF SHUE ET AL. (JGR-A, V.103, P. 17691, 1998).
!
! (2) CALCULATES THE GSW POSITION OF A POINT {XMGNP,YMGNP,ZMGNP},
LYING AT THE MODEL
!      MAGNETOPAUSE AND ASYMPTOTICALLY TENDING TO THE NEAREST
BOUNDARY POINT WITH
!      RESPECT TO THE OBSERVATION POINT {XGSW,YGSW,ZGSW}, AS IT
APPROACHES THE MAGNETO-
!      PAUSE.
!
!  INPUT: XN_PD - EITHER SOLAR WIND PROTON NUMBER DENSITY (PER C.C.) (IF
VEL>0) !
!                  OR THE SOLAR WIND RAM PRESSURE IN NANOPASCALS   (IF VEL<0)
!         BZIMF - IMF BZ IN NANOTESLAS  ( input 1 nT for now until we get a better
estimate...this is an approximation
!                                                  after looking at different models)
!               ! 2
!         VEL - EITHER SOLAR WIND VELOCITY (KM/SEC)
!                OR ANY NEGATIVE NUMBER, WHICH INDICATES THAT XN_PD STANDS
!                  FOR THE SOLAR WIND PRESSURE, RATHER THAN FOR THE DENSITY
!
```

```
!       XGSW,YGSW,ZGSW - GSW POSITION OF THE OBSERVATION POINT IN EARTH
RADII
!
!  OUTPUT: XMGNP,YMGNP,ZMGNP - GSW POSITION OF THE BOUNDARY POINT
!        DIST - DISTANCE (IN RE) BETWEEN THE OBSERVATION POINT
(XGSW,YGSW,ZGSW)
!               AND THE MODEL NAGNETOPAUSE
!        ID -  POSITION FLAG:  ID=+1 (-1) MEANS THAT THE OBSERVATION POINT
!        LIES INSIDE (OUTSIDE) OF THE MODEL MAGNETOPAUSE, RESPECTIVELY.
!
!  OTHER SUBROUTINES USED: T96_MGNP_08
!
!        AUTHOR:  N.A. TSYGANENKO,
!        DATE:    APRIL 4, 2003.
!        LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
      IMPLICIT REAL*8 (A-H,O-Z)
!
      IF (VEL.LT.0.) THEN
        P=XN_PD
      ELSE
        P=1.94D-6*XN_PD*VEL**2  ! P IS THE SOLAR WIND DYNAMIC PRESSURE (IN
nPa)
      ENDIF

!
!  DEFINE THE ANGLE PHI, MEASURED DUSKWARD FROM THE NOON-MIDNIGHT
MERIDIAN PLANE;
!  IF THE OBSERVATION POINT LIES ON THE X AXIS, THE ANGLE PHI CANNOT BE
UNIQUELY
!  DEFINED, AND WE SET IT AT ZERO:
!
      IF (YGSW.NE.0.D0.OR.ZGSW.NE.0.D0) THEN
         PHI=DATAN2(YGSW,ZGSW)
      ELSE
         PHI=0.D0
      ENDIF
!
!  FIRST, FIND OUT IF THE OBSERVATION POINT LIES INSIDE THE SHUE ET AL
BDRY
!  AND SET THE VALUE OF THE ID FLAG:
!
      ID=-1
      R0=(10.22D0+1.29D0*DTANH(.184D0*(BZIMF+8.14D0)))*P**(-.15151515D0)
      ALPHA=(0.58D0-0.007D0*BZIMF)*(1.D0+0.024*DLOG(P))
      R=DSQRT(XGSW**2+YGSW**2+ZGSW**2)
```

```
      RM=R0*(2.D0/(1.D0+XGSW/R))**ALPHA
      IF (R.LE.RM) ID=+1
!
!  NOW, FIND THE CORRESPONDING T96 MAGNETOPAUSE POSITION, TO BE USED
AS
!  A STARTING APPROXIMATION IN THE SEARCH OF A CORRESPONDING SHUE ET
AL.
!  BOUNDARY POINT:
!
      CALL T96_MGNP_08(P,-1.D0,XGSW,YGSW,ZGSW,XMT96,YMT96,ZMT96,DIST, &
        ID96)
!
      RHO2=YMT96**2+ZMT96**2
      R=DSQRT(RHO2+XMT96**2)
      ST=DSQRT(RHO2)/R
      CT=XMT96/R
!
!  NOW, USE NEWTON'S ITERATIVE METHOD TO FIND THE NEAREST POINT AT
THE
!   SHUE ET AL.'S BOUNDARY:
!
      NIT=0

  1   T=DATAN2(ST,CT)
      RM=R0*(2.D0/(1.D0+CT))**ALPHA

      F=R-RM
      GRADF_R=1.D0
      GRADF_T=-ALPHA/R*RM*ST/(1.D0+CT)
      GRADF=DSQRT(GRADF_R**2+GRADF_T**2)

      DR=-F/GRADF**2
      DT= DR/R*GRADF_T

      R=R+DR
      T=T+DT
      ST=DSIN(T)
      CT=DCOS(T)

      DS=DSQRT(DR**2+(R*DT)**2)

      NIT=NIT+1

      IF (NIT.GT.1000) THEN
         PRINT *, ' BOUNDARY POINT COULD NOT BE FOUND; ITERATIONS DO NOT
CONVERGE'
```

```
      ENDIF

      IF (DS.GT.1.D-4) GOTO 1

      XMGNP=R*DCOS(T)
      RHO=  R*DSIN(T)

      YMGNP=RHO*DSIN(PHI)
      ZMGNP=RHO*DCOS(PHI)

      DIST=DSQRT((XGSW-XMGNP)**2+(YGSW-YMGNP)**2+(ZGSW-ZMGNP)**2)

