Processing of DMSP Magnetic Data: Handbook of Programs, Tapes and Datasets

R.A. Langel, T.J. Sabaka, and J.R. Ridgway

February 1990
Processing of DMSP Magnetic Data: Handbook of Programs, Tapes and Datasets

R.A. Langel  
*Goddard Space Flight Center*  
*Greenbelt, Maryland*

T.J. Sabaka and J.R. Ridgway  
*Science Applications Research*  
*Lanham, Maryland*

---

NASA
National Aeronautics and Space Administration

*Goddard Space Flight Center*  
*Greenbelt, MD*

1990
Abstract

The DMSP F-7 satellite was an operational Air Force meteorological satellite which carried a magnetometer for geophysical measurements. The magnetometer was located within the body of the spacecraft in the presence of large spacecraft fields. In addition to stray magnetic fields, the data have inherent position and time inaccuracies. Algorithms were developed to identify and remove time varying magnetic field noise from the data. These algorithms are embodied in an automated procedure which fits a smooth curve through the data and then identifies outliers and which filters the predominant fourier components of noise from the data. Techniques developed for Magsat were then modified and used to attempt determination of the spacecraft fields, of any rotation between the magnetometer axes and the spacecraft axes, and of any scale changes within the magnetometer itself. Software setup and usage are documented and program listings are included in the Appendix. The initial and resulting data are archived on magnetic cartridge and the formats documented.
I. Introduction

The DMSP F7 spacecraft was launched on 18 Nov, 1983 into a 98.74 degree inclination orbit, with apogee 844 km altitude and perigee 822 km. (Rich, 1984). The primary purpose of the spacecraft was to obtain tropospheric meteorological data. However, a triaxial fluxgate magnetometer was included on the spacecraft in order to monitor the geophysical environment. This report is the second of two dealing with the examination of these magnetometer data to evaluate their usefulness in describing the earth's core-produced geomagnetic field. The first report "Processing of DMSP Magnetic Data and its Use in Geomagnetic Field Modeling" (Ridgway et. al, 1989), henceforth referred to as paper I, gave an overall summary of the processing methods and results and of the field modeling efforts. Some of the material in that report is duplicated in the present document. However the emphasis in this document is to describe the software utilized, the crucial data sets and the processing procedures. All pertinent data sets, code, JCL, etc. are stored on magnetic cartridge, as documented herin.  

The DMSP F7 magnetometer was mounted on the satellite body, as opposed to being attached to a boom, because of spacecraft engineering constraints (Rich, 1984). The magnetometer, a triaxial fluxgate, was aligned with the spacecraft X, Y and Z axes, which are defined as follows: X is vertically down, Y is along-track and Z is cross-track. The three sensor units were built by the Schonstedt Instrument Co., Reston, VA, in the 1960's. The electronics unit for the magnetometer was built by the Applied Physics Laboratory of Johns Hopkins University, Laurel, MD, based on the design of the MAGSAT fluxgate magnetometer.  

The magnetometer acquired field measurements at a rate of 20 samples per second. Measurements were in the form of counts, with one count equalling 12 nano-Teslas (nT). According to Rich (1984), the instrument was not intended to survey the main geomagnetic field, so it was not calibrated with high accuracy on the ground, nor recalibrated in orbit.  

Because of the close proximity of the magnetometer to on-board electronic instrumentation, its data were contaminated by non-random instrumental noise, with magnitudes of up to several thousand nT. The attitude of the spacecraft was measured to an accuracy of about 0.1 degree, or 360 arc-seconds. While this attitude accuracy is not as good as that obtained with MAGSAT, in principle it is of sufficient accuracy to enable meaningful vector measurements. In the absence of other near-Earth satellite magnetic field data for this time period, and in view of the success of methods used on MAGSAT to solve for spacecraft fields, it was decided to investigate the possibility of processing the DMSP F7 data to a stage where they may be useful for main field modeling.  

II. Transformation of On-tape Data to Magnetic Readings.  

The magnetometer data contained on the basic DMSP data tapes received from the Air Force is in the form of magnetometer counts, which must be converted into field values in nT in order to be useful. Data is arranged
on the tape as a header, containing time and position information for each minute of operation, followed by 60 magnetometer readings (1 per second). See description of tape format in Section VII. All times on the data are rounded to the nearest second. All positions on the original Air Force data are expressed in nautical miles, rounded down to the nearest nautical mile. Nautical miles were converted to kilometers (one nautical mile equals about 1.8 km) in subsequent data processing.

20 readings per second for each of the magnetometer X, Y and Z axes were originally recorded by DMSP, although only readings #1 and #11 were written on the tapes sent to the Geology and Geomagnetism branch at Goddard. For the Goddard main field studies, only the magnetometer reading associated with the header record was utilized. This preserved sufficient data density (one reading per minute) to fully describe the main geomagnetic field.

The magnetometer was calibrated prior to launch at the NASA Goddard Space Flight Center magnetic test chamber with the following results (Rich, 1984):

1) \[
\text{Measurement} = \text{Calibration Matrix} \times \text{Measurement} + \text{Bias}
\]

\[
\begin{align*}
(\text{radial}) \quad B_X &= \begin{bmatrix} 12.1001 & -0.0055 & 0.0193 \end{bmatrix} \times \begin{bmatrix} C_X \end{bmatrix} + \begin{bmatrix} -0.0653 \end{bmatrix} \\
(\text{along-trk}) \quad B_Y &= \begin{bmatrix} -0.0247 & 12.1863 & -0.0101 \end{bmatrix} \times \begin{bmatrix} C_Y \end{bmatrix} + \begin{bmatrix} 59.8733 \end{bmatrix} \\
(\text{cross-trk}) \quad B_Z &= \begin{bmatrix} 0.0069 & 0.0232 & 12.1735 \end{bmatrix} \times \begin{bmatrix} C_Z \end{bmatrix} + \begin{bmatrix} 39.4228 \end{bmatrix}
\end{align*}
\]

This equation is used to compute the magnetic field in spacecraft coordinates in nT, given a reading in magnetometer counts. However, this calibration does not take into account the field from the spacecraft, which adds greatly to the bias vector. This vector must be determined from in-flight data. Later work using the FIT program (see Field Value corrections) accomplished this, and re-determined the bias vector as: (89nT, 8457nT, -1441nT) for radial, along-track and cross-track measurements, respectively. These values are still somewhat approximate and require small corrections discussed in paper 1. In addition, for computational ease, it was decided to redefine the spacecraft system to be compatible with the Magsat coordinate system, so that the spacecraft X axis is defined as cross-track, Y is radially down and Z is along-track. The correct transformation of DMSP magnetometer counts to nano-teslas in spacecraft coordinates compatible with Magsat is thus:
2) Measurement = Calibration Matrix * Measurement + Bias 

\[
\begin{bmatrix}
(B_x) \\
(B_y) \\
(B_z)
\end{bmatrix} = 
\begin{bmatrix}
0.0069 & 0.0232 & 12.1735 \\
12.1001 & -0.0055 & 0.0193 \\
-0.0247 & 12.1863 & -0.0101
\end{bmatrix} 
\begin{bmatrix}
(C_x) \\
(C_y) \\
(C_z)
\end{bmatrix} + 
\begin{bmatrix}
-1441 \\
89 \\
8457
\end{bmatrix}
\]

All data discussed in the remainder of this report are assumed to have been processed through this equation and have units of nT. The \((B_x, B_y, B_z)\) measurement vector in equation #2 will henceforth have the label \(B_{spu}\), meaning the vector is in MAGSAT spacecraft coordinates and not yet processed through the final corrections.

III. Field Value Corrections.

According to Rich (1984), the DMSP magnetometer may be misaligned relative to the spacecraft by as much as 0.5 degree per axis, with the misalignment measured to an accuracy of about 0.1 degree. Also, bending of the spacecraft body may result in further misalignment. In addition, the values of the three magnetic components may be in error by a fixed bias or by a multiplying factor. The FIT program has the capability to solve for corrections in these parameters in conjunction with the least squares main field solution. The theory of this adjustment is as follows (see also Estes, 1983):

The FIT program computes three types of adjustments to vector satellite magnetometer data: 1) A diagonal calibration matrix containing "slope" parameters, which is multiplied times the measured vector to correct for magnetometer drift, 2) a bias correction vector which is subtracted from the measured vector to correct for constant magnitude offsets, and 3) a rotation matrix which is multiplied times the measured vector to correct for angular offsets of the magnetometer from ideal satellite coordinates. These adjustment parameters are applied to the measured uncorrected data in spacecraft coordinates according to the equation:

\[
B_{spc} = TSM \times TCAL \times (B_{spu} - bias)
\]

where:
- \(B_{spu}\) is the uncorrected measurement vector in spacecraft coordinates, as given in equation 2).
- \(B_{spc}\) is the corrected measurement vector in spacecraft coordinates.
- \(bias\) is a vector of magnetometer bias corrections in addition to those given in equation 2).
- \(TCAL\) is the calibration correction matrix of slope parameters.
- \(TSM\) is the rotation correction matrix.
The elements of **bias** are: \( (BS_1, BS_2, BS_3) \), where \( BS_i \) are component biases derived in the FIT program, with values derived and discussed in paper 1.

\[
\text{TCAL has elements: } \begin{bmatrix}
1/SL_1 & 0 & 0 \\
0 & 1/SL_2 & 0 \\
0 & 0 & 1/SL_3
\end{bmatrix}
\]

where \( SL_i \) are slopes derived in the FIT program. \( SL_1 \) and \( BS_1 \) are applied to the satellite X axis, \( SL_2 \) and \( BS_2 \) to satellite Y axis, and \( SL_3 \) and \( BS_3 \) to the satellite Z axis components.

The elements of **TSM** are based on three Euler angles \( (\epsilon_x, \epsilon_y, \epsilon_z) \) solved in execution of the FIT program. (Note: In the FIT program, as of 2/28/88, \( \epsilon_x \) is denoted \( \epsilon_2 \), \( \epsilon_y \) is denoted \( \epsilon_1 \), and \( \epsilon_z \) is denoted \( \epsilon_3 \).) Using the notation \( TSM_{ij} \), where \( i \) is the matrix row and \( j \) the matrix column, these are:

\[
\begin{align*}
TSM_{11} &= \cos \epsilon_y \cos \epsilon_z \\
TSM_{12} &= \cos \epsilon_y \cos \epsilon_x \sin \epsilon_z + \sin \epsilon_y \sin \epsilon_x \\
TSM_{13} &= -\cos \epsilon_y \sin \epsilon_x \sin \epsilon_z + \sin \epsilon_y \cos \epsilon_x \\
TSM_{21} &= -\sin \epsilon_z \\
TSM_{22} &= \cos \epsilon_x \cos \epsilon_z \\
TSM_{23} &= -\sin \epsilon_x \cos \epsilon_z \\
TSM_{31} &= -\sin \epsilon_y \cos \epsilon_z \\
TSM_{32} &= \cos \epsilon_y \sin \epsilon_x - \sin \epsilon_y \sin \epsilon_z \cos \epsilon_x \\
TSM_{33} &= \cos \epsilon_y \cos \epsilon_x + \sin \epsilon_y \sin \epsilon_x \sin \epsilon_z
\end{align*}
\]

The \( TSM \) matrix in the FIT program is derived from 3 rotation matrices (denoted \( R_x, R_y, R_z \)) in the spacecraft coordinate system. \( R_x \) is a left-handed rotation through the angle \( \epsilon_x \), about the spacecraft X axis. \( R_y \) is a left-handed rotation about the spacecraft Y axis, with angle of rotation \( \epsilon_y \). \( R_z \) is a right-handed rotation through the angle \( \epsilon_z \), about the spacecraft Z axis.
The matrices $R_x$, $R_y$ and $R_z$ are thus

$$
\begin{align*}
R_x &= \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \xi_x & -\sin \xi_x \\
0 & \sin \xi_x & \cos \xi_x
\end{bmatrix} \\
R_y &= \begin{bmatrix}
\cos \xi_y & 0 & \sin \xi_y \\
0 & 1 & 0 \\
-\sin \xi_y & 0 & \cos \xi_y
\end{bmatrix} \\
R_z &= \begin{bmatrix}
\cos \xi_z & \sin \xi_z & 0 \\
-\sin \xi_z & \cos \xi_z & 0 \\
0 & 0 & 1
\end{bmatrix}
\end{align*}
$$

Rotations provided by $R_x$, $R_y$ and $R_z$ are illustrated in Figures 1a), b) and c) in paper 1.

TSM is created by rotating about the $X$ axis first, then about $Z$, then about $Y$, e.g.,

$$
6) \quad \text{TSM} = R_y R_z R_x
$$

An older version of the FIT program, documented in Estes (1983), used a different order of rotation, about the $Z$ axis first, about the new $X$ axis second, and finally about the new $Z$ axis. It is now known that this $Z$-$X$-$Z$ rotation order fails to adequately resolve the first and third angles when they are large, and so the present order of rotation was instituted, with successful results.

The relation of euler angles $\xi_x$, $\xi_y$, $\xi_z$ to "roll, pitch, yaw" notation is dependent on the spacecraft axis designations. For example, since the MAGSAT $Z$-axis is pointed in its along-track direction, $\xi_z$ is roll, $\xi_y$(radial) is yaw, and $\xi_x$(cross track) is pitch.

The FIT program calculates some field quantities in earth-fixed cartesian coordinates. The coordinate origin is at earth's center; the $X$ axis points along $0^\circ$ longitude; the $Y$ axis points along $90^\circ$ meridian; and the $Z$ axis points along the geographic north pole. Information is therefore required on the relation between the corrected spacecraft measurement vector, $B_{spc}$, and its analog in earth-fixed coordinates $B_{ef}$, for every data point. This information is contained in transformation matrix $TGS$:

$$
7) \quad B_{ef} = TGS \times B_{spc}
$$
The TGS matrix itself is an approximation computed from the formula:

8) \[
\begin{bmatrix}
-n_x & -r_x & v_x \\
-n_y & -r_y & v_y \\
-n_z & -r_z & v_z
\end{bmatrix}
\]

where:
\[r_x = \cos \phi \cos \lambda, \quad r_y = \cos \phi \sin \lambda, \quad r_z = \sin \phi\]
\[n_x = \cos \phi_n \cos \lambda_n, \quad n_y = \cos \phi_n \sin \lambda_n, \quad n_z = \sin \phi_n\]
\[v_x = n_y r_z - r_y n_z, \quad v_y = n_z r_x - r_z n_x, \quad v_z = n_x r_y - r_x n_y\]

\(\phi, \lambda, \phi_n\) and \(\lambda_n\) are defined as:
\[
\begin{align*}
\phi & = \text{Geocentric latitude} \\
\lambda & = \text{East longitude} \\
\phi_n & = -8.74^\circ, \text{ the inclination of the vector normal to the orbit} \\
\lambda_n & = \lambda \pm \arccos[-\tan(\phi_n) \times \tan(\phi)], \text{ where + is used for a descending orbit (N to S) and - for an ascending orbit.}
\end{align*}
\]

It should be noted that this equation is not accurate for latitudes greater than about 75°. DMSP data between 75° and 81.26° were therefore not utilized in subsequent analyses. Derivation of this restriction and of the TGS transformation elements themselves may be found in Appendix B of paper 1.

Combining equations #3 and #7 yields:

9) \[\text{Bef} = \text{TGS} \times \text{TSM} \times \text{TCAL} \times (\text{Bspu} - \text{bias})\]

\(\text{Bspu}\) and \(\text{TGS}\) are read or computed from input data, and \(\text{TSM, TCAL}\) and \(\text{bias}\) are solved for in execution of the FIT program.

IV. Noise Sources in the Magnetic Data.

The DMSP data were examined initially by Rich (1984), who found three sources of magnetic noise. The first two are high frequency sinusoidal signals with periods of 0.576 and 3.456 seconds. These are caused by the rotating X-ray scanner, designated the SSB/S instrument, which is mounted 10 to 15 inches from the magnetometer sensors and generates a small magnetic field. These high frequency noise sources are of magnitude less than about 30 nT and are not a concern in the present study.
The third noise source found by Rich is the operation of the satellite torquing coils. These are turned on for durations of about 4 minutes at various times throughout the DMSP mission. When the coils are on, the magnetic field data is offset by a constant level shift of 3000 to 14,000 nT. This type of noise is screened out by the "gross outlier" criterion for reducing the data.

A fourth known noise source consists of fields in the 100 - 150 nT range which result from turning on transmitters and tape recorders when over a tracking station. It is assumed that most data so affected will be eliminated in the various outlier tests.

A noise source not discussed by Rich(1984) was discovered by examining orbital plots of residual data, i.e. data which have had a preliminary field model subtracted. These residual data show strong periodic trends with amplitudes of up to 70 nT, after other corrections were applied. An in-depth discussion of this periodic noise and its removal is found in paper I.

V. Automated Clean-Up Procedure for DMSP Data.

Preliminary field modeling

A test model was generated from DMSP data at epoch 1984.04. The $g_{10}$ term from this model equaled -29,900.4 nT. A model derived from observatory data at the same epoch yielded $g_{10}$ equal to -29,883.4. The closeness of the two terms suggests the apparent adequacy of the DMSP data for main field modeling. A calibration of the 1984.04 data, using the second procedure described in the previous section, was also executed with the following results: $SL_1 = 0.9955$, $SL_2 = 0.9996$, $SL_3 = 1.0025$. The nearness of these values to unity again suggests that the DMSP magnetometer measured the magnetic field accurately for that selection of data. These results indicated that DMSP data might be useful for main field modeling, in spite of the large spacecraft fields. All of these studies were conducted using only a few days of data. On the basis of these results, it was decided to proceed with a larger quantity of data.

