Doc. No.: DS-JSO-TN-0008

Cluster
Joint
Science
Operations
Centre

Coordinate transformations for Cluster

prepared by Mike Hapgood

Issue 1.3   2 October 1997

DOCUMENT CHANGE RECORD

Version	Date	Notes/remarks
Issue 1.0	7-Sep-1994	Issue for IWG#13
Issue 1.1	16-Nov-1994	Clarified despin transformation in section 4: * terminology consistent with current FGM terms * role of SR1 as a convenient construct made explicit
Issue 1.2	3-Jun-1997	Corrected equations 3 and 7 following comments by Iannis Dandouras. Added reference to paper on role of precession in coordinate transformations. Updated Appendix A so that it matches the operational JSOC software for coordinate transformations. Updated specification of geomagnetic pole to the Cluster-II era.
Issue 1.3	2-Oct-1997	Delete text on fixed position for geomagnetic pole - reflects adoption of time-varying position by JSOC and UK-CSC(QSAS). Update Appendix A to show consequential change in interfaces.

Margin bars indicate changes with respect to previous issue.

REFERENCE DOCUMENTS

[COORD1] Russell, C.T., "Geophysical Coordinate Transformations". *Cosmic Electrodyn.* **2**, 184-196, 1971.

[COORD2] Hapgood, M.A., "Space Physics Coordinate Transformations: A User Guide." *Planet. Space Sci.* **40**, 711-717, 1992.

[COORD3] Hapgood, M.A., "Space Physics Coordinate Transformations: The role of precession." *Ann. Geophysicae.* **13**, 713-716, 1995.

[DDID] "Data Delivery Interface Document", CL-ESC-ID-0001, Issue 2.3, 7-June-1994.

[FGMPRO] "Fluxgate magnetometer data processing for Cluster", CL-IGM-SN-0001, Issue 1, Rev. 1, 8-June-1993.

[IGRF95] International Association of Geomagnetism and Aeronomy (IAGA) Division V - Working Group 8, "International Geomagnetic Reference Field, 1995 revision." *Geophys. J. Int.* **125**, 318-321, 1996.

1. INTRODUCTION

The note proposes a set of algorithms that can support the coordinate transformations required to manipulate orbit and scientific data from the Cluster mission. The CSDS Implementation Working Group is invited to endorse these algorithms and, if appropriate, to recommend them as a standard for CSDS data handling.

This notes makes extensive use of the matrix notation presented in [COORD2] - namely that the expression <æ,axis> denotes the matrix required for a simple rotation through angle æ around one of the principal axes, i.e. one of X, Y and Z. Thus, for a rotation about the Z axis we have:

$$< z ,Z \geq \begin{bmatrix} \cos z & \sin z & 0 \\ -\sin z & \cos z & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{0}$$

Note the signs of the two sin æ terms in equation (1) define the sense of positive rotation.

All the coordinate transformations required for Cluster can be factorised into simple matrices of the form shown in equation (1). This factorisation of transformations into sequences of simple matrices is strongly recommended. It allows complex transformations to be performed by repeated use of simple functions, e.g. for matrix multiplication, to construct simple matrices. Thus the complexity of software is reduced, thereby aiding its production and maintenance. In particular, the factorisation eliminates the need to code complex trigonometric functions.

2. Scientific coordinate systems

The use of matrix methods to carry out coordinate transformations in space physics was introduced by Russell [COORD1]. A recent paper by Hapgood [COORD2] has extended Russell's work by giving a more comprehensive and up-to-date specification of the required matrices. It is recommended that the CSDS adopt the algorithms in [COORD2] which support transformations between the following systems:

- GEI geocentric equatorial inertial - in mean epoch of date
- GEO geographic
- GSE geocentric solar ecliptic
- GSM geocentric solar magnetospheric
- SM solar magnetic
- MAG geomagnetic

- HAE heliocentric aries ecliptic
- HEE heliocentric earth ecliptic
- HEEQ heliocentric earth equatorial

Coordinate transformation software, implementing the algorithms of [COORD2], is available from the author. See appendix.