      RETURN
      END SUBROUTINE SHUETAL_MGNP_08
!
!==================================================================
===================
!
      SUBROUTINE
T96_MGNP_08(XN_PD,VEL,XGSW,YGSW,ZGSW,XMGNP,YMGNP,ZMGNP, &
       DIST,ID)
!
!  FOR ANY POINT OF SPACE WITH GIVEN COORDINATES (XGSW,YGSW,ZGSW),
THIS SUBROUTINE DEFINES         20.20.20.20.20.
!  THE POSITION OF A POINT (XMGNP,YMGNP,ZMGNP) AT THE T96 MODEL
MAGNETOPAUSE WITH THE
!  SAME VALUE OF THE ELLIPSOIDAL TAU-COORDINATE, AND THE DISTANCE
BETWEEN THEM.  THIS IS
!  NOT THE SHORTEST DISTANCE D_MIN TO THE BOUNDARY, BUT DIST
ASYMPTOTICALLY TENDS TO D_MIN,
!  AS THE OBSERVATION POINT GETS CLOSER TO THE MAGNETOPAUSE.
!
!  INPUT: XN_PD - EITHER SOLAR WIND PROTON NUMBER DENSITY (PER C.C.) (IF
VEL>0)
!                 OR THE SOLAR WIND RAM PRESSURE IN NANOPASCALS   (IF VEL<0)
!        VEL - EITHER SOLAR WIND VELOCITY (KM/SEC)
!               OR ANY NEGATIVE NUMBER, WHICH INDICATES THAT XN_PD STANDS
!                 FOR THE SOLAR WIND PRESSURE, RATHER THAN FOR THE DENSITY
!
!        XGSW,YGSW,ZGSW - COORDINATES OF THE OBSERVATION POINT IN EARTH
RADII
!
!  OUTPUT: XMGNP,YMGNP,ZMGNP - GSW POSITION OF THE BOUNDARY POINT,
HAVING THE SAME
!         VALUE OF TAU-COORDINATE AS THE OBSERVATION POINT
(XGSW,YGSW,ZGSW)
```

```
!        DIST -  THE DISTANCE BETWEEN THE TWO POINTS, IN RE,
!        ID -    POSITION FLAG; ID=+1 (-1) MEANS THAT THE POINT
(XGSW,YGSW,ZGSW)
!        LIES INSIDE (OUTSIDE) THE MODEL MAGNETOPAUSE, RESPECTIVELY.
!
!  THE PRESSURE-DEPENDENT MAGNETOPAUSE IS THAT USED IN THE T96_01
MODEL
!  (TSYGANENKO, JGR, V.100, P.5599, 1995; ESA SP-389, P.181, OCT. 1996)
!
!   AUTHOR:  N.A. TSYGANENKO
!   DATE:    AUG.1, 1995, REVISED APRIL 3, 2003.
!   LAST MODIFICATION:    MARCH 21, 2008 (DOUBLE-PRECISION VERSION)
!
!  DEFINE SOLAR WIND DYNAMIC PRESSURE (NANOPASCALS, ASSUMING 4% OF
ALPHA-PARTICLES),
!   IF NOT EXPLICITLY SPECIFIED IN THE INPUT:
!
      IMPLICIT REAL*8 (A-H,O-Z)
!
      IF (VEL.LT.0.D0) THEN
       PD=XN_PD
      ELSE
       PD=1.94D-6*XN_PD*VEL**2
!
      ENDIF
!
!  RATIO OF PD TO THE AVERAGE PRESSURE, ASSUMED EQUAL TO 2 nPa:

      RAT=PD/2.0D0
      RAT16=RAT**0.14D0

! (THE POWER INDEX 0.14 IN THE SCALING FACTOR IS THE BEST-FIT VALUE
OBTAINED FROM DATA
!    AND USED IN THE T96_01 VERSION)
!
!  VALUES OF THE MAGNETOPAUSE PARAMETERS FOR  PD = 2 nPa:
!
      A0=70.D0
      S00=1.08D0
      X00=5.48D0
!
!   VALUES OF THE MAGNETOPAUSE PARAMETERS, SCALED BY THE ACTUAL
PRESSURE:
!
      A=A0/RAT16
      S0=S00
```

```
      X0=X00/RAT16
      XM=X0-A
!
!  (XM IS THE X-COORDINATE OF THE "SEAM" BETWEEN THE ELLIPSOID AND THE
CYLINDER)
!
!     (FOR DETAILS OF THE ELLIPSOIDAL COORDINATES, SEE THE PAPER:
!      N.A.TSYGANENKO, SOLUTION OF CHAPMAN-FERRARO PROBLEM FOR AN
!      ELLIPSOIDAL MAGNETOPAUSE, PLANET.SPACE SCI., V.37, P.1037, 1989).
!
       IF (YGSW.NE.0.D0.OR.ZGSW.NE.0.D0) THEN
          PHI=DATAN2(YGSW,ZGSW)
       ELSE
          PHI=0.D0
       ENDIF
!
       RHO=DSQRT(YGSW**2+ZGSW**2)
!
       IF (XGSW.LT.XM) THEN
           XMGNP=XGSW
           RHOMGNP=A*DSQRT(S0**2-1.D0)
           YMGNP=RHOMGNP*DSIN(PHI)
           ZMGNP=RHOMGNP*DCOS(PHI)
           DIST=DSQRT((XGSW-XMGNP)**2+(YGSW-YMGNP)**2+(ZGSW-ZMGNP)**2)
           IF (RHOMGNP.GT.RHO) ID=+1
           IF (RHOMGNP.LE.RHO) ID=-1
           RETURN
       ENDIF
!
          XKSI=(XGSW-X0)/A+1.D0
          XDZT=RHO/A
          SQ1=DSQRT((1.D0+XKSI)**2+XDZT**2)
          SQ2=DSQRT((1.D0-XKSI)**2+XDZT**2)
          SIGMA=0.5D0*(SQ1+SQ2)
          TAU=0.5D0*(SQ1-SQ2)
!
!  NOW CALCULATE (X,Y,Z) FOR THE CLOSEST POINT AT THE MAGNETOPAUSE
!
          XMGNP=X0-A*(1.D0-S0*TAU)
          ARG=(S0**2-1.D0)*(1.D0-TAU**2)
          IF (ARG.LT.0.D0) ARG=0.D0
          RHOMGNP=A*DSQRT(ARG)
          YMGNP=RHOMGNP*DSIN(PHI)
          ZMGNP=RHOMGNP*DCOS(PHI)
!
```