The preliminary field model was removed from January 14-18 DMSP data to create residual data. Upon examination, these orbits of data showed strong periodicities. A spectral decomposition of the data revealed noise sources with periods equal to the orbit period (100 minutes) and subharmonics of 1/2, 1/3 and 1/4 of the orbit period. Figure 1 displays a typical residual orbit from this time period. The X and Y components most clearly demonstrate this periodic noise. Figure 2 is the associated spectrum. Peaks in the spectrum display these dominant noise periods quite noticeably. As will be seen below, one cause of the periodic noise is the need for adjustment of the Euler angle values in the TSM matrix in equation 3). The other causes of this periodic noise is unknown. The peak-to-peak amplitude of the noise is about 300 nT before Euler angle correction and about 50-70 nT after that correction.
Automated procedure description

A five-stage clean-up procedure was followed to remove data spikes and periodic noise from the DMSP data. In this procedure the data were processed through a data cleaning and filtering program written by T.J. Sabaka called FILTER. The stages are as follows: 1) Fit a span of DMSP data, covering several days, with a preliminary field model. Subtract the field model to get residual data. Reject data points above 75° absolute latitude, and reject "gross outliers", i.e., residual data with absolute values greater than a specified cutoff. Fit the residual data with a spline function and reject points which deviate more than 2 standard deviations from that function. 2) Add residual data which is not rejected back to the preliminary field model. Then fit a new field model to this data with epoch equal to the average time of that data span. Solve for constant main field coefficients, magnetometer angle adjustments and biases. 3) The new field model is reformatted for further use. 4) Correct the original data with the angle and bias solutions. Use the computed field from stage #3 to re-create the residual data, and re-do step #1, i.e. reject gross outliers and spline outliers. 5) Fit a Fourier function, which is composed of the 4 dominant noise periods (25 minutes, 33 minutes, 50 minutes and 100 minutes) in a least-squares manner to the residual data. Reject outliers according to the Fourier fit, using the $2\sigma$ criterion as for the spline fit. Then subtract the Fourier function from the data. Add the result back to the computed field model from step #2, to create the final, corrected data set.

Stage 5) is somewhat ad hoc. Such periodic variations could arise from source corrections we have either overlooked or been unable to apply. For example, comparison of Figures 3a and 3b shows that much of the large periodic oscillation results from unadjusted Euler angles. It is both more meaningful and reliable to correct the euler angles than to remove the variations via the Fourier fit. For this reason the ad hoc Fourier fit correction is applied last.

Figure 3 shows the same profile from Figure 2, after it has undergone the data cleaning process. Most of the periodic noise is gone, and the major spikes and outliers have been removed.

Table 1 summarizes the five stages. The input and output files are indicated for each stage.
### TABLE 1: FIVE STAGE CLEAN-UP PROCESS

<table>
<thead>
<tr>
<th>NAME</th>
<th>INPUT FILE</th>
<th>OUTPUT FILE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(DTAPE.PROCESS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STAGE 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1) A.F. Tape DATE</td>
<td>DATE.STEP1.OUTBIN</td>
<td>Translates magnetometer counts to nT. Fits residual orbits with a B spline, and flags outliers and points with non-determinable velocities. Puts data into FIT binary format.</td>
<td></td>
</tr>
<tr>
<td>2) DATA.MISC (VBS, irecl=11204, (CAL84FID) Blksize=22412)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3) XRTJS.BSPINFO.DATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) XRTJS.DMSP.STEP2.DATA</td>
<td>&quot; &quot; &quot; 3 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot; &quot; &quot; 4 &quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&quot; &quot; &quot; 5 &quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STAGE 2</strong></td>
<td>DATE.STEP1.OUTBIN</td>
<td>DATE.STEP2.COEFFS</td>
<td>Fits a field model to flagged data. Solves for magnetometer corrections (euler angles, magnetometer biases).</td>
</tr>
<tr>
<td><strong>STAGE 3</strong></td>
<td>DATE.STEP2.COEFFS</td>
<td>DATE.STEP3.COEFFS</td>
<td>Reformats field model.</td>
</tr>
<tr>
<td><strong>STAGE 4</strong></td>
<td>Same as STAGE 1 except DATE.STEP3.COEFFS is used in place of DATA.MISC(CAL84FID).</td>
<td>DATE.STEP4.OUTF (FB, irecl=240, Blksize=4800)</td>
<td>Same function as STEP1, but with a different field model.</td>
</tr>
<tr>
<td><strong>STAGE 5</strong></td>
<td>DATE.STEP4.OUTF, plus same files as STEP4.</td>
<td>1) DATE.STEP5.OUTF</td>
<td>Fits orbits with periodic fourier function, and removes this function from data. Data is output both in formatted and binary (FIT) formats.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) DATE.STEP5.OUTBIN</td>
<td></td>
</tr>
</tbody>
</table>

(Note: STEP1,STEP4,STEP5 utilize load module XRJRR.SATFILT, which contains the FILTER program. STEP2 utilizes module XRJRR.FIT.DMSP.LOAD2, which contains the old FIT program.)

The output from STAGE1, DATE.STEP1.OUTBIN, is not saved.
The output from STAGE2, DATE.STEP2.COEFFS, becomes file 1 on the output tape.
The output from STAGE3, DATE.STEP3.COEFFS, becomes file 2 on the output tape.
The output from STAGE4, DATE.STEP4.OUTF, becomes file 3 on the output tape.
The output from STAGE5, DATE.STEP5.OUTBIN, becomes file 4 on the output tape.
The output from STAGE5, DATE.STEP5.OUTF, becomes file 5 on the output tape.
Use of the FILTER Program

As noted, the five stages of the cleanup procedure are based on the program FILTER. Filter is comprised of five steps, each modularly designed. [Not to be confused with the "stages" in Table 1 and accompanying text.]

STEP 1: Involves reading of an original satellite magnetic data tape and transforming the raw magnetometer counts to magnetic field values in the spacecraft coordinate system.

STEP 2: Involves the location and padding of time gaps in the data and the determination of the direction of the spacecraft velocity vector at each measurement location.

STEP 3: Involves the transformation of the magnetic field measurements from spacecraft to topocentric coordinate system from which residual measurements are determined from a given field model. Data locations at which any vector residual exceeds the specified tolerance are flagged as outliers.

STEP 4: Involves fitting a trend to the magnetic field residuals with B-Splines and/or fourier waveforms, with the option of flagging points whose trend residuals exceed a given tolerance and the option of detrending the original data.

STEP 5: Involves outputting a final modified satellite magnetic tape in three basic forms:
   1) EBCDIC tape in topocentric coordinates
   2) EBCDIC tape in desired spacecraft coordinates
   3) Binary tape in old fit program format (Magsat convention)

Program FILTER may run in one of four modes indicated by the input variable IMODE:

IMODE = 0: Perform steps 1, 2, 3, 4, and 5.
IMODE = 1: Perform steps 4 and 5.
IMODE = 3: Perform steps 1, 2, 3, and 4.

The reader will note that program FILTER is very general. It use in processing DMSP data is a special case.

The correspondence between IMODE and the STAGE's of Table 1 is as follows:
   STAGE 1: IMODE = 0
   STAGE 4: IMODE = 0
   STAGE 5: IMODE = 1

STAGE 2 consists of running the old version of the FIT (Main Field Modeling) Program. For future work, this step must be modified to use the new FIT program.
STAGE 3 is a simple program which modifies the format of the SHA coefficients output from STAGE 2. When output they are in the standard format for the old FIT program. This stage converts the coefficients into the format needed by the program FID. FID is the standard program to compute magnetic field at a specified time and location from a set of SHA coefficients.

Automated Procedure Deck Setup

Typical setup (JCL) decks, annotated, for the five stages are as follows:

A. For STAGE's 1, 4, and 5, using program FILTER:

In these listings XRJR. SATFILT is the location of the load module for the FILTER program. The source code is presently in XRTJS.DMSP.FILT.CNTL. In the future both, as well as the run decks printed on the next few pages, will be saved on a cartridge. Details will be given later in this report.

SATFILT is a combination of the basic programs FILTER and BSPLYN3, with slight modifications for use with DMSP data. These modifications have to do with data plotting. The original programs produced plot output for use in the WOLFPLOT plotting package. The modifications permit plotting using the DIUTIL plot package. The modifications are used in conjunction with program ADDFLAG whose purpose is to create an ASCII file from program FILTER output which has both a residual field (core model subtracted) and a B-spline fit to that field, versus time in minutes, for one orbit. Points which are outliers from the B-spline fit are flagged. INOTE is the flag. When a data point has a value of INOTE of 1, 2, or 6, the data point is not output. The code to produce the plotting output requires the following code additions:

In BSPLT:

```
**********************************************************************
WRITE(25,665)
665 FORMAT(1X,'RAW DATA (FIRST) AND B-SPLINE FIT')
DO 31 KKK=I,NOBS
31 WRITE(25,666)XS(KKK),SS(KKK),VS(KKK)
WRITE(25,667) BLANK
666 FORMAT(3F10.3)
667 FORMAT(A1)
**********************************************************************
DO 59 JL=1,LTYPE
NKNT=KK(JL)
```
This code outputs to unit 25, which should be allocated in the JCL as a fixed-block ASCII file. After FILTER has been run and the output of unit 25 saved, ADDFLAG should be run.

Inputs to ADDFLAG: Unit 10 - file created by FILTER unit 25
Unit 15 - file created by FILTER unit 15
=DATE.STEP4.OUTF

Output from ADDFLAG: Unit 20. Another fixed-block ASCII file with format:

Title (72A1)
Time (min), Residual (nT), B-spline value (nT), Flag - (3F10.3,I5)
... Data values repeated N times ...
Blank line

This sequence is repeated three times, one for each component (X, Y, Z).

The ASCII file is plotted with DIUTIL plotting program DPLOT1, which inputs it on unit 8 and outputs a plot file using standard DIUTIL commands.

ADDFLAG2: is a simpler version of ADDFLAG, which outputs an ASCII file of Fourier detrended points after running FILTER, step 5. ADDFLAG2 does not require special code to be inserted into FILTER. It writes out only points with INOTE = 0,3,4,5.

Input to ADDFLAG2: Unit 15 - file created by FILTER unit15 = DATE.STEP5.OUTF.

Output from ADDFLAG2: An ASCII file in the same format as that produced by ADDFLAG and which may be plotted by program DPLOT1.
Note that Unit 10 contains the input DMSP data and that OPTCD=Q means that an ASCII file is expected.

Unit 15 is FILTER output in EBCDIC, in topocentric coordinates. Unit 17 is output in old FIT format, Binary, using Magsat coordinates. Unit 12 is the input field model coefficients and Unit 22 contains B-Spline and Fourier series information.

Other programs related to FILTER and to DMSP data processing are: BSIG, and POWPLT. BSIG calculates the mean and standard deviation of DMSP data relative to a given field model. It also calculates the dipole latitude of the data and will not use data above a specified dipole latitude. It only considers "good" points, i.e. with INOTE =0. It presently compares the DMSP data to the field model values contained on DATE.STEP5.OUTF; however, it may be modified to use an arbitrary field model by removing the comment cards from the section of code which calls the FID program. Inputs are on Unit 8 and Unit 10: Unit 8 is used for field model coefficients in FID format, if using an arbitrary model, otherwise this unit is not needed. Unit 10 = DATE.STEP(1,4,5).OUTF is an ASCII file output from step 1, 4 or 5. The output is in printed format only.
POWPLT is a wolfplot power spectrum plotting routine. It may run on any file which has the correct input format, but was created specifically to plot power spectra output by the following code inserted into program FILTER, subroutine SPECT, following the code before the asterisk:

```plaintext
IF(PLT.EQ.1) CALL PLOT(PERIOD,AMP,LTOTL,' ')
IF(PLT.EQ.2) CALL PLOT(PERIOD,PHI,LTOTL,' ')
IF(PLT.EQ.3) CALL PLOT(PERIOD,POWER,LTOTL,' ')
```

When FILTER is run with the inserted code, a fixed block ASCII file in the following format is output to unit 26:

- Title (A72)
- Period Power (2E15.8, repeated N times)
- Blank line
- >entire sequence repeated three times, once for each component.

This file is then input into POWPLT, which outputs a plot file onto unit 8, which for WOLFPLOT is a plot tape.

Use of Dst. Dst is added to DMSP data for use in the old FIT program. The program which does this is DSTADD. DSTADD requires Dst values in a certain format: (2X, I2, I3, 2X, 24I4), where the first variable is year past 1900, the second is day of year, followed by 24 Dst values for that day. The original data tape containing Dst (TD5696) is not in this format, but must be processed through DSTI (located on DMSP.PROGRAMS) to create a file suitable for input into DSTADD. DSTI also windows Dst values according to date.

The functioning of Program FILTER depends on the input variables specified in various NAMELIST statements. The following pages are a Glossary of the various variables that can be set in this manner.
NAMELIST IOFILE -
-------------

IST1  - INPUT UNIT FOR ORIGINAL RAW DATA TAPE(S) IN STEP1.

IST2  - INPUT UNIT IN STEP2, OUTPUT UNIT IN STEP1, MAGNETIC FIELD IN FIT/MAGSAT COORDINATES.

IST3  - INPUT UNIT IN STEP3, OUTPUT UNIT IN STEP2, VELOCITY DIRECTIONS AND PADDED TIME-GAPS.

IST4  - INPUT UNIT IN STEP4, OUTPUT UNIT IN STEP3, MAGNETIC FIELD AND RESIDUALS IN TOPOCENTRIC COORDINATES.

IOR   - FILTER INPUT UNIT, SAME AS IST4 IN OPERATION MODE 0 AND 3.

IOW   - FILTER OUTPUT UNIT, INPUT UNIT IN STEP5.

IOF   - OUTPUT UNIT IN STEP5, FORMATTED MAGNETIC FIELD IN FIT/MAGSAT OR TOPOCENTRIC COORDINATES DEPENDING ON IBTBS VALUE.

IOD   - OUTPUT UNIT IN STEP5, FORMATTED MAGNETIC FIELD IN DESIRED SPACECRAFT COORDINATES.

IOB   - OUTPUT UNIT IN STEP5, BINARY MAGNETIC FIELD IN PROGRAM FIT FORMAT.

ISC1  - FILTER SCRATCH UNIT.

ISC2  - FILTER SCRATCH UNIT.

ISC3  - SCRATCH UNIT USED IN SUBPROGRAM DPINFO TO STORE VARIOUS DATA PARAMETERS.
NAMELIST FIELDP -

JJ - FID INPUT POSITION COORDINATES: (0) GEODETIC
     (1) GEOCENTRIC.

MM - FID EQUATORIAL RADIUS AND RECIPROCAL FLATTENING:
     (0) DEFAULT AE = 6378.16 KM, FLAT = 298.25 (1) INPUT
     VALUES.

NMX - MAXIMUM DEGREE OF FID MODEL EVALUATION.

NEXT - EXTERNAL FIELD MODEL: (0) DO NOT EVALUATE (1) EVALUATE.

IOCF - INPUT UNIT IN FID FOR COMPUTED MAGNETIC FIELD MODEL.

IDST - INDUCED FIELD COEFFICIENTS: (0) DO NOT EVALUATE
     (1) EVALUATE.

DST - DST VALUE.

LL - FID FIELD EVALUATION MODE: (-1) EVALUATE AT OLD TIME
     (0) EVALUATE (1) READ FIELD MODEL AND EVALUATE.

NAMELIST BSPLIN -

H - ARRAY CONTAINING NUMBER OF INTERNAL KNOTS FOR B-SPLINE
    FUNCTIONS FITTING X, Y, AND Z COMPONENTS, RESPECTIVELY.

NN - ARRAY CONTAINING ORDER OF B-SPLINE FUNCTIONS FITTING X,
    Y, AND Z COMPONENTS, RESPECTIVELY.

NT - ARRAY CONTAINING NUMBER OF FOURIER WAVEFORMS FITTING X,
    Y, AND Z COMPONENTS, RESPECTIVELY.

KA - B-SPLINE INTERNAL KNOT ADJUSTMENT FOR BEST FIT WITH
    RESPECT TO WEIGHTED RMS: (0) DO NOT ADJUST (1) ADJUST

ITERMX - MAXIMUM NUMBER OF ITERATIONS IN UNIVARIANT SEARCH FOR
        OPTIMUM B-SPLINE KNOT POSITIONS.

LGRMAX - MAXIMUM NUMBER OF ITERATIONS IN LAGRANGIAN INTERPOLATIVE
        SEARCH FOR BEST POSITION OF A PARTICULAR KNOT WITH
        RESPECT TO WEIGHTED RMS.

EPS - KNOT ADJUSTMENT TOLERANCE WITHIN WHICH THE KNOT POSITION
      IS CONSIDERED TO HAVE CONVERGED.

KO - BOOLEAN NUMBER IN WHICH EACH DIGIT GOVERNS THE ADJUSTMENT
     OF A PARTICULAR INTERNAL KNOT POSITION, WITH LEFT-MOST
     DIGIT CORRESPONDING TO LEFT-MOST KNOT: (0) ADJUST
     (1) DO NOT ADJUST.
IOBS - INPUT UNIT CONTAINING B-SPLINE KNOT POSITIONS, FOURIER WAVEFORM FREQUENCIES, AND SIGMAS FOR OBSERVED MAGNETIC FIELD VALUES.

NAMELIST TRFORM -
==================
EU - FIT EULER ANGLES (DEGREES).
QI - GSFC NOMINAL BIAS CORRECTIONS IN ORIGINAL SATELLITE COORDINATES (NT).
QF - FIT MAGNETOMETER BIAS ADJUSTMENTS (NT).
CF - FIT CALIBRATION SLOPE ADJUSTMENT MATRIX.
CA - CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATES.
RF - ROTATION MATRIX FROM ORIGINAL SATELLITE TO FIT/MAGSAT COORDINATES.
RC - ROTATION MATRIX FROM FIT/MAGSAT TO DESIRED SATELLITE COORDINATES.