3. Precession

Cluster orbit and attitude data are supplied by ESOC in the geocentric equatorial inertial system of epoch J2000.0. Thus to apply the algorithms of [COORD2] these data must first be transformed to the geocentric equatorial inertial system of the mean epoch of date. This transformation is a small adjustment that allows for the precession of the Earth's rotation axis. The Cluster orbit software provided by ESOC includes a Fortran subroutine (PR2000.FOR) that will generate the matrix **P** required to carry out this transformation - see [DDID].

The role of precession in the space physics coordinate transformations specified in both [COORD1] and [COORD2] is discussed in [COORD3].

4. Converting from spacecraft systems to inertial systems

Appendix H of [DDID] defines the Cluster Spin Reference (SR) System $\{X_{SR}, Y_{SR}, Z_{SR}\}$ whose Z axis is the maximum principal inertia axis of the spacecraft, which is the spin axis of the spacecraft when nutation and oscillations have been damped out. We can transform vectors from this system to an inertial frame - using the information provided in the Cluster SATT files ([DDID], Appendix E.5) - as follows.

4.1 Despin

First we must despin the vectors. To do this we define a despun version of the SR system. This new version has the same Z axis as the SR system; but the X axis is the intersection of XY plane of the SR system with the SR meridian containing the Sun (i.e. the Sun Reference pulse is generated when the Sun Sensor crosses this meridian). We term this system SR2. To convert from SR to SR2 we simply rotate around the common Z axis by the spin phase ã:

$$V_{SR2} = < -g, Z > V_{SR} \qquad \mathbf{0}$$

Note that the SR2 system is identical with the SCS (spacecraft-sun) system[1] coordinate system used in FGM processing **[FGMPRO]**.

[1]formerly termed the Non-Spinning Spacecraft (NSS) system - with a different order of axes.

4.2 SR2 to inertial

To specify the transformation from SR2 to inertial coordinates, we introduce a further variant of the Spin Reference System. This has the same Z axis but the X axis is the intersection of the XY plane of the inertial system with the XY plane of the SR system. We term this system SR1. The positive sense of X_{SR1} is parallel to the vector product $Z_{SR} \times Z_I$. Note that SR1 has no physical significance. It is just a means to construct the transformation from SR2 to inertial.

<u>4.2.1 SR1 to inertial</u>

To transform a vector **V** from the SR1 coordinate system to the inertial system use the following equation:

$$V_I = < 90 - a, Z >< 90 - d, X > V_{SR} 1 \qquad \mathbf{0}$$

where á is the right ascension of the Z_{SR} axis and ä is its declination. These two quantities are available in the Cluster SATT file as fields SPRASC and SPDECL - see [DDID], p. 77. These values are expressed in the inertial system of J2000.0 and thus equation (3) will yield $\mathbf{V_I}$ in the same system. The precession matrix **P** must be used convert $\mathbf{V_I}$ to the mean of date inertial system before converting to the various scientific coordinate systems.

<u>4.2.2 SR2 to SR1</u>

In principle this is a simple transformation:

$$V_{SR} 1 = < -q, Z > V_{SR} 2 \qquad \mathbf{0}$$

However the determination of the rotation angle è requires some subtlety. The X_{SR2} axis is defined by the SR meridian plane containing the Sun, i.e. the GSE direction {1,0,0}. Thus, if we convert this vector to the SR1 system, we can derive the longitude of the Sun in the SR1 system. This longitude is identical to the required rotation è.

