

CENTRE NATIONAL DE
LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE
VERSAILLES SAINT QUENTIN

INSTITUT PIERRE-SIMON LAPLACE

**CENTRE D'ETUDE DES ENVIRONNEMENTS
TERRESTRE et PLANETAIRES**

ROCOTLIB :

**a Coordinate Transformation Library
for Solar-Terrestrial studies**

by

Patrick ROBERT

Version 1.8 – November 2003

Update of RI-CETP/1/2003

CONTENTS

FOREWORD	5
PART I: DEFINITIONS AND MATHEMATICAL FORMULAS	7
I- DESCRIPTION OF COORDINATE SYSTEMS	9
1) Geocentric Equatorial Inertial system (GEI)	9
2) Geographic system (GEO).....	10
3) Geomagnetic system (MAG).....	10
4) Geocentric Solar Ecliptic system (GSE).....	11
5) Geocentric Solar Equatorial system (GSEQ)	11
6) Geocentric Solar Magnetospheric system (GSM)	12
7) Solar Magnetic system (SM)	12
8) Dipole Meridian system (DM)	13
9) Vertical Dusk Horizontal system (VDH).....	13
10) Spin Reference system (SR).....	14
11) Spin Reference 2 system (SR2).....	14
12) Magnetic Field Aligned system (MFA).....	15
II- DIAGRAM OF TRANSFORMATIONS	16
1) General remarks	16
2) Schematic diagram of transformations.....	16
3) List of transformations	17
III- MATHEMATICAL EXPRESSIONS OF TRANSFORMATION MATRIX.....	18
1) General remarks on transformation matrix.....	18
2) GEI to GEO transformation.....	19
3) GEI to MAG transformation.....	20
4) GEI to SM transformation	22
5) GEI to GSM transformation.....	24
6) GEI to GSE transformation	26
7) GEI to GSEQ transformation	27
8) GSE to GSEQ transformation	28
9) GSE to GSM transformation	30
10) GSM to SM transformation	34
11) SM to MAG transformation	36
12) GEO to MAG transformation.....	40
13) GEO to SM transformation.....	41
14) GEO to GSM transformation.....	43
15) GEO to GSE transformation	45
16) GEO to GSEQ transformation.....	47
17) GEO to DM transformation	49
18) GEO to VDH transformation	51
19) SR2 to GSE transformation	53
20) SR2 to MFA transformation	55
21) SR to SR2 transformation	59
IV- CALENDAR CONVERSIONS	60
1) General remarks	60

2) List of calendar conversions and definitions.....	60
• compute if a given year is or not a leap year (coleapy).....	60
• compute day of the year from given date (cdoyear)	60
• compute date from given day of the year and year (cdatdoy).....	60
• compute Julian day 1950 from given date (cjd1950)	60
• compute Julian day 2000 from given date (cjd2000)	61
• compute date from given 1950 Julian day (cdatj50)	61
• compute date from given 2000 Julian day (cdatj00)	61
• compute decimal hour from given hour, minute and second (chouday).....	61
• compute hour, minute and second from given decimal hour (ctimhou).....	61
• compute millisec. of the day from jour, hour, min,sec.,ms. (cmilday)	61
• compute time from millisec. of the day (ctimmil).....	61
• compute week number in a year from a given date (cweedoy)	61
• compute day of the week for a given date (dowweek)	61
• compute date from the week number of the year (cdatweek)	62
• compute number of day in a month (cnbdmon)	62
• compute US day name (cusdayn).....	62
• compute french day name (cfrdayn)	62
• compute US month name (cusmonn)	62
• compute french month name (cfrmonn).....	62
PART II: USER'S MANUAL	63
I- DESCRIPTION OF AVAILABLE MODULES	65
1) General remarks	65
2) Description of "Basic Computation" subroutines	66
3) Description of "Calendar" subroutines.....	69
4) Description of "Give" subroutines.....	73
5) Description of "Read and check" subroutines	75
6) Description of "Transformations" subroutines	76
7) Description of "print information" subroutine	84
II - DIRECTIONS FOR USE.....	85
1) Package delivered for installation	85
2) Make the rocotlib.o object file	87
3) Example of Fortran user program	88
4) Output of "print information" subroutine	91
5) Check program	92
• general remarks.....	92
• input rocotche.in data file of check program	92
• output rocotche.out file of check program.....	93
Summary of available subroutine.....	106
• "Basic computation" subroutines :	106
• "Calendar" subroutines :	106
• "Give" subroutines :	106
• "Read and check" subroutines :	107
• "Transformation" subroutines :	107
• "Print information" subroutine :	108
Bibliography	109

FOREWORD

The ROCOTLIB library (RObert's COordinate Transformation LIBrary) is a set of software modules to perform transformations between the various coordinate systems used in geophysical and magnetospheric studies. Most of the frames of reference are geocentric, and are thus independent of the position of the point of observation ; nevertheless, some local frames are also considered.

In addition to coordinate transformations, the library also provides a set of modules to perform format conversions and other operations associated with epoch, date and time.

This library was originally developed in 1992 by P. Robert, CNRS/CRPE, with support from ESA within the framework of preparation of the CLUSTER mission; since then it has been regularly updated by the author. Original document is entitled:

« Document de travail DT/CRPE/1231, CLUSTER Software Tools, Part 1 : Coordinate Transformation Library, Version 1.1 », by Patrick ROBERT, RPE/TID, CNRS/CNET/CRPE, Juillet 1993.

ROCOTLIB exists in both FORTRAN 77 and in FORTRAN 90, and of course can be run on any computer where these compilers are available. It will soon be available in the IDL and PV-Wave programming languages. Each transformation or module corresponds to a subroutine in FORTRAN, and to a procedure in IDL or PV-Wave. The package delivered to the user includes sources and makefiles of the library, examples of its use, a test program and the corresponding test output file to check the validity of the installation on the user's machine. The test programme has been developed and tested using the following FORTRAN compilers :

- SunOS 5.8 and SunOS 5, Sun WorkShop Compiler FORTRAN 77 V 5.0
- The same systems, with FORTRAN 90 V2.0
- LINUX /Intel i686 , g77 - GNU project FORTRAN Compiler (v0.5.24)
- DEC OSF/1, Digital FORTRAN 90 for Digital UNIX Alpha Systems V4.0

This document is the full documentation of the ROCOTLIB library and includes two parts:

- the first defines the various coordinate systems considered, and gives the mathematical formulas and matrices to pass from one system to another.
- The second part is the user's manual for the FORTRAN library. It give the complete list of available modules, the role for each one, and a description of the input and output variables. The example programs and the test program are also commented.

Documentation and code source are available on <http://cdpp.cesr.fr>.

This library is a living product, and can be completed in the future by other transformations, or derived applications. Any comments are welcome.

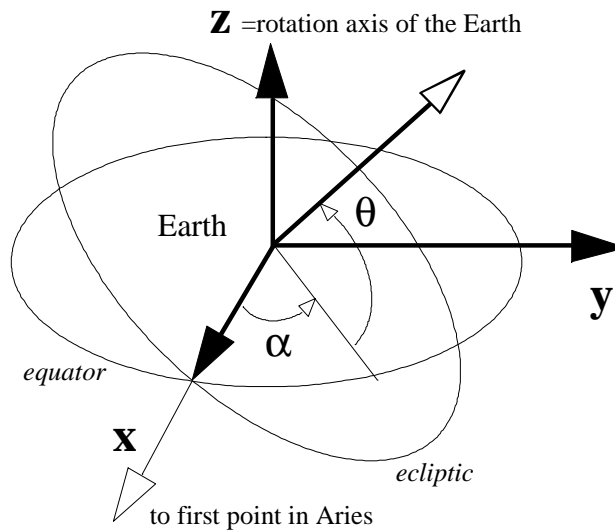
ROCOTLIB COORDINATE TRANSFORMATION LIBRARY

PART I: DEFINITIONS AND MATHEMATICAL FORMULAS

I- DESCRIPTION OF COORDINATE SYSTEMS

Most of the coordinate systems described are geocentric, with any exceptions as the dipole meridian system and the VDH system, which are local coordinate systems and thus depend of the position of the point of observation.

1) Geocentric Equatorial Inertial system (GEI)



The Z-axis is parallel to the rotation axis of the Earth.

The X-axis is defined by the intersection of the equator plane and the ecliptic plane, and is pointing towards the first point of Aries (Sun position at the vernal equinox).

one can define the *right ascension* α and the *declination* θ as:

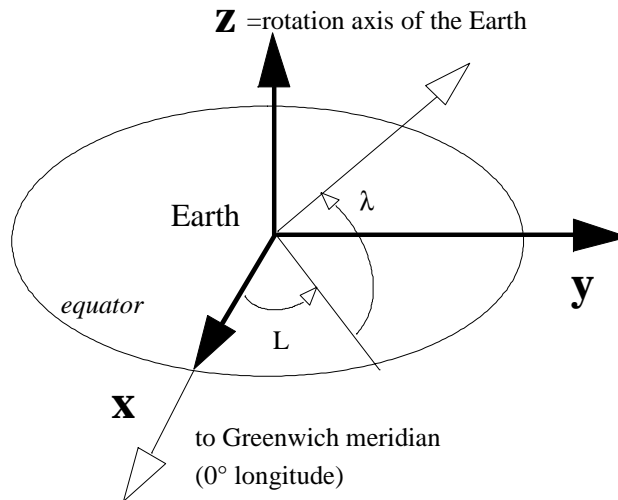
$$\text{right ascension } \alpha = \tan^{-1}(V_y/V_x)$$

$$\begin{aligned} &\text{with } \alpha \text{ in } [0^\circ, 180^\circ] \text{ for } V_y > 0 \\ &\alpha \text{ in } [180^\circ, 360^\circ] \text{ for } V_y < 0 \end{aligned}$$

$$\text{declination } \theta = \sin^{-1}(V_z/V)$$

$$\text{with } \theta \text{ in } [-90^\circ, 90^\circ]$$

2) Geographic system (GEO)



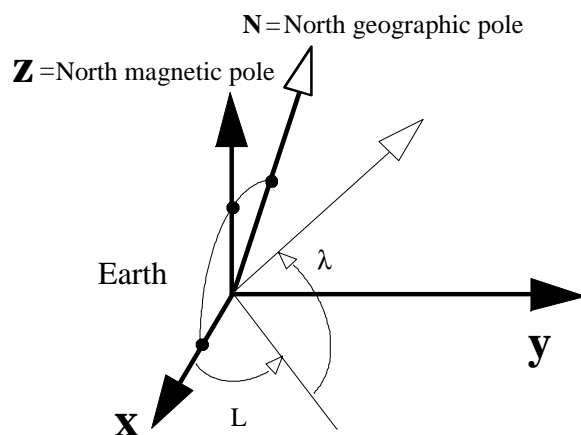
The Z-axis is parallel to the rotation axis of the Earth.

The X and Y axis are included in the equator plane.

The X axis is pointing from the centre of the Earth to the Greenwich meridian (0° longitude).

The GEO system is fixed with the rotating Earth. Longitude L and latitude λ are defined in this system in the same way as right ascension and declination in GEI system.

3) Geomagnetic system (MAG)



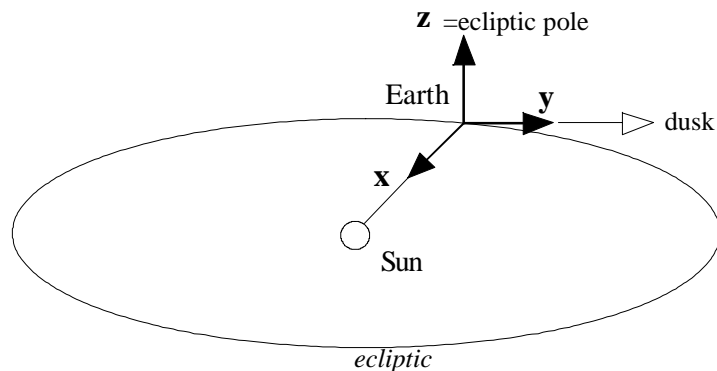
The Z-axis is parallel to the magnetic dipole axis.

If \underline{N} is the North geographic pole, the \underline{N} , \underline{Z} and \underline{X} vector are in the same plane.

The Y-axis is defined as $\underline{Y} = -\underline{Z} \times \underline{N}$

The MAG system is fixed with the rotating Earth. The magnetic longitude L and magnetic latitude λ are defined in this system in the same way as right ascension and declination in GEI system.

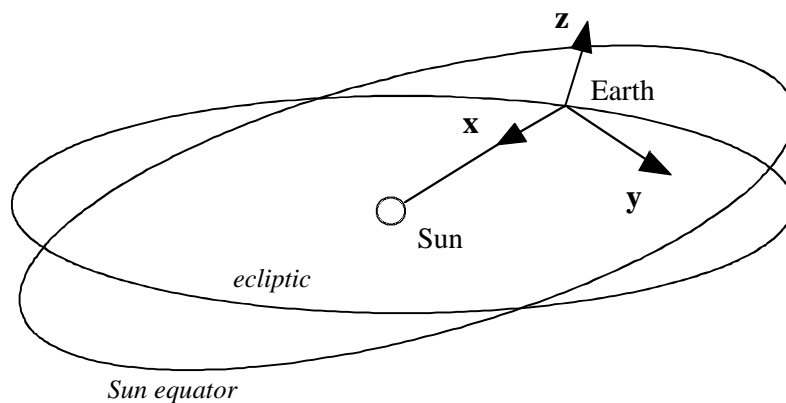
4) Geocentric Solar Ecliptic system (GSE)



The X-axis is pointing from the Earth towards the Sun.
The X-axis and the Y-axis are include in the ecliptic plane.
The Y-axis is pointing toward the dusk, opposing to the planetary motion.

The Z-axis is parallel to the ecliptic pole. The GSE system has a yearly rotation with respect to the inertial system.

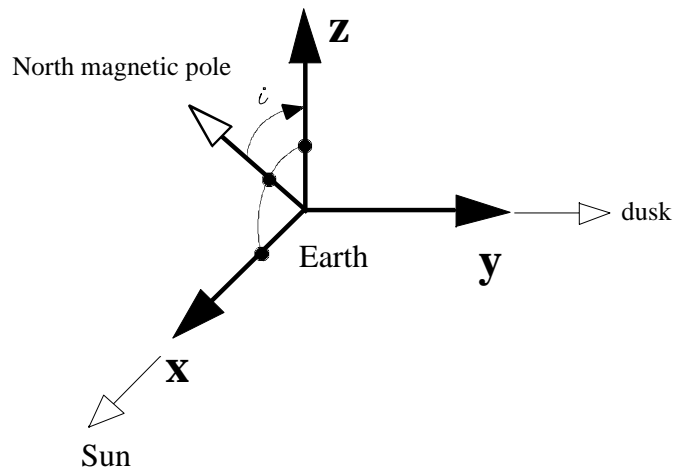
5) Geocentric Solar Equatorial system (GSEQ)



The X-axis is pointing toward the Sun and include in the ecliptic plane.
The Y-axis is parallel to the Sun's equatorial plane (inclined to the ecliptic)

X-axis is not necessarily in the Sun's equatorial plane;
Z-axis is not necessarily be parallel to the Sun's axis of rotation (which is perpendicular to Y, and thus in the X-Z plane);
Z-axis is chosen to be in the same sense as the ecliptic pole, i.e. northwards.

6) Geocentric Solar Magnetospheric system (GSM)



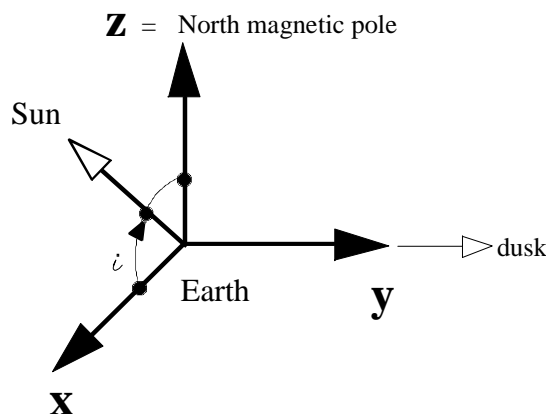
The X-axis is pointing from the Earth towards the Sun.

The X-Z plane contains the dipole axis.

The Y-axis is perpendicular to the Earth's magnetic dipole, towards the dusk and include in the magnetic equator plane.

The positive Z-axis is chosen to be in the same sense as the northern magnetic pole; the dipole tilt angle i is positive when the north magnetic pole is tilted towards the Sun. In addition to a yearly period due to the motion of the Earth about the Sun, the GSM system rocks about the Solar direction with a 24 h period.

7) Solar Magnetic system (SM)



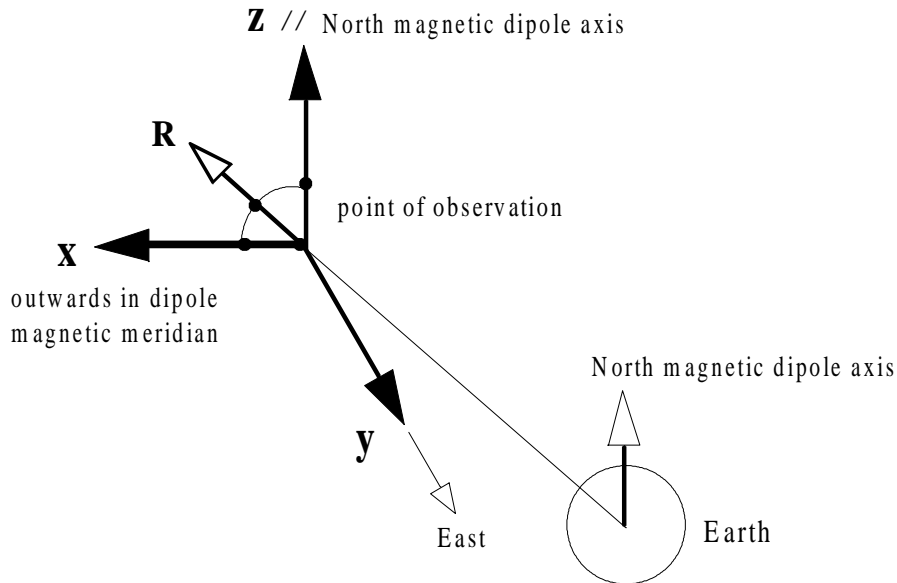
The Z-axis is parallel to the North magnetic dipole.

The X-Z plane contains the direction of the Sun.

The Y-axis is perpendicular to the Earth-Sun line toward dusk.

The SM system rotates with both a yearly and a daily period with respect to the inertial system.

8) Dipole Meridian system (DM)



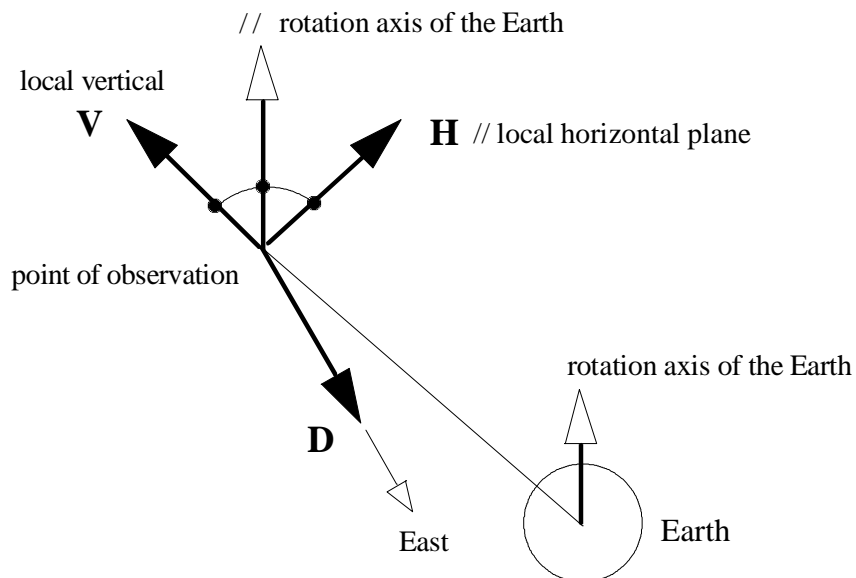
The Z-axis is parallel to the North magnetic dipole axis.

The X-Z plane contains the direction **R** of the point of observation, from the Earth, and is a dipole magnetic meridian plane.

The Y-axis is perpendicular to the **R** vector, eastwards.

This system is a local coordinate system, which is dependent of the position of the point of observation from the Earth.

9) Vertical Dusk Horizontal system (VDH)



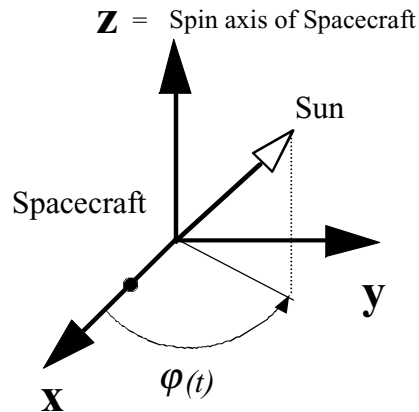
The V-axis is the outwards local vertical, to the point of observation.

The H-axis is parallel to the horizontal local plane, positive to the North.

The V-H plane is a geographic meridian plane. The D-axis is azimuthal, eastwards.

As DM system, this system is a local coordinate system, which is dependent of the position of the point of observation from the Earth.

10) Spin Reference system (SR)



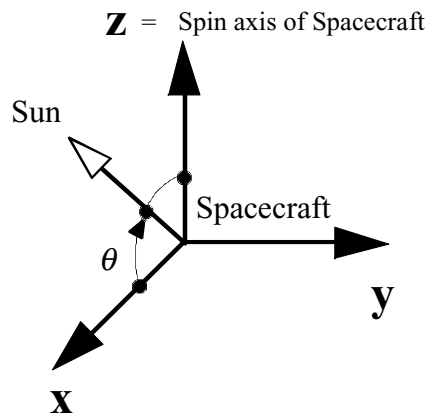
This is a spinning local system close to the measurement antenna of a spacecraft.

The Z-axis is the spin axis of the spacecraft.

The X-axis and Y-axis are perpendicular to the spin axis, and rotate at the spin frequency of the spacecraft.

The definition of the SR system need the knowledge of the spin axis in a fixed frame of reference as the GEI inertial system, and the value of the spin phase φ at a given time.

11) Spin Reference 2 system (SR2)



This is a fixed system usefull for the spacecraft data processing. It is also called SCS, as “Spacecraft-Sun system”, or DS system (Despun Satellite).

The Z-axis is the spin axis of the spacecraft.

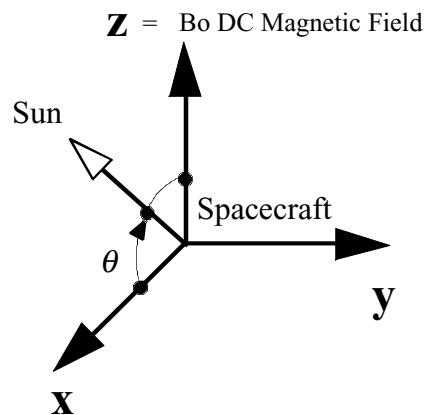
The X-Z plane contains the direction of the Sun.

The X-axis is towards the day side.

The Y-axis is perpendicular to the spacecraft-Sun line.

The SR2 system rotates with the same period than the orbital period of the spacecraft with respect to the inertial system, while the declination θ varies continuously.

12) Magnetic Field Aligned system (MFA)



This is a system useful for physics, but the meaning of the B_0 DC magnetic field must be known, as its time variation (see ref. [3]).

The Z-axis is the DC magnetic field vector.

The X-Z plane contains the direction of the Sun.

The X-axis is towards the day side.

The Y-axis is perpendicular to the spacecraft-Sun line.

The MFA system moves continuously with the time variation of the DC magnetic field.