```
!  NOW CALCULATE THE DISTANCE BETWEEN THE POINTS {XGSW,YGSW,ZGSW}
AND {XMGNP,YMGNP,ZMGNP}:
!   (IN GENERAL, THIS IS NOT THE SHORTEST DISTANCE D_MIN, BUT DIST
ASYMPTOTICALLY TENDS
!    TO D_MIN, AS WE ARE GETTING CLOSER TO THE MAGNETOPAUSE):
!
      DIST=DSQRT((XGSW-XMGNP)**2+(YGSW-YMGNP)**2+(ZGSW-ZMGNP)**2)
!
      IF (SIGMA.GT.S0) ID=-1   !  ID=-1 MEANS THAT THE POINT LIES OUTSIDE
      IF (SIGMA.LE.S0) ID=+1   !  ID=+1 MEANS THAT THE POINT LIES INSIDE
!                                   THE MAGNETOSPHERE
      RETURN
      END SUBROUTINE T96_MGNP_08
!
!==================================================================
===============
!
      SUBROUTINE T89C(IOPT,PARMOD,PS,X,Y,Z,BX,BY,BZ)
!
!
!   COMPUTES GSM COMPONENTS OF THE MAGNETIC FIELD PRODUCED BY
EXTRA-
!   TERRESTRIAL CURRENT SYSTEMS IN THE GEOMAGNETOSPHERE. THE MODEL
IS
!   VALID UP TO GEOCENTRIC DISTANCES OF 70 RE AND IS BASED ON THE MER-
!   GED IMP-A,C,D,E,F,G,H,I,J (1966-1974), HEOS-1 AND -2 (1969-1974),
!   AND ISEE-1 AND -2  SPACECRAFT DATA SET.
!
!   THIS IS A MODIFIED VERSION (T89c), WHICH REPLACED THE ORIGINAL ONE
!     IN 1992 AND DIFFERS FROM IT IN THE FOLLOWING:
!
!   (1)  ISEE-1,2 DATA WERE ADDED TO THE ORIGINAL IMP-HEOS DATASET
!   (2)  TWO TERMS WERE ADDED TO THE ORIGINAL TAIL FIELD MODES,
ALLOWING
!          A MODULATION OF THE CURRENT BY THE GEODIPOLE TILT ANGLE
!
!
!  REFERENCE FOR THE ORIGINAL MODEL: N.A. TSYGANENKO, A
MAGNETOSPHERIC MAGNETIC
!      FIELD MODEL WITH A WARPED TAIL CURRENT SHEET: PLANET.SPACE SCI.,
V.37,
!        PP.5-20, 1989.
!
!----INPUT PARAMETERS: IOPT - SPECIFIES THE GROUND DISTURBANCE LEVEL:
!
!   IOPT= 1       2       3       4       5       6     7
```

```
!               CORRESPOND TO:
!    KP= 0,0+  1-,1,1+  2-,2,2+  3-,3,3+  4-,4,4+  5-,5,5+  > =6-
!
!    PS - GEODIPOLE TILT ANGLE IN RADIANS
!    X, Y, Z  - GSM COORDINATES OF THE POINT IN EARTH RADII
!
!----OUTPUT PARAMETERS: BX,BY,BZ - GSM COMPONENTS OF THE MODEL
MAGNETIC
!                    FIELD IN NANOTESLAS
!
!   THE PARAMETER PARMOD(10) IS A DUMMY ARRAY.  IT IS NOT USED IN THIS
!       SUBROUTINE AND IS PROVIDED JUST FOR MAKING IT COMPATIBLE WITH
THE
!         NEW VERSION (4/16/96) OF THE GEOPACK SOFTWARE.
!
!   THIS RELEASE OF T89C IS DATED  FEB 12, 1996;
!----------------------------------------------------------------------
!
!
!            AUTHOR:     NIKOLAI A. TSYGANENKO
!                      HSTX CORP./NASA GSFC
!
      DIMENSION XI(4),F(3),DER(3,30),PARAM(30,7),A(30),PARMOD(*)
      DOUBLE PRECISION F,DER
       DATA PARAM/-116.53,-10719.,42.375,59.753,-11363.,1.7844,30.268,    &
     -0.35372E-01,-0.66832E-01,0.16456E-01,-1.3024,0.16529E-02,    &
     0.20293E-02,20.289,-0.25203E-01,224.91,-9234.8,22.788,7.8813,    &
     1.8362,-0.27228,8.8184,2.8714,14.468,32.177,0.01,0.0,    &
     7.0459,4.0,20.0,-55.553,-13198.,60.647,61.072,-16064.,    &
     2.2534,34.407,-0.38887E-01,-0.94571E-01,0.27154E-01,-1.3901,    &
     0.13460E-02,0.13238E-02,23.005,-0.30565E-01,55.047,-3875.7,    &
     20.178,7.9693,1.4575,0.89471,9.4039,3.5215,14.474,36.555,    &
     0.01,0.0,7.0787,4.0,20.0,-101.34,-13480.,111.35,12.386,-24699.,    &
     2.6459,38.948,-0.34080E-01,-0.12404,0.29702E-01,-1.4052,    &
     0.12103E-02,0.16381E-02,24.49,-0.37705E-01,-298.32,4400.9,18.692,    &
     7.9064,1.3047,2.4541,9.7012,7.1624,14.288,33.822,0.01,0.0,6.7442,    &
     4.0,20.0,-181.69,-12320.,173.79,-96.664,-39051.,3.2633,44.968,    &
     -0.46377E-01,-0.16686,0.048298,-1.5473,0.10277E-02,0.31632E-02,    &
     27.341,-0.50655E-01,-514.10,12482.,16.257,8.5834,1.0194,3.6148,    &
     8.6042,5.5057,13.778,32.373,0.01,0.0,7.3195,4.0,20.0,-436.54,    &
     -9001.0,323.66,-410.08,-50340.,3.9932,58.524,-0.38519E-01,    &
     -0.26822,0.74528E-01,-1.4268,-0.10985E-02,0.96613E-02,27.557,    &
     -0.56522E-01,-867.03,20652.,14.101,8.3501,0.72996,3.8149,9.2908,    &
      6.4674,13.729,28.353,0.01,0.0,7.4237,4.0,20.0,-707.77,-4471.9,    &
     432.81,-435.51,-60400.,4.6229,68.178,-0.88245E-01,-0.21002,    &
     0.11846,-2.6711,0.22305E-02,0.10910E-01,27.547,-0.54080E-01,    &
```