NAMELIST CONTRL -
===================
IMODE - PROGRAM OPERATION MODE: (0) RAW-TO-FINAL FIT TAPE TOTAL PROCESSING (1) FILTER-TO-FINAL FIT TAPE PROCESSING (2) FILTER PROCESSING ONLY (3) RAW-TO-FILTER TAPE PROCESSING.
IFORM - ORIGINAL RAW DATA TAPE(S) FORMAT: (0) EARLY FORMAT -- 2 SAMPLES/SECOND (1) LATTER FORMAT -- 20 SAMPLES/SECOND
NDATAR - NUMBER OF DATA RECORDS PROCESSED AFTER EPHEMERIS RECORD.
INPUTF - NUMBER OF INPUT FILES TO BE PROCESSED.
IARC - ARC PROCESSING LENGTH: (0) ENTIRE ARC (1) ARC SEGMENT BETWEEN BEGINNING AND ENDING TIMES ONLY.
IYRBEG - BEGINNING ARC TIME YEAR SINCE 1900.
IDYBEG - BEGINNING ARC TIME DAY NUMBER.
ISCBEG - BEGINNING ARC TIME SECONDS.
IYREND - ENDING ARC TIME YEAR SINCE 1900.
IDYEND - ENDING ARC TIME DAY NUMBER.
ISCEND - ENDING ARC TIME SECONDS.
ORBINC - SATELLITE ORBIT INCLINATION ANGLE (DEGREES).
ERAD - MEAN EARTH RADIUS (KM).
IEPDay - FILTER REFERENCE DAY NUMBER.
INCREM - FILTER WINDOW LENGTH (SECONDS).
INTRVL - FILTER WINDOW NUMBER FROM BEGINNING OF REFERENCE DAY.
IMETH - FILTER METHOD: (0) DETREND (1) DETREND AND FLAG OUTLIERS (2) FLAG OUTLIERS (3) NO MODIFICATION.
ISPEC - FFT SPECTRAL ANALYSIS: (0) NO ANALYSIS (1) ZERO-MEAN ANALYSIS (2) DIRECT ANALYSIS.
NEXTIN - NUMBER OF SUCCESSIVE FILTER WINDOWS TO BE PROCESSED DURING THIS RUN BEGINNING WITH WINDOW NUMBER "INTRVL".
IBTBS - FINAL TAPE OUTPUT COORDINATES: (0) FORMATTED TOPOCENTRIC (1) FORMATTED/BINARY FIT/MAGSAT (2) SAME AS 1, PLUS FORMATTED DESIRED SATELLITE.
SIGMLT - OUTLIER MULTIPLICATION FACTOR FOR TREND RESIDUAL SIGMA.
NFLAGK - DATA QUALITY FLAG RETENTION CODE FOR FILTER: EACH DIGIT INDICATES FLAG TO BE RETAINED FOR TREND FITTING.
IOWIOF - UNIT IOW INTERVALS FOR FINAL PROCESSING: (0) INTRVL ONLY (1) INTRVL AND PRECEEDING (2) ALL.
IOF1ST - OUTPUT DATA FLAG FOR UNITS IOF AND IOB: (0) DATA WILL BE APPENDED (1) DATA WILL BE FIRST.
IOD1ST - OUTPUT DATA FLAG FOR UNIT IOD: (0) DATA WILL BE APPENDED (1) DATA WILL BE FIRST.
IOW1ST - OUTPUT DATA FLAG FOR UNIT IOW: (0) DATA WILL BE APPENDED (1) DATA WILL BE FIRST.
NAMELIST OUTLIM -
                        ===========
DXOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC X COMPONENT (NT).
DYOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Y COMPONENT (NT).
DZOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Z COMPONENT (NT).
DBOL  - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC B COMPONENT (NT).

XWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT X COMPONENT.

YWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT Y COMPONENT.

ZWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT Z COMPONENT.

BWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT B COMPONENT.

ABVLAT - FILTER GEOCENTRIC LATITUDE TOLERANCE FOR ALL COMPONENTS.

TRNLAT - GEODETIC LATITUDE ABOVE WHICH SATELLITE VELOCITY DIRECTION IS INDETERMINABLE.

ITMGAP - TIME-GAP TOLERANCE INCREMENT FOR DATA (SECONDS).
...... RUN DECK FOR STAGE 1 ......

//XRJRRST1 JOB (F8002,X22,80),STEP1,TIME=(5,00),CLASS=A,MSGCLASS=X
/*JOBPARM LINES=100
//GO EXEC PGM=LOAD1,REGION=3000K
//STEPLIB DD DISP=SHR,DSN=XRJRR.SATFILT
/*
/* UNIT FOR INPUT PARAMETERS FOR PROGRAM FILTER FOLLOWS
/*
//GO.FT05F001 DD *
DMSP MAR 19-21, 1984. STEP1 WITHOUT COORDINATE SWITCH! - RF=I MATRIX!
&CONTRL IMODE=0, IARC=0, ORBINC=98.74, IEPDAY=79,
INCREM=21600, INTRVL=1, IMETH=2, ISPEC=1, NEXTIN=12,
IFIST=1, INPUTF=1, &END
&IOFILE IOR=13, IOW=14, &END
&BSPLIN H(1)=17, H(2)=17, H(3)=17, NT(1)=0, NT(2)=0, NT(3)=0,
NN(1)=4, NN(2)=4, NN(3)=4, &END
&OUTLIM, &END
&FIELDP, &END
&TRFORM RF(1,1)=0.0, RF(1,2)=0.0, RF(1,3)=1.0,
RF(2,1)=1.0, RF(2,2)=0.0, RF(2,3)=0.0,
RF(3,1)=0.0, RF(3,2)=1.0, RF(3,3)=0.0,
CA(1,1)=12.1001, CA(1,2)=-0.0055, CA(1,3)= 0.0193,
CA(2,1)=-0.0247, CA(2,2)=12.1863, CA(2,3)=-0.0101,
CA(3,1)= 0.0069, CA(3,2)= 0.0232, CA(3,3)=12.1735,
QI(1)=89.0, QI(2)=8457.0, QI(3)=-1441.0,
EU(1)=0.00, EU(2)=0.00, EU(3)=0.00, &END
/*
/* PRINTER OUTPUT UNIT FOLLOWS
/*
//GO.FT06F001 DD SYSOUT=*,SPACE=(CYL,(20,9),RLSE)
/*
/* PLOT TAPE UNIT FOLLOWS (RARELY USED, SO DUMMY OUT)
/*
//GO.FT08F001 DD UNIT=(1600, ,DEFER),LABEL=(1, ,OUT),
//DCB=(RECFM=WBS, LRECL=364, BLKSIZE=368, DEN=3), VOL=SER=JRR001
//GO.FT08F001 DD DUMMY
/*
/* NEW-FORMAT SATELLITE MAGNETIC TAPE UNIT FOLLOWS
/*
//GO.FT10F001 DD DISP=(OLD,KEEP),UNIT=6250,LABEL=(1, ,IN),
//DCB=(RECFM=FB, LRECL=75, BLKSIZE=1875, DEN=4, OPTCD=Q), VOL=SER=DT0031
/*
/* PERMANENT RE-USABLE DATA SETS FOLLOW (LEAVE AS IS ON STEP1)
/*
//GO.FT11F001 DD DSN=XRTJS.DMSP.STEP2.DATA,DISP=SHR
//DCB=(RECFM=FB, LRECL=240, BLKSIZE=4800),UNIT=SYSDA,
//SPACE=(TRK,(20,10), RLSE), VOL=SER=SACC09
//GO.FT12F001 DD DSN=XRTJS.DMSP.STEP3.DATA,DISP=SHR
//DCB=(RECFM=FB, LRECL=240, BLKSIZE=4800),UNIT=SYSDA,
//SPACE=(TRK,(20,10), RLSE), VOL=SER=SACC05
//GO.FT13F001 DD DSN=XRTJS.DMSP.STEP1S.JAN17.DATA,DISP=SHR
//GO.FT13F001 DD DSN=XRTJS.DMSP.STEP4.DATA,DISP=SHR

19
//GO.FT14F001 DD DSN=XRTJS.DMSP.STEP5.DATA,DISP=SHR

//*  
//* RUN-SPECIFIC OUTPUT DATA SETS FOLLOW  
//*  
//GO.FT15F001 DD DUMMY,DSN=XRSHS.SEP1684.STEP1.OUTF,DISP=(NEW,CATLG),  
// DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,  
// SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC02  
//GO.FT17F001 DD DSN=XRJRR.EUTEST.MAR1984.STEP1,DISP=SHR  
//*DCB=(RECFM=VBS,LRECL=11204,BLKSIZE=22412),UNIT=SYSDA,  
//*SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC06  
//*  
//* SCRATCH DATA SETS FOLLOW  
//* (BINARY, WILL NEVER LOOK AT)  
//GO.FT18F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//GO.FT19F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//GO.FT20F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//*  
//* INPUT MAGNETIC FIELD DATA SET Follows  
//* (KEEP AS IS FOR STEP1)  
//GO.FT21F001 DD DSN=XRJRR.DATA.MISC(CAL84FID),DISP=SHR  
//*  
//* INPUT TREND-FIT DATA SET Follows  
//*  
//GO.FT22F001 DD DSN=XRTJS.BSPINFO.DATA,DISP=SHR  
//*  
//* SYSTEM DUMP FOR ABEND-AID Follows  
//*  
//GO.SYSUDUMP DD DUMMY  
// EXEC NOTIFYTS

20
...... RUN DECK FOR STAGE 4 ......
//XRJRRST4 JOB (F8002.X22,80),STEP4,TIME=(10,00),CLASS=F,MSGCLASS=X
/*JOBPARMLINES=100
//GO EXEC PGM=LOAD1,REGION=3000K
//STEPLIB DD DISP=SHR,DSN=XRJRR.SATFILT
/*
/* UNIT FOR INPUT PARAMETERS FOR PROGRAM FILTER FOLLOWS
/*
//GO.FT05F001 DD *
&CONTRL IMODE=0, IFORM=1, IARC=0, ORBINC=98.74, IEPDAY=260,
INCREM=21600, INTRVL=1, IMETH=2, ISPEC=1, NEXTIN=12,
IOFIST=1, INPUTF=1, &END
&IOFILE IOR=13, IOW=14, &END
&BSPLIN H(1)=17, H(2)=17, H(3)=17, NT(1)=0, NT(2)=0, NT(3)=0,
NN(1)=4, NN(2)=4, NN(3)=4, &END
&OUTLIM, &END
&FIELDP, &END
&TRFORM RF(1,1)=0.0, RF(1,2)=0.0, RF(1,3)=1.0,
RF(2,1)=1.0, RF(2,2)=0.0, RF(2,3)=0.0,
RF(3,1)=0.0, RF(3,2)=1.0, RF(3,3)=0.0,
CA(1,1)=12.1001, CA(1,2)=-0.0055, CA(1,3)= 0.0193,
CA(2,1)=-0.0247, CA(2,2)=12.1863, CA(2,3)=-0.0101,
CA(3,1)= 0.0069, CA(3,2)= 0.0232, CA(3,3)=12.1735,
QI(1)=89.0, QI(2)=8457.0, QI(3)=-1441.0,
EU(1)=-.47839 EU(2)=-.09246, EU(3)=-0.00609,
QF(1)=-18.4, QF(2)=-9.46, QF(3)=-2.35, &END
//*
//* PRINTER OUTPUT UNIT FOLLOWS
//*
//GO.FT06F001 DD SYSOUT=*,SPACE=(CYL,(20,9),RLSE)
//*
//* PLOT TAPE UNIT FOLLOWS (RARELY USED, SO DUMMY OUT)
//*
//GO.FT08F001 DD UNIT=(1600,,DEFER),LABEL=(1,NL,,OUT),
//DCB=(RECFM=VBS,LRECL=364,BLKSIZE=368,UNIT=3),VOL=SER=JRR001
//GO.FT08F001 DD DUMMY
//*
//* NEW-FORMAT SATELLITE MAGNETIC TAPE UNIT FOLLOWS
//*
//GO.FT10F001 DD DISP=(OLD,KEEP),UNIT=6250,LABEL=(1,NL,,IN),
//DCB=(RECFM=FB,LRECL=75,BLKSIZE=1875,UNIT=4,OPTCD=Q),VOL=SER=DT0108
//*
//* PERMANENT RE-USABLE DATA SETS FOLLOW (LEAVE AS IS ON STEP1)
//*
//GO.FT11F001 DD DSN=XRTJS.DMSP.STEP2.DATA,DISP=SHR
//DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,
//SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC09
//GO.FT12F001 DD DSN=XRTJS.DMSP.STEP3.DATA,DISP=SHR
//DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,
//SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC05
//O.FT13F001 DD DSN=XRTJS.DMSP.STEP4.DATA,DISP=SHR
//GO.FT13F001 DD DSN=XRTJS.DMSP.STEP4.DATA,DISP=SHR
**RUN-SPECIFIC OUTPUT DATA SETS FOLLOW**

```plaintext
//GO.FT15F001 DD DSN=XRTJS.DMSP.STEPS.DATA,DISP=SHR
// DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,
// SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC04
//GO.FT17F001 DD DUMMY
```

**SCRATCH DATA SETS FOLLOW**

```plaintext
//GO.FT18F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)
//GO.FT19F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)
//GO.FT20F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)
```

**INPUT MAGNETIC FIELD DATA SET FOLLOWS**

```plaintext
//GO.FT21F001 DD DSN=XRTJS.SEP1684.STEP3.COEFFS,DISP=SHR
```

**INPUT TREND-FIT DATA SET FOLLOWS**

```plaintext
//GO.FT22F001 DD DSN=XRTJS.BSPINFO.DATA,DISP=SHR
```

**SYSTEM DUMP FOR ABEND-AID FOLLOWS**

```plaintext
//GO.SYSUDUMP DD DUMMY
// EXEC NOTIFYTS
```
...... RUN DECK FOR STAGE 5 ......

//XRJRRST5 JOB (F8002,X22,50),STEP5,TIME=(10,00),CLASS=F,MSGCLASS=X
/*JOBPARM LINES=150
//GO EXEC PGM=LOAD1,REGION=3000K
//STEPLIB DD DISP=SHR,DSN=XRJRR.SATFILT
/*
/* UNIT FOR INPUT PARAMETERS FOR PROGRAM FILTER FOLLOWS
/*
//GO.FT05F001 DD *
MAY 6-8,1984. STEP5. FOURIER REMOVAL STEP, USING NEW FIELD MODEL.
&CONTRL IMODE=1,IFORM=I,IMARC=0, ORBINC=98.74, IEPDAY=127,INCREM=21600,
INTRL=1, IMETH=1, ISPEC=1, NEXTIN=12,SIGMLT=2.8,
IOFIST=1, INPUTF=1, &END
&IOFILE IOR=13, IOW=14, &END
&BSPLIN H(1)=17, H(2)=17, H(3)=17, NT(1)=4, NT(2)=4, NT(3)=4,
NN(1)=0, NN(2)=0, NN(3)=0, &END
&OUTLIM , &END
&FIELDP , &END
&TRFORM RF(1,1)=0.0, RF(1,2)=0.0, RF(1,3)=1.0,
RF(2,1)=1.0, RF(2,2)=0.0, RF(2,3)=0.0,
RF(3,1)=0.0, RF(3,2)=1.0, RF(3,3)=0.0,
CA(1,1)=12.1001, CA(1,2)=0.0055, CA(1,3)=0.0193,
CA(2,1)=0.0247, CA(2,2)=12.1863, CA(2,3)=0.0101,
CA(3,1)=0.0069, CA(3,2)=0.0232, CA(3,3)=12.1735,
QI(1)=89.0, QI(2)=8457.0, QI(3)=-1441.0,
EU(1)=0.000, EU(2)=0.000, EU(3)=0.000, &END
/*
/* PRINTER OUTPUT UNIT FOLLOWS
/*
//GO.FT06F001 DD SYSOUT=*,SPACE=(CYL,(20,9),RLSE)
/*
/* PLOT TAPE UNIT FOLLOWS
/*
//GO.FT08F001 DD UNIT=(16000,DEFER),LABEL=(1,NL,OUT),
//*DCB=(RECFM=VBS,LRECL=364,BLKSIZE=368,DEN=3),VOL=SER=JRR001
//GO.FT08F001 DD DUMMY
/*
/* DO NOT USE UNIT#10 FOR THIS STEP.
//GO.FT10F001 DD DUMMY
/*
/* PERMANENT RE-USABLE DATA SETS FOLLOW
/*
//GO.FT11F001 DD DSN=XRTJS.DMSP.STEP2.DATA,DISP=SHR
//*DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,
//*SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC09
//GO.FT12F001 DD DSN=XRTJS.DMSP.STEP3.DATA,DISP=SHR
//*DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800),UNIT=SYSDA,
//*SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC05
/*
///*** INPUT ON UNIT #13, WHICH IS THE OUTPUT FROM STEP4 (FORMATTED).
//GO.FT13F001 DD DSN=XRSHS.MAY684.STEP4.OUTF,DISP=SHR
//*
//GO.FT14F001 DD DSN=XRTJS.DMSP.STEP5.DATA,DISP=SHR
//DCB=(RECFM=FB,LRECL=240,BLKSIZ=4800),UNIT=SYSDA,
//SPACE=(TRK,(20,10),RLSE),VOL=SER=SACC08
//*
//*
//** RUN-SPECIFIC OUTPUT DATA SETS FOLLOW
//** (OUTPUT BOTH FORMATTED AND BINARY DATA SETS).
//GO.FT15F001 DD DSN=XRSHS.MAY684.STEP5.OUTF,DISP=(NEW,CATLG),
//VOL=SER=SACC04,SPACE=(TRK,(20,10),RLSE),UNIT=SYSDA,
//DCB=(RECFM=FB,LRECL=240,BLKSIZ=4800)
//GO.FT17F001 DD DSN=XRSHS.MAY684.STEP5.OUTBIN,DISP=(NEW,CATLG),
//VOL=SER=SACC04,SPACE=(TRK,(20,10),RLSE),UNIT=SYSDA,
//DCB=(RECFM=VBS,LRECL=11204,BLKSIZ=22412)
//*
//** SCRATCH DATA SETS FOLLOW
//** (BINARY, WILL NEVER LOOK AT)
//GO.FT18F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
//DCB=(RECFM=FB,LRECL=80,BLKSIZ=7200)
//GO.FT19F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
//DCB=(RECFM=FB,LRECL=80,BLKSIZ=7200)
//GO.FT20F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(2,1),RLSE),
//DCB=(RECFM=FB,LRECL=80,BLKSIZ=7200)
//*
//** INPUT MAGNETIC FIELD DATA SET FOLLOWS
//** (THIS IS THE FIELD OUTPUT FROM STEP3 )
//GO.FT21F001 DD DSN=XRSHS.MAY684.STEP3.COEFFS,DISP=SHR
//*
//** INPUT TREND-FIT DATA SET FOLLOWS
//*
//GO.FT22F001 DD DSN=XRTJS.BSPINFO.DATA,DISP=SHR
//*
//** SYSTEM DUMP FOR ABEND-AID FOLLOWS
//*
//GO.SYSUDUMP DD DUMMY
// EXEC NOTIFYTS
B. For STAGE 2 there are two deck setups, one using the load module and one using the source code.