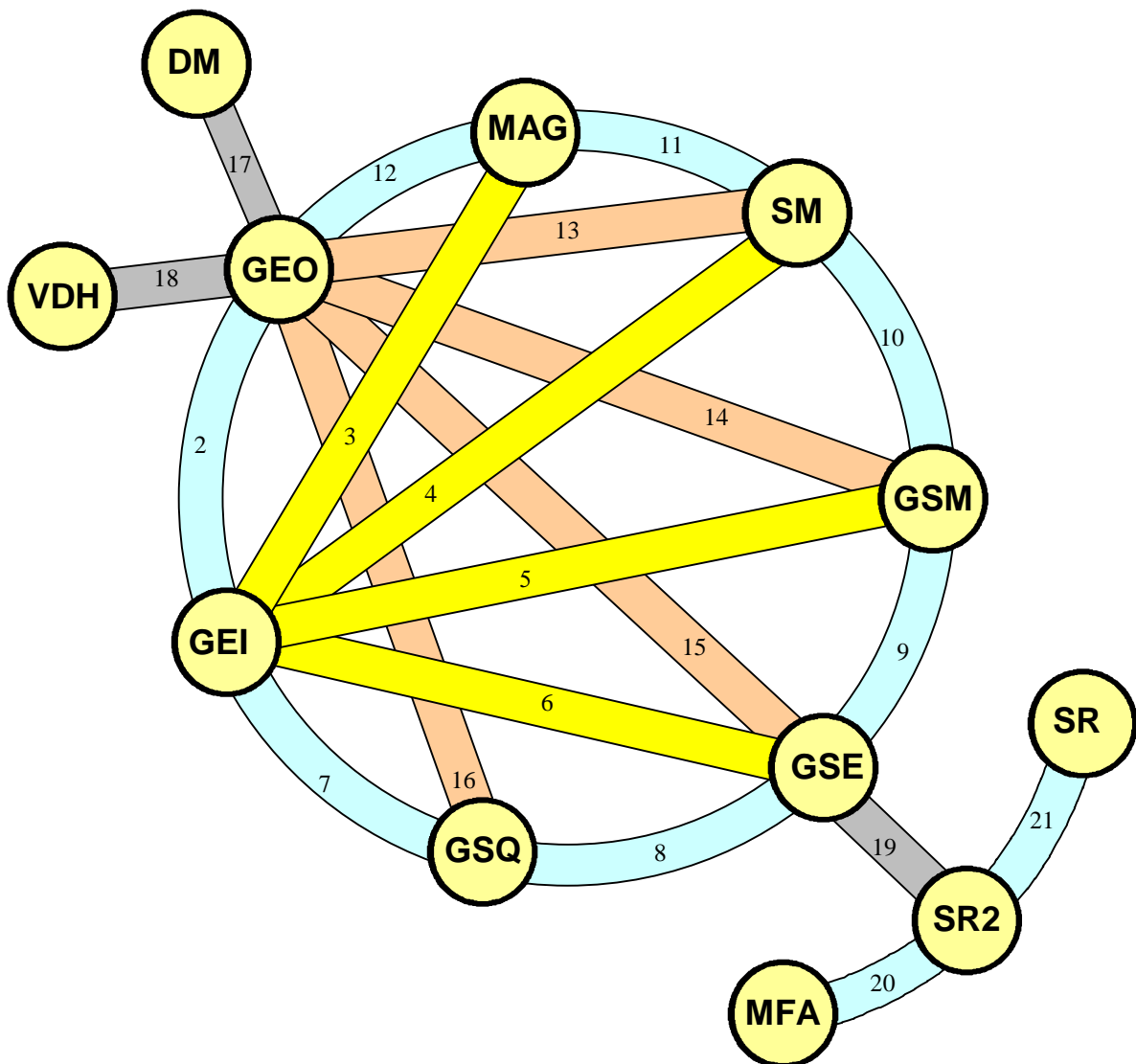
[3] CLUSTER DATA PROCESSING, Transformation of a STAFF waveform into a Magnetic Field Aligned coordinate system, by Patrick Robert and C. de Villedary, Rapport interne CNRS-UVSQ/CETP n° RI-CETP/6/2000, Octobre 2000.

II- DIAGRAM OF TRANSFORMATIONS

1) General remarks

Among all coordinates systems described in section I, we have single out two particular systems more frequently encountered: the Geographic system (GEO) and the Geocentric Equatorial Inertial system (GEI). From these two system, we have computed all transformations to directly convert these coordinates into any else other. Furthermore, other direct "circular" transformations between the other system are also given, as explained on the schematic diagram below.

2) Schematic diagram of transformations



3) List of transformations

All different coordinate transformations are listed below; number correspond to § number of section III, and are mentioned in the above schematic diagram.

	<i>name</i>	<i>input coordinates</i>		<i>output coordinates</i>
2)	geigeo geogei	Geocentric Equatorial Inertial (GEI) Geographical (GEO)	→ →	Geographical (GEO) Geocentric Equatorial Inertial (GEI)
3)	geimag maggei	Geocentric Equatorial Inertial (GEI) Magnetic dipole (MAG)	→ →	Magnetic dipole (MAG) Geocentric Equatorial Inertial (GEI)
4)	geisma smagei	Geocentric Equatorial Inertial (GEI) Solar Magnetic (SM)	→ →	Solar Magnetic (SM) Geocentric Equatorial Inertial (GEI)
5)	geigsm gsmgei	Geocentric Equatorial Inertial (GEI) Geocentric Solar Magnetospheric (GSM)	→ →	Geocentric Solar Magnetospheric (GSM) Geocentric Equatorial Inertial (GEI)
6)	geigse gsegei	Geocentric Equatorial Inertial (GEI) Geocentric Solar Ecliptic (GSE)	→ →	Geocentric Solar Ecliptic (GSE) Geocentric Equatorial Inertial (GEI)
7)	geigsq gsqgei	Geocentric Equatorial Inertial (GEI) Geocentric Solar Equatorial (GSQ)	→ →	Geocentric Solar Equatorial (GSQ) Geocentric Equatorial Inertial (GEI)
8)	gsegsq gsqgse	Geocentric Solar Ecliptic (GSE) Geocentric Solar Equatorial (GSQ)	→ →	Geocentric Solar Equatorial (GSQ) Geocentric Solar Ecliptic (GSE)
9)	gsegsm gsmgse	Geocentric Solar Ecliptic (GSE) Geocentric Solar Magnetospheric (GSM)	→ →	Geocentric Solar Magnetospheric (GSM) Geocentric Solar Ecliptic (GSE)
10)	gsmsma smagsm	Geocentric Solar Magnetospheric (GSM) Solar Magnetic (SM)	→ →	Solar Magnetic (SM) Geocentric Solar Magnetospheric (GSM)
11)	smamag magsma	Solar Magnetic (SM) Magnetic dipole (MAG)	→ →	Magnetic dipole (MAG) Solar Magnetic (SM)
12)	geomag mageo	Geographical (GEO) Magnetic dipole (MAG)	→ →	Magnetic dipole (MAG) Geographical (GEO)
13)	geosma smageo	Geographical (GEO) Solar Magnetic (SM)	→ →	Solar Magnetic (SM) Geographical (GEO)
14)	geogsm gsmgeo	Geographical (GEO) Geocentric Solar Magnetospheric (GSM)	→ →	Geocentric Solar Magnetospheric (GSM) Geographical (GEO)
15)	geogse gsegeo	Geographical (GEO) Geocentric Solar Ecliptic (GSE)	→ →	Geocentric Solar Ecliptic (GSE) Geographical (GEO)
16)	geogsq gsqgeo	Geographical (GEO) Geocentric Solar Equatorial (GSQ)	→ →	Geocentric Solar Equatorial (GSQ) Geographical (GEO)
17)	geodme dmegeo	Geographical (GEO) Dipole Meridian (DM)	→ →	Dipole Meridian (DM) Geographical (GEO)
18)	geovdh vdhgeo	Geographical (GEO) Vertical Dusk Horizontal (VDH)	→ →	Vertical Dusk Horizontal (VDH) Geographical (GEO)
19)	gsesr2 sr2gse	Geocentric Solar Ecliptic (GSE) Spin Reference 2 (SR2)	→ →	Spin Reference 2 (SR2) Geocentric Solar Ecliptic (GSE)
20)	sr2mfa	Spin Reference 2 (SR2)	→	Magnetic Field Aligned
21)	sresr2 sr2sre	Spin Reference (SR) Spin Reference 2 (SR2)	→ →	Spin Reference 2 (SR2) Spin Reference (SR)

III- MATHEMATICAL EXPRESSIONS OF TRANSFORMATION MATRIX

1) General remarks on transformation matrix

To obtain the matrix to transform a vector expressed in a coordinate system A into another system B, the simplest way is to express the directions of the 3 axis of system B in the coordinate system A.

Indeed, if we notes the coordinates of these 3 unit axes \mathbf{X}_B , \mathbf{Y}_B , \mathbf{Z}_B in coordinate system A as:

$$\mathbf{X}_B = \begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix}_{(A)} \quad \mathbf{Y}_B = \begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix}_{(A)} \quad \mathbf{Z}_B = \begin{pmatrix} Z1 \\ Z2 \\ Z3 \end{pmatrix}_{(A)}$$

any vector \mathbf{V} can be expressed in system B as:

$$\begin{aligned} V_1^{(B)} &= \mathbf{X}_B \cdot \mathbf{V} = X_1^{(A)} V_1^{(A)} + X_2^{(A)} V_2^{(A)} + X_3^{(A)} V_3^{(A)} \\ V_2^{(B)} &= \mathbf{Y}_B \cdot \mathbf{V} = Y_1^{(A)} V_1^{(A)} + Y_2^{(A)} V_2^{(A)} + Y_3^{(A)} V_3^{(A)} \\ V_3^{(B)} &= \mathbf{Z}_B \cdot \mathbf{V} = Z_1^{(A)} V_1^{(A)} + Z_2^{(A)} V_2^{(A)} + Z_3^{(A)} V_3^{(A)} \end{aligned}$$

Thus the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(B)} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(A)}$$

Similarly the transformation from system B to A is:

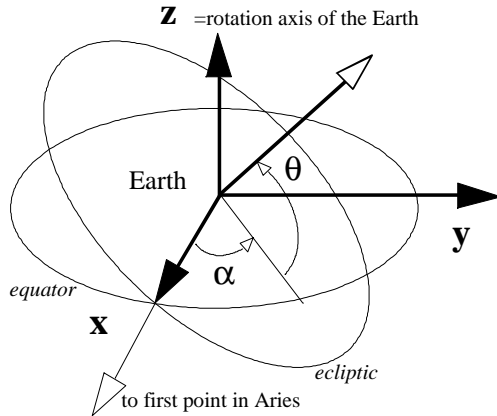
$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(A)} = \begin{pmatrix} X1 & Y1 & Z1 \\ X2 & Y2 & Z2 \\ X3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(B)}$$

All transformation matrix have the following properties, useful for error checking:

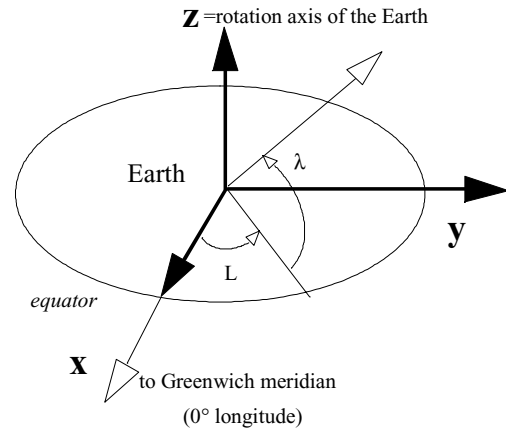
- 1) Each row and column is a unit vector;
- 2) The dot products of any two rows or any two columns is zero;
- 3) The cross product of any two rows or columns equals the third row or column or its negative (Row 1 cross row 2 equals row3; row 2 cross row 1 equal minus row3).

2) GEI to GEO transformation

Geocentric Equatorial Inertial system



Geographic system



GEO and GEI system have their Z-axis in common, so the only difference is a rotation around Z-axis of θ angle, thus the matrix transformation is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

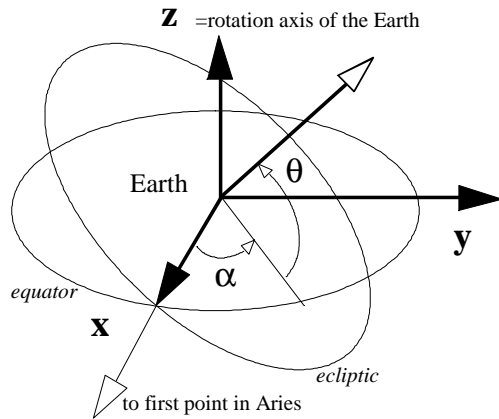
The θ angle is the angle between the Greenwich meridian and the first point in Aries, measured Eastward, in the Earth's equator, from the first point in Aries.

θ is called Greenwich Mean Sideral Time; GMST is a function of the time of the day and the time of year, since the sideral day (duration of a day relative to inertial space) is less than 24h.

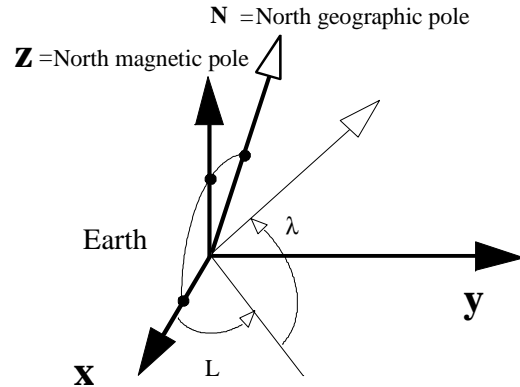
Practically, GMST is computed from *csundir* subroutine.

3) GEI to MAG transformation

Geocentric Equatorial Inertial



Geomagnetic



Transformation from GEI to MAG system requires a knowledge of the dipole direction in GEI system, noted as **M**.

The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, **D** is computed for a given time and year from *cdipdir* subroutine.

To know the $\mathbf{M} = \mathbf{D}$ vector in GEI system, which is the Z-axis of MAG system, we use the GEO to GEI transformation computed § III-2, so:

$$\mathbf{Z} = \mathbf{M} = \mathbf{D}_{\text{GEI}} = \begin{pmatrix} D_1 \cos\theta - D_2 \sin\theta \\ D_1 \sin\theta + D_2 \cos\theta \\ D_3 \end{pmatrix}$$

where θ is the Greenwich Mean Sideral Time computed from *csundir* subroutine.

We can deduce then the Y-axis of SM system in GEI coordinates from the cross product between North geographic pole (**N**) and North magnetic pole (**M**) :

$$\mathbf{Y} = \mathbf{N} \times \mathbf{M} / |\mathbf{N} \times \mathbf{M}|$$

Normalizing factors occurs because **N** and **M** are not necessarily perpendicular; since **N** has two components equal to zero, cross product is easy and we found:

$$\mathbf{Y} = \begin{pmatrix} -M_2 \\ M_1 \\ 0 \end{pmatrix} \cdot 1 / (M_1^2 + M_2^2)^{1/2}$$

X-axis is deduced from:

$$\mathbf{X} = \mathbf{Y} \times \mathbf{M}$$

so:

$$\mathbf{X} = \begin{pmatrix} M1M3 \\ M2M3 \\ -(M1^2 + M2^2) \end{pmatrix} \cdot 1 / (M1^2 + M2^2)^{1/2}$$

All coordinates of X-Y-Z axis of MAG system in GEI coordinates being known, the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ M1 & M2 & M3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

Similarly the transformation from system MAG to GEI is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} X1 & Y1 & M1 \\ X2 & Y2 & M2 \\ X3 & Y3 & M3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)}$$

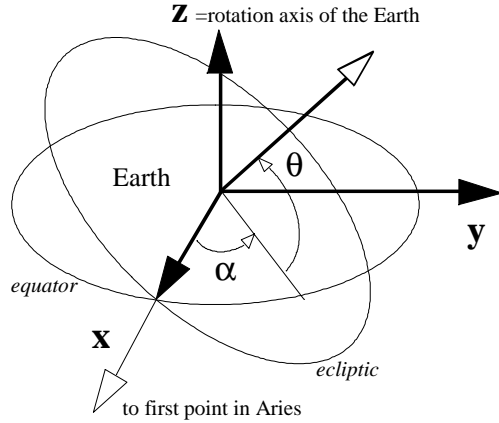
By replacing corresponding values, with $Q = (M1^2 + M2^2)^{1/2}$, we can obtain the fully expanded expression:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} (D1D3\cos\theta - D2D3\sin\theta) / Q & (D1D3\sin\theta + D2D3\cos\theta) / Q & -Q \\ -(D2\cos\theta + D1\sin\theta) / Q & (-D2\sin\theta + D1\cos\theta) / Q & 0 \\ D1\cos\theta - D2\sin\theta & D1\sin\theta + D2\cos\theta & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

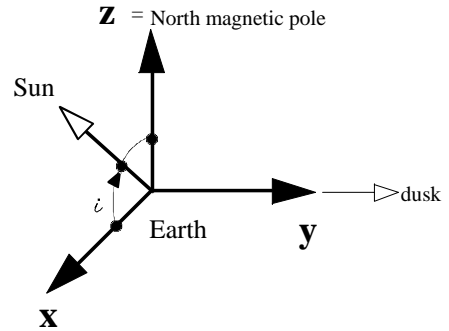
$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} (D1D3\cos\theta - D2D3\sin\theta) / Q & -(D2\cos\theta + D1\sin\theta) / Q & D1\cos\theta - D2\sin\theta \\ (D1D3\sin\theta + D2D3\cos\theta) / Q & (-D2\sin\theta + D1\cos\theta) / Q & D1\sin\theta + D2\cos\theta \\ -Q & 0 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)}$$

4) GEI to SM transformation

Geocentric Equatorial Inertial



Solar Magnetic



Transformation from GEI system to SM system requires a knowledge of Sun direction and magnetic dipole direction in GEI system.

In GEI system, the direction of the Sun is computed from *csundir* subroutine:

$$\mathbf{S} = (S_1, S_2, S_3)$$

The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, D is computed for a given time and year from *cdipdir* subroutine.

To know the $\mathbf{M}=\mathbf{D}$ vector in GEI system, which is the Z-axis of SM system, we use the GEO to GEI transformation computed § III-2, so:

$$\mathbf{Z} = \mathbf{M} = \mathbf{D}_{\text{GEI}} = \begin{pmatrix} D_1 \cos \theta & -D_2 \sin \theta \\ D_1 \sin \theta & +D_2 \cos \theta \\ & D_3 \end{pmatrix}$$

where θ is the Greenwich Mean Sideral Time computed from *csundir* subroutine.

We can deduce then the Y-axis of SM system in GEI coordinates as:

$$\mathbf{Y} = \mathbf{M} \times \mathbf{S} / |\mathbf{M} \times \mathbf{S}|$$

(normalizing factors occurs because \mathbf{M} and \mathbf{S} are not necessarily perpendicular)

X-axis is deduced from:

$$\mathbf{X} = \mathbf{Y} \times \mathbf{M}$$

All coordinates of X-Y-Z axis of SM system in GEI coordinates being known, the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

Similarly the transformation from system SM to GEI is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} X1 & Y1 & Z1 \\ X2 & Y2 & Z2 \\ X3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

with respectively:

$$\begin{pmatrix} Z1 \\ Z2 \\ Z3 \end{pmatrix} = \begin{pmatrix} M1 \\ M2 \\ M3 \end{pmatrix}$$

$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix} = \begin{pmatrix} M2S3 - M3S2 \\ M3S1 - M1S3 \\ M1S2 - M2S1 \end{pmatrix} \cdot 1/Q$$

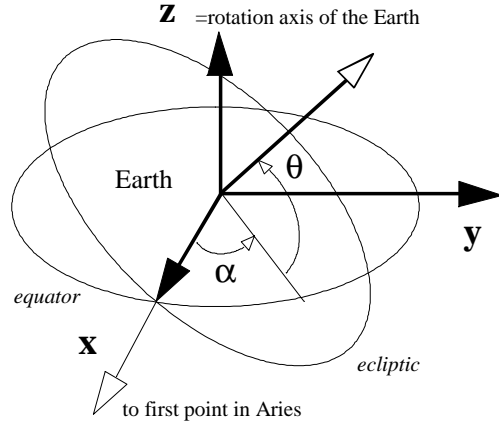
$$\begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = \begin{pmatrix} Y2M3 - Y3M2 \\ Y3M1 - Y1M3 \\ Y1M2 - Y2M1 \end{pmatrix}$$

$$\begin{pmatrix} M1 \\ M2 \\ M3 \end{pmatrix} = \begin{pmatrix} D1\cos\theta & - D2\sin\theta \\ D1\sin\theta & + D2\cos\theta \\ & D3 \end{pmatrix}$$

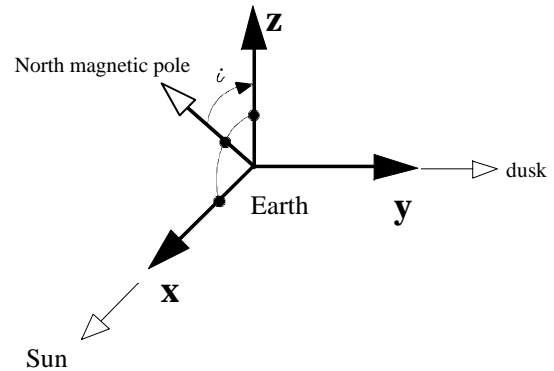
$$Q = \left[(M2S3 - M3S2)^2 + (M3S1 - M1S3)^2 + (M1S2 - M2S1)^2 \right]^{1/2}$$

5) GEI to GSM transformation

Geocentric Equatorial Inertial



Geocentric Solar Magnetospheric



Transformation from GEI system to GSM system requires a knowledge of Sun direction and magnetic dipole direction in GEI system.

In GEI system, the direction of X-axis of GSM system is the direction of the Sun computed from *csundir* subroutine:

$$\mathbf{X} = \mathbf{S} = (S_1, S_2, S_3)$$

The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, \mathbf{D} is computed for a given time and year from *cdipdir* subroutine.

To know the $\mathbf{M} = \mathbf{D}$ vector in GEI system, we use the GEO to GEI transformation computed in § III-2, so:

$$\mathbf{M} = \mathbf{D}_{\text{GEO}} = \begin{pmatrix} D_1 \cos \theta & - D_2 \sin \theta \\ D_1 \sin \theta & + D_2 \cos \theta \\ & & D_3 \end{pmatrix}$$

where θ is the Greenwich Mean Sideral Time computed from *csundir* subroutine.