```
    -424.23,1100.2,13.954,7.5337,0.89714,3.7813,8.2945,5.174,14.213,    &
    25.237,0.01,0.0,7.0037,4.0,20.0,-1190.4,2749.9,742.56,-1110.3,    &
    -77193.,7.6727,102.05,-0.96015E-01,-0.74507,0.11214,-1.3614,    &
    0.15157E-02,0.22283E-01,23.164,-0.74146E-01,-2219.1,48253.,    &
    12.714,7.6777,0.57138,2.9633,9.3909,9.7263,11.123,21.558,0.01,    &
    0.0,4.4518,4.0,20.0/

      DATA IOP/10/
!
       IF (IOP.NE.IOPT) THEN
!
          ID=1
          IOP=IOPT
          DO 1 I=1,30
  1        A(I)=PARAM(I,IOPT)
!
       ENDIF
!
      XI(1)=X
      XI(2)=Y
      XI(3)=Z
      XI(4)=PS
       CALL T89(ID,A,XI,F,DER)
        IF (ID.EQ.1) ID=2
      BX=F(1)
      BY=F(2)
      BZ=F(3)
      RETURN
     END SUBROUTINE T89C
!
!==================================================================
================
!
       SUBROUTINE  T89 (ID, A, XI, F, DER)
!
!       ***  N.A. Tsyganenko ***  8-10.12.1991  ***
!
!     Calculates dependent model variables and their deriva-
!  tives for given independent variables and model parame-
!  ters.  Specifies model functions with free parameters which
!  must be determined by means of least squares fits (RMS
!  minimization procedure).
!
!     Description of parameters:
!
!  ID  - number of the data point in a set (initial assignments are performed
```

```
!       only for ID=1, saving thus CPU time)
!  A   - input vector containing model parameters;
!  XI  - input vector containing independent variables;
!  F   - output double precision vector containing
!       calculated values of dependent variables;
!  DER   - output double precision vector containing
!       calculated values for derivatives of dependent
!       variables with respect to model parameters;
!
! - - - - - - - - - - - - - - - - - - - - - - - - - - - -
!
!     T89 represents external magnetospheric magnetic field
!  in Cartesian SOLAR MAGNETOSPHERIC coordinates (Tsyganenko N.A.,
!  Planet. Space Sci., 1989, v.37, p.5-20; the "T89 model" with the warped
!  tail current sheet) + A MODIFICATION ADDED IN APRIL 1992 (SEE BELOW)
!
!     Model formulas for the magnetic field components contain in total
!  30 free parameters (17 linear and 13 nonlinear parameters).
!     First 2 independent linear parameters A(1)-A(2) correspond to contribu-
!  tion from the tail current system, then follow A(3) and A(4) which are the
!  amplitudes of symmetric and antisymmetric terms in the contribution from
!  the closure currents; A(5) is the ring current amplitude. Then follow the
! coefficients A(6)-A(15) which define Chapman-Ferraro+Birkeland current field.
!    The coefficients c16-c19  (see Formula 20 in the original paper),
!   due to DivB=0 condition, are expressed through A(6)-A(15) and hence are not
!    independent ones.
!  A(16) AND A(17) CORRESPOND TO THE TERMS WHICH YIELD THE TILT ANGLE
DEPEN-
!    DENCE OF THE TAIL CURRENT INTENSITY (ADDED ON APRIL 9, 1992)
!
!     Nonlinear parameters:
!
!    A(18) : DX - Characteristic scale of the Chapman-Ferraro field along the
!        X-axis
!    A(19) : ADR (aRC) - Characteristic radius of the ring current
!    A(20) : D0 - Basic half-thickness of the tail current sheet
!    A(21) : DD (GamRC)- defines rate of thickening of the ring current, as
!             we go from night- to dayside
!    A(22) : Rc - an analog of "hinging distance" entering formula (11)
!    A(23) : G - amplitude of tail current warping in the Y-direction
!    A(24) : aT - Characteristic radius of the tail current
!    A(25) : Dy - characteristic scale distance in the Y direction entering
!                in W(x,y) in (13)
!    A(26) : Delta - defines the rate of thickening of the tail current sheet
!                in the Y-direction (in T89 it was fixed at 0.01)
!    A(27) : Q - this parameter was fixed at 0 in the final version of T89;
```