Note that for Unit 10 the tape is a dummy. The additional "information" is irrelevant to this run, but might be useful in other applications. These setups include an input set of SHA coefficients as a starting model for FIT and a list of observatories and their biases as determined in an earlier FIT. The basic program is the old FIT program.
..... STAGE 2 Run Deck with load module .....  

//XRJRRRTS2 JOB (G0111,X22,20),EUTST,TIME=(7,00),NOTIFY=XRJRR,CLASS=0,  
// MSGCLASS=X  
//*JOBPARM LINES=15  
//*  
//**XRJRR.DTAPE.PROCESS(STEP2) -- USE OF A LOAD MODULE (FIT.DMSP.LOAD2)  
//** INPUT TO THIS STEP IS FILE "OUTBIN" ON UNIT#17, FROM STEP1.  
//** INPUT IS ON UNIT #19.  
//** THE JOB SETUP PARAMETERS ARE ON UNIT #5.  
//GO EXEC PGM=FIT,REGION=3000K  
//STEPLIB DD DISP=SHR, DSN=XRJRR.FIT.DMSP.LOAD2  
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,7,2),RLSE),  
// DCB=(RECFM=VBST,LRECL=200,BLKSIZE=12004)  
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,7,2),RLSE),  
// DCB=(RECFM=VBST,LRECL=200,BLKSIZE=12004)  
//GO.FT06F001 DD SYSOUT=*  
//GO.FT07F001 DD DUMMY,SYSPRT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280),  
// SPACE=(CYL,0,1),RLSE)  
//GO.FT10F001 DD DUMMY,DSN=POG6CQ,UNIT=(9TRACK,DEFER),DISP=(OLD,KEEP),  
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,IN),  
// VOL=SER=MAG001  
//GO.FT10F002 DD DUMMY,DSN=POG6MQ,UNIT=(9TRACK,DEFER),DISP=(OLD,KEEP),  
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(2,SL,IN),  
// VOL=SER=MAG001  
//GO.FT10F003 DD DUMMY,DSN=POG246,UNIT=(9TRACK,DEFER),DISP=(OLD,KEEP),  
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(3,SL,IN),  
// VOL=SER=MAG001  
//*  
//* UNIT 11 IS A NORMAL MATRIX FILE. THIS IS NEEDED IN STEP 2  
//* ONLY IF STATISTICS ON THE INPUT DATA ARE DESIRED.  
//FT11F001 DD DSN='XRJRR.FIT.OUT.NMATX',DISP=SHR  
//*  
//* UNIT 12 IS A SCRATCH FILE.  
//FT12F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,90,20),RLSE),  
// DCB=(RECFM=VBS,LRECL=100,BLKSIZE=7204)  
//GO.FT13F001 DD DUMMY  
//FT15F001 DD DUMMY  
//FT16F001 DD DUMMY  
//FT17F001 DD DUMMY  
//FT18F001 DD DUMMY  
//*  
//* BINARY INPUT DATA follows. MUST BE IN FIT BINARY FORMAT.  
//*T19F001 DD DSN=XRJRR.EUTEST.MAR1984.STEP1,DISP=SHR  
//FT19F001 DD DSN=XRJRR.MAR1984.STEP1.OUTBIN,DISP=SHR  
//*  
//*FT22F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,40,10),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//FT23F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,40,10),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//FT24F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,40,10),RLSE),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)  
//*
// OUTPUT COEFFICIENTS FOLLOW.
// FT25F001 DD DSN=XRJRR.FIT.DMSP.COEFFS,UNIT=SYSDA,DISP=SHR
// *VOL=SER=SACCO9,DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
// *SPACE=(TRK,(2,2),RLSE),DISP=(NEW,CATLG)

// FT35F001 DD DUMMY
// FT36F001 DD DUMMY
// FT40F001 DD DUMMY
// FT45F001 DD DUMMY
// SYSUDUMP DD DUMMY
// FT05F001 DD *
&CONTRL NSIML=0, IRSTRT=0,NOISE=0,RTIM=9999.,
EULER=2,IBIAS=1,
ITER=2,
NSKIP=1,
END
MARCH 19-21, 1984, TEST OF DMSP EULER ANGLE SOLUTION.
&FIELD MONO=2,
NMAXR=0, NMAXTR=0,
BGNTIM=0.,
EXTFLD=0,NEXT=0
PRCORL=4,
IDST=0,
EPOCH=1984.22,AVETIM=1984.22,
NMAX=11,NMAXT=9,NMAXTT=0,NMXTTT=0,NMAXIV=0,NMAXV=0,
END
&LIMERR ERRLIM=2*20.,8*1000.,3*1000.,2*.0012,2*30.,
NTDATA=1,NAT(1)=8,NAT(2)=2,APRIOR=0,
NSTOP=0,
AYSTAT=1, BIASAP=5000.,
NDGEN=5, &END

2 1 29878.2 0.000000E+00 26.9879 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
2 2 1924.05 5526.55 7.95582 -19.3154 0.000000E+00 0.000000E+00 0.000000E+00
3 1 2063.34 0.000000E+00 -16.6929 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
3 2 3044.32 -2183.87 4.24784 -13.6396 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
3 3 1682.87 -291.646 5.04403 -22.9796 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
4 1 1279.34 0.000000E+00 -558737 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
4 2 2200.81 -317.451 -5.07226 4.55282 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
4 3 1250.13 -282.905 -2.80991 3.00112 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
4 4 831.335 -289.166 6.04653 -9.23767 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
5 1 943.053 0.000000E+00 -1.34613 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
5 2 776.331 230.858 1.48226 4.66531 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
5 3 370.782 -248.342 -6.77951 2.08782 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
5 4 424.398 64.1152 -1.36526 2.80991 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
5 5 831.335 -294.299 -6.07800 0.717217 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 1 1211.934 0.000000E+00 1.48473 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 2 358.879 45.6865 0.409051 -1.126245 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 3 252.241 145.820 -2.20934 -0.996403 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 4 90.4987 -152.348 -4.06068 -4.44105 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 5 162.388 -77.5140 -6.77951 0.529454E-01 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
6 6 65.1517 97.0991 -.127636 1.24753 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
7 1 50.2750 0.000000E+00 0.582770 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
7 2 65.8066 -14.4218 0.735771E-01 0.834850E-01 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
-&END