We can deduce then the Y-axis and Z-axis of GSM system in GEI coordinates as:

$$\mathbf{Y} = \mathbf{M} \times \mathbf{S} / |\mathbf{M} \times \mathbf{S}|$$

(normalizing factors occurs because \mathbf{M} and \mathbf{S} are not necessarily perpendicular)

and

$$\mathbf{Z} = \mathbf{S} \times \mathbf{Y}$$

All coordinates of X-Y-Z axis of GSM system in GEI coordinates being known, the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)} = \begin{pmatrix} S1 & S2 & S3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

Similarly the transformation from system GSM to GEI is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} S1 & Y1 & Z1 \\ S2 & Y2 & Z2 \\ S3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

with respectively:

$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix} = \begin{pmatrix} M2S3 - M3S2 \\ M3S1 - M1S3 \\ M1S2 - M2S1 \end{pmatrix} \cdot 1/Q$$

$$\begin{pmatrix} Z1 \\ Z2 \\ Z3 \end{pmatrix} = \begin{pmatrix} S2Y3 - S3Y2 \\ S3Y1 - S1Y3 \\ S1Y2 - S2Y1 \end{pmatrix}$$

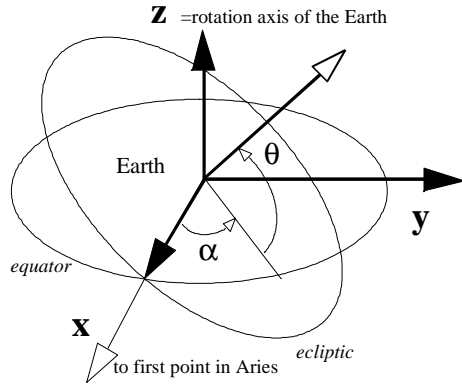
where

$$\begin{pmatrix} M1 \\ M2 \\ M3 \end{pmatrix} = \begin{pmatrix} D1\cos\theta - D2\sin\theta \\ D1\sin\theta + D2\cos\theta \\ D3 \end{pmatrix}$$

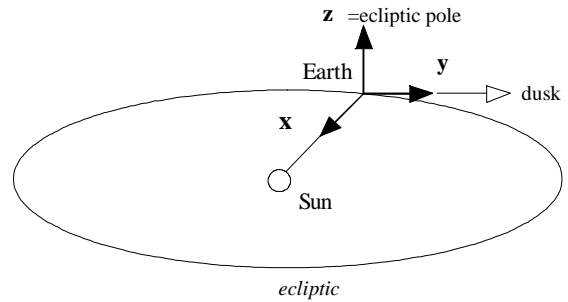
$$Q = [(M2S3 - M3S2)^2 + (M3S1 - M1S3)^2 + (M1S2 - M2S1)^2]^{1/2}$$

6) GEI to GSE transformation

Geocentric Equatorial Inertial



Geocentric Solar Ecliptic



In GEI system, the direction of X-axis of GSE system is the direction \mathbf{S} of the SUN , computed from *csundir* subroutine:

$$\mathbf{X}=\mathbf{S}=(S_1, S_2, S_3)$$

The direction of the Z-axis of GSE is the direction of ecliptic pole, which is a known constant value:

$$\mathbf{Z}=\mathbf{E}=(E_1, E_2, E_3) = (0, -0.398, 0.917)$$

The third axis, Y, is deduced from $\mathbf{Y}=\mathbf{Z} \times \mathbf{X} = \mathbf{E} \times \mathbf{S}$, thus:

$$\mathbf{Y}=\mathbf{E} \times \mathbf{S} = \begin{pmatrix} E_2S_3 - E_3S_2 \\ E_3S_1 - E_1S_3 \\ E_1S_2 - E_2S_1 \end{pmatrix}$$

Thus the transform matrix of any vector \mathbf{V} is:

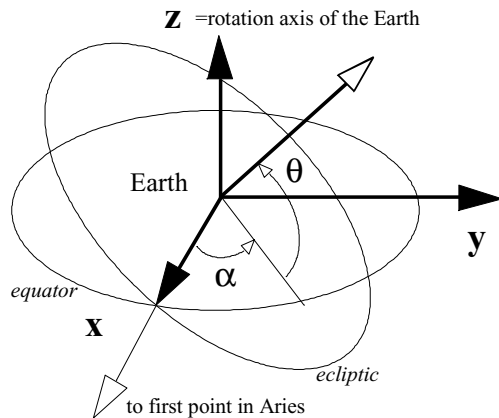
$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} S_1 & S_2 & S_3 \\ Y_1 & Y_2 & Y_3 \\ E_1 & E_2 & E_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GEI)}$$

Similarly the transformation from system GSE to GEI is:

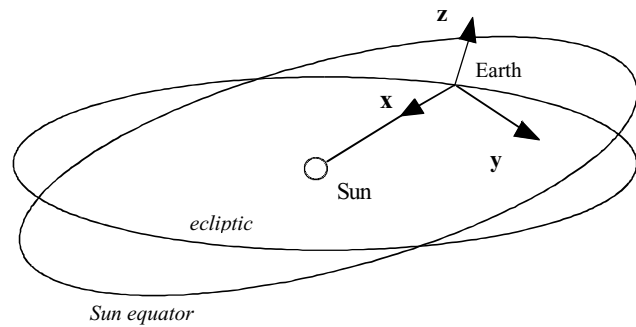
$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} S_1 & Y_1 & E_1 \\ S_2 & Y_2 & E_2 \\ S_3 & Y_3 & E_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GSE)}$$

7) GEI to GSEQ transformation

Geocentric Equatorial Inertial



Geocentric Solar Equatorial



In GEI system, the direction of X-axis of GSEQ system is the direction of the SUN computed from **csundir** subroutine:

$$\mathbf{X} = \mathbf{S} = (S1, S2, S3)$$

The direction of the rotation axis of the SUN in GEI system is a known constant value:

$$\mathbf{R} = (R1, R2, R3) = (0.122, -0.424, 0.899)$$

Since Y-axis of GSEQ is parallel to the Sun's equatorial plane, the direction of Y-axis in GEI is $\mathbf{R} \times \mathbf{S}$; nevertheless the cross product must be normalized to have a Y unit axis, because \mathbf{R} and \mathbf{S} are not necessarily perpendicular, so:

$$\mathbf{Y} = (\mathbf{R} \times \mathbf{S}) / |\mathbf{R} \times \mathbf{S}|$$

and $\mathbf{Z} = \mathbf{S} \times \mathbf{Y}$

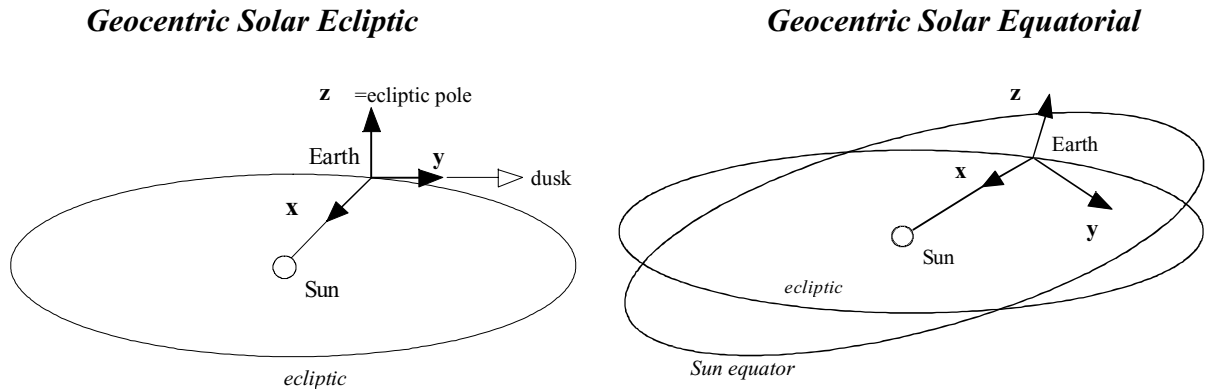
Thus the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)} = \begin{pmatrix} S1 & S2 & S3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

Similarly the transformation from system GSEQ to GEI is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} S1 & Y1 & Z1 \\ S2 & Y2 & Z2 \\ S3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)}$$

8) GSE to GSEQ transformation



The only difference between GSE and GSEQ systems is a rotation about the common X-axis, to have the Y-GSEQ axis parallel to the Sun equator plane.

So, if θ is the rotation angle, the transformation matrix of any vector V is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)}$$

computation of θ angle:

θ is the (Y_{GSE}, Y_{GSEQ}) angle, so $\sin\theta = |Y_{GSE} \times Y_{GSEQ}|$

To compute θ , we use the following known vectors in GEI system:

- 1) the direction \mathbf{S} of the SUN, computed from *csundir* subroutine:

$$\mathbf{S} = (S1, S2, S3)$$

- 2) the direction of ecliptic pole, which is a known constant value:

$$\mathbf{E} = (E1, E2, E3) = (0, -0.398, 0.917)$$

3) the direction of the rotation axis of the Sun, which is also a constant value:

$$\mathbf{R} = (R_1, R_2, R_3) = (0.122, -0.424, 0.899)$$

To compute $\sin \theta = |\mathbf{Y}_{GSE} \times \mathbf{Y}_{GSEQ}|$ we use the following properties:

$$\mathbf{Y}_{GSE} = \mathbf{Z}_{GSE} \times \mathbf{X}_{GSE} = \mathbf{E} \times \mathbf{S}$$

and since \mathbf{R} is in the X-Z plane in the GSEQ system:

$$\mathbf{Y}_{GSEQ} = (\mathbf{R} \times \mathbf{S}) / |\mathbf{R} \times \mathbf{S}|$$

$$\text{so we have } \sin \theta = |(\mathbf{E} \times \mathbf{S}) \times (\mathbf{R} \times \mathbf{S})| / |\mathbf{R} \times \mathbf{S}|$$

$$\text{since } (\mathbf{A} \times \mathbf{B}) \times (\mathbf{C} \times \mathbf{D}) = (\mathbf{A} \times \mathbf{B} \cdot \mathbf{D})\mathbf{C} - (\mathbf{A} \times \mathbf{B} \cdot \mathbf{C})\mathbf{D}$$

$$\text{and } (\mathbf{A} \times \mathbf{B}) \cdot \mathbf{C} = (\mathbf{A} \cdot \mathbf{B}) \times \mathbf{C}$$

$$\text{thus } (\mathbf{E} \times \mathbf{S}) \times (\mathbf{R} \times \mathbf{S}) = (\mathbf{E} \times \mathbf{S} \cdot \mathbf{S})\mathbf{R} - (\mathbf{E} \times \mathbf{S} \cdot \mathbf{R})\mathbf{S} = (\mathbf{R} \times \mathbf{E} \cdot \mathbf{S})\mathbf{S}$$

and finally, as \mathbf{S} is a unit vector:

$$\sin \theta = (\mathbf{R} \times \mathbf{E}) \cdot \mathbf{S} / |\mathbf{R} \times \mathbf{S}|$$

For numerical applications, expression $\sin \theta = (\mathbf{R} \times \mathbf{E}) \cdot \mathbf{S} / |\mathbf{R} \times \mathbf{S}|$ can be extended as:

$$\sin \theta = \frac{[(R_2E_3 - R_3E_2)S_1 + (R_3E_1 - R_1E_3)S_2 + (R_1E_2 - R_2E_1)S_3]}{[(R_2S_3 - R_3S_2)^2 + (R_3S_1 - R_1S_3)^2 + (R_1S_2 - R_2S_1)^2]^{1/2}}$$

with numerical values above, this becomes:

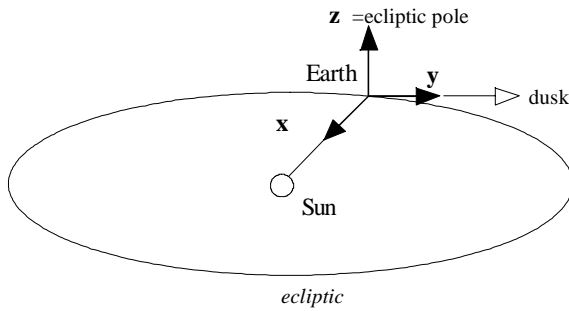
$$\sin \theta = (-0.031, -0.112, -0.049) \cdot \mathbf{S} / |(0.122, -0.424, 0.899) \times \mathbf{S}|$$

Since the Sun's spin axis is inclined 7.25° to the ecliptic, θ changes from -7.25 to 7.25 each year, from approximately December 5 to June 5.

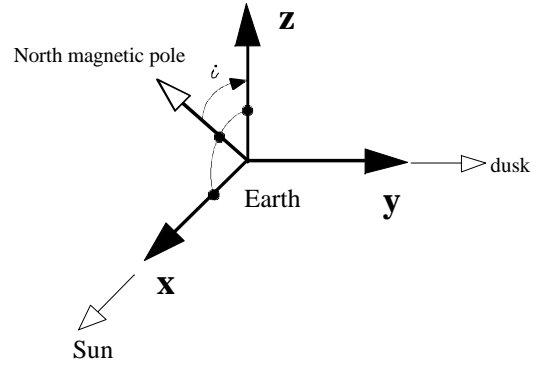
The Sun's spin axis is directed most towards the Earth on approximately September 5 at which time the Earth reaches its northern most heliographic latitude. At this time $\theta=0$.

9) GSE to GSM transformation

Geocentric Solar Ecliptic



Geocentric Solar Magnetospheric



GSE and GSM systems have their X-axis in common, so the only difference is a rotation around the X-axis of the angle ξ , thus the matrix transformation is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\xi & \sin\xi \\ 0 & -\sin\xi & \cos\xi \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\xi & -\sin\xi \\ 0 & \sin\xi & \cos\xi \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

Nevertheless the ξ angle cannot be obtained from a simple equation.

To compute the rotation terms of transformation matrix, we use the GSE to GEI and the GEI to GSM previous matrix transformations, given in § III-6 and III-5.

These transformations are noted (see § III-6 and III-5):

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} S1 & S2 & S3 \\ Y1 & Y2 & Y3 \\ E1 & E2 & E3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)}$$

with $\mathbf{y} = (\mathbf{E} \times \mathbf{S})$ in GEI system

and:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEI)} = \begin{pmatrix} S1 & Y1 & Z1 \\ S2 & Y2 & Z2 \\ S3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

with $\mathbf{Y} = (\mathbf{M} \times \mathbf{S}) / |(\mathbf{M} \times \mathbf{S})|$ in GEI system
and $\mathbf{Z} = (\mathbf{S} \times \mathbf{Y})$ in GEI system

we can write the GSM to GSE transformation as:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} S1 & S2 & S3 \\ Y1 & Y2 & Y3 \\ E1 & E2 & E3 \end{pmatrix} \begin{pmatrix} S1 & Y1 & Z1 \\ S2 & Y2 & Z2 \\ S3 & Y3 & Z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

which give

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} S \bullet S & S \bullet Y & S \bullet Z \\ Y \bullet S & Y \bullet Y & Y \bullet Z \\ E \bullet S & E \bullet Y & E \bullet Z \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

which becomes, since \mathbf{S} and \mathbf{Y} are unit vectors perpendicular between us, as \mathbf{S} and \mathbf{Z} :

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & Y \bullet Y & Y \bullet Z \\ 0 & E \bullet Y & E \bullet Z \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

Of course the final matrix does not depend on the \mathbf{S} vector.

Computation of $\cos \xi$

We have to equalize $\mathbf{Y} \bullet \mathbf{Y}$ and $\mathbf{E} \bullet \mathbf{Z}$ terms as $\cos \xi$.

Taking $\mathbf{Y} \bullet \mathbf{Y} = (\mathbf{E} \times \mathbf{S}) \bullet (\mathbf{M} \times \mathbf{S})$

we compute $\mathbf{E} \bullet \mathbf{Z} = \mathbf{E} \bullet (\mathbf{S} \times \mathbf{Y}) = \mathbf{E} \bullet [(\mathbf{S} \times (\mathbf{M} \times \mathbf{S}))]$

since $\mathbf{A} \bullet (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) \bullet \mathbf{C}$

we effectively found $\mathbf{E} \bullet \mathbf{Z} = (\mathbf{E} \times \mathbf{S}) \bullet (\mathbf{M} \times \mathbf{S}) = \mathbf{Y} \bullet \mathbf{Y}$

then:

$$\mathbf{Y} \bullet \mathbf{Y} = \mathbf{E} \bullet \mathbf{Z} = \cos \xi$$

by replacing the corresponding values, we set:

$$\cos \xi = (\mathbf{E} \times \mathbf{S}) \cdot (\mathbf{M} \times \mathbf{S}) / |(\mathbf{M} \times \mathbf{S})|$$

since

$$\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) \cdot \mathbf{C}$$

we found

$$\cos \xi = -(\mathbf{E} \times \mathbf{S}) \times \mathbf{S} / |(\mathbf{M} \times \mathbf{S})|$$

and finally

$\cos \xi = \mathbf{E} \cdot \mathbf{M} / (\mathbf{M} \times \mathbf{S}) $

E and **M** are known since:

1) the direction of ecliptic pole in GEI system is a known constant value:

$$\mathbf{E} = (E_1, E_2, E_3) = (0, -0.398, 0.917)$$

2) **M** is the dipole direction in GEI system, computed § III-5 as:

$$\mathbf{M} = (M_1, M_2, M_3) = ((D_1 \cos \theta - D_2 \sin \theta), (D_1 \sin \theta + D_2 \cos \theta), D_3)$$

3) the geographic coordinates of the dipole axis **D** is computed for a given time and year from **cdipdir** subroutine; for instance for IGRF epoch 1965 we have:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

4) the θ Greenwich Mean Sideral Time is computed for a given time and year from **csundir** subroutine.

Computation of $\sin \xi$

Similarly one has to ensure that the $\mathbf{E} \cdot \mathbf{Y}$ and $-\mathbf{y} \cdot \mathbf{Z}$ terms are equal.

Taking

$$\mathbf{y} \cdot \mathbf{Z} = (\mathbf{E} \times \mathbf{S}) \cdot (\mathbf{S} \times \mathbf{Y})$$

since

$$\mathbf{A} \cdot (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) \cdot \mathbf{C}$$

we have

$$\mathbf{y} \cdot \mathbf{Z} = [(\mathbf{E} \times \mathbf{S}) \times \mathbf{S}] \cdot \mathbf{Y}$$

and find:

$$\mathbf{y} \cdot \mathbf{Z} = -\mathbf{E} \cdot \mathbf{Y} = -\sin \xi$$

Similarly computation of $\sin \xi$ is made as:

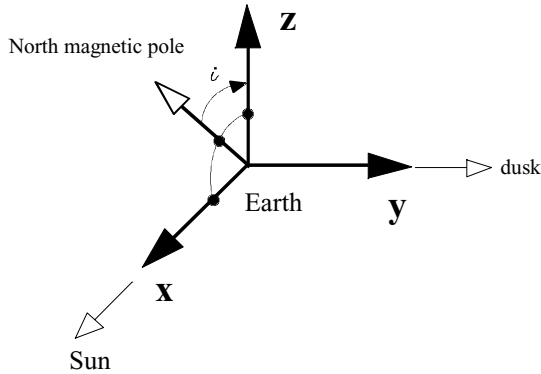
$$\sin \xi = \mathbf{E} \cdot \mathbf{Y}$$

thus

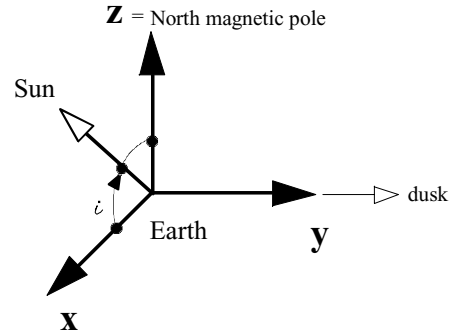
$$\sin \xi = \mathbf{E} \cdot (\mathbf{M} \times \mathbf{S}) / |(\mathbf{M} \times \mathbf{S})|$$

10) GSM to SM transformation

Geocentric Solar Magnetospheric



Solar Magnetic



The GSM and SM system have the Y-axis in common, then the transformation matrix is a simple rotation of μ angle, which is named the dipole tilt angle.

Thus the transformation matrix of any vector V is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)} = \begin{pmatrix} \cos\mu & 0 & -\sin\mu \\ 0 & 1 & 0 \\ \sin\mu & 0 & \cos\mu \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)} = \begin{pmatrix} \cos\mu & 0 & \sin\mu \\ 0 & 1 & 0 \\ -\sin\mu & 0 & \cos\mu \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

μ can be obtained from $\sin \mu = \mathbf{S} \cdot \mathbf{D}$, where \mathbf{S} is the direction of the sun and \mathbf{D} the dipole direction, both in GEO system for instance.

\mathbf{S} can be computed from GEI to GEO transformation given in § III-2; then we have:

$$\mathbf{S} = (S1, S2, S3) = ((S1\cos\theta + S2\sin\theta), (-S1\sin\theta + S2\cos\theta), S3)$$

where S is the direction of the Sun in GEI system, computed from *csundir* subroutine, such as the Greenwich Mean Sideral Time θ .

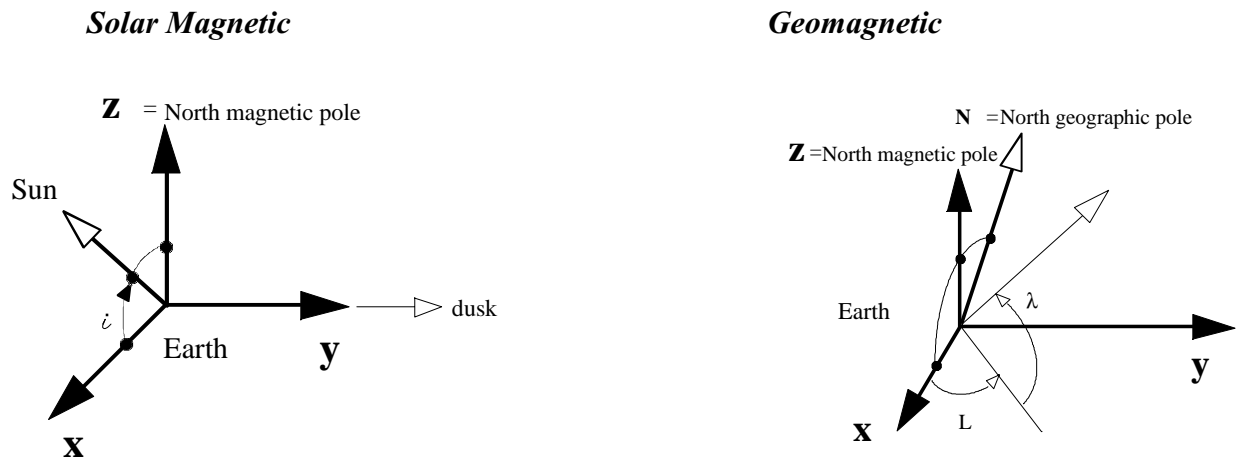
D is obtained from the International Geomagnetic Reference Field (IGRF); practically, **D** is computed for a given time and year from *cdipdir* subroutine; value for 1965.0 is:

$$\mathbf{D} = (D1, D2, D3) = (0.06859, -0.18602, 0.98015)$$

finally, rotation matrix elements are:

$$\begin{aligned} \sin \mu &= S1D1 + S2D2 + S3D3 \\ \cos \mu &= (1 - \sin^2 \mu)^{1/2} \end{aligned}$$

11) SM to MAG transformation



The SM and MAG system have the Z-axis in common, then the transformation matrix is a simple rotation of ϕ angle, thus the transformation matrix of any vector V is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)}$$

Nevertheless the angle ϕ is not derivable from a simple equation.

To compute the rotation terms of transformation matrix, we use the MAG to GEO and the GEO to SM matrix transformations, given in § III-12 and III-13.

The first transformation is noted (see § III-12):

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ D1 & D2 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

with $\mathbf{X} = (\mathbf{Y} \times \mathbf{D})$ in GEO System
and $\mathbf{Y} = (\mathbf{N} \times \mathbf{D}) / |\mathbf{N} \times \mathbf{D}|$

N and **D** vector are respectively the North geographic pole and the magnetic dipole in geographic system.

Practically, geographic dipole direction **D** is computed for a given time and year from **cdipdir** subroutine; value for 1965.0 is:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

we deduce from $\mathbf{Y} = (\mathbf{N} \times \mathbf{D}) / |\mathbf{N} \times \mathbf{D}|$:

$$\mathbf{Y} = (-D_2, D_1, 0) / (D_1^2 + D_2^2)^{1/2}$$

The second transformation is noted (see § III-13):

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} x1 & y1 & D1 \\ x2 & y2 & D2 \\ x3 & y3 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

$$\text{with } \mathbf{x} = (\mathbf{y} \times \mathbf{D}) \text{ in GEO system} \\ \mathbf{y} = (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}|$$

S is the direction of the sun in GEO system, which can be computed from GEI to GEO transformation given in § III-2, then we have:

$$\mathbf{S} = (S1, S2, S3) = ((S1 \cos \theta + S2 \sin \theta), (-S1 \sin \theta + S2 \cos \theta), S3)$$

S is the direction of the Sun in GEI system, computed from **csundir** subroutine, such as the Greenwich Mean Sideral Time θ .