```
!            initially it was introduced for making Dy to depend on X
!    A(28) : Sx (Xo) - enters in W(x,y) ; see (13)
!    A(29) : Gam (GamT) - enters in DT in (13) and defines rate of tail sheet
!            thickening on going from night to dayside; in T89 fixed at 4.0
!    A(30) : Dyc - the Dy parameter for closure current system; in T89 fixed
!             at 20.0
!  - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
!
          IMPLICIT  REAL * 8  (A - H, O - Z)
!
      REAL  A(*), XI(*)
!
          DIMENSION  F(3), DER(3,30)
!
      INTEGER  ID, I, L
      DATA A02,XLW2,YN,RPI,RT/25.D0,170.D0,30.D0,0.31830989D0,30.D0/
      DATA XD,XLD2/0.D0,40.D0/
!
!   The last four quantities define variation of tail sheet thickness along X
!
     DATA SXC,XLWC2/4.D0,50.D0/
!
!   The two quantities belong to the function WC which confines tail closure
!    current in X- and Y- direction
!
     DATA DXL/20.D0/
!
!
      IF (ID.NE.1)  GOTO  3
         DO  2  I = 1, 30
          DO  1  L = 1, 3
 1        DER(L,I) = 0.0D0
 2      CONTINUE
!
     DYC=A(30)
     DYC2=DYC**2
     DX=A(18)
     HA02=0.5D0*A02
     RDX2M=-1.D0/DX**2
     RDX2=-RDX2M
     RDYC2=1.D0/DYC2
     HLWC2M=-0.5D0*XLWC2
     DRDYC2=-2.D0*RDYC2
     DRDYC3=2.D0*RDYC2*DSQRT(RDYC2)
     HXLW2M=-0.5D0*XLW2
     ADR=A(19)
```

```
D0=A(20)
DD=A(21)
RC=A(22)
G=A(23)
AT=A(24)
DT=D0
DEL=A(26)
P=A(25)
Q=A(27)
SX=A(28)
GAM=A(29)
HXLD2M=-0.5D0*XLD2
ADSL=0.D0
XGHS=0.D0
H=0.D0
HS=0.D0
GAMH=0.D0
W1=-0.5D0/DX
DBLDEL=2.D0*DEL
W2=W1*2.D0
W4=-1.D0/3.D0
W3=W4/DX
W5=-0.5D0
W6=-3.D0
AK1=A(1)
AK2=A(2)
AK3=A(3)
AK4=A(4)
AK5=A(5)
AK6=A(6)
AK7=A(7)
AK8=A(8)
AK9=A(9)
AK10=A(10)
AK11=A(11)
AK12=A(12)
AK13=A(13)
AK14=A(14)
AK15=A(15)
 AK16=A(16)
 AK17=A(17)
SXA=0.D0
SYA=0.D0
SZA=0.D0
AK610=AK6*W1+AK10*W5
AK711=AK7*W2-AK11
```

```
      AK812=AK8*W2+AK12*W6
      AK913=AK9*W3+AK13*W4
      RDXL=1.D0/DXL
      HRDXL=0.5D0*RDXL
      A6H=AK6*0.5D0
      A9T=AK9/3.D0
      YNP=RPI/YN*0.5D0
      YND=2.D0*YN
!
 3    CONTINUE
!
         X  = XI(1)
            Y  = XI(2)
            Z  = XI(3)
         TILT=XI(4)
            TLT2=TILT**2
            SPS = DSIN(TILT)
         CPS = DSQRT (1.0D0 - SPS ** 2)
!
      X2=X*X
      Y2=Y*Y
      Z2=Z*Z
      TPS=SPS/CPS
      HTP=TPS*0.5D0
      GSP=G*SPS
      XSM=X*CPS-Z*SPS
      ZSM=X*SPS+Z*CPS
!
!   CALCULATE THE FUNCTION ZS DEFINING THE SHAPE OF THE TAIL CURRENT
SHEET
!    AND ITS SPATIAL DERIVATIVES:
!
      XRC=XSM+RC
      XRC16=XRC**2+16.D0
      SXRC=DSQRT(XRC16)
      Y4=Y2*Y2
      Y410=Y4+1.D4
      SY4=SPS/Y410
      GSY4=G*SY4
      ZS1=HTP*(XRC-SXRC)
      DZSX=-ZS1/SXRC
      ZS=ZS1-GSY4*Y4
      D2ZSGY=-SY4/Y410*4.D4*Y2*Y
      DZSY=G*D2ZSGY
!
!   CALCULATE THE COMPONENTS OF THE RING CURRENT CONTRIBUTION:
```

```
!
      XSM2=XSM**2
      DSQT=DSQRT(XSM2+A02)
      FA0=0.5D0*(1.D0+XSM/DSQT)
      DDR=D0+DD*FA0
      DFA0=HA02/DSQT**3
      ZR=ZSM-ZS
      TR=DSQRT(ZR**2+DDR**2)
      RTR=1.D0/TR
      RO2=XSM2+Y2
      ADRT=ADR+TR
      ADRT2=ADRT**2
      FK=1.D0/(ADRT2+RO2)
      DSFC=DSQRT(FK)
      FC=FK**2*DSFC
      FACXY=3.0D0*ADRT*FC*RTR
      XZR=XSM*ZR
      YZR=Y*ZR
      DBXDP=FACXY*XZR
      DER(2,5)=FACXY*YZR
      XZYZ=XSM*DZSX+Y*DZSY
      FAQ=ZR*XZYZ-DDR*DD*DFA0*XSM
      DBZDP=FC*(2.D0*ADRT2-RO2)+FACXY*FAQ
      DER(1,5)=DBXDP*CPS+DBZDP*SPS
      DER(3,5)=DBZDP*CPS-DBXDP*SPS
!
!  CALCULATE THE TAIL CURRENT SHEET CONTRIBUTION:
!
      DELY2=DEL*Y2
      D=DT+DELY2
      IF (DABS(GAM).LT.1.D-6) GOTO 8
      XXD=XSM-XD
      RQD=1.D0/(XXD**2+XLD2)
      RQDS=DSQRT(RQD)
      H=0.5D0*(1.D0+XXD*RQDS)
      HS=-HXLD2M*RQD*RQDS
      GAMH=GAM*H
      D=D+GAMH
      XGHS=XSM*GAM*HS
      ADSL=-D*XGHS
   8  D2=D**2
      T=DSQRT(ZR**2+D2)
      XSMX=XSM-SX
      RDSQ2=1.D0/(XSMX**2+XLW2)
      RDSQ=DSQRT(RDSQ2)
      V=0.5D0*(1.D0-XSMX*RDSQ)
```