0

0

}&EUORBS Y1=0.D0,Z1=0.D0,Z2=0.D0,
EUD(1,1)=0.D0,EUD(2,1)=0.D0,EUD(3,1)=0.D0,
EUD(1,2)=0.D0,EUD(2,2)=0.D0,EUD(3,2)=0.D0,
BS1=0.D0,BS2=0.D0,BS3=0.D0,
BST1=0.D0,BST2=0.D0,BST3=0.D0,
}&END
<table>
<thead>
<tr>
<th>Location</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABINGER</td>
<td>0.80</td>
<td>0.44</td>
<td>0.16</td>
</tr>
<tr>
<td>ABISKO</td>
<td>1.22</td>
<td>1.37</td>
<td>2.06</td>
</tr>
<tr>
<td>ADAK</td>
<td>5.92</td>
<td>2.05</td>
<td>8.17</td>
</tr>
<tr>
<td>ADDIS ABABA</td>
<td>3.26</td>
<td>5.11</td>
<td>3.15</td>
</tr>
<tr>
<td>AGINCOURT</td>
<td>2.52</td>
<td>2.08</td>
<td>5.08</td>
</tr>
<tr>
<td>ALERT</td>
<td>5.24</td>
<td>8.18</td>
<td>19.86</td>
</tr>
<tr>
<td>ALIBAG</td>
<td>7.16</td>
<td>2.47</td>
<td>6.23</td>
</tr>
<tr>
<td>ALIBAG II</td>
<td>1.17</td>
<td>0.96</td>
<td>1.18</td>
</tr>
<tr>
<td>ALMA ATA</td>
<td>3.16</td>
<td>3.05</td>
<td>2.76</td>
</tr>
<tr>
<td>ALMERIA</td>
<td>3.37</td>
<td>1.48</td>
<td>2.91</td>
</tr>
<tr>
<td>AMBERLEY II</td>
<td>2.60</td>
<td>0.96</td>
<td>2.90</td>
</tr>
<tr>
<td>ANCHORAGE</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>ANNAMALAINAGAR</td>
<td>7.13</td>
<td>7.74</td>
<td>10.89</td>
</tr>
<tr>
<td>APIA III</td>
<td>3.53</td>
<td>1.66</td>
<td>2.27</td>
</tr>
<tr>
<td>AQUILA</td>
<td>2.42</td>
<td>1.23</td>
<td>1.92</td>
</tr>
<tr>
<td>ARGENTINE ISLND</td>
<td>3.98</td>
<td>1.39</td>
<td>2.61</td>
</tr>
<tr>
<td>ARTI</td>
<td>2.63</td>
<td>1.12</td>
<td>3.36</td>
</tr>
<tr>
<td>ASO</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>AVERROES</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BAKER LAKE</td>
<td>3.13</td>
<td>3.28</td>
<td>4.44</td>
</tr>
<tr>
<td>BANGUI</td>
<td>10.96</td>
<td>4.67</td>
<td>4.06</td>
</tr>
<tr>
<td>BANGUI II</td>
<td>0.47</td>
<td>4.88</td>
<td>1.82</td>
</tr>
<tr>
<td>BANGUI III</td>
<td>6.46</td>
<td>5.64</td>
<td>3.32</td>
</tr>
<tr>
<td>BARROW II</td>
<td>5.03</td>
<td>3.04</td>
<td>27.69</td>
</tr>
<tr>
<td>BARROW III</td>
<td>3.06</td>
<td>2.81</td>
<td>6.57</td>
</tr>
<tr>
<td>BARTER ISLAND</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BEIJING</td>
<td>4.25</td>
<td>1.86</td>
<td>3.35</td>
</tr>
<tr>
<td>BELOIT</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BELSK</td>
<td>2.14</td>
<td>1.73</td>
<td>1.93</td>
</tr>
<tr>
<td>BEREZNAIYKI</td>
<td>2.34</td>
<td>5.63</td>
<td>4.87</td>
</tr>
<tr>
<td>BIG DELTA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BINZA</td>
<td>6.19</td>
<td>4.99</td>
<td>2.95</td>
</tr>
<tr>
<td>BJORNOYA</td>
<td>2.01</td>
<td>3.52</td>
<td>4.16</td>
</tr>
<tr>
<td>BOULDER</td>
<td>2.53</td>
<td>1.45</td>
<td>1.93</td>
</tr>
<tr>
<td>BUDAKESZI</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BUDKOV</td>
<td>1.92</td>
<td>0.88</td>
<td>1.67</td>
</tr>
<tr>
<td>BURLINGTON</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>BYRD II</td>
<td>4.00</td>
<td>1.95</td>
<td>7.55</td>
</tr>
<tr>
<td>CAMBRIDGE BAY</td>
<td>3.18</td>
<td>2.05</td>
<td>2.89</td>
</tr>
<tr>
<td>CANBERRA</td>
<td>0.62</td>
<td>2.85</td>
<td>1.64</td>
</tr>
<tr>
<td>CARROLLTON</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>CASEY</td>
<td>48.80</td>
<td>0.82</td>
<td>53.70</td>
</tr>
<tr>
<td>CASPER</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>CASTELLACCIO</td>
<td>1.93</td>
<td>0.64</td>
<td>6.00</td>
</tr>
<tr>
<td>CASTLE ROCK</td>
<td>5.87</td>
<td>4.57</td>
<td>4.37</td>
</tr>
<tr>
<td>CHA PA</td>
<td>5.96</td>
<td>3.54</td>
<td>7.86</td>
</tr>
<tr>
<td>Location</td>
<td>2.96</td>
<td>1.38</td>
<td>4.96</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>CHAMBON FORET</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>CHANGCHUN</td>
<td>2.23</td>
<td>2.03</td>
<td>7.54</td>
</tr>
<tr>
<td>CHELTENHAM</td>
<td>2.91</td>
<td>1.35</td>
<td>2.26</td>
</tr>
<tr>
<td>CHELYUSKIN II</td>
<td>2.23</td>
<td>2.03</td>
<td>7.54</td>
</tr>
<tr>
<td>COIMBRA</td>
<td>2.89</td>
<td>1.59</td>
<td>3.87</td>
</tr>
<tr>
<td>COSTA RICA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>DALLAS</td>
<td>2.45</td>
<td>1.03</td>
<td>1.95</td>
</tr>
<tr>
<td>DAVIS</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>DIKSON II</td>
<td>4.30</td>
<td>2.79</td>
<td>9.86</td>
</tr>
<tr>
<td>DOMBAS II</td>
<td>2.57</td>
<td>1.53</td>
<td>1.84</td>
</tr>
<tr>
<td>DOURBES</td>
<td>1.81</td>
<td>1.11</td>
<td>2.27</td>
</tr>
<tr>
<td>DRUZHNAYA</td>
<td>5.40</td>
<td>5.56</td>
<td>15.94</td>
</tr>
<tr>
<td>DUMONT DURVILLE</td>
<td>4.85</td>
<td>4.75</td>
<td>14.95</td>
</tr>
<tr>
<td>DUSHETI</td>
<td>4.23</td>
<td>1.92</td>
<td>4.69</td>
</tr>
<tr>
<td>DYMER</td>
<td>1.76</td>
<td>1.08</td>
<td>2.05</td>
</tr>
<tr>
<td>EBRO</td>
<td>3.16</td>
<td>1.49</td>
<td>1.99</td>
</tr>
<tr>
<td>EIGHTS</td>
<td>4.19</td>
<td>0.42</td>
<td>4.63</td>
</tr>
<tr>
<td>ELISABETHVILLE</td>
<td>2.31</td>
<td>1.15</td>
<td>3.06</td>
</tr>
<tr>
<td>ESKDALEMUIR</td>
<td>1.96</td>
<td>1.23</td>
<td>2.60</td>
</tr>
<tr>
<td>ESPANOLA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>EYREWELL</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>FANNING ISLAND</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>FORT CHURCHILL</td>
<td>3.67</td>
<td>2.72</td>
<td>3.93</td>
</tr>
<tr>
<td>FORT YUKON</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>FREDERICKSBURG</td>
<td>3.21</td>
<td>1.08</td>
<td>2.90</td>
</tr>
<tr>
<td>FUQUENE</td>
<td>2.46</td>
<td>2.25</td>
<td>7.60</td>
</tr>
<tr>
<td>FURSTNFELDBRUCK</td>
<td>2.39</td>
<td>0.96</td>
<td>2.09</td>
</tr>
<tr>
<td>GIBILMANNA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>GNANGARA</td>
<td>2.05</td>
<td>1.01</td>
<td>2.07</td>
</tr>
<tr>
<td>GODHAVN</td>
<td>1.86</td>
<td>1.49</td>
<td>4.21</td>
</tr>
<tr>
<td>GONZALEZ VIDELA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>GORNOTAYEZHNAYA</td>
<td>3.21</td>
<td>1.25</td>
<td>5.64</td>
</tr>
<tr>
<td>GREAT WHALE R</td>
<td>6.43</td>
<td>2.01</td>
<td>4.20</td>
</tr>
<tr>
<td>GROCKA</td>
<td>2.27</td>
<td>0.83</td>
<td>2.40</td>
</tr>
<tr>
<td>GUAM</td>
<td>3.65</td>
<td>1.24</td>
<td>2.16</td>
</tr>
<tr>
<td>GUANGZHOU</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>HALLETT STATION</td>
<td>11.86</td>
<td>20.93</td>
<td>109.30</td>
</tr>
<tr>
<td>HALEY BAY</td>
<td>9.35</td>
<td>27.52</td>
<td>31.37</td>
</tr>
<tr>
<td>HATIZYO</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>HAVANA</td>
<td>2.23</td>
<td>1.57</td>
<td>4.43</td>
</tr>
<tr>
<td>HEALY</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>HEL</td>
<td>2.00</td>
<td>1.61</td>
<td>2.28</td>
</tr>
<tr>
<td>HELWAN</td>
<td>0.45</td>
<td>2.41</td>
<td>2.19</td>
</tr>
<tr>
<td>HERMANUS</td>
<td>2.90</td>
<td>1.24</td>
<td>1.32</td>
</tr>
<tr>
<td>HOLLANDIA</td>
<td>5.51</td>
<td>2.49</td>
<td>7.33</td>
</tr>
<tr>
<td>HONGKONG</td>
<td>25.75</td>
<td>9.58</td>
<td>8.04</td>
</tr>
<tr>
<td>HONOLULU IV</td>
<td>3.40</td>
<td>0.82</td>
<td>1.46</td>
</tr>
<tr>
<td>HUANCAYO</td>
<td>3.61</td>
<td>1.54</td>
<td>2.20</td>
</tr>
<tr>
<td>HURBANOV0</td>
<td>2.89</td>
<td>1.06</td>
<td>2.98</td>
</tr>
<tr>
<td>HYDERABAD</td>
<td>3.98</td>
<td>14.84</td>
<td>2.58</td>
</tr>
<tr>
<td>Place</td>
<td>3.68</td>
<td>6.56</td>
<td>1.92</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>ISLA DE PASCUA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>ISTANBL KNDILLI</td>
<td>2.07</td>
<td>1.47</td>
<td>4.34</td>
</tr>
<tr>
<td>JAIPUR</td>
<td>9.67</td>
<td>0.61</td>
<td>23.81</td>
</tr>
<tr>
<td>JARVIS ISLAND</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>JASSY</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>JULIANEHAAB II</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>KAKIOKA</td>
<td>3.46</td>
<td>1.47</td>
<td>3.42</td>
</tr>
<tr>
<td>KANOSTA</td>
<td>3.44</td>
<td>0.78</td>
<td>1.20</td>
</tr>
<tr>
<td>KANOSAN</td>
<td>3.94</td>
<td>1.34</td>
<td>2.36</td>
</tr>
<tr>
<td>KELES</td>
<td>3.07</td>
<td>0.71</td>
<td>4.35</td>
</tr>
<tr>
<td>KERGUELEN</td>
<td>3.36</td>
<td>2.81</td>
<td>2.65</td>
</tr>
<tr>
<td>KIEV</td>
<td>6.72</td>
<td>3.56</td>
<td>2.44</td>
</tr>
<tr>
<td>KLYUCHI</td>
<td>3.22</td>
<td>0.83</td>
<td>2.74</td>
</tr>
<tr>
<td>KODAIKANAL</td>
<td>5.09</td>
<td>5.08</td>
<td>4.30</td>
</tr>
<tr>
<td>KOROR</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>KOTZEBUE</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>KRASNAYA PAKHRA</td>
<td>1.84</td>
<td>1.33</td>
<td>2.84</td>
</tr>
<tr>
<td>KSARA</td>
<td>6.68</td>
<td>1.83</td>
<td>19.61</td>
</tr>
<tr>
<td>L AMERICA III</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>L AMERICA V</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>LA PAZ</td>
<td>3.80</td>
<td>1.77</td>
<td>12.02</td>
</tr>
<tr>
<td>LA QUIACA II</td>
<td>4.87</td>
<td>1.69</td>
<td>2.39</td>
</tr>
<tr>
<td>LANZHOU</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>LAUDER</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>LAZAREVA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>LEADVILLE</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>LEIRVOGUR</td>
<td>2.82</td>
<td>1.43</td>
<td>3.73</td>
</tr>
<tr>
<td>LERWICK</td>
<td>1.79</td>
<td>1.38</td>
<td>2.05</td>
</tr>
<tr>
<td>LHASA</td>
<td>5.53</td>
<td>2.18</td>
<td>4.19</td>
</tr>
<tr>
<td>LOGRONO</td>
<td>2.64</td>
<td>2.48</td>
<td>2.01</td>
</tr>
<tr>
<td>LOPARSKOYE</td>
<td>2.37</td>
<td>2.19</td>
<td>5.10</td>
</tr>
<tr>
<td>LOVO</td>
<td>1.93</td>
<td>1.03</td>
<td>2.10</td>
</tr>
<tr>
<td>LUANDA BELAS</td>
<td>6.44</td>
<td>2.74</td>
<td>10.29</td>
</tr>
<tr>
<td>LUNFING</td>
<td>3.37</td>
<td>1.85</td>
<td>2.44</td>
</tr>
<tr>
<td>LVVOV</td>
<td>3.07</td>
<td>1.02</td>
<td>4.37</td>
</tr>
<tr>
<td>LWIRO</td>
<td>6.12</td>
<td>1.79</td>
<td>2.14</td>
</tr>
<tr>
<td>M BOUR</td>
<td>3.17</td>
<td>4.87</td>
<td>1.91</td>
</tr>
<tr>
<td>MACQUARIE ISLND</td>
<td>3.03</td>
<td>5.87</td>
<td>4.65</td>
</tr>
<tr>
<td>MAGADAN</td>
<td>1.63</td>
<td>1.08</td>
<td>30.43</td>
</tr>
<tr>
<td>MAJURO</td>
<td>0.89</td>
<td>7.10</td>
<td>1.10</td>
</tr>
<tr>
<td>MANHAY</td>
<td>3.54</td>
<td>4.62</td>
<td>6.27</td>
</tr>
<tr>
<td>MAPUTO</td>
<td>4.15</td>
<td>3.33</td>
<td>4.11</td>
</tr>
<tr>
<td>MARION ISLAND</td>
<td>5.96</td>
<td>0.99</td>
<td>1.91</td>
</tr>
<tr>
<td>MARTIN VIVIES</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>MAURITIUS II</td>
<td>8.38</td>
<td>4.52</td>
<td>13.17</td>
</tr>
<tr>
<td>MAWSON</td>
<td>3.30</td>
<td>4.36</td>
<td>7.49</td>
</tr>
<tr>
<td>MEANOOK</td>
<td>3.74</td>
<td>2.30</td>
<td>6.57</td>
</tr>
<tr>
<td>MEMAMBETSU</td>
<td>3.46</td>
<td>0.76</td>
<td>2.81</td>
</tr>
<tr>
<td>MIDWAY</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>MIRNYY</td>
<td>5.10</td>
<td>4.38</td>
<td>11.53</td>
</tr>
<tr>
<td>MISALLAT</td>
<td>3.68</td>
<td>2.24</td>
<td>3.91</td>
</tr>
<tr>
<td>MOCA</td>
<td>3.22</td>
<td>2.65</td>
<td>1.68</td>
</tr>
<tr>
<td>Location</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>MODIIM</td>
<td>7.51</td>
<td>7.79</td>
<td>22.67</td>
</tr>
<tr>
<td>MOLODEZHNYAYA</td>
<td>6.27</td>
<td>6.29</td>
<td>7.48</td>
</tr>
<tr>
<td>MONTE CAPELLINO</td>
<td>2.35</td>
<td>2.61</td>
<td>3.66</td>
</tr>
<tr>
<td>MOULD BAY</td>
<td>4.33</td>
<td>3.51</td>
<td>3.78</td>
</tr>
<tr>
<td>MUNTINLUPA</td>
<td>3.65</td>
<td>2.38</td>
<td>3.53</td>
</tr>
<tr>
<td>NAGYCENK</td>
<td>3.77</td>
<td>2.88</td>
<td>6.36</td>
</tr>
<tr>
<td>NAIROBI</td>
<td>3.25</td>
<td>1.77</td>
<td>4.59</td>
</tr>
<tr>
<td>NANTES</td>
<td>2.16</td>
<td>1.17</td>
<td>2.08</td>
</tr>
<tr>
<td>NEWPORT</td>
<td>2.18</td>
<td>0.99</td>
<td>2.25</td>
</tr>
<tr>
<td>NIEMEGK</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>NITZANIM</td>
<td>6.13</td>
<td>21.64</td>
<td>5.21</td>
</tr>
<tr>
<td>NORTHWAY</td>
<td>3.49</td>
<td>4.57</td>
<td>16.75</td>
</tr>
<tr>
<td>NORWAY STATION</td>
<td>2.71</td>
<td>1.53</td>
<td>2.52</td>
</tr>
<tr>
<td>NURMIJARVI</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>OASIS</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>ORCADAS DEL SUR</td>
<td>5.17</td>
<td>1.34</td>
<td>3.72</td>
</tr>
<tr>
<td>PAMATAI</td>
<td>3.97</td>
<td>1.34</td>
<td>1.98</td>
</tr>
<tr>
<td>PAMATAI II</td>
<td>2.63</td>
<td>0.84</td>
<td>2.50</td>
</tr>
<tr>
<td>PANAGYURISHTYE</td>
<td>3.79</td>
<td>1.61</td>
<td>2.78</td>
</tr>
<tr>
<td>PARAMARIBO</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>PATRICK</td>
<td>2.78</td>
<td>1.43</td>
<td>3.11</td>
</tr>
<tr>
<td>PATRONY</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>PENDELI</td>
<td>4.78</td>
<td>1.59</td>
<td>3.44</td>
</tr>
<tr>
<td>PILAR</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>PIONERSKAYA</td>
<td>3.06</td>
<td>1.02</td>
<td>3.71</td>
</tr>
<tr>
<td>PLATAINE</td>
<td>5.78</td>
<td>7.06</td>
<td>6.51</td>
</tr>
<tr>
<td>PLATHE</td>
<td>2.73</td>
<td>1.86</td>
<td>3.19</td>
</tr>
<tr>
<td>PLATEAU</td>
<td>2.20</td>
<td>2.18</td>
<td>6.76</td>
</tr>
<tr>
<td>PLESHEHITZI</td>
<td>4.04</td>
<td>22.09</td>
<td>6.98</td>
</tr>
<tr>
<td>PORT MORESBY</td>
<td>0.67</td>
<td>1.86</td>
<td>1.70</td>
</tr>
<tr>
<td>PORT-ALFRED</td>
<td>2.21</td>
<td>1.60</td>
<td>4.58</td>
</tr>
<tr>
<td>PRICE</td>
<td>2.22</td>
<td>1.57</td>
<td>2.67</td>
</tr>
<tr>
<td>PRUHONICE</td>
<td>8.85</td>
<td>5.99</td>
<td>9.82</td>
</tr>
<tr>
<td>QUETTA</td>
<td>3.70</td>
<td>3.36</td>
<td>17.15</td>
</tr>
<tr>
<td>REGENSBERG</td>
<td>3.05</td>
<td>0.89</td>
<td>3.27</td>
</tr>
<tr>
<td>RESOLUTE BAY</td>
<td>5.24</td>
<td>14.55</td>
<td>3.29</td>
</tr>
<tr>
<td>ROBURENT</td>
<td>4.64</td>
<td>13.43</td>
<td>7.17</td>
</tr>
<tr>
<td>ROI BAUDOUIL</td>
<td>8.25</td>
<td>4.81</td>
<td>18.79</td>
</tr>
<tr>
<td>RUDE SKOV</td>
<td>3.22</td>
<td>2.34</td>
<td>2.45</td>
</tr>
<tr>
<td>SABHAWALA</td>
<td>4.68</td>
<td>2.38</td>
<td>0.71</td>
</tr>
<tr>
<td>SAN FERNANDO</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>SAN JOSE LAS LA</td>
<td>3.56</td>
<td>1.47</td>
<td>1.91</td>
</tr>
<tr>
<td>SITKA</td>
<td>2.24</td>
<td>1.31</td>
<td>2.24</td>
</tr>
<tr>
<td>Location</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>SODANKYLA</td>
<td>2.83</td>
<td>1.31</td>
<td>2.88</td>
</tr>
<tr>
<td>SOUTH GEORGIA</td>
<td>4.57</td>
<td>3.17</td>
<td>2.57</td>
</tr>
<tr>
<td>SOUTH POLE</td>
<td>5.00</td>
<td>5.06</td>
<td>11.67</td>
</tr>
<tr>
<td>SREDNIKAN IV</td>
<td>6.37</td>
<td>2.49</td>
<td>13.26</td>
</tr>
<tr>
<td>STEKOLINIY</td>
<td>2.74</td>
<td>1.76</td>
<td>2.36</td>
</tr>
<tr>
<td>STEPANOVKA</td>
<td>2.69</td>
<td>1.10</td>
<td>3.29</td>
</tr>
<tr>
<td>STONYHURST</td>
<td>3.19</td>
<td>9.57</td>
<td>2.83</td>
</tr>
<tr>
<td>SURURLI</td>
<td>8.88</td>
<td>2.44</td>
<td>18.22</td>
</tr>
<tr>
<td>SWIDER</td>
<td>2.39</td>
<td>0.97</td>
<td>7.13</td>
</tr>
<tr>
<td>SYOWA BASE</td>
<td>3.13</td>
<td>8.86</td>
<td>11.35</td>
</tr>
<tr>
<td>TAHITI</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>TAMANRASSET</td>
<td>6.62</td>
<td>9.97</td>
<td>14.19</td>
</tr>
<tr>
<td>TANANARIVE</td>
<td>4.79</td>
<td>2.97</td>
<td>7.43</td>
</tr>
<tr>
<td>TANGERANG</td>
<td>26.70</td>
<td>14.07</td>
<td>43.78</td>
</tr>
<tr>
<td>TATUOCA</td>
<td>5.15</td>
<td>2.33</td>
<td>5.57</td>
</tr>
<tr>
<td>TEHRAN</td>
<td>4.92</td>
<td>3.50</td>
<td>7.66</td>
</tr>
<tr>
<td>TEHRAN II</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>TENERIFE</td>
<td>9.17</td>
<td>10.70</td>
<td>14.67</td>
</tr>
<tr>
<td>TEOLOYUCAN</td>
<td>16.56</td>
<td>11.22</td>
<td>20.84</td>
</tr>
<tr>
<td>THULE II</td>
<td>1.43</td>
<td>0.87</td>
<td>4.68</td>
</tr>
<tr>
<td>TIHANY</td>
<td>4.45</td>
<td>2.07</td>
<td>4.57</td>
</tr>
<tr>
<td>TIKHAYA BAY</td>
<td>0.97</td>
<td>1.43</td>
<td>0.47</td>
</tr>
<tr>
<td>TIKSI</td>
<td>4.97</td>
<td>1.35</td>
<td>9.00</td>
</tr>
<tr>
<td>TOLEDO</td>
<td>3.17</td>
<td>1.19</td>
<td>5.87</td>
</tr>
<tr>
<td>TOMSK</td>
<td>4.22</td>
<td>1.81</td>
<td>6.39</td>
</tr>
<tr>
<td>TOOLANGI</td>
<td>1.58</td>
<td>1.30</td>
<td>1.67</td>
</tr>
<tr>
<td>TRELEW</td>
<td>3.37</td>
<td>2.76</td>
<td>2.25</td>
</tr>
<tr>
<td>TRIVANDRUM</td>
<td>5.51</td>
<td>7.37</td>
<td>5.38</td>
</tr>
<tr>
<td>TROMSO</td>
<td>3.97</td>
<td>1.76</td>
<td>5.17</td>
</tr>
<tr>
<td>TSUMEB</td>
<td>2.82</td>
<td>1.19</td>
<td>1.64</td>
</tr>
<tr>
<td>TUCSON</td>
<td>2.89</td>
<td>0.94</td>
<td>2.27</td>
</tr>
<tr>
<td>TULSA</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>UELEN</td>
<td>4.75</td>
<td>10.61</td>
<td>5.52</td>
</tr>
<tr>
<td>UJJAIN</td>
<td>12.76</td>
<td>3.68</td>
<td>7.07</td>
</tr>
<tr>
<td>ULAN BATOR</td>
<td>11.40</td>
<td>1.92</td>
<td>2.89</td>
</tr>
<tr>
<td>Urumqi</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>VALENTIA</td>
<td>2.06</td>
<td>0.92</td>
<td>1.62</td>
</tr>
<tr>
<td>VANNOVSKAYA</td>
<td>4.42</td>
<td>2.79</td>
<td>6.00</td>
</tr>
<tr>
<td>VASSOURAS</td>
<td>4.20</td>
<td>1.68</td>
<td>2.57</td>
</tr>
<tr>
<td>VICTORIA</td>
<td>3.02</td>
<td>1.57</td>
<td>2.87</td>
</tr>
<tr>
<td>VOROSHILOV</td>
<td>5.21</td>
<td>0.27</td>
<td>0.0</td>
</tr>
<tr>
<td>VOSTOK</td>
<td>11.89</td>
<td>8.99</td>
<td>14.45</td>
</tr>
<tr>
<td>VOYEYKOV</td>
<td>2.54</td>
<td>1.19</td>
<td>3.25</td>
</tr>
<tr>
<td>VYKHODNOY</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>VYSOKAY DUBRAVA</td>
<td>2.37</td>
<td>0.94</td>
<td>5.94</td>
</tr>
<tr>
<td>WATHEROO</td>
<td>3.63</td>
<td>1.56</td>
<td>7.82</td>
</tr>
<tr>
<td>WIEN KOBENZL</td>
<td>2.29</td>
<td>0.93</td>
<td>1.97</td>
</tr>
<tr>
<td>WILKES</td>
<td>5.37</td>
<td>3.33</td>
<td>11.61</td>
</tr>
<tr>
<td>WINGST</td>
<td>1.75</td>
<td>0.80</td>
<td>2.38</td>
</tr>
<tr>
<td>WITTEVEEN</td>
<td>3.66</td>
<td>1.00</td>
<td>2.24</td>
</tr>
<tr>
<td>WUHAN</td>
<td>9999.00</td>
<td>9999.00</td>
<td>9999.00</td>
</tr>
<tr>
<td>Location</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>YAKUTSK</td>
<td>3.93</td>
<td>2.34</td>
<td>5.80</td>
</tr>
<tr>
<td>YANGI-BAZAR</td>
<td>3.11</td>
<td>0.97</td>
<td>3.32</td>
</tr>
<tr>
<td>YELLOW-KNIFE</td>
<td>2.74</td>
<td>3.15</td>
<td>5.39</td>
</tr>
<tr>
<td>YUZHNO SAKHALSK</td>
<td>3.13</td>
<td>1.46</td>
<td>1.18</td>
</tr>
<tr>
<td>YUZHNO SAKH II</td>
<td>3.89</td>
<td>2.35</td>
<td>4.28</td>
</tr>
<tr>
<td>YUZHNO SAK III</td>
<td>5.45</td>
<td>6.38</td>
<td>2.87</td>
</tr>
<tr>
<td>ZAYMISHCHE</td>
<td>3.49</td>
<td>1.02</td>
<td>4.41</td>
</tr>
<tr>
<td>ZUY</td>
<td>4.07</td>
<td>0.87</td>
<td>2.54</td>
</tr>
</tbody>
</table>

// EXEC NOTIFYTS
..... STAGE 2 Run Deck with source deck ..... 