The required transformation is:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = < 90 - f, X >< a - 90, Z > P^{-1} T_2^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{0}$$

where **P** is the precession matrix and $\mathbf{T}_2$ is the transformation matrix from GEI mean-of-date to GSE specified in [COORD2]. è is then given by:

$$q = \begin{cases} \arccos(x/\sqrt{x^2+y^2}) & \text{if} \quad y > 0 \\ 360 - \arccos(x/\sqrt{x^2+y^2}) & \text{if} \quad y \leq 0 \end{cases} \qquad \mathbf{0}$$

4.3 Summary

The transformation from SR to inertial coordinates may be written:

$$V_I = <90 - a, Z><90 - d, X><-q, Z><-g, Z> V_{SR} \qquad \mathbf{0}$$

Thus the SR system may be related to any other spacecraft-based coordinate system.

Appendix - Coordinate transformation software

This is available from the author as a subroutine library in the form of Fortran 90 source code. The present version supports only transformations between the scientific coordinate systems as specified in section 2; it would be straightforward to add precession and the SR2 spacecraft systems.

External interfaces

1. Subroutine STARTCT(CONTROL_CHANNEL,IO_STATUS) initialises the software. It must be called at start of the application program. CONTROL_CHANNEL is the Fortran channel number on which data file TERMS2.DAT (see below) is read; IO_STATUS is status code returned by the open operation on TERMS.DAT (and is zero if it opened ok).

2. Subroutine CALANGLE(MJD,UT,STATUS) calculates all transformation angles for the date and time of interest. It also calculates the position of the geomagnetic pole at that time. It must be called whenever the time changes.

a. MJD is a 4 byte integer that contains the Modified Julian Date at 00:00 UTC on the day of interest. This uses the normal textbook definition of MJD as the number of days from 00:00 UT on 17 November 1858. Do not confuse with the MJD2000 used by the ESOC orbit software (MJD = MJD2000 + 51544).

b. UT is a 4 byte real that contains the Universal Time, measured in hours, from 00:00 UTC on the day of interest.
c. STATUS is a status code generated by the software which calculates the position of the pole; it is non-zero if the given date is outside the range of dates for which the position is calculable.

3. Subroutine TRANSYS (X,SYS,FORM,RCODE,OUTSYS,OUTFORM) transforms a vector. CALANGLE must be called first to calculate the angles used in the transformation.

a. X is a 4 byte real array of 3 elements containing the vector of interest. Contains initial vector on entry and transformed vector on exit.
b. SYS is a 4 byte string containing the name of the coordinate system associated with vector X, i.e. initial system on entry and final system on exit. See comments in the source code for allowed values of SYS.
c. FORM is a 3 byte string indicating whether X is expressed in Cartesian form (FORM=' CA'; X={x,y,z}) or Polar form (FORM=' PO'; X={latitude, longitude, magnitude}). Contains initial form on entry and final form on exit.
d. RCODE is a 4 byte integer used as a return code.
e. OUTSYS is a 4 byte string containing the name of the coordinate system in which vector X is required. Unchanged on exit.
f. OUTFORM is a 3 byte string containing the form in which vector X is required. Unchanged on exit.

Internal interfaces

1. Data file TERMS2.DAT is read on the Fortran channel assigned in the call to STARTCT. The file is opened and closed from within the calling routine.

2. Common block TABLE contains specifications of the coordinate systems and the transformations between them.

3. Common block MPOLE contains the co-latitude and longitude of the north pole of the magnetic dipole of the Earth.

4. Common block ANGLE contains the 12 rotation angles needed to construct the coordinate transformations.

Other interfaces

Some of the modules in the library may be of interest:

1. Integer function MJD(YEAR,MONTH,DAY) calculates the Modified Julian Date given 4 byte integer values of YEAR (all 4 digits), MONTH and DAY.

2. Subroutine MAKTRAN(INSYS,OUTSYS,T) calculates the matrix T to transform a cartesian vector from the system described in string INSYS to the system described in string OUTSYS. INSYS and OUTSYS are 4 byte strings and are unchanged on exit. T is a 4 byte real array with 3 x 3 elements.