```
      DVX=HXLW2M*RDSQ*RDSQ2
      OM=DSQRT(DSQRT(XSM2+16.D0)-XSM)
      OMS=-OM/(OM*OM+XSM)*0.5D0
      RDY=1.D0/(P+Q*OM)
      OMSV=OMS*V
      RDY2=RDY**2
      FY=1.D0/(1.D0+Y2*RDY2)
      W=V*FY
      YFY1=2.D0*FY*Y2*RDY2
      FYPR=YFY1*RDY
      FYDY=FYPR*FY
      DWX=DVX*FY+FYDY*Q*OMSV
      YDWY=-V*YFY1*FY
      DDY=DBLDEL*Y
      ATT=AT+T
      S1=DSQRT(ATT**2+RO2)
      F5=1.D0/S1
      F7=1.D0/(S1+ATT)
      F1=F5*F7
      F3=F5**3
      F9=ATT*F3
      FS=ZR*XZYZ-D*Y*DDY+ADSL
      XDWX=XSM*DWX+YDWY
      RTT=1.D0/T
      WT=W*RTT
      BRRZ1=WT*F1
      BRRZ2=WT*F3
      DBXC1=BRRZ1*XZR
      DBXC2=BRRZ2*XZR
      DER(2,1)=BRRZ1*YZR
      DER(2,2)=BRRZ2*YZR
        DER(2,16)=DER(2,1)*TLT2
        DER(2,17)=DER(2,2)*TLT2
      WTFS=WT*FS
      DBZC1=W*F5+XDWX*F7+WTFS*F1
      DBZC2=W*F9+XDWX*F1+WTFS*F3
      DER(1,1)=DBXC1*CPS+DBZC1*SPS
      DER(1,2)=DBXC2*CPS+DBZC2*SPS
      DER(3,1)=DBZC1*CPS-DBXC1*SPS
      DER(3,2)=DBZC2*CPS-DBXC2*SPS
        DER(1,16)=DER(1,1)*TLT2
        DER(1,17)=DER(1,2)*TLT2
        DER(3,16)=DER(3,1)*TLT2
        DER(3,17)=DER(3,2)*TLT2
!
!  CALCULATE CONTRIBUTION FROM THE CLOSURE CURRENTS
```

```
!
      ZPL=Z+RT
      ZMN=Z-RT
      ROGSM2=X2+Y2
      SPL=DSQRT(ZPL**2+ROGSM2)
      SMN=DSQRT(ZMN**2+ROGSM2)
      XSXC=X-SXC
      RQC2=1.D0/(XSXC**2+XLWC2)
      RQC=DSQRT(RQC2)
      FYC=1.D0/(1.D0+Y2*RDYC2)
      WC=0.5D0*(1.D0-XSXC*RQC)*FYC
      DWCX=HLWC2M*RQC2*RQC*FYC
      DWCY=DRDYC2*WC*FYC*Y
      SZRP=1.D0/(SPL+ZPL)
      SZRM=1.D0/(SMN-ZMN)
      XYWC=X*DWCX+Y*DWCY
      WCSP=WC/SPL
      WCSM=WC/SMN
      FXYP=WCSP*SZRP
      FXYM=WCSM*SZRM
      FXPL=X*FXYP
      FXMN=-X*FXYM
      FYPL=Y*FXYP
      FYMN=-Y*FXYM
      FZPL=WCSP+XYWC*SZRP
      FZMN=WCSM+XYWC*SZRM
      DER(1,3)=FXPL+FXMN
      DER(1,4)=(FXPL-FXMN)*SPS
      DER(2,3)=FYPL+FYMN
      DER(2,4)=(FYPL-FYMN)*SPS
      DER(3,3)=FZPL+FZMN
      DER(3,4)=(FZPL-FZMN)*SPS
!
!   NOW CALCULATE CONTRIBUTION FROM CHAPMAN-FERRARO SOURCES + ALL
OTHER
!
         EX=DEXP(X/DX)
         EC=EX*CPS
         ES=EX*SPS
         ECZ=EC*Z
         ESZ=ES*Z
            ESZY2=ESZ*Y2
            ESZZ2=ESZ*Z2
            ECZ2=ECZ*Z
            ESY=ES*Y
!
```

```
            DER(1,6)=ECZ
            DER(1,7)=ES
            DER(1,8)=ESY*Y
            DER(1,9)=ESZ*Z
            DER(2,10)=ECZ*Y
            DER(2,11)=ESY
            DER(2,12)=ESY*Y2
            DER(2,13)=ESY*Z2
            DER(3,14)=EC
            DER(3,15)=EC*Y2
            DER(3,6)=ECZ2*W1
            DER(3,10)=ECZ2*W5
            DER(3,7)=ESZ*W2
            DER(3,11)=-ESZ
            DER(3,8)=ESZY2*W2
            DER(3,12)=ESZY2*W6
            DER(3,9)=ESZZ2*W3
            DER(3,13)=ESZZ2*W4
!
!  FINALLY, CALCULATE NET EXTERNAL MAGNETIC FIELD COMPONENTS,
!    BUT FIRST OF ALL THOSE FOR C.-F. FIELD:
!
     SX1=AK6*DER(1,6)+AK7*DER(1,7)+AK8*DER(1,8)+AK9*DER(1,9)
     SY1=AK10*DER(2,10)+AK11*DER(2,11)+AK12*DER(2,12)+AK13*DER(2,13)