//XRJRRXYZ JOB (F8002,X22,50),STEP2B,TIME=(6,0),NOTIFY=XRJRR,CLASS=O, 
// MSGCLASS=X  
/*JOBPARM LINES=60 
/* XRJRR.DTAPE.PROCESS(STEP2B) 
/* TEST FOR EULER ANGLE, BIAS SOLUTION. MARCH 19-21, 1984 
// EXEC SYSIN 
//SYSIN DD DSN=XRJRR.FIT.FILES(UPDMSP2),DISP=SHR 
//TPSY EXEC PGM=TPSYS,REGION=150K 
//STEPLIB DD DSN=YCWDW.TPSYS.LOAD,DISP=SHR 
//FT10F001 DD DSN=YCDMM.FIT.FORT,UNIT=SYSDA,DISP=SHR 
//FT11F001 DD UNIT=SYSDA,DSN=YCDMM.FIT.SRCE,SPACE=(CYL,(5,1),RLSE), 
// DBC=(RECFM=FB,LRECL=80,BLKSIZE=7280),DISP=(,PASS) 
//FT06F001 DD SYSOUT=* 
//FT05F001 DD DISP=SHR,DSN=&&DATA5 
// EXEC OFORTH,PARM='XREF,LINECNT=60',REGION=2000K 
//SYSIN DD DISP=(OLD,DELETE),DSN=YCDMM.FIT.SRCE 
//RESULT EXEC OLINKH,COND=(9,LT) 
//NEWLIN DD DSN=YCWDW.FITQ.LOAD,DISP=SHR 
//SYSLMOD DD DSN=YCDMM.FIT.T.LOAD,DISP=(,PASS) 
//OBJECT DD * 
// INCLUDE NEWLIN(FIT8305) 
ENTRY MAIN 
NAME FIT(R) 
REALY EXEC PGM=FIT,REGION=3000K 
STEPLIB DD DISP=SHR,DSN=YCDMM.FIT.LOAD 
GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2),RLSE), 
 DBC=(RECFM=VBS,LRECL=200,BLKSIZE=12004) 
GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(7,2),RLSE), 
 DBC=(RECFM=VBS,LRECL=200,BLKSIZE=12004) 
GO.FT06F001 DD SYSOUT=* 
GO.FT07F001 DD DUMMY,SYSSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280), 
 SPACE=(CYL,(0,1),RLSE) 
GO.FT08F001 DD DUMMY,DSN=POG6CQ,UNIT=(9TRACK,,DEFER),DISP=(OLD,KEEP), 
 DBC=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN), 
 VOL=SER=MAG001 
GO.FT09F002 DD DUMMY,DSN=POG6MQ,UNIT=(9TRACK,,DEFER),DISP=(OLD,KEEP), 
 DBC=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(2,SL,,IN), 
 VOL=SER=MAG001 
GO.FT10F003 DD DUMMY,DSN=POG246,UNIT=(9TRACK,,DEFER),DISP=(OLD,KEEP), 
 DBC=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(3,SL,,IN), 
 VOL=SER=MAG001 
GO.FT11F001 DD DSN='XRJRR.FIT.OUT.NMATX',DISP=SHR 
//FT12F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(90,20),RLSE), 
// DBC=(RECFM=VBS,LRECL=100,BLKSIZE=7204) 
GO.FT13F001 DD DUMMY 
GO.FT15F001 DD DUMMY 
GO.FT16F001 DD DUMMY 
GO.FT17F001 DD DUMMY 
GO.FT18F001 DD DUMMY 
/*
/* BINARY INPUT DATA FOLLOWS.*/

//FT20F001 DD DSN=XJRJRR.MAR1984.STEPL.OUTBIN,DISP=SHR

//

//FT22F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(40,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)

//FT23F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(40,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)

//FT24F001 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(TRK,(40,10),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7200)

/* OUTPUT COEFFICIENTS FOLLOW.*/

//FT25F001 DD DUMMY,DSN=XJRJRR.FIT.DMSP.COEFFS,UNIT=SYSDA,DISP=SHR
//*VOL=SER=SACC09,DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
//*SPACE=(TRK,(2,2),RLSE),DISP=(NEW,CATLG)

/*

//FT35F001 DD DUMMY
//FT36F001 DD DUMMY
//FT40F001 DD DUMMY
//FT41F001 DD DUMMY
//FT45F001 DD DUMMY
//FT46F001 DD DUMMY
//FT65F001 DD DUMMY
//SYSUDUMP DD DUMMY
//FT05F001 DD *
&CONTRL
  NSIML=0, IRSTRT=0,NOISE=0,RTIM=9999.,
  EULER=2,IBIAS=1,
  ITR=2,
  NSKIP=1,
&END

DMSP MAR19-21, 1984 SOLVE FOR EU ANGLES, BIASES. LAT CUT=75, DEGREE=4.
&FIELD
  MONO=2,
  NMAXR=0, NMAXT=0,
  BGN=0.,
  EXTLD=0,NEXT=0
  PRCORL=4,
  IDST=0,
  EPOCH=1984.22,AVETIM=1984.22,
  NMAX=11,NMAXT=9,NMAXTT=0,NMXTT=0,NMAXIV=0,NMAXV=0,
&END
&LIMERR
  ERRLIM=2*90.,8*1000., 3*1000.,2*.0012,2*30.,
  NTDATA=1,NAT(1)=8,NAT(2)=2,APRIOR=0,
  NSTP=0,
  AYSTAT=1, BIASAP=5000.,
  NDGEN=5, &END

2 1-29878.2 0.000000E+00 26.9879 0.000000E+000.000000E+000.000000E+00
2 2-1924.05 5526.55 7.95582 -19.3154 0.000000E+000.000000E+00

36
C. STAGE 3 is a simple program to change the format of the SHA coefficients.

...... Run Deck for STAGE 3 .......

//XRJRST3 JOB (F8002,X22,10),STEP3,TIME=(0,10),CLASS=O,MSGCLASS=X,
// NOTRF=XRJRR
//* STEP3. INPUT IS "COEFFS" FILE ON UNIT #25 FROM STEP2.
//* THIS PROGRAM READS GAUSS COEFFICIENTS IN FIT FORMAT AND WRITES
//* THEM OUT IN FID FORMAT.
// EXEC FORTRAN,PARM='XREF'
//SYSIN DD *
   INTEGER WORD(20)
   DATA NZERO/0/,MZERO/0/
C SET SOME MORE FID PARAMETERS
   DATA MODEXT/0/,K/0/,ABAR/6371.2/,MODIND/0/
C READ IN TITLE FROM FIT FORMAT DATA, ON UNIT #1.
   READ(1,101) (WORD(I),I=1,20)
101 FORMAT(20A4)
C READ IN FIT INPUT PARAMETERS
   READ(I,102) NMAX,NMAXT,NMAXTT,NMAXT3,EPOCH
102 FORMAT(412,16X,FI0.0)
C*** NOTE: FOR THIS VERSION OF THE PROGRAM ONLY, SET NMAXT=9. ****
   NMAXT=9
C WRITE OUT FID PARAMETERS
   WRITE(2,201) NMAX,NMAXT,NMAXTT,NMAXT3,MODEXT,K,EPOCH,ABAR,
             MODIND
201 FORMAT(612,2F6.I,I2)
C WRITE OUT TITLE
   WRITE(2,101) (WORD(I),I=1,20)
   ICOUNT=0
C READ IN GAUSS COEFFICIENTS AND WRITE OUT
   READ(I,103) N,M,G,H,GT,HT,GTT,HTT
103 FORMAT(213,6FI2.0)
   WRITE(2,203) N,M,G,H,GT,HT,GTT,HTT
203 FORMAT(213,6FII.4)
C IF (N .LT. NMAX) GO TO 3
IF (M .LT. NMAX) GO TO 3
   GO TO 5
C
3 ICOUNT = ICOUNT + 1
   NF = N
   MF = M
   GO TO 1
C
5 ICOUNT = ICOUNT + 1
C PUT ZEROS AT THE BOTTOM OF THE DATA LIST
   WRITE(2,203) NZERO,MZERO
   WRITE(2,203) NZERO,NZERO
   NF = N
   MF = M
   WRITE(6,601) ICOUNT,NF,MF

37
601 FORMAT(///,10X,'NUMBER OF COEFFICIENTS IS: ',I3,/, > 10X,'MAX DEGREE= ',I3,3X,'MAX ORDER= ',I3)
C
STOP
END
// EXEC LINKGO,REGION.GO=200K
//GO.FT01F001 DD DSN=XRSHS.JAN1885.STEP2.COEFFS,DISP=SHR
//GO.FT02F001 DD DSN=XRSHS.JAN1885.STEP3.COEFFS,DISP=(NEW,CATLG),
// DCB=(RECFM=FB,LRECL=80,BLKSIZ=8000),SPACE=(TRK,(2,2),RLSE),
// VOL=SER=SACC01,UNIT=SYSDA
// EXEC NOTIFYTS

38
VI. Further Data Processing

Each of these sets was then further processed by rejecting data with high Kp or large DST indices. The DST index was added to each data point, and data in each set were sorted by geographical location (equal area bin). Finally, data in each bin were rejected, until a specified number (9 for dip-latitude > 30 degrees, 3 for dip-lat < 30) per bin was obtained. The 15 sets were then concatenated into a single file, ready for input into the FIT program.

Table 2 summarizes this process and indicates the input and output data sets used in the various programs.

Programs and Processing Steps

TABLE 2: PROCESSING OF DMSP DATA IN PREPARATION FOR FIELD MODELING

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>INPUT</th>
<th>OUTPUT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSTADD</td>
<td>DATE.STEP5.OUTBIN</td>
<td>TEMPFILE #1</td>
<td>Adds DST values, equal-area geographic bin numbers to data. Reject selected lengths of data with high Kp and large DST indices. Sorts data by bin number.</td>
</tr>
<tr>
<td></td>
<td>Dst tape</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BINSIFT</td>
<td>TEMPFILE #1</td>
<td>DATE.SIFT.DATA (VBS,lrec=116, Blksize=11604)</td>
<td>Reduces the # of points in each equal-area bin down to a specified level. Point rejection criteria: 1) points flagged by STEP5, 2) DST beyond the -5 to 20 nT range, 3) Random rejection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The output from this program, DATE.SIFT.DATA, becomes file 6 on the output tape.

FITprep | DATE.SIFT.DATA (14 separate dates) | DMSP.FITPRP (VBS,lrec=11204, Blksize=22412) | Concatenates individual data sets into 1 file; puts data into FIT format (100 points per logical record). |
EUTRANS  
DMSP.FITPRP  TEMPFILE #2  Applies calibration values calculated in a FIT run to update the data.

XYZTRANS  
TEMPFILE #2  DMSP.FITXYZ  (VBS,lrecl=11204, Blksize=22412)  Transforms spacecraft coordinates to XYZ (topocentric) coordinates. Data is in old FIT format.

Format Information

XRJRR.DMSP.FITPRP - Same format as File#6, except that data have been re-concatenated into 100 points per record.

XRJRR.DMSP.FITXYZ - Same format as DMSP.FITPRP, except that A(11,I) holds X (north) component in topocentric coordinates, A(12,I) holds Y (east) component, A(13,I) holds Z (radial) component.

VII. Data Tapes and Cartridges

Programs and Related Information

All programs and related data sets known to be relevant to DMSP data processing and evaluation have been collected and saved on Cartridge S01000. TABLE 3 summarizes the contents of this cartridge.

Note that particular JCL was necessary to copy a load module onto the cartridge and that restoration of that load module also requires specific JCL. This JCL is given as follows:
1) To copy an IBM load module from a partitioned data set on a disk to a tape or cartridge file:

```
//XR1RBFAT JOB (F8002,X22,10), 'IEBCPY', CLASS=A,
// MSGCLASS=X, TIME=(,30), NOTIFY=XR1RB
/*JOBPARM LINES=50
/* XR1RB.LIB.CNTL(IEBCPY)
/*====================================================================

/*
/* THIS COPIES A LOAD MODULE ON DISK TO TAPE OR CARTRIDGE
/*
/*====================================================================

/* EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR, DSN=XRIRB.FIT.DMSP.LOAD2
//SYSUT2 DD DISP=(NEW,PASS),
//  UNIT=3480, VOL=SER=S01000, LABEL=(4, SL,, OUT),
// DSN=OLDFIT3
/*

2) To copy a load module on a tape or cartridge file to a partitioned data set on disk.

```

```
//XR1RBFAT JOB (F8002,X22,10), 'CPYDSK', CLASS=A,
// MSGCLASS=X, TIME=(,30), NOTIFY=XR1RB
/*JOBPARM LINES=50
/* XR1RB.LIB.CNTL(CPYDSK)
/*====================================================================

/*
/* THIS COPIES A LOAD MODULE ON TAPE OR CARTRIDGE TO DISK
/* PARTITIONED
/*====================================================================

/* EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=* 
//SYSIN DD DUMMY
//SYSUT1 DD DISP=(NEW,PASS),
//  UNIT=3480, VOL=SER=S01000, LABEL=(4, SL,, IN),
// DSN=OLDFIT3
//SYSUT2 DD DSN=XR1RB.TEMPORY.NAME, SPACE=(TRK,(14,5,1), RLSE),
// DISP=(NEW, CATLG), DCB=(RECFM=U, BLKSIZE=19069), UNIT=SYSDA
```
<table>
<thead>
<tr>
<th>PROGRAM NAME OR FILE IDENTIFIER</th>
<th>SOURCE</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTER</td>
<td>XRTJS.DMSP.FILT.CNTL</td>
<td></td>
</tr>
<tr>
<td>Files for old FIT Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLD FIT ONE</td>
<td>XRTJR.FIT.FILES(UPDMSP2)</td>
<td>Update Deck</td>
</tr>
<tr>
<td>OLD FIT TWO</td>
<td>YCDMM.FIT.FORT</td>
<td>Standard Fit Source Code</td>
</tr>
<tr>
<td>OLD FIT THREE</td>
<td>XRTJR.FIT.DMSP.LOAD2</td>
<td>Load Module for fit used for DMSP</td>
</tr>
<tr>
<td>OLD FIT FOUR</td>
<td>YCWDW.TPSYS.LOAD</td>
<td>Standard Update Sys.</td>
</tr>
<tr>
<td>OLD FIT FIVE</td>
<td>YCWDW.FITQ.LOAD</td>
<td>Contains additional FIT items such as assembler version of DLOOP and FMOVE, FREAD, etc.</td>
</tr>
<tr>
<td>Program to reformat SHA Coefficients</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEP3</td>
<td>XRTJR.DTAPE.PROCESS(STEP3)</td>
<td>STAGE3 Run Deck</td>
</tr>
<tr>
<td>Programs to work with cleaned up data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BINSIFT</td>
<td>XRTJR.DTAPE.PROCESS(BINSIFT)</td>
<td>Run Deck</td>
</tr>
<tr>
<td>DSTADD</td>
<td>XRTJR.DTAPE.PROCESS(DSTADD)</td>
<td>Run Deck</td>
</tr>
<tr>
<td>FITPREP</td>
<td>XRTJR.DTAPE.PROCESS(FITPREP)</td>
<td>Run Deck</td>
</tr>
<tr>
<td>EUTRANS</td>
<td>XRTJR.DTAPE.PROCESS(EUTRANS)</td>
<td>Run Deck</td>
</tr>
<tr>
<td>XYZTRANS</td>
<td>XRTJR.DTAPE.PROCESS(XYZTRANS)</td>
<td>Run Deck</td>
</tr>
<tr>
<td>Deck Setups for the STAGES of Table 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEP1</td>
<td>XRTJR.DTAPE.PROCESS(STEP1)</td>
<td>Run Deck for STAGE1</td>
</tr>
<tr>
<td>STEP2</td>
<td>XRTJR.DTAPE.PROCESS(STEP2)</td>
<td>Run Deck for STAGE2, using Load Module</td>
</tr>
<tr>
<td>STEP2B</td>
<td>XRTJR.DTAPE.PROCESS(STEP2B)</td>
<td>Run Deck for STAGE2, using source module</td>
</tr>
<tr>
<td>STEP4</td>
<td>XRTJR.DTAPE.PROCESS(STEP4)</td>
<td>Run Deck for STAGE4</td>
</tr>
<tr>
<td>STEP5</td>
<td>XRTJR.DTAPE.PROCESS(STEP5)</td>
<td>Run Deck for STAGE5</td>
</tr>
<tr>
<td>Source and Load Modules for Stages 1, 4, and 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CODE1</td>
<td>XRTJR.DTAPE.PROCESS</td>
<td>Modified source code of FILTER and of BSPLYNE3 for DMSP processing.</td>
</tr>
</tbody>
</table>
| SATFILT                       | XRTJR.SATFILT | Load module containing DMSP version of FILTER and BSPLINE.
........ Data or Model Input ...........
20 CAL84FID XRJRR.DTAPE.PROCESS(CAL84FID) Initial field model
21 DST81 XRJRR.DST81 Yearly Dst
22 FITPRP XRJRR.DMSP.FITPRP Output Data
23 FITXYZ XRJRR.DMSP.FITXYZ Output Data
24 BSPINFO XRTJS.BSPINFO.DATA Bspline and Fourier parameters

........ Miscellaneous Ridgway Programs ...........
From XRJRR.DMSP.PROGRAMS(....)
25 ADDFLAG Flags Bspline outliers from FILTER before plotting
26 ADDFLAG2 Same as ADDFLAG, except all bad points flagged
27 BSIG Computes statistics on DMSP data relative to selected field model.
28 LOOK Printout of STAGE5 output.
29 POWFLT Power spectrum plotting routine.