Knowing all elements for MAG to GEO and GEO to SM transformations, we can write the SM to MAG transformation as:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ D1 & D2 & D3 \end{pmatrix} \begin{pmatrix} x1 & y1 & D1 \\ x2 & y2 & D2 \\ x3 & y3 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

which give

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} \mathbf{X} \cdot \mathbf{x} & \mathbf{X} \cdot \mathbf{y} & \mathbf{X} \cdot \mathbf{D} \\ \mathbf{Y} \cdot \mathbf{x} & \mathbf{Y} \cdot \mathbf{y} & \mathbf{Y} \cdot \mathbf{D} \\ \mathbf{D} \cdot \mathbf{x} & \mathbf{D} \cdot \mathbf{y} & \mathbf{D} \cdot \mathbf{D} \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

In other hand we can write the following equivalences:

$$\mathbf{X} \cdot \mathbf{D} = (\mathbf{Y} \times \mathbf{D}) \cdot \mathbf{D} = \mathbf{Y} \cdot (\mathbf{D} \times \mathbf{D}) = 0$$

$$\mathbf{Y} \cdot \mathbf{D} = (\mathbf{N} \times \mathbf{D}) \cdot \mathbf{D} / |\mathbf{N} \times \mathbf{D}| = \mathbf{N} \cdot (\mathbf{D} \times \mathbf{D}) / |\mathbf{N} \times \mathbf{D}| = 0$$

$$\mathbf{D} \cdot \mathbf{x} = \mathbf{D} \cdot (\mathbf{y} \times \mathbf{D}) = -\mathbf{D} \cdot (\mathbf{D} \times \mathbf{y}) = -(\mathbf{D} \times \mathbf{D}) \cdot \mathbf{y} = 0$$

$$\mathbf{D} \cdot \mathbf{y} = \mathbf{D} \cdot (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}| = (\mathbf{D} \times \mathbf{D}) \cdot \mathbf{S} / |\mathbf{D} \times \mathbf{S}| = 0$$

$$\mathbf{D} \cdot \mathbf{D} = 1$$

then we have the following matrix:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(MAG)} = \begin{pmatrix} \mathbf{x} \cdot \mathbf{x} & \mathbf{x} \cdot \mathbf{y} & 0 \\ \mathbf{y} \cdot \mathbf{x} & \mathbf{y} \cdot \mathbf{y} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

Computation of $\cos \varphi$

We have to equalize $\mathbf{X} \cdot \mathbf{x}$ and $\mathbf{Y} \cdot \mathbf{y}$ terms as $\cos \varphi$.

Taking
$$\mathbf{X} \cdot \mathbf{x} = (\mathbf{Y} \times \mathbf{D}) \cdot (\mathbf{y} \times \mathbf{D}) = -(\mathbf{Y} \times \mathbf{D}) \cdot (\mathbf{D} \times \mathbf{y})$$

$$\mathbf{X} \cdot \mathbf{x} = -(\mathbf{Y} \times \mathbf{D}) \times \mathbf{D} \cdot \mathbf{y}$$

and since \mathbf{Y} and \mathbf{D} are perpendicular:

$$\boxed{\mathbf{X} \cdot \mathbf{x} = \mathbf{Y} \cdot \mathbf{y} = \cos \varphi}$$

coordinates of \mathbf{Y} axe is given in § III-12 as:

$$\mathbf{Y} = (-D_2, D_1, 0) / (D_1^2 + D_2^2)^{1/2}$$

and

$$\mathbf{y} = (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}|$$

then we have:

$$\cos \varphi = \mathbf{Y} \cdot \mathbf{y} = \mathbf{Y} \cdot (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}| = (\mathbf{Y} \times \mathbf{D}) \cdot \mathbf{S} / |\mathbf{D} \times \mathbf{S}|$$

The $(\mathbf{Y} \times \mathbf{D})$ vector can be expanded as:

$$(\mathbf{Y} \times \mathbf{D}) = (D_1 D_3, D_2 D_3, -(D_1^2 + D_2^2)) / (D_1^2 + D_2^2)^{1/2}$$

and we deduce:

$$\boxed{\cos \varphi = [(D_1 D_3 S_1 + D_2 D_3 S_2 - (D_1^2 + D_2^2) S_3)] / Q}$$

with

$$Q = (D_1^2 + D_2^2)^{1/2} \cdot [(D_2 S_3 - D_3 S_2)^2 + (D_3 S_1 - D_1 S_3)^2 + (D_1 S_2 - D_2 S_1)^2]^{1/2}$$

$$(S_1, S_2, S_3) = ((S_1 \cos \theta + S_2 \sin \theta), (-S_1 \sin \theta + S_2 \cos \theta), S_3)$$

Computation of $\sin \varphi$

We have to equalize $\mathbf{X} \cdot \mathbf{y}$ and $-\mathbf{Y} \cdot \mathbf{x}$ terms as $\sin \varphi$.

Taking $\mathbf{X} \cdot \mathbf{y} = (\mathbf{Y} \times \mathbf{D}) \cdot \mathbf{y}$

and $-\mathbf{Y} \cdot \mathbf{x} = -\mathbf{Y} \cdot (\mathbf{y} \times \mathbf{D}) = \mathbf{Y} \cdot (\mathbf{D} \times \mathbf{y}) = (\mathbf{Y} \times \mathbf{D}) \cdot \mathbf{y}$

yet we have well:

$$\mathbf{X} \cdot \mathbf{y} = -\mathbf{Y} \cdot \mathbf{x} = \sin \varphi$$

$\sin \varphi$ is then computed from $\sin \varphi = (\mathbf{Y} \times \mathbf{D}) \cdot \mathbf{y}$

from above we have:

$$(\mathbf{Y} \times \mathbf{D}) = (D_1 D_3, D_2 D_3, -(D_1^2 + D_2^2)) / (D_1^2 + D_2^2)^{1/2}$$

and

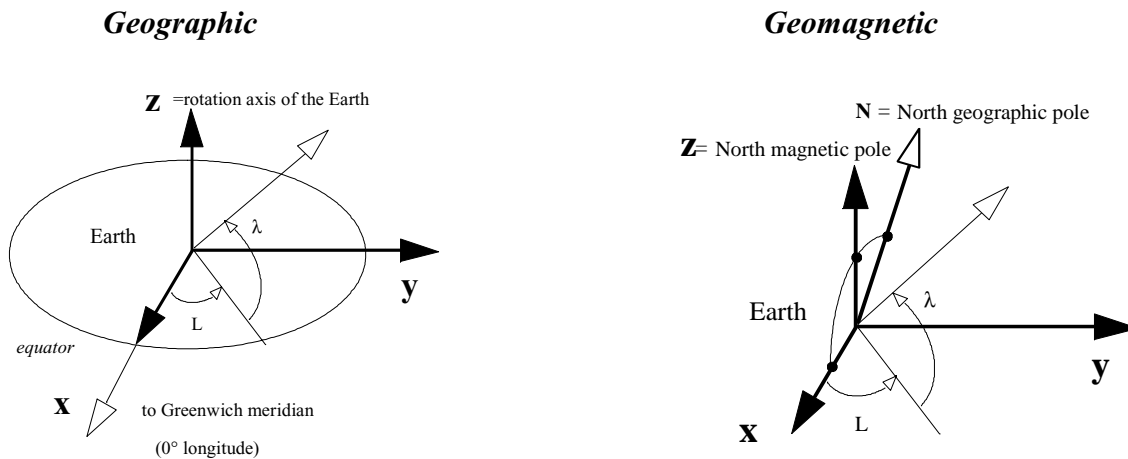
$$\mathbf{y} = (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}|$$

we deduce:

$$\sin \varphi = (D_2 S_1 - D_1 S_2) / Q$$

with Q such as above.

12) GEO to MAG transformation



In geomagnetic coordinates, Z-axis is parallel to the magnetic dipole axis \mathbf{D} , and the Y-axis is perpendicular to the North geographic pole \mathbf{N} , then we have in geographic system:

$$\begin{aligned} \mathbf{Z} &= \mathbf{D} \\ \mathbf{Y} &= (\mathbf{N} \times \mathbf{D}) / |\mathbf{N} \times \mathbf{D}| \\ \text{and } \mathbf{X} &= \mathbf{Y} \times \mathbf{D} \end{aligned}$$

The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude.

Practically, geographic dipole direction \mathbf{D} is computed for a given time and year from *cdipdir* subroutine; value for 1965.0 is:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

we deduce: $\mathbf{Y} = (-D_2, D_1, 0) / (D_1^2 + D_2^2)^{1/2}$

and: $\mathbf{X} = (Y_2 D_3, -Y_1 D_3, Y_1 D_2 - Y_2 D_1)$

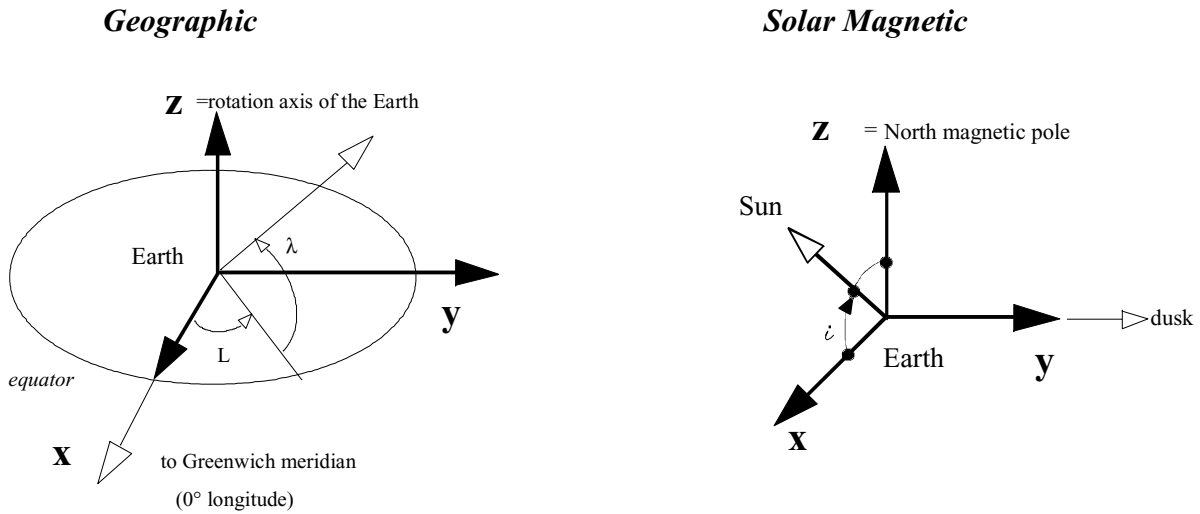
Thus the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(\text{MAG})} = \begin{pmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ D_1 & D_2 & D_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(\text{GEO})}$$

Similarly the transformation from system MAG to GEO is:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(\text{GEO})} = \begin{pmatrix} X_1 & Y_1 & D_1 \\ X_2 & Y_2 & D_2 \\ X_3 & Y_3 & D_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(\text{MAG})}$$

13) GEO to SM transformation



In GEO system, the direction of the Z-axis of SM system is the dipole direction \mathbf{D} . The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, \mathbf{D} is computed for a given time and year from *cdipdir* subroutine.

We can then deduce in GEO system the Y-axis of SM system as:

$$\mathbf{y} = (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}|$$

where \mathbf{S} is the direction of the sun in GEO system, which can be computed from GEI to GEO transformation given in § III-2, then we have:

$$\mathbf{S} = (S_1, S_2, S_3) = ((S_1 \cos \theta + S_2 \sin \theta), (-S_1 \sin \theta + S_2 \cos \theta), S_3)$$

\mathbf{S} is the direction of the Sun in GEI system, computed from *csundir* subroutine, such as the Greenwich Mean Sideral Time θ ; normalizing factors occurs because \mathbf{D} and \mathbf{S} are not necessarily perpendicular.

The third axis X is computed from:

$$\mathbf{x} = (\mathbf{y} \times \mathbf{D})$$

and the GEO to SM transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)} = \begin{pmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ D1 & D2 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

Similarly the SM to GEO transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} x1 & y1 & D1 \\ x2 & y2 & D2 \\ x3 & y3 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(SM)}$$

with respectively:

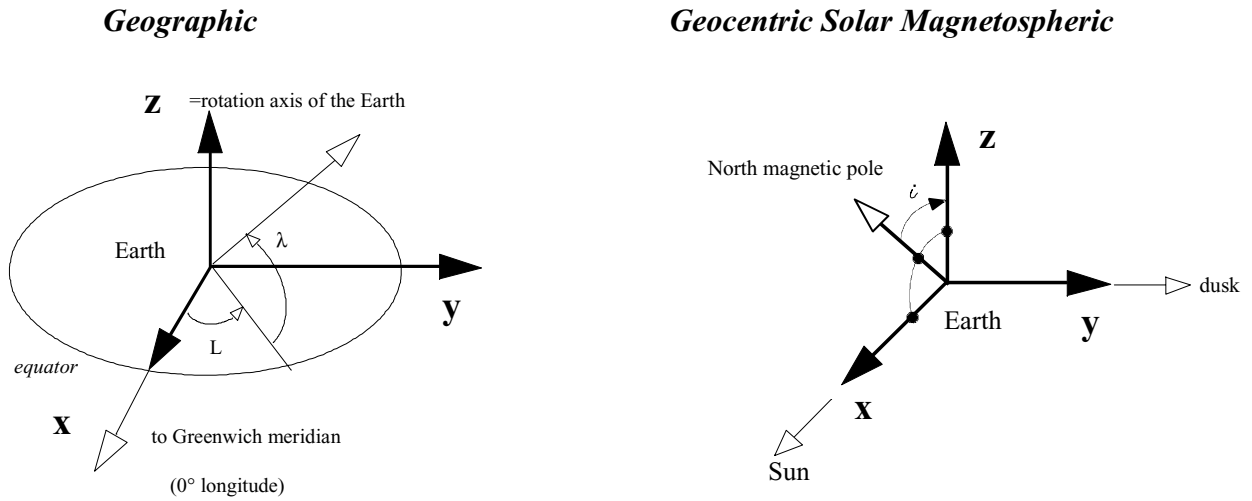
$$\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} y2D3 - y3D2 \\ y3D1 - y1D3 \\ y1D2 - y2D1 \end{pmatrix}$$

$$\begin{pmatrix} y1 \\ y2 \\ y3 \end{pmatrix} = \begin{pmatrix} D2S3 - D3S2 \\ D3S1 - D1S3 \\ D1S2 - D2S1 \end{pmatrix} \cdot 1/Q$$

$$Q = [(D2S3 - D3S2)^2 + (D3S1 - D1S3)^2 + (D1S2 - D2S1)^2]^{1/2}$$

$$\mathbf{S} = (S1, S2, S3) = ((S1\cos\theta + S2\sin\theta) , (-S1\sin\theta + S2\cos\theta) , S3)$$

14) GEO to GSM transformation



In GEO system, the direction of the X-axis of GSM system is the direction of the sun \mathbf{S} , which can be computed from GEI to GEO transformation given in § III-2; then we have:

$$\mathbf{x} = \mathbf{S} = (S_1, S_2, S_3) = ((S_1 \cos \theta + S_2 \sin \theta), (-S_1 \sin \theta + S_2 \cos \theta), S_3)$$

\mathbf{S} is the direction of the Sun in GEI system, computed from *csundir* subroutine, such as the Greenwich Mean Sidereal Time θ .

The Y axis in GEO system can be deduced from:

$$\mathbf{y} = (\mathbf{D} \times \mathbf{S}) / |\mathbf{D} \times \mathbf{S}|$$

where \mathbf{D} is the dipole direction; the geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, \mathbf{D} is computed for a given time and year from *cdipdir* subroutine; normalizing factors in cross product occurs because \mathbf{D} and \mathbf{S} are not necessarily perpendicular.

And finally the third axis Z is computed from:

$$\mathbf{z} = \mathbf{S} \times \mathbf{y}$$

and the GEO to GSM transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)} = \begin{pmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ z1 & z2 & z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

Similarly the GSM to GEO transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSM)}$$

with respectively:

$$\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} S1 \\ S2 \\ S3 \end{pmatrix}$$

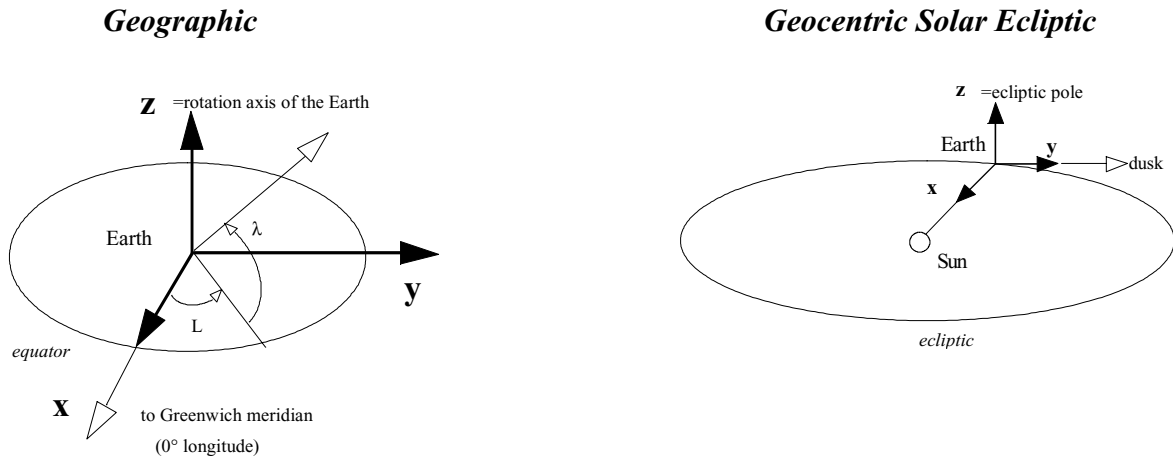
$$\begin{pmatrix} y1 \\ y2 \\ y3 \end{pmatrix} = \begin{pmatrix} D2S3 - D3S2 \\ D3S1 - D1S3 \\ D1S2 - D2S1 \end{pmatrix} \cdot 1/Q$$

$$\begin{pmatrix} z1 \\ z2 \\ z3 \end{pmatrix} = \begin{pmatrix} x2y3 - x3y2 \\ x3y1 - x1y3 \\ x1y2 - x2y1 \end{pmatrix}$$

$$Q = [(D2S3 - D3S2)^2 + (D3S1 - D1S3)^2 + (D1S2 - D2S1)^2]^{1/2}$$

$$\mathbf{S} = (S1, S2, S3) = ((S1 \cos \theta + S2 \sin \theta), (-S1 \sin \theta + S2 \cos \theta), S3)$$

15) GEO to GSE transformation



In GEO system, the direction of the X-axis of GSE system is the direction of the sun \mathbf{S} , which can be computed from GEI to GEO transformation given in § III-2; then we have:

$$\mathbf{X} = \mathbf{S} = (S_1, S_2, S_3) = (S_1 \cos \theta + S_2 \sin \theta, -S_1 \sin \theta + S_2 \cos \theta, S_3)$$

\mathbf{S} is the direction of the Sun in GEI system, computed from *csundir* subroutine, such as the Greenwich Mean Sideral Time θ .

The Z axis is the direction of the ecliptic pole, which is a known constant value in GEI system:

$$\mathbf{E} = (E_1, E_2, E_3) = (0, -0.398, 0.917)$$

from GEI to GEO transformation we have:

$$\mathbf{Z} = \mathbf{E} = (E_1, E_2, E_3) = (E_1 \cos \theta + E_2 \sin \theta, -E_1 \sin \theta + E_2 \cos \theta, E_3)$$

And finally the Y axis in GEO system can be deduced from:

$$\mathbf{Y} = \mathbf{E} \times \mathbf{S}$$

and the GEO to GSE transformation is given by:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GSE)} = \begin{pmatrix} S_1 & S_2 & S_3 \\ Y_1 & Y_2 & Y_3 \\ E_1 & E_2 & E_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GEO)}$$

Similarly the GSE to GEO transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} S1 & Y1 & E1 \\ S2 & Y2 & E2 \\ S3 & Y3 & E3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSE)}$$

with respectively:

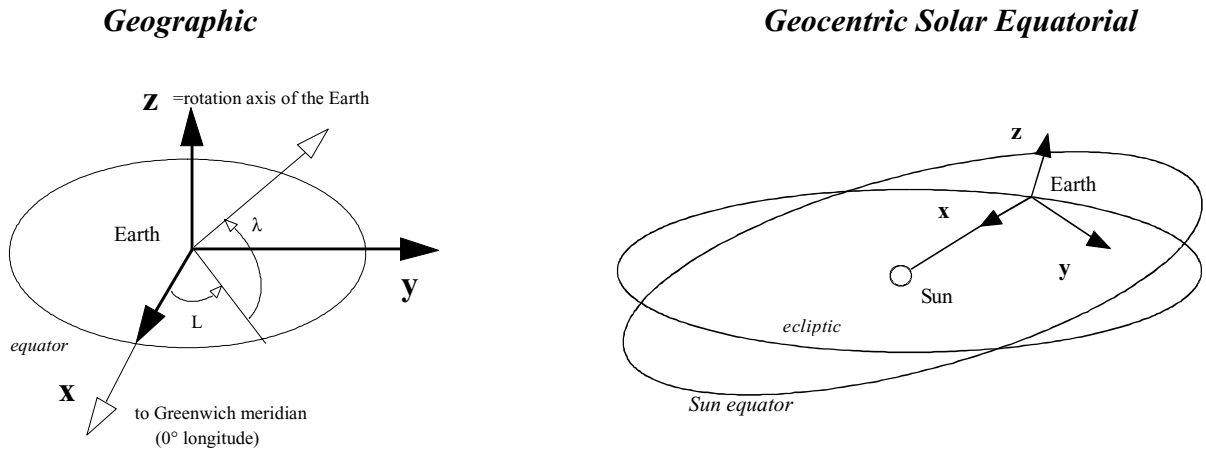
$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix} = \begin{pmatrix} E2S3 - E3S2 \\ E3S1 - E1S3 \\ E1S2 - E2S1 \end{pmatrix}$$

$$\mathbf{S} = (S1, S2, S3) = ((S1\cos\theta + S2\sin\theta), (-S1\sin\theta + S2\cos\theta), S3)$$

$$\mathbf{E} = (E1, E2, E3) = ((E1\cos\theta + E2\sin\theta), (-E1\sin\theta + E2\cos\theta), E3)$$

$$\mathbf{E} = (E1, E2, E3) = (0, -0.398, 0.917)$$

16) GEO to GSEQ transformation



In GEO system, the direction of the X-axis of GSEQ system is the direction of the sun \mathbf{S} , which can be computed from GEI to GEO transformation given in § III-2; then we have:

$$\mathbf{x} = \mathbf{S} = (S_1, S_2, S_3) = ((S_1 \cos \theta + S_2 \sin \theta), (-S_1 \sin \theta + S_2 \cos \theta), S_3)$$

\mathbf{S} is the direction of the Sun in GEI system, computed from *csundir* subroutine, such as the Greenwich Mean Sideral Time θ .

The Sun equator plane is defined from the direction of the rotation axis of the SUN which is in GEI system a known constant value:

$$\mathbf{R} = (R_1, R_2, R_3) = (0.122, -0.424, 0.899)$$

from GEI to GEO transformation we obtain this vector in GEO system as:

$$\mathbf{R} = (R_1, R_2, R_3) = ((R_1 \cos \theta + R_2 \sin \theta), (-R_1 \sin \theta + R_2 \cos \theta), R_3)$$

Then the Y axis in GEO system can be deduced from:

$$\mathbf{y} = \mathbf{R} \times \mathbf{S} / |\mathbf{R} \times \mathbf{S}|$$

normalizing factors in cross product occurs because \mathbf{R} and \mathbf{S} are not necessarily perpendicular.