SZ1=AK14*DER(3,14)+AK15*DER(3,15)+AK610*ECZ2+AK711*ESZ+AK812*ESZY2+AK
913*ESZZ2
      BXCL=AK3*DER(1,3)+AK4*DER(1,4)
      BYCL=AK3*DER(2,3)+AK4*DER(2,4)
      BZCL=AK3*DER(3,3)+AK4*DER(3,4)
      BXT=AK1*DER(1,1)+AK2*DER(1,2)+BXCL +AK16*DER(1,16)+AK17*DER(1,17)
      BYT=AK1*DER(2,1)+AK2*DER(2,2)+BYCL +AK16*DER(2,16)+AK17*DER(2,17)
      BZT=AK1*DER(3,1)+AK2*DER(3,2)+BZCL +AK16*DER(3,16)+AK17*DER(3,17)
      F(1)=BXT+AK5*DER(1,5)+SX1+SXA
      F(2)=BYT+AK5*DER(2,5)+SY1+SYA
      F(3)=BZT+AK5*DER(3,5)+SZ1+SZA
!
      RETURN
      END SUBROUTINE  T89
!
!=====================================================================
================
!
SUBROUTINE PARTICLAC(Particle,deltat,Totalt,Vel,R,angle,Energy,C,Magpos,Magcomp,L)
      !L= switch between single or group and proton or electron and possibilties
      !Vel in km/s,angle in degrees
```

```fortran
      Implicit none
      Real*8, dimension(:, :) :: Particle,Magpos,Magcomp
      Real*8, dimension(:) :: C,Vel,R,angle,Energy
      Real*8, dimension(1:2) :: D
      Real*8 :: deltat,X,Y,Z,PX,PY,PZ,DX,DY,DZ,B,BX,BY,BZ,tg,vx,vy,vz,rc,ac,ec
      Integer i,j,k,l,n,q,s,h,u,za,zb,zc,zd,ze,zf,num_points,num_shls,Cnum,Totalt

      num_points= size(Magpos(1,:))
      num_shls= size(Magpos(:,1))
      Cnum= (size(C)/3)*size(Energy)*size(R)*size(angle)
      D(1)= 1.0E10
      Vel = Vel/6378.0d0 !change 2 radii

      If(mod(L,2)==0.and.L>0) GOTO 10
      If(mod(L,2)/=0.and.L>0) GOTO 20
      If(mod(L,2)==0.and.L<0) GOTO 30
      If(mod(L,2)/=0.and.L<0) GOTO 40

10    do u=1,cnum,1!size(particle(: ,1)),1 !cases

         vx= Vel(2)*C(15-2)
         vy= Vel(2)*C(15-1)
         vz= Vel(2)*C(15)
         rc= R(3)
         ac= angle(1)
         ec= Energy(2)
         Particle(1:Cnum , 1)= 1.0d0 !rc*cos(ac)
       Particle(1:Cnum , 2)= 1.0d0 !rc*sin(ac)
       Particle(1:Cnum , 3)= 0.0d0

       do n= 2,size(particle(1 ,:)),3 ! positions
         do q= 1,num_shls
           do i=2,(num_points*3),3
             j=((i+1)/3)

                PX= Particle( u, n-1)
             PY= Particle( u , n )
             PZ= Particle( u, n+1)
             X = Magpos( q , i+1)
             Y = Magpos( q , i )
             Z = Magpos( q , i-1)

                IF(X<0.AND.PX<0.OR.X>0.AND.PX>0) THEN
                  DX = ABS(ABS(X)-ABS(PX))
                ELSE
                  DX = ABS(((-1)*X)-PX)
```

```
            ENDIF
            IF(Y<0.AND.PY<0.OR.Y>0.AND.PY>0) THEN
              DY = ABS(ABS(Y)-ABS(PY))
            ELSE
              DY = ABS(((-1)*Y)-PY)
            ENDIF
            IF(Z<0.AND.PZ<0.OR.Z>0.AND.PZ>0) THEN
              DZ = ABS(ABS(Z)-ABS(PZ))
            ELSE
              DZ = ABS(((-1)*Z)-PZ)
            ENDIF

            D(2)=DX+DY+DZ

            IF(D(2).le.D(1)) Then
              K=j*3
              H=Q
            ENDIF
            D(1) = D(2)
         enddo
          enddo
      ! i nd q
        s = ((n+1)/3)
        BX=Magcomp(h,k-2)
        BY=Magcomp(h,k-1)
        BZ=Magcomp( h, k )
        B= (BX**2) + (BY**2) + (BZ**2)
        tg= 1.0d0/(0.0152d0*B)
        Deltat=tg/100.0d0

        Particle( u, n+2) = PX + (Vy*Bz-Vz*By)
        Particle( u, n+3) = PY + (-1)*(Vx*Bz-Vz*Bx)
        Particle( u, n+4) = PZ + (Vx*By-Vy*Bx)

      enddo
    enddo

20   GOTO 30
30   GOTO 40
40   continue
     return
END SUBROUTINE PARTICLAC
!
!=================================================================================
================
!
```

```
SUBROUTINE CHG_DAY (MONTH,DAY,IYEAR,IDAY)       !!!!!!!!!!!!!
24.24.24.24.24.
     INTEGER :: MONTH,DAY,IDAY,IYEAR
     INTEGER :: PDAY,I