........ DMSP version of the BSPLINE Program ............
30 BSPLINE XRTJS.LIB.MAG(BSPLYN3) General program for B-Spline and Fourier fitting.

Processed Data

The following Table is a list of the original tapes as received from AFGL and a list of the tapes onto which the data was processed. The processed data tape is a six file tape; the formats of the six files are described in the following paragraphs. For permanent storage, the tapes were copied onto the indicated Cartridge: the first cartridge file contains the contents of the (one file) raw data tape from AFGL; the second through seventh cartridge files are files one through six of the processed data tape.

<table>
<thead>
<tr>
<th>DATE</th>
<th>RAW DATA TAPE</th>
<th>TAPE ON WHICH PROCESSED DATA IS STORED</th>
<th>CARTRIDGE ON WHICH TAPES ARE COPIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/7-9/84</td>
<td>DT0030</td>
<td>DT0119</td>
<td>S01011=S01012</td>
</tr>
<tr>
<td>1/17-18/84</td>
<td>MAG025,MAG026</td>
<td>DT0120</td>
<td>S01013=S01014</td>
</tr>
<tr>
<td>3/19-21/84</td>
<td>DT0031</td>
<td>DT0121</td>
<td>S01015=S01016</td>
</tr>
<tr>
<td>5/6-8/84</td>
<td>DT0105</td>
<td>DT0122</td>
<td>S01017=S01018</td>
</tr>
<tr>
<td>6/20-23/84</td>
<td>DT0106</td>
<td>DT0123</td>
<td>S01019=S01020</td>
</tr>
<tr>
<td>8/20-23/84</td>
<td>DT0107</td>
<td>DT0124</td>
<td>S01021=S01022</td>
</tr>
<tr>
<td>9/16,17/84</td>
<td>DT0108</td>
<td>DT0125</td>
<td>S01023=S01024</td>
</tr>
<tr>
<td>1/18-20/85</td>
<td>DT0109</td>
<td>DT0127</td>
<td>S01025=S01026</td>
</tr>
<tr>
<td>5/23-25/85</td>
<td>DT0111</td>
<td>DT0128</td>
<td>S01027=S01028</td>
</tr>
</tbody>
</table>

(Data on these dates was "bad", not used in final modeling)
The input tapes from AFGL were one file of ASCII data with the following format:

Tape characteristics: Non-labeled, 6250 BPI, Recfm=FB, Lrecl=75, Blksize = 1875, ASCII (OPTCD=Q).

Data format:

Header record -- every 60 seconds:

<table>
<thead>
<tr>
<th>COLS</th>
<th>VARIABLE</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>IYR</td>
<td>INT</td>
<td>Year</td>
</tr>
<tr>
<td>5-8</td>
<td>IDAYD</td>
<td>INT</td>
<td>Day number</td>
</tr>
<tr>
<td>9-14</td>
<td>IETIME</td>
<td>INT</td>
<td>Time of record (seconds U.T.)</td>
</tr>
<tr>
<td>15-18</td>
<td>IALT</td>
<td>INT</td>
<td>Altitude (Nautical miles)</td>
</tr>
<tr>
<td>19-28</td>
<td>GLAT</td>
<td>REAL</td>
<td>Geographic latitude.</td>
</tr>
<tr>
<td>29-38</td>
<td>GLONG</td>
<td>REAL</td>
<td>Geographic longitude.</td>
</tr>
<tr>
<td>39-48</td>
<td>GMLAT</td>
<td>REAL</td>
<td>Corrected geomagnetic latitude.</td>
</tr>
<tr>
<td>49-58</td>
<td>GMLONG</td>
<td>REAL</td>
<td>Corrected geomagnetic longitude.</td>
</tr>
<tr>
<td>59-68</td>
<td>XMLT</td>
<td>REAL</td>
<td>Corrected geomagnetic local time.</td>
</tr>
<tr>
<td>72-75</td>
<td>NS</td>
<td>INT</td>
<td>Number of data records following header (usually = 60).</td>
</tr>
</tbody>
</table>

Data record -- every second

<table>
<thead>
<tr>
<th>COLS</th>
<th>VARIABLE</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>IDSEC</td>
<td>INT</td>
<td>Time of data record(sec U.T.) (I6)</td>
</tr>
<tr>
<td>12-29</td>
<td>X1,Y1,Z1</td>
<td>INT</td>
<td>Magnetometer counts for first of 20 samples per second; 3 axes (316).</td>
</tr>
<tr>
<td>34-51</td>
<td>X2,Y2,Z2</td>
<td>INT</td>
<td>Magnetometer counts for eleventh of 20 samples per second; 3 axes (316).</td>
</tr>
<tr>
<td>56-75</td>
<td>NF(1-10)</td>
<td>INT</td>
<td>Ten data quality flags (10I2).</td>
</tr>
</tbody>
</table>

The raw data tape from AFGL becomes the first file, ASCII, on the output cartridge. Files 1 through 5 on the output tape become files 2 through 6 on the cartridge. File 6 on the output tape, to be described later in this Section, becomes file 7 on the cartridge.
Storage of processed data: On the "processed data" tapes, and corresponding cartridges, the files are set up as follows (add one file number for cartridge files):

<table>
<thead>
<tr>
<th>File#</th>
<th>Data File</th>
<th>File characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATE.STEP2.COEFFS</td>
<td>FB, Lrecl=80, Blksize=5440</td>
</tr>
<tr>
<td>2</td>
<td>DATE.STEP3.COEFFS</td>
<td>FB, Lrecl=80, Blksize=5520</td>
</tr>
<tr>
<td>3</td>
<td>DATE.STEP4.OUTF</td>
<td>FB, Lrecl=240, Blksize=4800</td>
</tr>
<tr>
<td>4</td>
<td>DATE.STEP5.OUTBIN</td>
<td>VBS, Lrecl=11204, Blksize=22412</td>
</tr>
<tr>
<td>5</td>
<td>DATE.STEP5.OUTF</td>
<td>FB, Lrecl=240, Blksize=4800.</td>
</tr>
<tr>
<td>6</td>
<td>DATE.SIFT.DATA</td>
<td>VBS, Lrecl=116, Blksize=11604</td>
</tr>
</tbody>
</table>

Formats of processed data files:

i) File #1 - Standard FIT coefficient format.

ii) File #2 - Standard FDG coefficient format.

iii) File #3,5 - Fixed block format:

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>TYPE</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>INT</td>
<td>I2</td>
<td>Year</td>
</tr>
<tr>
<td>3-6</td>
<td>INT</td>
<td>I4</td>
<td>Day</td>
</tr>
<tr>
<td>7-12</td>
<td>INT</td>
<td>I6</td>
<td>Seconds of day.</td>
</tr>
<tr>
<td>13-19</td>
<td>Real</td>
<td>F7.2</td>
<td>Geographic latitude.</td>
</tr>
<tr>
<td>20-26</td>
<td>Real</td>
<td>F7.2</td>
<td>Geocentric latitude.</td>
</tr>
<tr>
<td>27-33</td>
<td>Real</td>
<td>F7.2</td>
<td>Longitude.</td>
</tr>
<tr>
<td>34-40</td>
<td>Real</td>
<td>F7.2</td>
<td>Dip-latitude.</td>
</tr>
<tr>
<td>41-47</td>
<td>Real</td>
<td>F7.2</td>
<td>Dip-longitude.</td>
</tr>
<tr>
<td>48-54</td>
<td>Real</td>
<td>F7.2</td>
<td>Altitude.</td>
</tr>
<tr>
<td>55-61</td>
<td>Real</td>
<td>F7.2</td>
<td>Geocentric Radius.</td>
</tr>
<tr>
<td>62-69</td>
<td>Real</td>
<td>F8.1</td>
<td>Cross-track spacecraft mag. component.</td>
</tr>
<tr>
<td>70-77</td>
<td>Real</td>
<td>F8.1</td>
<td>Radial spacecraft mag. component.</td>
</tr>
<tr>
<td>78-85</td>
<td>Real</td>
<td>F8.1</td>
<td>Along-track spacecraft mag. component.</td>
</tr>
<tr>
<td>86-93</td>
<td>Real</td>
<td>F8.1</td>
<td>Total field intensity.</td>
</tr>
<tr>
<td>94-101</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual cross-track component.</td>
</tr>
<tr>
<td>102-109</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual radial</td>
</tr>
<tr>
<td>110-117</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual along-track</td>
</tr>
<tr>
<td>118-125</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual total field intensity.</td>
</tr>
<tr>
<td>126-133</td>
<td>Real</td>
<td>F8.1</td>
<td>X (north) component.</td>
</tr>
<tr>
<td>134-141</td>
<td>Real</td>
<td>F8.1</td>
<td>Y (east) component.</td>
</tr>
<tr>
<td>142-149</td>
<td>Real</td>
<td>F8.1</td>
<td>Z (down) component.</td>
</tr>
<tr>
<td>150-157</td>
<td>Real</td>
<td>F8.1</td>
<td>Total field intensity.</td>
</tr>
<tr>
<td>158-165</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual X component.</td>
</tr>
<tr>
<td>166-173</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual Y component.</td>
</tr>
<tr>
<td>174-181</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual Z component.</td>
</tr>
<tr>
<td>182-189</td>
<td>Real</td>
<td>F8.1</td>
<td>Residual total field intensity.</td>
</tr>
<tr>
<td>190-197</td>
<td>Real</td>
<td>F8.1</td>
<td>Model field, X component.</td>
</tr>
<tr>
<td>198-205</td>
<td>Real</td>
<td>F8.1</td>
<td>Model field, Y component.</td>
</tr>
<tr>
<td>206-213</td>
<td>Real</td>
<td>F8.1</td>
<td>Model field, Z component.</td>
</tr>
<tr>
<td>214-221</td>
<td>Real</td>
<td>F8.1</td>
<td>Model field, total intensity.</td>
</tr>
<tr>
<td>222-226</td>
<td>INT</td>
<td>I5</td>
<td>Velocity direction flag (=+, -1)</td>
</tr>
<tr>
<td>227-231</td>
<td>INT</td>
<td>I5</td>
<td>Data quality flag (=0 if good, 1-7 if bad)</td>
</tr>
<tr>
<td>232-239</td>
<td>INT</td>
<td>4I2</td>
<td>Data indicator flags for X, Y, Z, B.</td>
</tr>
</tbody>
</table>
iv) File #4 - Binary FIT format (100 points per record, 28 real*4 words per point):

One of the common formats into the (old) field modeling program is called FIT format. Data from the POGO, Magsat and DMSP F-7 satellites are generally in this format, or a variation thereof. These files are binary with 100 points per record and with each point having 28 REAL*4 words of data, as follows:

```
REAL*4 A(28,100)
INTEGER IA(28,100)
EQUIVALENCE (A(1,1), IA(1,1))
```

<table>
<thead>
<tr>
<th>ARRAY LOCATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA(1,1)</td>
<td>Modified Julian Day.</td>
</tr>
<tr>
<td>IA(2,1)</td>
<td>Milliseconds of Day.</td>
</tr>
<tr>
<td>A(3,1)</td>
<td>Not used.</td>
</tr>
<tr>
<td>A(4,1)</td>
<td>In some cases not used, in others fraction of day.</td>
</tr>
<tr>
<td>A(5,1)</td>
<td>Time in years from 1900.</td>
</tr>
<tr>
<td>A(6,1)</td>
<td>Geocentric latitude.</td>
</tr>
<tr>
<td>A(7,1)</td>
<td>Longitude.</td>
</tr>
<tr>
<td>A(8,1)</td>
<td>Not used.</td>
</tr>
<tr>
<td>A(9,1)</td>
<td>Not used.</td>
</tr>
<tr>
<td>A(10,1)</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

If data are in geocentric coordinates:
- A(11,1) North component, -B\theta, or Satellite X-axis component.
- A(12,1) East component, B\phi, or Satellite Y-axis component.
- A(13,1) Satellite Z-axis component (along-track).

If data are in spacecraft coordinates:
- A(11,1) Cross track component
- A(12,1) Radially down component
- A(13,1) Along track component

- A(14,1) Scalar total intensity.
- IA(15,1) Geocentric altitude (meters) above earth radius, earth radius taken to be 6371.0 km. (Note: this standard Earth radius was used for Magsat. Other data used 6371.2. User beware.)
- A(16,1) Not used.
- A(17,1) Not used.
Used only for DMSP = Data quality classification flag (0-7):
0 = Data is adequate quality
1 = Residual from field model exceeds a specified cutoff (Gross outlier)
2 = Padded time gap value (data does not actually exist here on tape.)
3 = Outlier from B-spline function
4 = Outlier from Fourier function
5
6 = Latitude of data exceeds specified geocentric latitude cutoff.
7 = Direction of satellite indeterminable

=0 except for DMSP where it indicates satellite velocity vector direction (=+,-i), + means going north; - means going south, if zero the direction is undetermined.

Magnetic latitude outlier flag for sat. X axis. (0 = no data; 2 = data)

(0 = no data; 2 = data)

(0 = no data; 2 = data)

Not used.
Not used.
Not used.

v) File #6 - Binary "pseudo-FIT" format (1 point per record, 28 real*4 words per point): Same as FIT format, except that IA(16) holds the geographic equal-area bin number, and IA(17) holds the DST hourly index.
Unprocessed Data

The following Table is a list of original tapes from AFGL which were received after processing was suspended.

<table>
<thead>
<tr>
<th>DATE</th>
<th>RAW DATA TAPE</th>
<th>CARTRIDGE ON WHICH TAPES ARE COPIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/1/85</td>
<td>DT0116</td>
<td>S01037=S01038</td>
</tr>
<tr>
<td>10/16-18/84</td>
<td>DT0181 (M4784)</td>
<td>S01043=S01044</td>
</tr>
<tr>
<td>1/11-14/86</td>
<td>DT0182 (M5098)</td>
<td>S01045=S01046</td>
</tr>
<tr>
<td>3/9-12/86</td>
<td>DT0183 (M5097)</td>
<td>S01047=S01048</td>
</tr>
<tr>
<td>5/13-15/86</td>
<td>DT0184 (M4785)</td>
<td>S01049=S01050</td>
</tr>
<tr>
<td>5/28-30/86</td>
<td>DT0185 (M5099)</td>
<td>S01051=S01052</td>
</tr>
<tr>
<td>6/25-26/86</td>
<td>DT0186 (M5100)</td>
<td>S01053=S01054</td>
</tr>
<tr>
<td>7/14-15/86</td>
<td>DT0187 (M5101)</td>
<td>S01055=S01056</td>
</tr>
<tr>
<td>1/16</td>
<td>DT0188 (M5102)</td>
<td>S01057=S01058</td>
</tr>
<tr>
<td>8/16-18/86</td>
<td>DT0189 (M5103)</td>
<td>S01059=S01060</td>
</tr>
<tr>
<td>9/8/86</td>
<td>DT0190 (M5218)</td>
<td>S01061=S01062</td>
</tr>
<tr>
<td>9/16,22/86</td>
<td>DT0191 (M5104)</td>
<td>S01063=S01064</td>
</tr>
<tr>
<td>10/10-12/86</td>
<td>DT0192 (M5215)</td>
<td>S01065=S01066</td>
</tr>
<tr>
<td>11/8-10/86</td>
<td>DT0193 (M5216)</td>
<td>S01067=S01068</td>
</tr>
<tr>
<td>11/21-23/86</td>
<td>DT0194 (M5217)</td>
<td>S01069=S01070</td>
</tr>
<tr>
<td>12/5-6/86</td>
<td>DT0195 (M5270)</td>
<td>S01071=S01072</td>
</tr>
<tr>
<td>??????????</td>
<td>DT0196 (M4050)</td>
<td>S01073=S01074</td>
</tr>
<tr>
<td>2/15/85</td>
<td>DT0216 (M4786)</td>
<td>S01075=S01076</td>
</tr>
<tr>
<td>4/18/85</td>
<td>DT0217 (M4788)</td>
<td>S01077=S01078</td>
</tr>
<tr>
<td>2/16/86</td>
<td>DT0218 (M4787)</td>
<td>S01079=S01080</td>
</tr>
</tbody>
</table>

These tapes and cartridges are one file, ASCII, in the same format as the first file on the tapes with processed data.
VIII. Individual Epoch DMSP Field Models

The correction procedure was applied to 15 sub-sets of DMSP data, each containing several days of data. These are the data in the processed tapes and cartridges of the previous section. Subset epochs ranged from January, 1984 through November, 1985. Each data set was chosen from a magnetically quiet period as determined by the world-wide Kp index. Results of STAGE 2, which solves for the field model and magnetometer adjustment parameters, are summarized in Table 4.