The third axis Z is computed from:

$$\mathbf{z} = \mathbf{S} \times \mathbf{y}$$

and the GEO to GSEQ transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)} = \begin{pmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ z1 & z2 & z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

Similarly the GEQ to GEO transformation is given by:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GSEQ)}$$

with respectively:

$$\begin{pmatrix} x1 \\ x2 \\ x3 \end{pmatrix} = \begin{pmatrix} S1 \\ S2 \\ S3 \end{pmatrix}$$

$$\begin{pmatrix} y1 \\ y2 \\ y3 \end{pmatrix} = \begin{pmatrix} R2S3 & -R3S2 \\ R3S1 & -R1S3 \\ R1S2 & -R2S1 \end{pmatrix} \cdot 1/Q$$

$$\begin{pmatrix} z1 \\ z2 \\ z3 \end{pmatrix} = \begin{pmatrix} S2y3 - S3y2 \\ S3y1 - S1y3 \\ S1y2 - S2y1 \end{pmatrix}$$

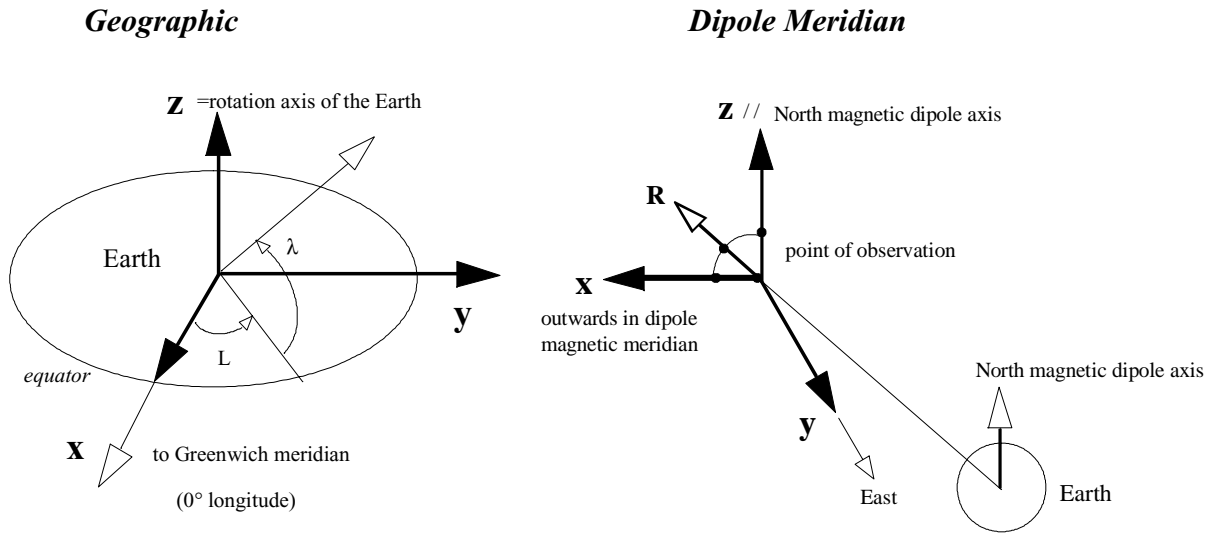
$$Q = [(R2S3 - R3S2)^2 + (R3S1 - R1S3)^2 + (R1S2 - R2S1)^2]^{1/2}$$

$$\mathbf{S} = (S1, S2, S3) = ((S1 \cos \theta + S2 \sin \theta), (-S1 \sin \theta + S2 \cos \theta), S3)$$

$$\mathbf{R} = (R1, R2, R3) = ((R1 \cos \theta + R2 \sin \theta), (-R1 \sin \theta + R2 \cos \theta), R3)$$

$$\mathbf{R} = (R1, R2, R3) = (0.122, -0.424, 0.899)$$

17) GEO to DM transformation



Dipole meridian system is a local coordinate system, and varies with the position of the point of observation relative to the centre of the Earth; this position is noted in GEO system as:

$$\mathbf{R} = (R_1, R_2, R_3)$$

To transform GEO coordinate to DM coordinates, we need the dipole position in GEO system which is the Z axis of DM system. The geographic coordinates of the dipole axis can be known, for instance for IGRF epoch 1965, as 11.435° colatitude and -69.761° east longitude, thus:

$$\mathbf{Z} = \mathbf{D} = (D_1, D_2, D_3) = (0.06859, -0.18602, 0.98015)$$

Practically, \mathbf{D} is computed for a given time and year from *cdipdir* subroutine.

We can deduce then the Y-axis of DM system in GEO coordinates as:

$$\mathbf{Y} = \mathbf{D} \times \mathbf{R} / |\mathbf{D} \times \mathbf{R}|$$

normalizing factors occurs because \mathbf{D} and \mathbf{R} are not necessarily perpendicular and because \mathbf{R} is not a unit vector.

The third axis X is deduced from: $\mathbf{X} = \mathbf{Y} \times \mathbf{D}$

All coordinates of X-Y-Z axis of DM system in GEO coordinates being known, the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(DM)} = \begin{pmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ D_1 & D_2 & D_3 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix}_{(GEO)}$$

Similarly the transformation from system DM to GEO is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} X1 & Y1 & D1 \\ X2 & Y2 & D2 \\ X3 & Y3 & D3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(DM)}$$

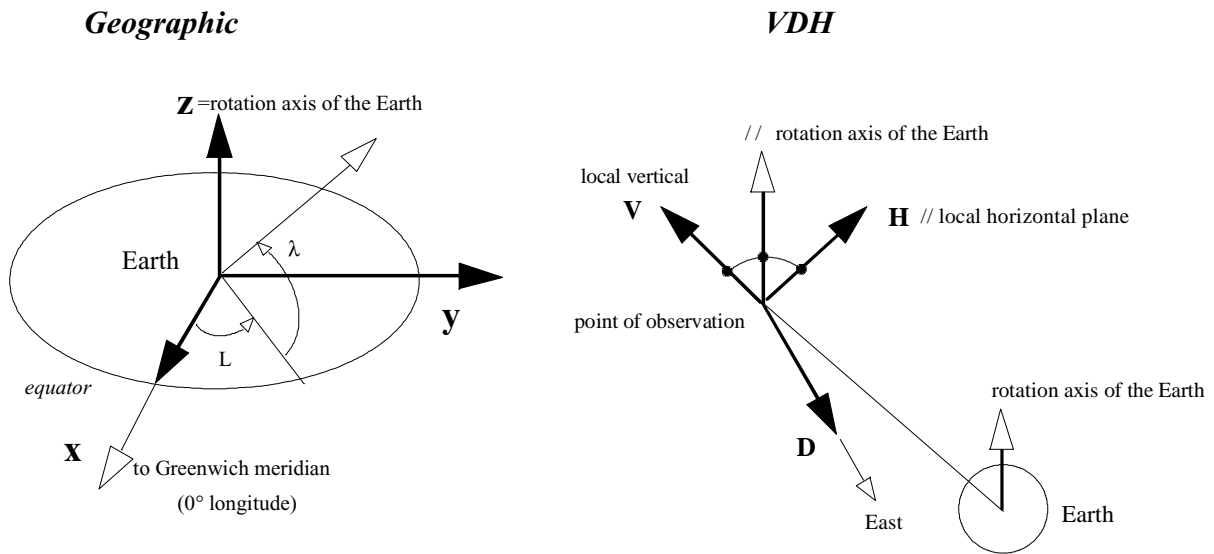
with respectively:

$$\begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = \begin{pmatrix} Y2D3 - Y3D2 \\ Y3D1 - Y1D3 \\ Y1D2 - Y2D1 \end{pmatrix}$$

$$\begin{pmatrix} Y1 \\ Y2 \\ Y3 \end{pmatrix} = \begin{pmatrix} D2R3 - D3R2 \\ D3R1 - D1R3 \\ D1R2 - D2R1 \end{pmatrix} \cdot 1/Q$$

$$Q = \left[(D2R3 - D3R2)^2 + (D3R1 - D1R3)^2 + (D1R2 - D2R1)^2 \right]^{1/2}$$

18) GEO to VDH transformation



VDH system is a local coordinate system, and varies with the position of the point of observation relative to the centre of the Earth; this position is noted in GEO system as:

$$\mathbf{R} = (R_1, R_2, R_3)$$

and we have directly

$$\mathbf{V} = \mathbf{R} / (R_1^2 + R_2^2 + R_3^2)^{1/2}$$

Since D is perpendicular to V and to the rotation axis of the Earth, we have:

$$\mathbf{D} = \mathbf{Z}_E \times \mathbf{R} / |\mathbf{Z}_E \times \mathbf{R}|$$

which give:

$$\mathbf{D} = \begin{pmatrix} -R_2 \\ R_1 \\ 0 \end{pmatrix} \cdot 1 / (R_1^2 + R_2^2)^{1/2}$$

The third axis H is deduced from:

$$\mathbf{H} = \mathbf{V} \times \mathbf{D}$$

which give:

$$\mathbf{H} = \begin{pmatrix} -R_1 R_3 \\ -R_2 R_3 \\ R_1^2 + R_2^2 \end{pmatrix} \cdot 1 / Q$$

with

$$Q = [(R_1^2 + R_2^2)(R_1^2 + R_2^2 + R_3^2)]^{1/2}$$

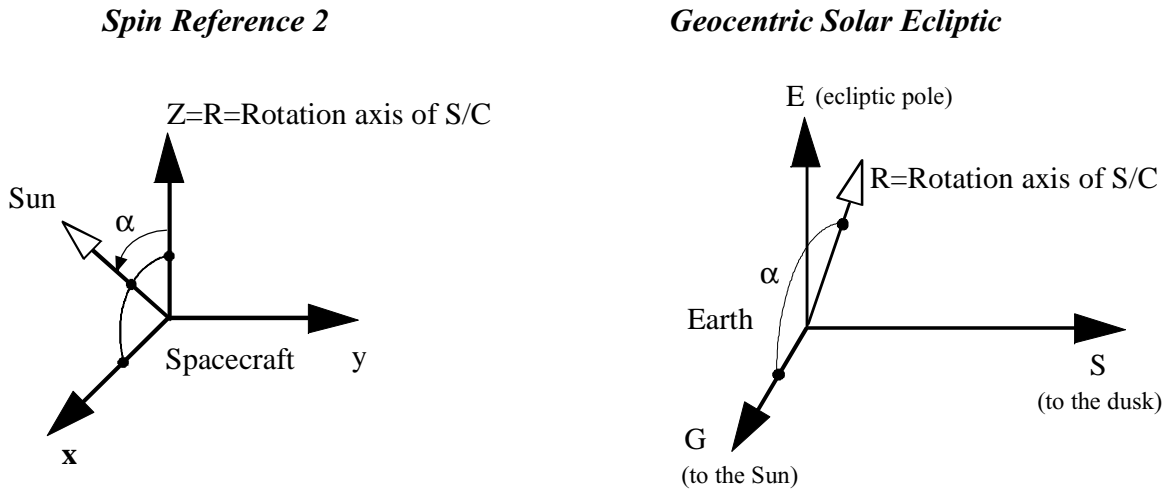
All coordinates of X-Y-Z axis of VDH system in GEO coordinates being known, the transform matrix of any vector \mathbf{V} is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(VDH)} = \begin{pmatrix} V1 & V2 & V3 \\ D1 & D2 & D3 \\ H1 & H2 & H3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)}$$

Similarly the transformation from system VDH to GEO is:

$$\begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(GEO)} = \begin{pmatrix} V1 & D1 & H1 \\ V2 & D2 & H2 \\ V3 & D3 & H3 \end{pmatrix} \begin{pmatrix} V1 \\ V2 \\ V3 \end{pmatrix}_{(VDH)}$$

19) SR2 to GSE transformation



The transformation of any vector expressed in GSE system into the SR2 system can be written as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} X1 & X2 & X3 \\ Y1 & Y2 & Y3 \\ Z1 & Z2 & Z3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE}$$

where $(X1, X2, X3)$ are the components of the \mathbf{X} axis of the SR2 system expressed in GSE coordinates, and in the same way for \mathbf{Y} and \mathbf{Z} axis.

In SR2 system, \mathbf{Z} is the rotation axis of the spacecraft, and must be known in GSE system from auxiliary data of the Spacecraft. We can write it as:

$$\mathbf{Z} = \begin{pmatrix} Z1 \\ Z2 \\ Z3 \end{pmatrix} = \mathbf{R} = \begin{pmatrix} Rx \\ Ry \\ Rz \end{pmatrix}_{GSE}$$

The components of the \mathbf{Y} axis can be found by the relation $\mathbf{Y} = (\mathbf{R} \times \mathbf{S}) / |\mathbf{R} \times \mathbf{S}|$ which is very simple since the direction of the Sun \mathbf{S} is the \mathbf{X} axis of GSE system, expressed as $(1,0,0)$, and we deduce:

$$\mathbf{Y} = \frac{1}{\sqrt{Ry^2 + Rz^2}} \begin{pmatrix} 0 \\ Rz \\ -Ry \end{pmatrix}_{GSE}$$

Finally we complete the system by computing $\mathbf{X} = \mathbf{Y} \times \mathbf{Z}$, so:

$$\mathbf{X} = a \begin{pmatrix} Ry^2 + Rz^2 \\ -RxRy \\ -RxRz \end{pmatrix}_{GSE} \quad \text{with } a = \frac{1}{\sqrt{Ry^2 + Rz^2}}$$

Thus, the transformation of GSE coordinate into SR2 coordinates can be written as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} a(Ry^2 + Rz^2) & -a(RxRy) & -aRxRz \\ 0 & aRz & -aRy \\ Rx & Ry & Rz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE}$$

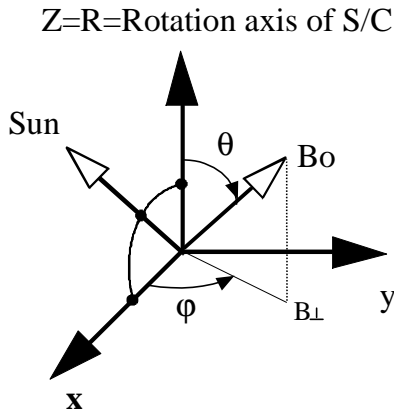
where **R** vector is the spin axis (or Rotation axis) expressed in GSE coordinate system, and extracted from auxiliary SPD data.

Note that, if useful, the reverse transform is simply:

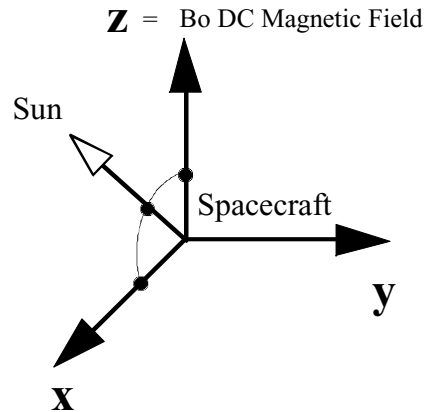
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{GSE} = \begin{pmatrix} a(Ry^2 + Rz^2) & 0 & Rx \\ -aRxRy & aRz & Ry \\ -aRxRz & -aRy & Rz \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

20) SR2 to MFA transformation

Spin Reference 2

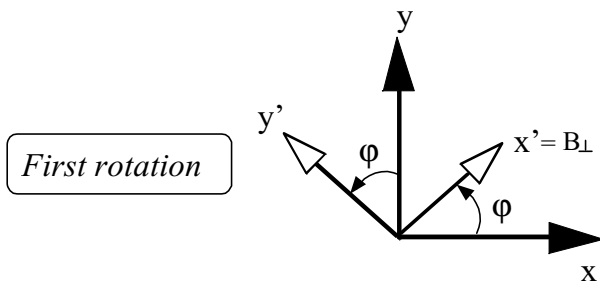


Magnetic Field Aligned



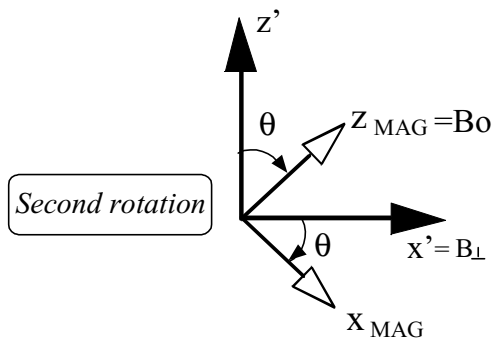
The Magnetic Field Aligned system is defined as a system where Z is along the DC magnetic field, noted as Bo, and X in the plane containing the DC magnetic field vector and the direction of the Sun. A problem of definition can appear when the DC-MF is aligned with the direction of the Sun, which is not an unlikely case. When this situation arrives, the direction of the normal to the ecliptic plane could be used instead of the direction of the Sun, but is not still the case in this version.

To obtain the alignment of Z with Bo, we apply two rotations. The first one, in the x-y plane, align the x axis with the perpendicular component of Bo in the x-y plane:



$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

A second rotation, in the x'-z' plane, leads the new z axis to be aligned with Bo:



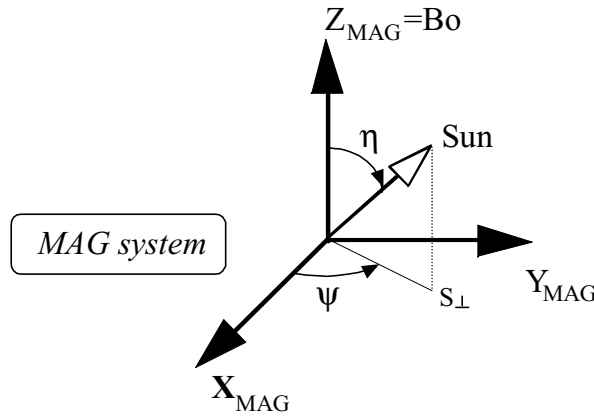
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MAG} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}$$

and the transformation from SR2 to a new MAG is given by:

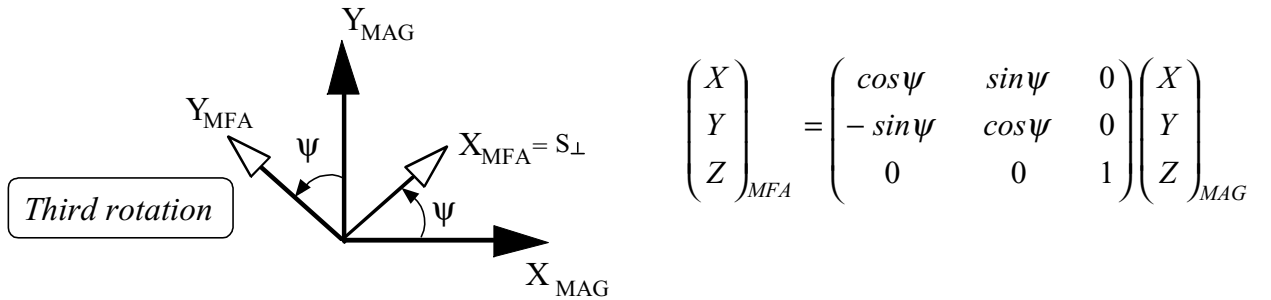
$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MAG} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

At this level, the Z axis of new MAG system is aligned with the Bo DC local magnetic field, but the MAG system is still not the MFA system, because the MFA system is defined as having the X in the plane containing the DC magnetic field vector and the direction of the Sun. So we must apply a third rotation to pass from the MAG system to the MFA system.

This third rotation in the x-y plane depends of the direction of the Sun in the MAG system, defined by its spherical angles η and ψ :



and we pass from the MAG system to the MFA system by a rotation of ψ angle around the Z axis:



So, the total transformation from SR2 system to MFA system is:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{MFA} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

Computation of the angles θ and φ

θ and φ are respectively the polar and azimuthal angles of the Bo DC magnetic field vector in SR2 system, which has been computed in section 4. If we note Bx, By and Bz the components of Bo in SR2 system, we have:

$$\sin\theta = \frac{\sqrt{Bx^2 + By^2}}{Bo}, \quad \cos\theta = \frac{Bz}{Bo}$$

$$\sin\varphi = \frac{By}{\sqrt{Bx^2 + By^2}}, \quad \cos\varphi = \frac{Bx}{\sqrt{Bx^2 + By^2}}$$

Computation of the angle ψ

ψ is the azimuthal angle of the direction of the Sun in MAG system . If we note Sx_M, Sy_M, Sz_M the Cartesian components of this direction in MAG system, we have:

$$\sin\psi = \frac{Sy_M}{\sqrt{Sx_M^2 + Sy_M^2}}, \quad \cos\psi = \frac{Sx_M}{\sqrt{Sx_M^2 + Sy_M^2}}$$

Computation of Sx_M, Sy_M, Sz_M .

To compute the components of the direction of the Sun in MAG system, we use the transformation from SR2 to MAG, computed in the beginning of this annexe, as:

$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2}$$

where Sx, Sy, Sz are the components of the direction of the Sun in SR2 system, which leads to:

$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} Sx \cos\theta \cos\varphi - Sz \sin\theta \\ -Sx \sin\varphi \\ Sx \sin\theta \cos\varphi - Sz \cos\theta \end{pmatrix}$$

Sx, Sy, Sz are deduced by the transformation of GSE to SR2 given in annexe 1, thus:

$$\begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2} = \begin{pmatrix} a(Ry^2 + Rz^2) & -a(RxRy) & -aRxRz \\ 0 & aRz & -aRy \\ Rx & Ry & Rz \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}_{GSE} \quad \text{with } a = \frac{1}{\sqrt{Ry^2 + Rz^2}}$$

R vector being the Rotation axis expressed in GSE coordinate. Note that in GSE, the direction of the Sun is simply (1,0,0), thus the direction of the Sun in SR2 system is:

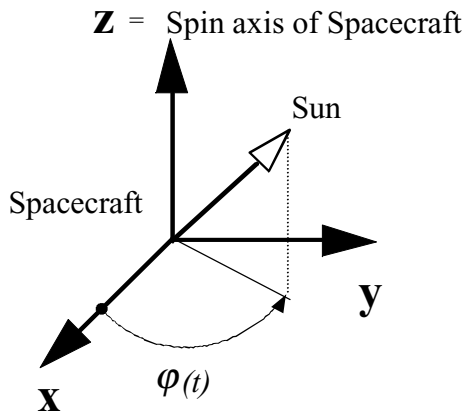
$$\begin{pmatrix} Sx \\ Sy \\ Sz \end{pmatrix}_{SR2} = \begin{pmatrix} \sqrt{Ry^2 + Rz^2} \\ 0 \\ Rx \end{pmatrix}$$

which leads to the result:

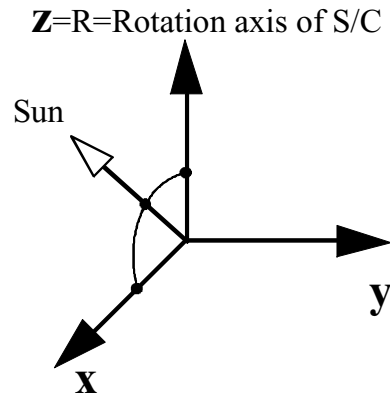
$$\begin{pmatrix} Sx_M \\ Sy_M \\ Sz_M \end{pmatrix}_{MAG} = \begin{pmatrix} \sqrt{Ry^2 + Rz^2} \cos\theta \cos\varphi - Rx \sin\theta \\ -\sqrt{Ry^2 + Rz^2} \sin\varphi \\ \sqrt{Ry^2 + Rz^2} \sin\theta \cos\varphi - Rx \cos\theta \end{pmatrix}$$

21) SR to SR2 transformation

Spin Reference System



Spin Reference 2 System



By the way of the Sun Reference Pulse, we know the spin phase $\varphi(t_0)$ at a given time t_0 . The spin phase $\varphi(t_0)$ is the azimuthal angle of the direction of the Sun in the SR system, for the known time t_0 . At any time t , knowing the spin frequency f_s , the $\varphi(t)$ angle can be computed by:

$$\varphi(t) = \varphi(t_0) - 2\pi f_s (t - t_0)$$

The spacecraft is supposed rotating in the direct x to y sens. If it rotates in the other sens, the sign of f_s must be changed. Note that at each time multiple of $t = t_0 + k/f_s$ (k being any signed integer), $\varphi(t)$ has the same value as $\varphi(t_0)$.