     PDAY=0

     DO I=1,12,1
          IF(MONTH.LT.1.OR.MONTH.GT.12) THEN
                  PRINT*, "INVAILD NUMBER"
                     EXIT
             ELSEIF((I-1).EQ.1) THEN
                     PDAY=PDAY+31
             ELSEIF((I-1).EQ.2) THEN
                  PDAY=PDAY+27
                  IF (MOD(IYEAR,4).EQ.0) THEN
                       PDAY=PDAY+1
                  ENDIF
             ELSEIF(I.EQ.4.OR.I.EQ.6.OR.I.EQ.8.OR.I.EQ.9.OR.I.EQ.11) THEN
                  PDAY= PDAY+31
             ELSEIF(I.EQ.5.OR.I.EQ.7.OR.I.EQ.10) THEN
                  PDAY= PDAY+30
         ELSEIF(I.EQ.MONTH) THEN
               EXIT
          ELSE
               EXIT
          ENDIF
     ENDDO

     IDAY= DAY + PDAY
     RETURN
END SUBROUTINE CHG_DAY

SUBROUTINE FILL_MAG(Mag,R)             !!!!!!!!!!!!!
25.25.25.25.25.
     Real*8,Dimension(:) :: Mag
     Real*8 :: pi,a,R,ptsd
     Integer :: i,j,k
     pi= 3.14159265358979323846264338327950d0
     a = (size(Mag)/3.0d0)
     ptsd=(360.0d0/sqrt(a))
     j=0
     k=ptsd
!     SAS=4pir2
!     =360/sqrt(a)
     do i = 2,(size(Mag)-1),3    ! make array = A(R,Theta,Phi,R,Theta,Phi,etc.) for point
```

```
        Mag(i-1)= R !in Radii 6378 in kilometers
           if((ptsd*(((i+1)/3)-j)).ge.360.0d0) then
          j= ((i+1)/3)-1
             k=k + ptsd
           endif
           Mag(i)= (90-(ptsd*(((i+1)/3)-j)))*(pi/180.0d0) !Theta in radians;
           Mag(i+1)= ((ptsd+k )*(pi/180.0d0)) !Phi in radians
      enddo

!        Mag(i)= ((90 -( (90/a)*((i+1)/3) ))* (pi/180.0d0))!Theta in radians;
!                      ! creates evenly divided points above or below equator!! not both!
!                   !gets amount of points,divides to get even distribution,creats new
!                   !point at with new loop,subtracts for colatitude,multipy 2 =radians
!        Mag(i+1)= (((360/a)*((i+1)/3))* (pi/180.0d0))!Phi in radians
END SUBROUTINE FILL_MAG

SUBROUTINE TRACEPARTI(particle)

END SUBROUTINE TRACEPARTI

SUBROUTINE PRINT_MAG(Mag)             !!!!!!!!!!!!
27.27.27.27.27.
      Real*8, Dimension(:) :: Mag
      Integer i

      do i=1,(size(Mag)-2),3
        !write(unit= , fmt= , rec= , advance= )
        write(*,"(f15.12)",advance = "no") Mag(i),Mag(i+1),Mag(i+2)
      enddo

END SUBROUTINE PRINT_MAG


!SUBROUTINE FILL_OBSERV(Observ,Vel,DeltaT,num_points)
!      Real*8, Dimension(:) :: Observ
!      Real*8 :: X,Y,Z,Vel,DeltaT,sun
!      Integer i,num_points
!      sun= 23386.489201406
!      do i = 2,(size(Observ)-1),3
!        Observ(i-1)= (sun/num_points)*((3*num_points)/(i+1))!X
!        Observ(i)=0!Y
!        Observ(i+1)=0!Z
!      enddo
!
!END SUBROUTINE FILL_OBSERV
```

```
!===========================================================================
===============
Endmodule
```

(B) NetLogo Code

```
globals [components particles positions]

to loadData
  file-close-all
  let mComponentScale 1 / 100000
  clear-turtles
  file-open "magdata.txt"
  set positions (read-from-string (word "[" file-read-line "]"))
  let mDummy file-read-line ; yo frankie, pay attentiion to this line if fortran changes
  set components (read-from-string (word "[" file-read-line "]"))
  let counter 0
  repeat length positions / 3
   [crt 1 [
       set shape "arrow"
       set size .025
       setxyz (item counter positions) (item (counter + 1) positions) (item (counter + 2) positions)
       ;;;  setheading item counter components
       hatch 1 [ set shape "default" create-link-with myself  set xcor xcor + item counter
components * mComponentScale
              set ycor ycor + item (counter + 1) components * mComponentScale set zcor zcor +
item (counter + 2) components * mComponentScale]
       ]
   set counter counter + 3
  ]
; ask turtles [set shape "circle"]
; ask turtles [hatch 1 [set xcor xcor + -.5  create-link-with myself]]
; ask turtles with [who > 2499] [die]
; ask links [set color scale-color red link-length .3 .6]
; ask turtles [set color gray]
; ask turtles [set shape "circle" set size .025]
 ;;ask one-of links [show zcor]
 ask links [ifelse ([zcor] of one-of both-ends > 0) [set color blue][set color red]]
 ;;ask turtles [ht]

 ;let positions []

end
```

```
;to magsphere_1
;  let mComponentScale 1 / 100000
;  file-open "magsphere_1.txt"
;  set positions (read-from-string (word "[" file-read-line "]"))
;  let mDummy file-read-line
;  set components (read-from-string (word "[" file-read-line "]"))
;  let counter 0
;  repeat length positions / 3
;  [crt 1 [
;       set shape "arrow"
;       set size .025
;       setxyz (item counter positions) (item (counter + 1) positions) (item (counter + 2) positions)
;       ;;;  setheading item counter components
;       hatch 1 [ set shape "default" create-link-with myself  set xcor xcor + item counter
components * mComponentScale
;                 set ycor ycor + item (counter + 1) components * mComponentScale set zcor zcor +
item (counter + 2) components * mComponentScale]
;       ]
;   set counter counter + 3
;  ]
;  ask turtles [set shape "arrow"]
;
;end
```