The only one difference between the SR system and the SR2 system is a rotation of the φ angle around the spin axis, so we have:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2} = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR}$$

and the inverse transformation is obviously:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR} = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{SR2}$$

IV- CALENDAR CONVERSIONS

1) General remarks

Most space missions deliver or use for their processing data date and time in various format, such as decimal Julian day, month-day in the month, year, decimal hours, hour-min-sec, and so on. In order to easily manipulate all these quantities, and to get transformations between else, special calendar conversions, described below, has been added to ROCOTLIB library.

2) List of calendar conversions and definitions

- *compute if a given year is or not a leap year (coleapy)*

A given year (as 1990) is a leap year if this number is divisible by 4; if the year is a secular year (as 1900 or 2000), this year is a leap year if this number is divisible by 400.

For instance, 1988, 1992, 2000 are leap years, 1900, 1989, 2100 are not.

- *compute day of the year from given date (cdoyear)*

This conversion compute the day of the year, equal to 1 for January 1, for a date given as month, day in the month, and year.

For instance, the day of the year of October 17, 1990 is 290. This conversion counts the number of the day in a month, taking into account leap year since February has not a constant number of days (using coleapy).

- *compute date from given day of the year and year (cdatdoy)*

This is the inverse conversion of cdoyear, since one compute the corresponding month and day in the month for a given day of the year, and a given year. Year is necessary to know if the year is or not a leap year (using coleapy).

- *compute Julian day 1950 from given date (cjd1950)*

From a given date as month, day, year, this conversion compute the Julian day , i.e. day number from January 1, 1950, with Julian day equal to 1 for January 1, 1950 (use cdoyear and coleapy conversions).

- ***compute Julian day 2000 from given date (cjd2000)***

From a given date as month, day, year, this conversion compute the Julian day , i.e. day number from January 1, 2000, with Julian day equal to 1 for January 1, 2000 (use cdoyear and coleapy conversions).

- ***compute date from given 1950 Julian day (cdatj50)***

This is the inverse conversion of cjd1950, since one compute for a given Julian day the corresponding date as month, day in the month and year (use coleapy and cdatdoy conversions).

- ***compute date from given 2000 Julian day (cdatj00)***

This is the inverse conversion of cjd2000, since one compute for a given Julian day the corresponding date as month, day in the month and year (use coleapy and cdatdoy conversions).

- ***compute decimal hour from given hour, minute and second (chouday)***

This conversion makes the operation $hour + minute/60. + second/3600.$

- ***compute hour, minute and second from given decimal hour (ctimhou)***

Inverse of chouday.

- ***compute millisec. of the day from jour, hour, min,sec.,ms. (cmilday)***

As $jour*24*360000 + hour*3600000 +min*60000 +sec*1000 +ms$

- ***compute time from millisec. of the day (ctimmil)***

Inverse of cmilday

- ***compute week number in a year from a given date (cweedoy)***

A week is always beginning by a Monday.

- ***compute day of the week for a given date (doweek)***

With 1 for Monday, etc.

- ***compute date from the week number of the year (cdatweek)***

Date is computed for the first day of the week, always a Monday.

- ***compute number of day in a month (cnbdmon)***

Dependent of the year because of leap years.

- ***compute US day name (cusdayn)***

Example : « Monday » for 1

- ***compute french day name (cfrdayn)***

Example : « Lundi » for 1

- ***compute US month name (cusmonn)***

Example : « January » for 1

- ***compute french month name (cfrmonn)***

Example : « Janvier » for 1

ROCOTLIB COORDINATE TRANSFORMATION LIBRARY

PART II: USER'S MANUAL

I- DESCRIPTION OF AVAILABLE MODULES

1) General remarks

ROCOTLIB library delivers 5 kinds of subroutines:

a) “Basic computation subroutines”, such as `cxxxxx` (`ctimpar`, `csundir` etc.). It only do computations without any particular previous call. One has include in this category the particular '`ctimpar`' subroutine which set for a given date and time all the time varying quantities needed for the transformations matrix (results are stored in 15 common transparent to the user). The `csundir` and `cdipdir` subroutine compute the direction of the Sun in GEI, and the direction of the magnetic dipole axis in GEO, and are used in `ctimpar`, but can be used independently.

b) “Calendar subroutines”, such as `cxxxxx` (`cdoyear`, `ctimhou` etc.). It also only do computations without any particular previous call. These are useful for conversion of dates and times given in various format, or to compute a complete calendar with the days.

c) “Give subroutines”, such as `gxxxxx` (`gsundi`, `gdipdi`, etc...) which give in the output arguments useful parameters as direction of the Sun, direction of the dipole etc... in GEI and GEO systems; subroutine `ctimpar` must be called before using theses subroutines at each times where the date and time are changed;

d) “Read and check subroutines”, such as `rxxxxx` (`redate`, `retime`), which query and read on the standard input date and time, with automatic check of good format (no hour greater than 23, or minutes greater than 59, and so on), useful to write an interactive user program where date and time are an input.

e) “Transformations subroutines”, such as `txxxxxx` (`tgeigeo`, `tgeogei`, etc...) which transforms input Cartesian coordinates system into an another one following mathematical expressions given in Part I; as above, subroutine `ctimpar` must be called before using theses subroutines at each times where the date and time are changed.

One subroutine is out of class: a call to `plibinf` subroutine just print the information relative to the library (version number, etc.).

Section VI give an example of Fortran user program, directions for use, example of installation of the library on a UNIX system, and provide a test program code source to check the validity of the library.

2) Description of "Basic Computation" subroutines

ctimpar

subroutine ctimpar(iyear, imonth, iday, ihour, imin, isec)

compute_time_parameters: prepare matrix for coordinate transformations

prepare all time varying quantities for computations of coordinate transforms of the library, and store results in 15 common statements; use csundir and cdipdir subroutines. Quantities stored in commons are below:

sin and cos of GMST

ecliptic pole in GEI system

direction of the rotation axis of the sun in GEI system

dipole direction in GEI system

direction of the sun in GEO system

direction of the ecliptic in GEO system

direction of the rotation axis of the sun in GEO system

cross product MxS in GEI system

cross product ExS in GEI system

cross product RxS in GEI system

cross product RxE in GEI system

cross product DxS in GEO system

cross product ExS in GEO system

cross product RxS in GEO system

computation of gei to mag vectors

computation of gei to sm vectors

computation of gei to gsm vectors

computation of gei to gseq vectors

computation of tetq angle

computation of mu angle

computation of dzeta angle

computation of phi angle

computation of geo to mag vectors

computation of geo to sm vectors

computation of geo to gsm vectors

computation of geo to gsq vectors

input : iyear, imonth, iday : date as 1985 08 24
iyear must be >1901 and <2099)
ihour, imin, isec : time 23 57 36 (U.T.)

output: in common statements

ctimpa2

subroutine ctimpa2(jd1950, houday)

compute_time_parameters: prepare matrix for coordinate transformations

Same as ctimpar, but with different date-time input

input : Julian day from 1950, decimal hour in the day
output: in common statements

ctimpa3

subroutine ctimpa3(jd2000, houday)

compute_time_parameters: prepare matrix for coordinate transformations

Same as ctimpar, but with different date-time input

input : Julian day from 2000, decimal hour in the day
output: in common statements

cdipdir

subroutine cdipdir(iyear, idoty, d1, d2, d3)

compute_dipole_direction in GEO system

compute geodipole axis direction from International Geomagnetic Reference Field (IGRF) models for time interval 1965 to 1990. For time out of interval, computation is made for nearest boundary.

input : iyear : year (1965 - 1990)
 idoty : day of the year (1 for January 1)
output: d1,d2,d3 Cartesian dipole components in GEO

csundir

subroutine csundir(iyear, idoty, ihour, imin, isec,
 gst, slong, sra, sdec, obliq)

compute_sun_direction in GEI system

calculates four quantities in GEI system necessary for coordinate transformations dependent on sun position (and, hence, on universal time and season) (from C.T. Russel, cosmic electrodynamics, v.2, 184-196, 1971, revised P. Robert 1992 & 2000)

input : iyear : year (1901-2099)
 idoty : day of the year (1 for January 1)
 ihour, imin, isec : hours, minutes, seconds U.T.

output: gst Greenwich mean sidereal time (radians)
 slong longitude along ecliptic (radians)
 sra right ascension (radians)
 sdec declination of the sun (radians)

cangrat

subroutine cangrat(ux,uy,uz, vx,vy,vz, angle,ratio)

compute_angle_and_ratio between 2 vectors

input : ux,uy,uz, vx,vy,vz

output: angle (radians) and $|u|/|v|$

csunset

subroutine csunset(iyear,imon,iday,rlat,rlon,tmer,tris,tset,durd)

compute_Sunset and other Sun parameters

input : iyear,imon,iday : date

 rlat,rlon: geographic latitude and longitude in rad.

output: tmer,tris,tset,durd : Sun meridian time, Sunrise time,
 Sunset time and duration of the day, in Cha*8 variable

3) Description of "Calendar" subroutines

cdatdoy

subroutine cdatdoy(idoty,iyear,imonth,iday)

compute_date_from_day_of_the_year and for a given year.

input : idoty, iyear, (idoty=1 for January 1)
output: imonth, iday

cdatj00

subroutine cdatj00(jud00, iyear,imonth,iday)

compute_date_from_Julian_day with jud00=1 for January 1, 2000

input : jud00 Julian day (1= 1/1/2000)
output: iyear, imonth, iday

cdatj50

subroutine cdatj50(jud50, iyear,imonth,iday)

compute_date_from_Julian_day with jud50=1 for January 1, 1950

input : jud50 Julian day (1= 1/1/1950)
output: iyear, imonth, iday

cdatwee

subroutine cdatwee(iweek,iyear,imonth,iday)

compute_date_for_first_day_of_week_number always a Monday

input : iweek: week number in the year
output: iyear, imonth, iday

cdoweeek

subroutine cdoweeek(iyear,imonth,iday,idow)

compute_day_of_the_week from date

input : iyear, imonth, iday
output: idow : day of the week, 1 for Monday

cdoyear

subroutine cdoyear(iyear, imonth, iday, idoty)

compute_day_of_the_year with idoty=1 for January 1

input : iyear, imonth, iday, ex: 1990, 10, 17

output: idoty ex: 290

cfrdayn

subroutine cfrdayn(iday, cday, n)

compute_french_day_name as "Lundi" for iday= 1

input : iday, day number (1-7)

output: cday, as "Lundi" an n= number of character of the word

cfrmonn

subroutine cfrmonn(imonth, cmonth, n)

compute_french_month_name as "Janvier" for imonth=1

input : imonth, month (1-12)

output: cmonth, as "Janvier" an n= number of character of the word

chouday

subroutine chouday(ih, im, is, houday)

compute_hour_of_the_day from hours, minutes, seconds

input : ih, im, is

output: houday decimal hour of the day

cjd1950

subroutine cjd1950(iyear, imonth, iday, jud50)

compute_Julian_day with jud50=1 for January 1, 1950

input : iyear, imonth, iday ex: 1990, 10, 17

output: jud50

cjd2000

subroutine cjd2000(iyear, imonth, iday, jud00)

compute_Julian_day with jud00=1 for January 1, 2000

input : iyear, imonth, iday ex: 2001, 10, 17

output: jud50

cmilday

subroutine cmilday(ih,im,is,ims,milday)

compute_millisecond_of_the_day

input : ih,im,is,ims: time in hour,min,sec,millisecond
output: milday, millisecond. of the day (<86400000)

cnbdmon

subroutine cnbdmon(iyear,imonth,nbday)

compute_number_of_days_in_a_month

input : iyear,imonth
output: nbday, ex 31 for January

coleapy

subroutine coleapy(iyear,ileap)

compute_leap_year with ileap=1 for leap year, 0 if not

input : iyear (ex: 1980)
output: ileap (1 or 0 if iyear is or not a leap year)

ctimhou

subroutine ctimhou(houday,ih,im,is)

compute_time from decimal hour of the day

input : houday, decimal hour of the day
output: ih,im,is

ctimmil

subroutine ctimmil(milday,ih,im,is,ims)

compute_time from millisecond of the day

input : milday, millisecond of the day
output: ih,im,is,ims

cusdayn

subroutine cusdayn(iday,cday,n)

compute_US_day_name as "Monday" for iday= 1

input : iday, day number (1-7)
output: cday, as "Monday" an n= number of character of the word

cusmonn

subroutine cusmonn(imonth,cmonth,n)

compute_US_month_name as "January" for imonth=1

input : imonth, month (1-12)

output: cmonth, as "January" an n= number of character of the word

cweedoy

subroutine cweedoy(iyear,imonth,iday,iweek)

compute_week_of_the_year from date

input : iyear, imonth, iday

output: iweek : week of the year, 1 for first Monday

4) Description of "Give" subroutines

gdipdir

subroutine gdipdir(dxgei,dygei,dzgei,dxgeo,dygeo,dzgeo)

give_dipole_direction in GEI and GEO system

input : none

output: dxgei,dygei,dzgei Cartesian dipole GEI coord.
dxgeo,dygeo,dzgeo Cartesian dipole GEO coord.

gdiptan

subroutine gdiptan(diptan)

give_dipole_tilt_angle in GSM system (radians)

input : none

output: diptan=(D,Z) angle in GSM (radians)
(diptan>0 when the north magnetic pole
is tilted toward the Sun)

gecldir

subroutine gecldir(exgei,eygei,ezgei,exgeo,eygeo,ezgeo)

give_ecliptic_direction in GEI and GEO system

input : none

output: exgei,eygei,ezgei Cartesian eclip. GEI coord.
exgeo,eygeo,ezgeo Cartesian eclip. GEO coord.

gsrodir

subroutine gsrodir(rxgei,rygei,rzgei,rxgeo,rygeo,rzgeo)

give_sun_rotation_direction in GEI and GEO system

input : none

output: rxgei,rygei,rzgei Cartesian sun rot. GEI coord.
rxgeo,rygeo,rzgeo Cartesian sun rot. GEO coord.

gsundir

subroutine gsundir(sxgei,sygei,szgei,sxgeo,sygeo,szgeo)

give_sun_direction in GEI and GEO system

input : none

output: sxgei,sygei,szgei Cartesian sun GEI coord.
sxgeo,sygeo,szgeo Cartesian sun GEO coord.

gsunpar

subroutine gsunpar(gmst, slon, sras, sdec)

give_sun_parameter dependant of time in GEI system

input : none
output: gmst Greenwich mean sideral time(radians)
slon longitude along ecliptic (radians)
sras right ascension (radians)
sdec declination of the sun (radians)

gvernum

subroutine gvernum(vernum)

give_version_number of the library

input : none
output: vernum (ex 1.0)

5) Description of "Read and check" subroutines

recoor

subroutine recoor(x,y,z,cs)

read_coordinate values from standard input and check validity

read cs character variable from standard input;

if cs is eq to "c", then read Cartesian component of a vector from standard input.

If cs is eq to "s", else read spherical r, t, j components from standard input.

input : cs (char*1)

output: x,y,z, cs Cartesian components and cs

recys

subroutine recys(csys)

read_coordinate_system from terminal input and check validity

csys must have only the following values: gei, geo, mag, sma, gsm, gse, gsq

if not, ask again

input : none

output: csys ; print error if csys is not valid, and ask again

redate

subroutine redate(iyear, imonth, iday)

read_date from standard input and check validity

test if imonth is not greater than 12, test if iday is not greater then length of the month, taking into account the leap years; year must be greater or equal to 1900; use coleapy subroutine.

input : none

output: iyear, imonth, iday

print error if date is not valid, and ask again

retime

subroutine retime(ih, im, is)

read_time from standard input and check validity

ih must be between 1 and 23, im and is between 1 and 59

input : none

output: ih, im, is

print error if time is not valid, and ask again

6) Description of "Transformations" subroutines

Subroutines are sorted according with the schematic diagram of § II-2, or the liste given in § II-3

1

tcarsph

subroutine tcarsph(x,y,z,r,teta,phi)

transforms_car_to_sph : CAR → SPH system

input : x,y,z Cartesian coordinates
output: r,teta,phi spherical coordinates (radians)

tsphcar

subroutine tsphcar(r,teta,phi,x,y,z)

transforms_sph_to_car : SPH → CAR system

input : r,teta,phi spherical coordinates (radians)
output: x,y,z Cartesian coordinates

2

tgeigeo

subroutine tgeigeo(xgei,ygei,zgei,xgeo,ygeo,zgeo)

transforms_gei_to_geo : GEI → GEO system

input : xgei,ygei,zgei Cartesian gei coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

tgeogei

subroutine tgeogei(xgeo,ygeo,zgeo,xgei,ygei,zgei)

transforms_geo_to_gei : GEO → GEI system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xgei,ygei,zgei Cartesian gei coordinates

3

tgeimag

subroutine tgeimag(xgei,ygei,zgei,xmag,ymag,zmag)

transforms_gei_to_mag : GEI → MAG system

input : xgei,ygei,zgei Cartesian gei coordinates
output: xmag,ymag,zmag Cartesian mag coordinates

tmaggei

subroutine tmaggei(xmag,ymag,zmag,xgei,ygei,zgei)

transforms_mag_to_gei : MAG → GEI system

input : xmag,ymag,zmag Cartesian mag coordinates
output: xgei,ygei,zgei Cartesian gei coordinates

4

tgeisma

subroutine tgeisma(xgei,ygei,zgei,xsma,ysma,zsma)

transforms_gei_to_sma : GEI → SM system

input : xgei,ygei,zgei Cartesian gei coordinates

output: xsma,ysma,zsma Cartesian sma coordinates

tsmagei

subroutine tsmagei(xsma,ysma,zsma,xgei,ygei,zgei)

transforms_sma_to_gei : SM → GEI system

input : xsma,ysma,zsma Cartesian sma coordinates

output: xgei,ygei,zgei Cartesian gei coordinates

5

tgeigsm

subroutine tgeigsm(xgei,ygei,zgei,xgsm,ygsm,zgsm)

transforms_gei_to_gsm : GEI → GSM system

input : xgei,ygei,zgei Cartesian gei coordinates

output: xgsm,ygsm,zgsm Cartesian gsm coordinates

tgsmgei

subroutine tgsmgei(xgsm,ygsm,zgsm,xgei,ygei,zgei)

transforms_gsm_to_gei : GSM → GEI system

input : xgsm,ygsm,zgsm Cartesian gsm coordinates

output: xgei,ygei,zgei Cartesian gei coordinates

6

tgeigse

subroutine tgeigse(xgei,ygei,zgei,xgse,ygse,zgse)

transforms_gei_to_gse : GEI → GSE system

input : xgei,ygei,zgei Cartesian gei coordinates

output: xgse,ygse,zgse Cartesian gse coordinates

tgsegei

subroutine tgsegei(xgse,ygse,zgse,xgei,ygei,zgei)

transforms_gse_to_gei : GSE → GEI system

input : xgse,ygse,zgse Cartesian gse coordinates

output: xgei,ygei,zgei Cartesian gei coordinates

7

tgeigsq

subroutine tgeigsq(xgei,ygei,zgei,xgsq,ygsq,zgsq)

transforms_gei_to_gsq : GEI → GSEQ system

input : xgei,ygei,zgei Cartesian gei coordinates
output: xgsq,ygsq,zgsq Cartesian gsq coordinates

tgsqgei

subroutine tgsqgei(xgsq,ygsq,zgsq,xgei,ygei,zgei)

transforms_gsq_to_gei : GSEQ → GEI system

input : xgsq,ygsq,zgsq Cartesian gsq coordinates
output: xgei,ygei,zgei Cartesian gei coordinates

8

tgsegsq

subroutine tgsegsq(xgse,ygse,zgse,xgsq,ygsq,zgsq)

transforms_gse_to_gsq : GSE → GSEQ system

input : xgse,ygse,zgse Cartesian gse coordinates
output: xgsq,ygsq,zgsq Cartesian gsq coordinates

tgsqgse

subroutine tgsqgse(xgsq,ygsq,zgsq,xgse,ygse,zgse)

transforms_gsq_to_gse : GSEQ → GSE system

input : xgsq,ygsq,zgsq Cartesian gsq coordinates
output: xgse,ygse,zgse Cartesian gse coordinates

9

tgsegsm

subroutine tgsegsm(xgse,ygse,zgse,xgsm,ygsm,zgsm)

transforms_gse_to_gsm : GSE → GSM system

input : xgse,ygse,zgse Cartesian gse coordinates
output: xgsm,ygsm,zgsm Cartesian gsm coordinates

tgsmgse

subroutine tgsmgse(xgsm,ygsm,zgsm,xgse,ygse,zgse)

transforms_gsm_to_gse : GSM → GSE system

input : xgsm,ygsm,zgsm Cartesian gsm coordinates
output: xgse,ygse,zgse Cartesian gse coordinates

10

tgsmsma

subroutine tgsmsma(xgsm,ygsm,zgsm,xsma,ysma,zsma)

transforms_gsm_to_sma : GSM → SM system

input : xgsm,ygsm,zgsm Cartesian gsm coordinates

output: xsma,ysma,zsma Cartesian sma coordinates

tsmagsm

subroutine tsmagsm(xsma,ysma,zsma,xgsm,ygsm,zgsm)

transforms_sma_to_gsm : SM → GSM system

input : xsma,ysma,zsma Cartesian sma coordinates

output: xgsm,ygsm,zgsm Cartesian gsm coordinates

10 bis

tgsmsgsq

subroutine tgsmsgsq(xgsq,ygsq,zgsq,xmag,ymag,zmag)

transforms_gsm_to_gsq : GSM → GSQ system (via GSM→GSE→GSQ)

input : xgsm,ygsm,zgsm Cartesian gsm coordinates

output: xgsq,ygsq,zgsq Cartesian gsq coordinates

tgsqgsm

subroutine tgsqgsm(xgsq,ygsq,zgsq,xgsm,ygsm,zgsm)

transforms_gsq_to_gsm : GSQ → GSM system (via GSQ→GSE→GSM)

input : xgsq,ygsq,zgsq Cartesian gsq coordinates

output: xgsm,ygsm,zgsm Cartesian gsm coordinates

10 ter

tgsmmag

subroutine tgsmmag(xgsm,ygsm,zgsm,xmag,ymag,zmag)

transforms_gsm_to_mag : GSM → MAG system (via GSM→SM→MAG)

input : xgsm,ygsm,zgsm Cartesian gsm coordinates

output: xmag,ymag,zmag Cartesian mag coordinates

tmaggsm

subroutine tmaggsm(xmag,ymag,zmag,xgsm,ygsm,zgsm)

transforms_mag_to_gsm : MAG → GSM system (via MAG→SM→GSM)

input : xmag,ymag,zmag Cartesian mag coordinates

output: xgsm,ygsm,zgsm Cartesian gsm coordinates

11

tsmamag

subroutine tsmamag(xsma,ysma,zsma,xmag,ymag,zmag)

transforms_sma_to_mag : *SM* → *MAG* system

input : xsma,ysma,zsma Cartesian sma coordinates
output: xmag,ymag,zmag Cartesian mag coordinates

tmagsma

subroutine tmagsma(xmag,ymag,zmag,xsma,ysma,zsma)

transforms_mag_to_sma : *MAG* → *SM* system

input : xmag,ymag,zmag Cartesian mag coordinates
output: xsma,ysma,zsma Cartesian sma coordinates

12

tgeomag

subroutine tgeomag(xgeo,ygeo,zgeo,xmag,ymag,zmag)

transforms_geo_to_mag : *GEO* → *MAG* system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xmag,ymag,zmag Cartesian mag coordinates

tmaggeo

subroutine tmaggeo(xmag,ymag,zmag,xgeo,ygeo,zgeo)

transforms_mag_to_geo : *MAG* → *GEO* system

input : xmag,ymag,zmag Cartesian mag coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

13

tgeosma

subroutine tgeosma(xgeo,ygeo,zgeo,xsma,ysma,zsma)

transforms_geo_to_sma : *GEO* → *SM* system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xsma,ysma,zsma Cartesian sma coordinates

tsmageo

subroutine tsmageo(xsma,ysma,zsma,xgeo,ygeo,zgeo)

transforms_sma_to_geo : *SM* → *GEO* system

input : xsma,ysma,zsma Cartesian sma coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

14

tgeogsm

subroutine tgeogsm(xgeo,ygeo,zgeo,xgsm,ygsm,zgsm)

transforms_geo_to_gsm : GEO → GSM system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xgsm,ygsm,zgsm Cartesian gsm coordinates

tgsmgeo

subroutine tgsmgeo(xgsm,ygsm,zgsm,xgeo,ygeo,zgeo)

transforms_gsm_to_geo : GSM → GEO system

input : xgsm,ygsm,zgsm Cartesian gsm coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

15

tgeogse

subroutine tgeogse(xgeo,ygeo,zgeo,xgse,ygse,zgse)

transforms_geo_to_gse : GEO → GSE system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xgse,ygse,zgse Cartesian gse coordinates

tgsegeo

subroutine tgsegeo(xgse,ygse,zgse,xgeo,ygeo,zgeo)

transforms_gse_to_geo : GSE → GEO system

input : xgse,ygse,zgse Cartesian gse coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

16

tgeogsq

subroutine tgeogsq(xgeo,ygeo,zgeo,xgsq,ygsq,zgsq)

transforms_geo_to_gsq : GEO → GSEQ system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
output: xgsq,ygsq,zgsq Cartesian gsq coordinates

tgsggeo

subroutine tgsggeo(xgsq,ygsq,zgsq,xgeo,ygeo,zgeo)

transforms_gsq_to_geo : GSEQ → GEO system

input : xgsq,ygsq,zgsq Cartesian gsq coordinates
output: xgeo,ygeo,zgeo Cartesian geo coordinates

17

tgeodme

subroutine tgeodme(xgeo,ygeo,zgeo,rlat,rlong,xdme,ydme,zdme)

transforms_geo_to_dme : GEO → DM system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
 rlat,rlong latitude and longitude of the
 point of observation (radians)
output: xdme,ydme,zdme Cartesian dme coordinates

tdmegeo

subroutine tdmegeo(xdme,ydme,zdme,rlat,rlong,xgeo,ygeo,zgeo)

transforms_dme_to_geo : DM → GEO system

input : xdme,ydme,zdme Cartesian dme coordinates
 rlat,rlong latitude and longitude of the
 point of observation (radians)
output: xgeo,ygeo,zgeo Cartesian geo coordinates

18

tgeovdh

subroutine tgeovdh(xgeo,ygeo,zgeo,rlat,rlong,xvdh,yvdh,zvdh)

transforms_geo_to_vdh : GEO → VDH system

input : xgeo,ygeo,zgeo Cartesian geo coordinates
 rlat,rlong latitude and longitude of the
 point of observation (radians)
output: xvdh,yvdh,zvdh Cartesian vdh coordinates

tvdhgeo

subroutine tvdhgeo(xvdh,yvdh,zvdh,rlat,rlong,xgeo,ygeo,zgeo)

transforms_vdh_to_geo : VDH → GEO system

input : xvdh,yvdh,zvdh Cartesian vdh coordinates
 rlat,rlong latitude and longitude of the
 point of observation (radians)
output: xgeo,ygeo,zgeo Cartesian geo coordinates

19

tgse2

subroutine
tgse2(xgse,ygse,zgse,rotx,roty,rotz,xsr2,ysr2,zsr2)

transforms_gse_to_sr2 : GSE → SR2 system

input : xgse,ygse,zgse Cartesian GSE coordinates
 rotx,roty,rotz Cartesian GSE coordinates
 of the rotation axis of the S/C
output: xsr2,ysr2,zsr2 Cartesian SR2 coordinates

tsr2gse

subroutine tsr2gse(xsr2,ysr2,zsr2,rotx,roty,rotz, xgse,ygse,zgse)

transforms_sr2_to_gse : SR2 → GSE system

input : xsr2,ysr2,zsr2 Cartesian SR2 coordinates
 rotx,roty,rotz Cartesian GSE coordinates
 of the rotation axis of the S/C
output: xgse,ygse,zgse Cartesian GSE coordinates

20

tsr2mfa

subroutine tsr2mfa(xsr2,ysr2,zsr2,bx,by,bz,rotx,roty,rotz,
 xmfa,ymfa,zmfa)

transforms_sr2_to_mfa : SR2 → MFA system

input : xsr2,ysr2,zsr2 Cartesian SR2 coordinates
 bx,by,bz Cartesian SR2 coordinates
 of the DC magnetic field
 rotx,roty,rotz Cartesian GSE coordinates
 of the rotation axis of the S/C
output: xmfa,ymfa,zmfa Cartesian MFA coordinates

21

tsr2sre

subroutine tsr2sre(xsr2,ysr2,spifre,spipha,deltaT,xsre,ysre)

transforms_sr2_to_sre : SR2 → SR system

input : xsr2, ysr2 cartesian sr2 coordinates
 spifre spin frequency in Hz
 spipha spin phase in radians, growing with time
 spipha= positive angle between the xsr axis and
 the component of the direction of the Sun in the
 xsr-ysr plane.
 deltaT (T -To) (sec.), between the current time
 & the time where is measured the spin phase
output : xsre,ysre cartesian sr coordinates
Comment: Z component is unchanged (spin axis)

tsresr2

subroutine tsresr2(xsre,ysre,spifre,spipha,deltaT,xsr2,ysr2)

transforms_sre_to_sr2 : SR → SR2 system

input : xsre, ysre cartesian sr coordinates
 spifre spin frequency in Hz
 spipha spin phase in radians, growing with time
 spipha= positive angle between the xsr axis and
 the component of the direction of the Sun in the
 xsr-ysr plane.
 deltaT (T -To) (sec.), between the current time
 & the time where is measured the spin phase
output : xsr2,ysr2 cartesian sr coordinates
Comment: Z component is unchanged (spin axis)

7) Description of "print information" subroutine

subroutine prinfo

print_information and version of the library; result of this call is listed in section VI

input : none

output: print information on standard output

II - DIRECTIONS FOR USE

1) Package delivered for installation

This section correspond to the file readme.txt of the delivered UNIX standard package. To make installation, do the following steps:

- copy the rocotlib directory and all *.f and *.make files;
- compil the library and make the example, check and information programs by running "make_all.bat" shell file;
- execute the example, check and information program by running "run_all.bat" shell file.

Then the users can write their own programs from the rocotexp.f example program and its associated rocotexp.make shell file for making the executable file.

Contents of readme.txt file is following:

readme.txt file

```
*****
ROCOTLIB Package for UNIX Systems
V 1.8, P. Robert, CETP, December 2002
*****

I- COMPLETE PACKAGE
-----

After normal installation, the complete package
is the following:

readme.txt      this file.
rocotlib.list   list of availaible subroutines of Rocot Library
                (ascii file).
make_all.bat    make all object file and executable files for the
                3 applications below (rocotXXX.exe files).
run_all.bat     run the 3 given applications, and create the
                3 files rocotXXX.out

rocotlib.make   shell script to create rocotlib.o object file
rocotlib.f      fortran code   of Rocot Library
rocotlib.o      object file    of Rocot Library

rocotche.make   shell script to create rocotche.exe executable file
rocotche.o      object file    of "check" program
rocotche.exe    executable    of "check" program
rocotche.bat    shell script  to run rocotche.exe
rocotche.f      fortran code  of "check" program
rocotche.in     input file    of "check" program
rocotche.out    output file   of "check" program
rocotche.ou8    output/V 1.8  of "check" program
```

→ suite

```
rocotexp.make  shell script to create rocotexp.exe executable file
rocotexp.o     object file of "example" program
rocotexp.exe  executable of "example" program
rocotexp.bat  shell script to run rocotexp.exe
rocotexp.f    fortran code of "example" program
rocotexp.in   input of "example" program
rocotexp.out  output of "example" program
rocotexp.ou8  output/V 1.8 of "example" program

rocotinf.make  shell script to create rocotinf.exe executable file
rocotinf.o     object file of "information" program
rocotinf.exe  executable of "information" program
rocotinf.bat  shell script to run rocotinf.exe
rocotinf.f    fortran code of "information" program
rocotinf.out  output file of "information" program
rocotinf.ou8  output/V 1.8 of "information" program
```

II- INTALLATION

After copy of above files, all *.o and *.exe can be re-created by running make_all.bat.

All *.out files can be re-created by running run_all.bat.

III- APPLICATIONS

- 1) rocotche program is a check program of the library.
rocotche.bat create it by:

```
rocotche.exe <rocotche.in >rocotche.out
```

The rocotche.out created file can be compared with the initial rocotche.ou8 file; The file must be the same from a system to another one, except precision: for instance, you can find 0.999999 instead of 1.000000, and vice-versa.
No major change must be found.

For any other tests, it is possible, of course, to change input values into rocotche.in file or directly answer by typing data on terminal while directly running rocotche.exe.
Obviously, rocotche.out is changed.

- 2) rocotexp.exe is an example program showing how to use the library.
rocotexp.bat create it by:

```
rocotexp.exe <rocotexp.in >rocotexp.out
```

rocotexp.exe can be also used as an interactive program.

- 3) rocotinf.out is a program giving short informations on the library, in particular the version number.
rocotinf.bat create it by:

```
rocotinf.exe >rocotinf.out
```

IV- COMMENTS

for new application, compilation and link can be as following:

```
f77 -c toto.f --> create toto.o
f77 toto.o rocotlib.o -o toto.exe --> create toto.exe
```

Library has been developped on a UNIX machine with f77 compiler.
File rocotche.ou8 correspond to results with SUN Sparc Station 20 machine, with the version 1.8.

For more information, see paper documentation.

P. Robert, November 1992
revised March 1993
revised January 1995
revised July 2001
revised December 2002

2) Make the rocotlib.o object file

The rocotlib.f source code can be compiled under UNIX system generally as:

```
f77 -c -C rocotlib.f
```

which produce rocotexp.o binary file; The users can also use the rocotlib.make file included in the package, and given below, and type :

```
rocotlib.make or rocotlib.make f90
```

according to the choosed compiler F77 (default) or F90 or other.

rocotlib.make file

```
#!/bin/sh
echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
echo "          Rocotlib Package"
echo "    create object and executable files of rocotlib.f program"
echo "          P. Robert, June 2001"
echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
echo " "

# F77 par default

if test $# -eq 0
then FC=f77
else FC=$1
fi

FC1="$FC -c -C"

# creation de l'objet

if test -f rocotlib.o
then rm rocotlib.o
fi

if test -f rocotlib.f
then
echo "1) compil rocotlib.f, make rocotlib.o with $FC1"
$FC1 rocotlib.f
else
echo "No rocotlib.f file"
exit 1
fi
```

3) Example of Fortran user program

The rocotexp.f program is an example program; it can be modified by the user to see how to use the rocotlib library. Content is given below :

rocotexp.f source file

```

      program rocotexp
c
c  -----
c
c *   Rocot_Example of using Rocotlib
c
c  -----
c
c
      print*
      print*, '*****'
      print*
      print*, 'ROCOTEXP:'
      print*
c
      call redatte(iyear,imonth,iday)
      call rettime(ih,im,is)

      call cdoyear(iyear,imonth,iday,idoty)
      call ctimpar(iyear,imonth,iday,ih,im,is)
      call gsundir(sxgei,sygei,szgei,sxgeo,sygeo,szgeo)
      call tgeigsm(sxgei,sygei,szgei,sxgsm,sygsms,szgsms)

      print*
      print*, 'date: MM/JJ/YY=', imonth,iday,iyear
      print*, 'time: HH/MM/SS=', ih,im,is
      print*
      print*, 'Sun in GEI:',sxgei,sygei,szgei
      print*, 'Sun in GEO:',sxgeo,sygeo,szgeo
      print*, 'Sun in GSM:',sxgsm,sygsms,szgsms
      print*
      print*, '*****'
      print*

      stop 'rocotexp: normal termination'
      end
```

The rocotlib.o object file being existant, the user can compile the example program or its own program by a command such as :

```
f77 -c rocotexp.f
```

which produce rocotexp.o binary file, and then link with the rocotlib.o library by :

```
f77 rocotexp.o rocotlib.o -o rocotexp.exe
```

which produce rocotexp.exe executable file (rocotexp.o can be then erased). Both command can be grouped as:

```
f77 rocotexp.f rocotlib.o -o rocotexp.exe
```

(then rocotexp.o is not produced).

The users can also use the `rocotexp.make` file included in the package, and given below, and type :

```
rocotexp.make or rocotexp.make f90
```

according to the choosed compiler F77 (default) or F90 or other.

If the `rocotlib.o` file is not in the same directory than `rocotexp.f`, the corresponding line in `rocotexp.make` file must takes into account the path of the `rocotlib.o` library.

rocotexp.make file

```
#!/bin/sh
echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
echo "          Rocolib Package"
echo "    create object and executable files of rocotexp.f program"
echo "          P. Robert, June 2001"
echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
echo " "

# F77 par default

if test $# -eq 0
then FC=f77
else FC=$1
fi

FC1="$FC -c -C"
FC2="$FC -C"

# creation de l'objet

if test -f rocotexp.o
then rm rocotexp.o
fi

if test -f rocotexp.f
then
echo "1) compil rocotexp.f, make rocotexp.o with $FC1"
$FC1 rocotexp.f
else
echo "No rocotexp.f file"
exit 1
fi

# link avec rocotlib.o et creation de l'executable

if test -f rocotexp.exe
then rm rocotexp.exe
fi

if test -f rocotlib.o
then
echo " "
echo "2) link with rocotlib.o, make rocotexp.exe with $FC2"
$FC2 rocotexp.o rocotlib.o -o rocotexp.exe
else
echo "No rocotlib.o object library file"
exit 2
fi
```

The rocotexp program can be run interactively, the input being taken from the standard input, by:

```
rocotexp.exe
```

or can be run in batch mode by :

```
rocotexp.bat
```

the input data being taken from the rocotexp.in file. Then the results are in the rocotexp.out file, and must be as following :

rocotexp.out file

```
*****
ROCOTEXP:
iyear,imonth,iday ? (ex: 1990,10,17)
1990 7 14
hour, minute, second ? (ex: 10,45,50)
12 0 0

date: MM/JJ/YY= 7 14 1990
time: HH/MM/SS= 12 0 0

Sun in GEI: -0.371170 0.851934 0.369380
Sun in GEO: 0.928981 2.35213E-02 0.369380
Sun in GSM: 1.00000 -4.47035E-08 -2.98023E-08
*****
```

4) Output of "print information" subroutine

The small fortran program below (rocotinf.f file in standard package) give the following output on default output (rocotinf.out file) containing information, in particular the version number of the library.

rocotinf.f source file

```
      program rocotinf
c
c -----
c *   Rocot_Information: give information on Rocotlib
c                       (in particular version number)
c -----
c
c   call plibinf
c
c   stop 'rocotinf: normal termination'
c   end
```

rocotinf.out file

```
*****
Coordinates Transformation Library  ROCOTLIB
Revised Version  1.8 - December 2002

-----
initially supported by
EUROPEAN SPACE AGENCY

Study of the Cluster Mission
Planning Related Aspects
within the Numerical Simulations Network

Patrick ROBERT, CRPE, November 1992

-----

First version 1.0, PR, CRPE, November 1992
revised version 1.1, PR, CRPE, July 1993
revised version 1.2, PR, CETP, January 1995
revised version 1.3, PR, CETP, July 2000
revised version 1.4, PR, CETP, June 2001
revised version 1.5, PR, CETP, December 2001
revised version 1.6, PR, CETP, June 2002
revised version 1.7, PR, CETP, December 2002
revised version 1.8, PR, CETP, November 2003

Copyright 1992, Patrick ROBERT, CNRS-ESA, All Rights reserved

-----

For details, see the original document untitled:
"CLUSTER Software Tools, part I: Coordinate Transformation Library"
Document de travail DT/CRPE/1231
Patrick Robert, CRPE/TID, Juillet 1993

-----
```

5) Check program

• *general remarks*

The `rocotche.f` and `rocotche.exe` programs delivered in the standard package allows the user to check the validity of the library for a given machine and compiler. It run from an input date and time, an input vector test, and an input direction of observation. Next paragraph show examples of input and output data file (`rocotche.in` and `rocotche.out` files in standard package). Accuracy of the computation is one deduce of the Sun direction computation, which is 0.006 degrees.

Sections of `rocotche.out` file are as follow:

I) check calendar conversions of the library;

II) check basic time parameters as Sun direction, Dipole direction, Ecliptic pole, Sun equator pole;

III) check basic transformation, for instance verify that direction of the Sun in GSM system is well the X axis, that Dipole direction in MAG system is well the Z axis, and so on;

IV) check the usual transformations, by changing the input vector across all available paths showed on the schematic diagram of § II-2.

Mainly, the vector is transformed:

- a) as a "star" centred on GEI system (see schematic diagram);
- b) as a "counter-clockwise ring cumulative transformation", from GEO to GEO after a complete circle;
- c) as a "clockwise ring cumulative transformation", from GEO to GEO, in the other sense;
- d) as a "star" centred on GEO system;
- e) Finally, input direction of the point of observation is used to check local coordinate systems, i.e. DM and VDH.

At each "go and back", one check that the original vector is returned to the same value, taking into account accuracy of the computation (32 bits simple precision, 7 significative digits, and input Sun direction within 0.006 degrees).

• *input rocotche.in data file of check program*

```
1990 10 17    # iyear, imonth, iday
12 30 1      # ihour, imin, isec
 5. 30. 60.   # R, Theta, Phi (deg.) of test vector
45. 30.      # Geographic latitude and longitude (deg.) of point of observ.
2. 170. 10.   # R, Theta, Phi (deg.) of spin axis in GSE
0.25 30.     # Spin frequency (Hz), Spin phase (Deg.)
1.2345       # delta T from Spin phase time measurement
```

• *output rocotche.out file of check program*

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Coordinates Transformation Library  ROCOTLIB
Revised Version  1.8 - November 2003
-----
Rocotche program: Check transformations of Rocotlib
Patrick ROBERT, CRPE, November 1992
last revision for V 1.8 : November 2003
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

SUMMARY OF CHECK OPERATIONS :
*****

I) TEST OF ROCOTLIB CALENDAR CONVERSIONS

II) CHECK BASIC TIME PARAMETERS

III) CHECK BASIC TRANSFORMATIONS

IV) TEST OF ROCOTLIB TRANSFORMATION SUBROUTINES

A) STAR TRANSFORMATIONS AROUND  GEI  SYSTEM

    GEI -----> MAG
    GEI -----> SM
    GEI -----> GSM
    GEI -----> GSE

B) COUNTERCLOCKWISE RING CUMULATIVE TRANSFORM.

    GEO ->GEI ->GSQ ->GSE ->GSM ->SM ->MAG ->GEO

C) CLOCKWISE RING CUMULATIVE TRANSFORMATIONS

    GEO ->MAG ->SM ->GSM ->GSE ->GSQ ->GEI ->GEO

D) STAR TRANSFORMATIONS AROUND  GEO  SYSTEM

    GEO -----> SM
    GEO -----> GSM
    GEO -----> GSE
    GEO -----> GSQ

E) LOCAL SYSTEMS

    GEO ----> DM
    GEO ----> VDH

F) SPACECRAFT SYSTEMS

    GEO -> GSE -> SR2 -> SR -> SR2 -> GSE -> GEO
```

Suite →

```
INPUT DATA TEST :
*****

INPUT DATE AND TIME:
  iyear, imonth, iday ? (ex: 1990,10,17)
  1990 10 17
  hour, minute, second ? (ex: 10,45,50)
  12 30 1

iyear, imonth, iday: 1990 10 17
ih, im, is          : 12 30 01

INPUT VECTOR TEST:
r, teta, phi of any vector in GEO system ? (deg.)
  5.00000  30.000  60.000

INPUT GEOG. LAT AND LONG. FOR LOCAL SYSTEMS TEST:
lat. and long. of local direct. of observation ? (d.)
  45.000  30.000

INPUT SPIN DIRECTION for SR and SR2 SYSTEMS:
r, teta, phi of spin axis in GSE system ? (deg.)
  2.00000  170.000  10.000

SPIN FREQUENCY, SPIN PHASE & Deltat T :
  Spin frequency, Spin phase ? (Hz, deg.)
  0.25000  30.000
delta T ? (sec. from time of spin phase)
  1.23450

RUN OF CHECK PROGRAM
*****

Please wait...
```

Suite →

T E S T O F R O C O T L I B V e r s i o n 1 . 8

I) TEST OF ROCOTLIB CALENDAR CONVERSIONS

=====

INPUT DATE AND TIME:

year, month, day : 1990 10 17
hour, minute, second: 12 30 1

computed values:

decimal hour of the day: 12.5003
day of the year: 290
Julian day from 1-1-1950: 14899
Julian day from 1-1-2000: -3363
leap year (1=yes,0=no): 0

recompute date from year and day of the year: 1990 10 17
recompute date from Julian day 1950 : 1990 10 17
recompute date from Julian day 2000 : 1990 10 17
recompute time from decimal hour: 12 30 1

II) CHECK BASIC TIME PARAMETERS

=====

SUN PARAMETERS IN GEI SYSTEM

Greenwich Sideral Time (deg.): 213.253
ecliptic longitude (deg.): 203.879
right ascension (deg.): 202.100
declination (deg.): -9.265
DIPOLE TILT ANGLE (deg.): -3.750

SUN DIRECTION

in GEI:
X= -0.91444 r = 1.00000
Y= -0.37132 teta= 99.265
Z= -0.16100 phi = -157.900

in GEO:
X= 0.96832 r = 1.00000
Y= -0.19090 teta= 99.265
Z= -0.16100 phi = -11.152

DIPOLE DIRECTION

in GEI:
X= -0.14832 r = 1.00000
Y= 0.11554 teta= 10.837
Z= 0.98217 phi = 142.081

in GEO:
X= 0.06068 r = 1.00000
Y= -0.17795 teta= 10.837
Z= 0.98217 phi = -71.171

ECLIPTIC DIRECTION

in GEI:
X= 0.00000 r = 1.00000
Y= -0.39780 teta= 23.440
Z= 0.91747 phi = -90.000

in GEO:
X= 0.21812 r = 1.00000
Y= 0.33266 teta= 23.440
Z= 0.91747 phi = 56.747

SUN EQUATOR DIRECTION

in GEI:
X= 0.12170 r = 1.00000
Y= -0.42440 teta= 26.200
Z= 0.89726 phi = -74.000

in GEO:
X= 0.13094 r = 1.00000
Y= 0.42164 teta= 26.200
Z= 0.89726 phi = 72.747

III) CHECK BASIC TRANSFORMATIONS
=====

Check (S,E) angle is equal to 90 deg. : angle= 90.000 (deg.)

1) input vector: Sun direction in GEI system

X= -0.91444 r = 1.00000
Y= -0.37132 teta= 99.265
Z= -0.16100 phi = -157.900

in GEO system

X= 0.96832 r = 1.00000
Y= -0.19090 teta= 99.265
Z= -0.16100 phi = -11.152

check Y=0 in SM system with tgeisma:

X= 0.99786 r = 1.00000
Y= 0.00000 teta= 93.750
Z= -0.06540 phi = 0.000

with tgeosma:

X= 0.99786 r = 1.00000
Y= 0.00000 teta= 93.750
Z= -0.06540 phi = 0.000

check Y=0 and Z=0 in GSM system with tgeigsm:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

with tgeogsm:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

check Y=0 and Z=0 in GSE system with tgeigse:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

with tgeogse:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

check Y=0 and Z=0 in GSQ system with tgeigsq:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

with tgeogsq:

X= 1.00000 r = 1.00000
Y= 0.00000 teta= 90.000
Z= 0.00000 phi = 0.000

Suite →

2) input vector: Dipole direction in GEI system

X= -0.14832 r = 1.00000
Y= 0.11554 teta= 10.837
Z= 0.98217 phi = 142.081

in GEO system

X= 0.06068 r = 1.00000
Y= -0.17795 teta= 10.837
Z= 0.98217 phi = -71.171

check X=0 and Y=0 in MAG system with tgeimag:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

with tgeomag:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

check X=0 and Y=0 in SM system with tgeisma:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

with tgeosma:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

check Y=0 in GSM system with tgeigsm:

X= -0.06540 r = 1.00000
Y= 0.00000 teta= 3.750
Z= 0.99786 phi = 180.000

with tgeogsm:

X= -0.06540 r = 1.00000
Y= 0.00000 teta= 3.750
Z= 0.99786 phi = 180.000

Suite →

3) input vector: Ecliptic direction in GEI system

X= 0.00000 r = 1.00000
Y= -0.39780 teta= 23.440
Z= 0.91747 phi = -90.000

in GEO system

X= 0.21812 r = 1.00000
Y= 0.33266 teta= 23.440
Z= 0.91747 phi = 56.747

check X=0 and Y=0 in GSE system with tgeigse:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

with tgeogse:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

check X=0 in GSQ system with tgeigsq:

X= 0.00000 r = 1.00000
Y= -0.07931 teta= 4.549
Z= 0.99685 phi = -90.000

with tgeogsq:

X= 0.00000 r = 1.00000
Y= -0.07931 teta= 4.549
Z= 0.99685 phi = -90.000

4) input vector: Sun equator in GEI system

X= 0.12170 r = 1.00000
Y= -0.42440 teta= 26.200
Z= 0.89726 phi = -74.000

in GEO system

X= 0.13094 r = 1.00000
Y= 0.42164 teta= 26.200
Z= 0.89726 phi = 72.747

check and Y=0 in GSQ system with tgeigsq:

X= -0.09815 r = 1.00000
Y= 0.00000 teta= 5.633
Z= 0.99517 phi = 180.000

with tgeogsq:

X= -0.09815 r = 1.00000
Y= 0.00000 teta= 5.633
Z= 0.99517 phi = 180.000

Suite →

5) input vector: North geographic in GEO system

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

check X=0 and Y=0 in GEI system with tgeogei:

X= 0.00000 r = 1.00000
Y= 0.00000 teta= 0.000
Z= 1.00000 phi = 0.000

check Y=0 in MAG system with tgeimag:

X= -0.18801 r = 1.00000
Y= 0.00000 teta= 10.837
Z= 0.98217 phi = 180.000

with tgeomag:

X= -0.18801 r = 1.00000
Y= 0.00000 teta= 10.837
Z= 0.98217 phi = 180.000

Suite →

IV) TEST OF ROCOTLIB TRANSFORMATION SUBROUTINES

=====

1) input choosed test vector in GEO system :

X=	1.25000	r =	5.00000
Y=	2.16506	teta=	30.000
Z=	4.33013	phi =	60.000

2) converts GEO system to GEI system by tgeogei

X=	0.14185	r =	5.00000
Y=	-2.49597	teta=	30.000
Z=	4.33013	phi =	-86.747

A) STAR TRANSFORMATIONS AROUND GEI SYSTEM

=====

3
GEI -----> MAG
4
GEI -----> SM
5
GEI -----> GSM
6
GEI -----> GSE

3) converts GEI system to MAG system by tgeimag

X=	-2.43054	r =	5.00000
Y=	1.88187	teta=	37.936
Z=	3.94348	phi =	142.251

come back to GEI system by tmaggei

X=	0.14185	r =	5.00000
Y=	-2.49597	teta=	30.000
Z=	4.33013	phi =	-86.747

difference between first GEI vector: angle= 0.000 (deg.)
ratio= 1.00000

4) converts GEI system to SM system by tgeisma

X=	0.35862	r =	5.00000
Y=	3.05292	teta=	37.936
Z=	3.94348	phi =	83.300

come back to GEI system by tsmagei

X=	0.14185	r =	5.00000
Y=	-2.49597	teta=	30.000
Z=	4.33013	phi =	-86.747

difference between first GEI vector: angle= 0.000 (deg.)
ratio= 1.00000

5) converts GEI system to GSM system by tgeigsm

X=	0.09996	r =	5.00000
Y=	3.05292	teta=	37.655
Z=	3.95849	phi =	88.125

come back to GEI system by tgsmgei

X=	0.14185	r =	5.00000
Y=	-2.49597	teta=	30.000
Z=	4.33013	phi =	-86.747

difference between first GEI vector: angle= 0.000 (deg.)
ratio= 1.00000

Suite →

```
6) converts GEI system to GSE system by tgeigse
      X= 0.09996   r = 5.00000
      Y= 0.57634   teta= 6.718
      Z= 4.96567   phi = 80.160

      come back to GEI system by tgsegei
      X= 0.14185   r = 5.00000
      Y= -2.49597  teta= 30.000
      Z= 4.33013   phi = -86.747

      difference between first GEI vector: angle= 0.000 (deg.)
                                         ratio= 1.00000

B) COUNTERCLOCKWISE RING CUMULATIVE TRANSFORMATIONS
=====
# 2      7      8      9      10     11     12
GEO -> GEI -> GSQ -> GSE -> GSM -> SM -> MAG -> GEO

input choosed test vector in GEO system :
      X= 1.25000   r = 5.00000
      Y= 2.16506   teta= 30.000
      Z= 4.33013   phi = 60.000

2) converts GEO system to GEI system by tgeogei
      X= 0.14185   r = 5.00000
      Y= -2.49597  teta= 30.000
      Z= 4.33013   phi = -86.747

7) converts GEI system to GSQ system by tgeigsq
      X= 0.09996   r = 5.00000
      Y= 0.18069   teta= 2.367
      Z= 4.99573   phi = 61.047

8) converts GSQ system to GSE system by tgsqgse
      X= 0.09996   r = 5.00000
      Y= 0.57634   teta= 6.718
      Z= 4.96567   phi = 80.160

9) converts GSE system to GSM system by tgsegsm
      X= 0.09996   r = 5.00000
      Y= 3.05292   teta= 37.655
      Z= 3.95849   phi = 88.125

10) converts GSM system to SM system by tgsmsma
      X= 0.35862   r = 5.00000
      Y= 3.05292   teta= 37.936
      Z= 3.94348   phi = 83.300

11) converts SM system to MAG system by tsmamag
      X= -2.43054   r = 5.00000
      Y= 1.88187   teta= 37.936
      Z= 3.94348   phi = 142.251

12) converts MAG system to GEO system by tmaggeo
      X= 1.25000   r = 5.00000
      Y= 2.16506   teta= 30.000
      Z= 4.33013   phi = 60.000

      difference between first GEO vector: angle= 0.000 (deg.)
                                         ratio= 1.00000
```

Suite →

C) CLOCKWISE RING CUMULATIVE TRANSFORMATIONS

=====

12 11 10 9 8 7 2
GEO -> MAG -> SM -> GSM -> GSE -> GSQ -> GEI -> GEO

input choosed test vector in GEO system :

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

12) converts GEO system to MAG system by tgeomag

X= -2.43054 r = 5.00000
Y= 1.88187 teta= 37.936
Z= 3.94348 phi = 142.251

11) converts MAG system to SM system by tmagsma

X= 0.35862 r = 5.00000
Y= 3.05292 teta= 37.936
Z= 3.94348 phi = 83.300

10) converts SM system to GSM system by tsmagsm

X= 0.09996 r = 5.00000
Y= 3.05292 teta= 37.655
Z= 3.95849 phi = 88.125

9) converts GSM system to GSE system by tgsmsgse

X= 0.09996 r = 5.00000
Y= 0.57634 teta= 6.718
Z= 4.96567 phi = 80.160

8) converts GSE system to GSQ system by tgsegsq

X= 0.09996 r = 5.00000
Y= 0.18069 teta= 2.367
Z= 4.99573 phi = 61.047

7) converts GSQ system to GEI system by tgsqgei

X= 0.14185 r = 5.00000
Y= -2.49597 teta= 30.000
Z= 4.33013 phi = -86.747

difference between first GEI vector: angle= 0.000 (deg.)
ratio= 1.00000

2) converts GEI system to GEO system by tgeigeo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

Suite →

D) STAR TRANSFORMATIONS AROUND GEO SYSTEM

=====

13
GEO -----> SM
14
GEO -----> GSM
15
GEO -----> GSE
16
GEO -----> GSQ

input choosed test vector in GEO system :

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

13) converts GEO system to SM system by tgeosma

X= 0.35862 r = 5.00000
Y= 3.05292 teta= 37.936
Z= 3.94348 phi = 83.300

come back to GEO system by tsmageo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

14) converts GEO system to GSM system by tgeogsm

X= 0.09996 r = 5.00000
Y= 3.05292 teta= 37.655
Z= 3.95849 phi = 88.125

come back to GEO system by tgsageo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

15) converts GEO system to GSE system by tgeogse

X= 0.09996 r = 5.00000
Y= 0.57634 teta= 6.718
Z= 4.96567 phi = 80.160

come back to GEO system by tgsegeo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

16) converts GEO system to GSQ system by tgeogsq

X= 0.09996 r = 5.00000
Y= 0.18069 teta= 2.367
Z= 4.99573 phi = 61.047

come back to GEO system by tgsqgeo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

Suite →

E) LOCAL SYSTEMS TRANSFORMATIONS

=====

17

GEO -----> DM

18

GEO -----> VDH

1) first direction of observation in GEO system:

geog. lat., long. = 60.000 60.000

17) converts GEO system to DM system by tgeodme

X=	3.07392	r =	5.00000
Y=	0.00000	teta=	37.936
Z=	3.94348	phi =	0.000

come back to GEO system by tdmegeo

X=	1.25000	r =	5.00000
Y=	2.16506	teta=	30.000
Z=	4.33013	phi =	60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

18) converts GEO system to VDH system by tgeovdh

X=	5.00000	r =	5.00000
Y=	0.00000	teta=	90.000
Z=	0.00000	phi =	0.000

come back to GEO system by tvdhgeo

X=	1.25000	r =	5.00000
Y=	2.16506	teta=	30.000
Z=	4.33013	phi =	60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

2) second direction of observation in GEO system:

geog. lat., long. = 45.000 30.000

17) converts GEO system to DM system by tgeodme

X=	2.63031	r =	5.00000
Y=	1.59072	teta=	37.936
Z=	3.94348	phi =	31.164

come back to GEO system by tdmegeo

X=	1.25000	r =	5.00000
Y=	2.16506	teta=	30.000
Z=	4.33013	phi =	60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

18) converts GEO system to VDH system by tgeovdh

X=	4.59279	r =	5.00000
Y=	1.25000	teta=	72.170
Z=	1.53093	phi =	15.225

come back to GEO system by tvdhgeo

X=	1.25000	r =	5.00000
Y=	2.16506	teta=	30.000
Z=	4.33013	phi =	60.000

difference between first GEO vector: angle= 0.000 (deg.)
ratio= 1.00000

Suite →

F) LOCAL SPACECRAFT SYSTEMS TRANSFORMATIONS

=====

15 19 21 -21 -19 -15
GEO -> GSE -> SR2 -> SR -> SR2 -> GSE -> GEO

input choosed test vector in GEO system :

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

input choosed spin axis in GSE system :

X= 0.34202 r = 2.00000
Y= 0.06031 teta= 170.000
Z= -1.96962 phi = 10.000

input choosed Spin frequency, Spin phase, dt:

Spin frequency= 0.2500 Hz
Spin phase = 30.0000 deg.
delta T = 1.2345 sec.

15) converts GEO system to GSE system by tgeogse

X= 0.09996 r = 5.00000
Y= 0.57634 teta= 6.718
Z= 4.96567 phi = 80.160

19) converts GSE system to SR2 system by tgsesr2

X= 0.94425 r = 5.00000
Y= -0.72804 teta= 166.204
Z= -4.85575 phi = -37.633

21) converts SR2 system to SR system by tsr2sre

X= -0.57328 r = 5.00000
Y= -1.04547 teta= 166.204
Z= -4.85575 phi = -118.738

-21) converts SR system to SR2 system by tsresr2

X= 0.94425 r = 5.00000
Y= -0.72804 teta= 166.204
Z= -4.85575 phi = -37.633

-19) converts SR2 system to GSE system by tsr2gse

X= 0.09996 r = 5.00000
Y= 0.57634 teta= 6.718
Z= 4.96567 phi = 80.160

-15) converts GSE system to GEO system by tgsegeo

X= 1.25000 r = 5.00000
Y= 2.16506 teta= 30.000
Z= 4.33013 phi = 60.000

difference beetveen first GEO wector: angle= 0.000 (deg.)
ratio= 1.00000

end of test program

Summary of available subroutine

• **“Basic computation” subroutines :**

module	arguments	object	position (line #)
cangrat	(ux,uy,uz,vx,vy,vz,angle,ratio)	compute_angle_and_ratio between U and V vectors	42
cdipdir	(year,idoy,d1,d2,d3)	compute_dipole_direction in GEO system	271
csundir	(year,idoy,ih,im,is,qst,slong,sra,sdec,obliq)	compute_sun_direction in GEI system	774
csunset	(year,imon,iday,rlat,r lon,tmer,tris,tset,durd)	compute_sunset_time and others	857
ctimpa2	(jd1950,houday)	compute_time_parameters and time-dependent matrix	1112
ctimpa3	(jd2000,houday)	compute_time_parameters and time-dependent matrix	1143
ctimpar	(year,imonth,iday,ih,im,is)	compute_time_parameters and time-dependent matrix	1174

• **“Calendar” subroutines :**

module	arguments	object	position
cdatdoy	(idoy,iyear,imonth,iday)	compute_date_from_day_of_year and for a given year	97
cdatj00	(jd00,iyear,imonth,iday)	compute_date_from_julian_day_2000 with jd00=0 for jan. 1	155
cdatj50	(jd50,iyear,imonth,iday)	compute_date_from_julian_day_1950 with jd50=0 for jan. 1	179
cdatwee	(iweek,iyear,imonth,iday)	compute_date_for_first_day_of_week_number	225
cdowweek	(iyear,imonth,iday,idow)	compute_day_of_the_week	415
cdoyear	(iyear,imonth,iday,idoy)	compute_day_of_year with idoy=1 for january 1	453
cfrdayn	(iday,cday,nbcha)	compute_french_day_name, ex: 'Lundi' for iday=1	503
cfrmonn	(imonth,cmonth,nchar)	compute_French_month_name	537
chouday	(ih,im,is,houday)	compute_hour_of_day from hours, minutes, seconds	577
cjd1950	(iyear,imonth,iday,jd50)	compute_julian_day_1950 with jd50=0 for january 1, 1950	598
cjd2000	(iyear,imonth,iday,jd00)	compute_julian_day_2000 with jd00=0 for january 1, 2000	638
cmilday	(ih,im,is,ims,milday)	compute_millsec_of_day from hours, minutes, seconds, ms	670
cnbdmon	(iyear,imonth,nbday)	compute_number_of_day_of_the_month	691
coleapy	(iyear,ileap)	compute_leap_year with ileap=1 for leap year, 0 if not	731
ctimhou	(houday,ih,im,is)	compute_time from decimal hour of the day	1047
ctimmil	(milday,ih,im,is,ims)	compute_time from millsec. of the day	1079
cusdayn	(iday,cday,nbcha)	compute_US_day_name, ex: 'Monday' for iday=1	1442
cusmonn	(imonth,cmonth,nchar)	compute_US_month_name	1476
cweedoy	(iyear,imonth,iday,iweek)	compute_week_of_the_year	1514

• **“Give” subroutines :**

module	arguments	object	position
gdipdir	(dxgei,dygei,dzgei,dxgeo,dygeo,dzgeo)	give_dipole_direction in GEI and GEO system	1573
gdipatan	(diptan)	give_dipole_tilt_angle in radians	1604
gecldir	(exgei,eygei,ezgei,exgeo,eygeo,ezgeo)	give_ecliptic_direction in GEI and GEO system	1624
gsrodir	(rxgei,rygei,rzgei,rxgeo,rygeo,rzgeo)	give_sun_rotation_direction in GEI and GEO system	1658
gsundir	(sxgei,sygei,szgei,sxgeo,sygeo,szgeo)	give_sun_direction in GEI and GEO system	1689
gsunpar	(gmst,slon,sras,sdec,obli)	give_sun_parameter dependant of time in GEI system	1721
qvernum	(vernum,verdat)	give_version_number and modification date of the library	1751

• **“Read and check” subroutines :**

module	arguments	object	position
recoor	(x,y,z,cs)	read coordinate values from input	1857
recsys	(csys)	read coordinate system from input and check validity	1919
redate	(year,imonth,iday)	read_date from input and check validity	1960
retime	(ih,im,is)	read_time from input and check validity	2023

• **“Transformation” subroutines :**

module	arguments	object	position
tcarsph	(x,y,z,r,teta,phi)	transforms_car_to_sph: CAR → SPH system	2070
tdmegeo	(xdme,ydme,zdme,rlat,rlong,xgeo,ygeo,zgeo)	transforms_dme_to_geo: DM → GEO system	2113
tgeigeo	(xgei,ygei,zgei,xgeo,ygeo,zgeo)	transforms_gei_to_geo: GEI → GEO system	2161
tgeigse	(xgei,ygei,zgei,xgse,ygse,zgse)	transforms_gei_to_gse: GEI → GSE system	2187
tgeigsm	(xgei,ygei,zgei,xgsm,ygsm,zgsm)	transforms_gei_to_gsm: GEI → GSM system	2214
tgeigsq	(xgei,ygei,zgei,xgsq,ygsq,zgsq)	transforms_gei_to_gsq: GEI → GSEQ system	2241
tgeimag	(xgei,ygei,zgei,xmag,ymag,zmag)	transforms_gei_to_mag: GEI → MAG system	2268
tgeisma	(xgei,ygei,zgei,xsma,ysma,zsma)	transforms_gei_to_sma: GEI → SM system	2295
tgeodme	(xgeo,ygeo,zgeo,rlat,rlong,xdme,ydme,zdme)	transforms_geo_to_dme: GEO → DM system	2322
tgeoget	(xgeo,ygeo,zgeo,xgei,ygei,zgei)	transforms_geo_to_gei: GEO → GEI system	2369
tgeoqse	(xgeo,ygeo,zgeo,xgse,ygse,zgse)	transforms_geo_to_gse: GEO → GSE system	2395
tgeogsm	(xgeo,ygeo,zgeo,xgsm,ygsm,zgsm)	transforms_geo_to_gsm: GEO → GSM system	2422
tgeogsq	(xgeo,ygeo,zgeo,xgsq,ygsq,zgsq)	transforms_geo_to_gsq: GEO → GSEQ system	2449
tgeomag	(xgeo,ygeo,zgeo,xmag,ymag,zmag)	transforms_geo_to_mag: GEO → MAG system	2476
tgeosma	(xgeo,ygeo,zgeo,xsma,ysma,zsma)	transforms_geo_to_sma: GEO → SM system	2503
tgeovdh	(xgeo,ygeo,zgeo,rlat,rlong,xvdh,yvdh,zvdh)	transforms_geo_to_vdh: GEO → VDH system	2530
tgsegei	(xgse,ygse,zgse,xgei,ygei,zgei)	transforms_gse_to_gei: GSE → GEI system	2575
tgsegeo	(xgse,ygse,zgse,xgeo,ygeo,zgeo)	transforms_gse_to_geo: GSE → GEO system	2602
tgsegsm	(xgse,ygse,zgse,xgsm,ygsm,zgsm)	transforms_gse_to_gsm: GSE → GSM system	2629
tgsegsq	(xgse,ygse,zgse,xgsq,ygsq,zgsq)	transforms_gse_to_gsq: GSE → GSEQ system	2655
tgsestr2	(xgse,ygse,zgse,rotx,roty,rotz,xsr2,ysr2,zsr2)	transforms_gse_to_sr2: GSE → SR2 system	2681
tgsmgei	(xgsm,ygsm,zgsm,xgei,ygei,zgei)	transforms_gsm_to_gei: GSM → GEI system	2730
tgsmgeo	(xgsm,ygsm,zgsm,xgeo,ygeo,zgeo)	transforms_gsm_to_geo: GSM → GEO system	2755
tgsmgse	(xgsm,ygsm,zgsm,xgse,ygse,zgse)	transforms_gsm_to_gse: GSM → GSE system	2784
tgsmgsq	(xgsm,ygsm,zgsm,xgsq,ygsq,zgsq)	transforms_gsm_to_gsq: GSM → GSQ system	2810
tgsmmag	(xgsm,ygsm,zgsm,xmag,ymag,zmag)	transforms_gsm_to_mag: GSM → MAG system	2832
tgsmisma	(xgsm,ygsm,zgsm,xsma,ysma,zsma)	transforms_gsm_to_sma: GSM → SM system	2854
tgsggei	(xgsq,ygsq,zgsq,xgei,ygei,zgei)	transforms_gsq_to_gei: GSEQ → GEI system	2880
tgsggeo	(xgsq,ygsq,zgsq,xgeo,ygeo,zgeo)	transforms_gsq_to_geo: GSEQ → GEO system	2907
tgsggse	(xgsq,ygsq,zgsq,xgse,ygse,zgse)	transforms_gsq_to_gse: GSEQ → GSE system	2934
tgsggsm	(xgsq,ygsq,zgsq,xgsm,ygsm,zgsm)	transforms_gsq_to_gsm: GSQ → GSM system	2960
tmaggei	(xmag,ymag,zmag,xgei,ygei,zgei)	transforms_mag_to_gei: MAG → GEI system	2982
tmaggeo	(xmag,ymag,zmag,xgeo,ygeo,zgeo)	transforms_mag_to_geo: MAG → GEO system	3009
tmaggsm	(xmag,ymag,zmag,xgsm,ygsm,zgsm)	transforms_mag_to_gsm: MAG → GSM system	3036
tmagsma	(xmag,ymag,zmag,xsma,ysma,zsma)	transforms_mag_to_sma: MAG → SM system	3058
tsmagei	(xsma,ysma,zsma,xgei,ygei,zgei)	transforms_sma_to_gei: SM → GEI system	3084
tsmageo	(xsma,ysma,zsma,xgeo,ygeo,zgeo)	transforms_sma_to_geo: SM → GEO system	3111
tsmagsm	(xsma,ysma,zsma,xgsm,ygsm,zgsm)	transforms_sma_to_gsm: SM → GSM system	3138
tsmamag	(xsma,ysma,zsma,xmag,ymag,zmag)	transforms_sma_to_mag: SM → MAG system	3164
tsphcar	(r,teta,phi,x,y,z)	transforms_sph_to_car: SPH → CAR system	3190
tsr2gse	(xsr2,ysr2,zsr2,rotx,roty,rotz,xgse,ygse,zgse)	transforms_sr2_to_gse: SR2 → GSE system	3215
tsr2mfa	(xsr2,ysr2,zsr2,bx,by,bz,rox,roy,roz,xm,ym,zm)	transforms_sr2_to_mfa: SR2 → MFA system	3263
tsr2sre	(xsr2,ysr2,spifre,spipha,deltaT,xsre,ysre)	transforms_sr2_to_sre: SR2 → SRef. system	3332

tsresr2	(xsre,ysre,spifre,spipha,deltaT,xsr2,ysr2)	transforms_sre_to_sr2: SRef → SR2	system	3369
tvdhgeo	(xvdh,yvdh,zvdh,rlat,rlong,xgeo,ygeo,zgeo)	transforms_vdh_to_geo: VDH → GEO	system	3406

• ***“Print information” subroutine :***

module	arguments	object	position
plibinf		print_library_informations	1785

Bibliography

- [1] Geophysical coordinate transformations, C.T. Russell, cosmic electrodynamics, v.2, 184-196, 1971.
- [2] CLUSTER Software Tools, Part 1 : Coordinate Transformation Library, Version 1.1, by Patrick Robert, CNRS-CNET/CRPE, Document de travail DT/CRPE/1231, Juillet 1993.
- [3] CLUSTER DATA PROCESSING, Transformation of a STAFF waveform into a Magnetic Field Aligned coordinate system, by Patrick Robert and C. de Villedary, Rapport interne CNRS-UVSQ/CETP n° RI-CETP/6/2000, Octobre 2000.
- [4] ROCOTLIB: a Coordinate Transformation Library for Solar-Terrestrial studies, Version 1.7, by Patrick Robert, Rapport interne CNRS-UVSQ/CETP n° RI-CETP/02/2003, January 2003

NOTES

NOTES