Table 4

<table>
<thead>
<tr>
<th>Date</th>
<th>ga</th>
<th>gb</th>
<th>h</th>
<th>ex</th>
<th>ey</th>
<th>ez</th>
<th>BIAS1</th>
<th>BIAS2</th>
<th>BIAS3</th>
</tr>
</thead>
<tbody>
<tr>
<td>yrs</td>
<td>nT</td>
<td>nT</td>
<td>nT</td>
<td>deg</td>
<td>deg</td>
<td>deg</td>
<td>nT</td>
<td>nT</td>
<td>nT</td>
</tr>
<tr>
<td>84.02</td>
<td>29895</td>
<td>-1927</td>
<td>5522</td>
<td>-.038</td>
<td>-.449</td>
<td>0.005</td>
<td>12.0</td>
<td>2.6</td>
<td>-1.7</td>
</tr>
<tr>
<td>84.05</td>
<td>29893</td>
<td>-1928</td>
<td>5532</td>
<td>-.065</td>
<td>-.446</td>
<td>0.006</td>
<td>0.2</td>
<td>-.5</td>
<td>1.4</td>
</tr>
<tr>
<td>84.21</td>
<td>29887</td>
<td>-1935</td>
<td>5523</td>
<td>-.114</td>
<td>-.459</td>
<td>-.004</td>
<td>7.8</td>
<td>2.7</td>
<td>-8.7</td>
</tr>
<tr>
<td>84.34</td>
<td>29872</td>
<td>-1925</td>
<td>5516</td>
<td>-.172</td>
<td>-.457</td>
<td>-.013</td>
<td>11.4</td>
<td>-2.0</td>
<td>-12.5</td>
</tr>
<tr>
<td>84.47</td>
<td>29860</td>
<td>-1922</td>
<td>5514</td>
<td>-.138</td>
<td>-.451</td>
<td>-.016</td>
<td>2.4</td>
<td>-7.6</td>
<td>-11.2</td>
</tr>
<tr>
<td>84.63</td>
<td>29866</td>
<td>-1932</td>
<td>5503</td>
<td>-.063</td>
<td>-.474</td>
<td>-.009</td>
<td>-11.9</td>
<td>-12.1</td>
<td>-3.8</td>
</tr>
<tr>
<td>84.71</td>
<td>29866</td>
<td>-1927</td>
<td>5505</td>
<td>-.092</td>
<td>-.478</td>
<td>-.006</td>
<td>-18.4</td>
<td>-9.5</td>
<td>-2.4</td>
</tr>
<tr>
<td>85.05</td>
<td>29857</td>
<td>-1918</td>
<td>5496</td>
<td>-.050</td>
<td>-.496</td>
<td>-.022</td>
<td>-91.2</td>
<td>-56.7</td>
<td>20.0</td>
</tr>
<tr>
<td>85.34</td>
<td>29856</td>
<td>-1910</td>
<td>5492</td>
<td>-.129</td>
<td>-.471</td>
<td>-.013</td>
<td>-91.2</td>
<td>-67.0</td>
<td>4.1</td>
</tr>
<tr>
<td>85.45</td>
<td>29838</td>
<td>-1920</td>
<td>5494</td>
<td>-.125</td>
<td>-.467</td>
<td>-.017</td>
<td>-93.6</td>
<td>-68.3</td>
<td>1.3</td>
</tr>
<tr>
<td>85.46</td>
<td>29843</td>
<td>-1916</td>
<td>5491</td>
<td>-.130</td>
<td>-.472</td>
<td>-.020</td>
<td>-86.7</td>
<td>-65.9</td>
<td>1.3</td>
</tr>
<tr>
<td>85.60</td>
<td>29842</td>
<td>-1915</td>
<td>5495</td>
<td>-.098</td>
<td>-.475</td>
<td>-.009</td>
<td>-100.4</td>
<td>-74.2</td>
<td>4.6</td>
</tr>
<tr>
<td>85.75</td>
<td>29847</td>
<td>-1908</td>
<td>5490</td>
<td>-.074</td>
<td>-.520</td>
<td>0.013</td>
<td>-112.6</td>
<td>-72.3</td>
<td>7.3</td>
</tr>
<tr>
<td>85.82</td>
<td>29843</td>
<td>-1914</td>
<td>5484</td>
<td>-.081</td>
<td>-.511</td>
<td>0.021</td>
<td>-113.1</td>
<td>-63.7</td>
<td>12.8</td>
</tr>
<tr>
<td>85.90</td>
<td>29832</td>
<td>-1905</td>
<td>5489</td>
<td>-.030</td>
<td>-.518</td>
<td>0.032</td>
<td>-110.0</td>
<td>-57.5</td>
<td>25.7</td>
</tr>
</tbody>
</table>

Figures 4a) through 4e) are derived from Table 4. They display solutions for ga, gb, h, the three Euler angles, and the three biases for each DMSP data set throughout time. The main field coefficients decrease in magnitude with time as expected from earlier models, but the trend is not smooth. This could indicate that the data sub-sets have marginal geographic distribution, or that the DMSP data are not sufficiently stable over time. The Euler angle solutions are fairly consistent, with ey (yaw) varying slowly from -.44 to -.52 degrees, ex (pitch) averaging about -.1 degrees and ez (roll) averaging about zero. The bias values show a noticeable break between September, 1984 and January, 1985, most strongly in X and Y. Biases at January, 1985 depart sharply from the previous bias trend in all three components. This jump is evident in the biases only, and its cause is uncertain. One possible explanation is that on 30 October, 1984, the solar array panel was rotated 90°. This could result in a changed contribution to the bias field from the solar array since both its position and its total current were changed. Another, though less likely, spacecraft change that could contribute to the bias change is that on 7 November, 1984, the skew momentum wheel was reset so that it drew 100 ma less current.
The bias values in Table 4 are part of the value of the vector parameter bias to be used in equation 3), i.e. they are a small time dependent correction to be applied in addition to the large bias values of equation 2). A small further correction is derived in section IX of paper 1.
<table>
<thead>
<tr>
<th>Figure #</th>
<th>Caption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DMSP orbital X, Y, and Z magnetic component data which have had an estimated field model removed, revealing strong periodicities in the residuals. The dashed line is a spline fit to the residuals.</td>
</tr>
<tr>
<td>2</td>
<td>Power spectra of X and Y DMSP residual data from Figure 1.</td>
</tr>
<tr>
<td>3</td>
<td>Y-component of DMSP data from Figure 1, demonstrating removal of outliers, magnetometer rotation and bias correction, and subtraction of Fourier periodic function. The dashed line is a spline fit to the residuals.</td>
</tr>
<tr>
<td>4</td>
<td>Plots of $g_0$, $g_1$, $h_{11}$, Euler angles, and biases versus time (yrs), for field model solutions from 15 DMSP data sets spanning 1984 - 1986.</td>
</tr>
</tbody>
</table>
AXIAL DIPOLE COEFFICIENT

$g_1$ (nT)

YEAR

FIGURE 4a
APPENDIX

This appendix contains listings of the primary programs used in the processing of the DMSP data. Most of these are documented with internal comments.
PROGRAM FILTER

C
C PROGRAM TO PRE-PROCESS SATELLITE MAGNETIC VECTOR DATA. THE PROGRAM IS
C COMPRISED OF FIVE STEPS, EACH MODULARLY DESIGNED:

C STEP 1:
C ======
C
C PERFORMED IN SUBROUTINE STEP1, IT INVOLVES THE READING OF AN ORIGINAL
C SATELLITE MAGNETIC DATA TAPE, AND TRANSFORMING THE RAW MAGNETOMETER
C COUNTS TO MAGNETIC FIELD VALUES IN THE SPACECRAFT COORDINATE SYSTEM.
C
C STEP 2:
C ======
C
C PERFORMED IN SUBROUTINE STEP2, IT INVOLVES THE LOCATION AND PADDING OF
C SPACECRAFT VELOCITY VECTOR AT EACH MEASUREMENT LOCATION.
C
C STEP 3:
C ======
C
C PERFORMED IN SUBROUTINE STEP3, IT INVOLVES THE TRANSFORMATION OF THE
C MAGNETIC FIELD MEASUREMENTS FROM SPACECRAFT TO TOPOCENTRIC COORDINATE
C SYSTEM FROM WHICH RESIDUAL MEASUREMENTS ARE DETERMINED FROM A GIVEN
C FIELD MODEL. DATA LOCATIONS AT WHICH ANY VECTOR RESIDUAL EXCEEDS THE
C SPECIFIED TOLERANCE ARE FLAGGED AS OUTLIERS.
C
C STEP 4:
C ======
C
C PERFORMED IN SUBROUTINE STEP4, IT INVOLVES FITTING A TREND TO THE
C MAGNETIC FIELD RESIDUALS WITH B-SPLINES AND/OR FOURIER WAVEFORMS, WITH
C THE OPTION OF FLAGGING POINTS WHOSE TREND RESIDUALS EXCEED A GIVEN
C TOLERANCE AND THE OPTION OF DETRENDING THE ORIGINAL DATA.
C
C STEP 5:
C ======
C
C PERFORMED IN SUBROUTINE STEP5, IT INVOLVES OUTPUTTING A FINAL MODIFIED
C SATELLITE MAGNETIC DATA TAPE IN THREE BASIC FORMS:
C
C (1) EBCDIC TAPE IN TOPOCENTRIC COORDINATES
C (2) EBCDIC TAPE IN DESIRED SPACECRAFT COORDINATES
C (3) BINARY TAPE IN OLD FIT PROGRAM FORMAT (MAGSAT CONVENTION)
C
C PROGRAM FILTER MAY RUN IN ONE OF FOUR MODES INDICATED BY THE INPUT
C VARIABLE IMODE:
C
C IMODE = 0:
C ========
C
C PERFORM STEPS 1, 2, 3, 4, AND 5
C
C IMODE = 1:
C ========
PROGRAM FILTER OPERATION IS GOVERNED BY VARIABLES INPUT THROUGH FIVE NAMELIST CATEGORIES:

- **CONTRL:** Governs program mode and ephemeris processing details.
- **IOFILE:** Establishes program logical units.
- **BSPLIN:** Provides vector measurement sigmas and information concerning trend fitting via B-splines and/or Fourier waveforms.
- **OUTLIM:** Provides residual field tolerance levels, magnetic latitude tolerance levels, geocentric latitude tolerance level, geodetic latitude above which spacecraft velocity vector direction is indeterminable, and time gap tolerance level.
- **FIELDP:** Provides information for the application of the given magnetic field model to be used as a basis for residual field measurements.
- **TRFORM:** Provides various rotation angles, slopes, and biases used to transform the raw magnetometer counts to magnetic field values in the spacecraft coordinate system.

Program Filter requires up to three input data sets located on the following logical units:
C IST1:
C ===
C SATELLITE MAGNETIC VECTOR MEASUREMENTS IN RAW MAGNETOMETER COUNTS,
C CURRENTLY IN DMSP SATELLITE FORMAT.
C
C IOCF:
C ===
C MAGNETIC FIELD MODEL INFORMATION IN PROGRAM FID FORMAT.
C
C IOBS:
C ===
C B-SPLINE KNOT POSITIONS, FOURIER WAVEFORM FREQUENCIES, AND OBSERVATION
C SIGMAS FOR MAGNETIC FIELD VALUES.
C
C===============================================================================
C===============================================================================
C===============================================================================
Glossary of Program Filter Namelist Items
===============================================================================

NAMELIST IOFILE -
================

IST1 - INPUT UNIT FOR ORIGINAL RAW DATA TAPE(S) IN STEP1.

IST2 - INPUT UNIT IN STEP2, OUTPUT UNIT IN STEP1, MAGNETIC FIELD
       IN FIT/MAGSAT COORDINATES.

IST3 - INPUT UNIT IN STEP3, OUTPUT UNIT IN STEP2, VELOCITY
       DIRECTIONS AND PADDED TIME-GAPS.

IST4 - INPUT UNIT IN STEP4, OUTPUT UNIT IN STEP3, MAGNETIC FIELD
       AND RESIDUALS IN TOPOCENTRIC COORDINATES.

IOR - FILTER INPUT UNIT, SAME AS IST4 IN OPERATION MODE 0
      AND 3.

IOW - FILTER OUTPUT UNIT, INPUT UNIT IN STEP5.

IOF - OUTPUT UNIT IN STEP5, FORMATTED MAGNETIC FIELD IN FIT/
      MAGSAT OR TOPOCENTRIC COORDINATES DEPENDING ON IBTBS
      VALUE.

IOD - OUTPUT UNIT IN STEP5, FORMATTED MAGNETIC FIELD IN DESIRED
      SPACECRAFT COORDINATES.

IOB - OUTPUT UNIT IN STEP5, BINARY MAGNETIC FIELD IN PROGRAM
      FIT FORMAT.

ISC1 - FILTER SCRATCH UNIT.

ISC2 - FILTER SCRATCH UNIT.

ISC3 - SCRATCH UNIT USED IN SUBPROGRAM DPINFO TO STORE VARIOUS
       DATA PARAMETERS.

NAMELIST FIELDP -
JJ - FID INPUT POSITION COORDINATES: (0) GEODETIC
         (1) GEOCENTRIC.

MM - FID EQUATORIAL RADIUS AND RECIPROCAL FLATTENING:
         (0) DEFAULT AE = 6378.16 KM, FLAT = 298.25 (1) INPUT
         VALUES.

NMX - MAXIMUM DEGREE OF FID MODEL EVALUATION.

NEXT - EXTERNAL FIELD MODEL: (0) DO NOT EVALUATE (1) EVALUATE.

IOCF - INPUT UNIT IN FID FOR COMPUTED MAGNETIC FIELD MODEL.

IDST - INDUCED FIELD COEFFICIENTS: (0) DO NOT EVALUATE
         (1) EVALUATE.

DST - DST VALUE.

LL - FID FIELD EVALUATION MODE: (-1) EVALUATE AT OLD TIME
         (0) EVALUATE (1) READ FIELD MODEL AND EVALUATE.

NAMELIST BSPLIN -

H - ARRAY CONTAINING NUMBER OF INTERNAL KNOTS FOR B-SPLINE
       FUNCTIONS FITTING X, Y, AND Z COMPONENTS, RESPECTIVELY.

NN - ARRAY CONTAINING ORDER OF B-SPLINE FUNCTIONS FITTING X,
         Y, AND Z COMPONENTS, RESPECTIVELY.

NT - ARRAY CONTAINING NUMBER OF FOURIER WAVEFORMS FITTING X,
         Y, AND Z COMPONENTS, RESPECTIVELY.

KA - B-SPLINE INTERNAL KNOT ADJUSTMENT FOR BEST FIT WITH
        RESPECT TO WEIGHTED RMS: (0) DO NOT ADJUST (1) ADJUST

ITERMX - MAXIMUM NUMBER OF ITERATIONS IN UNIVARIANT SEARCH FOR
          OPTIMUM B-SPLINE KNOT POSITIONS.

LGRMAX - MAXIMUM NUMBER OF ITERATIONS IN LAGRANGIAN INTERPOLATIVE
          SEARCH FOR BEST POSITION OF A PARTICULAR KNOT WITH
          RESPECT TO WEIGHTED RMS.

EPS - KNOT ADJUSTMENT TOLERANCE WITHIN WHICH THE KNOT POSITION
       IS CONSIDERED TO HAVE CONVERGED.

KO - BOOLEAN NUMBER IN WHICH EACH DIGIT GOVERNS THE ADJUSTMENT
       OF A PARTICULAR INTERNAL KNOT POSITION, WITH LEFT-MOST
       DIGIT CORRESPONDING TO LEFT-MOST KNOT: (0) ADJUST
       (1) DO NOT ADJUST.

IOBS - INPUT UNIT CONTAINING B-SPLINE KNOT POSITIONS, FOURIER
         WAVEFORM FREQUENCIES, AND SIGMAs FOR OBSERVED MAGNETIC
         FIELD VALUES.

NAMELIST TRFORM -
EU - FIT EULER ANGLES (DEGREES).

QI - GSFC NOMINAL BIAS CORRECTIONS IN ORIGINAL SATELLITE COORDINATES (NT).

QF - FIT MAGNETOMETER BIAS ADJUSTMENTS (NT).

CF - FIT CALIBRATION SLOPE ADJUSTMENT MATRIX.

CA - CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATES.

RF - ROTATION MATRIX FROM ORIGINAL SATELLITE TO FIT/MAGSAT COORDINATES.

RC - ROTATION MATRIX FROM FIT/MAGSAT TO DESIRED SATELLITE COORDINATES.

NAMELIST CONTRL -

IMODE - PROGRAM OPERATION MODE: (0) RAW-TO-FINAL FIT TAPE TOTAL PROCESSING (1) FILTER-TO-FINAL FIT TAPE PROCESSING (2) FILTER PROCESSING ONLY (3) RAW-TO-FILTER TAPE PROCESSING.

IFORM - ORIGINAL RAW DATA TAPE(S) FORMAT: (0) EARLY FORMAT -- 2 SAMPLES/SECOND (1) LATTER FORMAT -- 20 SAMPLES/SECOND

NDATAR - NUMBER OF DATA RECORDS PROCESSED AFTER EPHEMERIS RECORD.

INPUTF - NUMBER OF INPUT FILES TO BE PROCESSED.

IARC - ARC PROCESSING LENGTH: (0) ENTIRE ARC (1) ARC SEGMENT BETWEEN BEGINNING AND ENDING TIMES ONLY.

IYRBEG - BEGINNING ARC TIME YEAR SINCE 1900.

IDYBEG - BEGINNING ARC TIME DAY NUMBER.

ISCBEG - BEGINNING ARC TIME SECONDS.

IYREND - ENDING ARC TIME YEAR SINCE 1900.

IDYEND - ENDING ARC TIME DAY NUMBER.

ISCEND - ENDING ARC TIME SECONDS.

ORBINC - SATELLITE ORBIT INCLINATION ANGLE (DEGREES).

ERAD - MEAN EARTH RADIUS (KM).

IEPDAY - FILTER REFERENCE DAY NUMBER.

INCREM - FILTER WINDOW LENGTH (SECONDS).

INTRVL - FILTER WINDOW NUMBER FROM BEGINNING OF REFERENCE DAY.

IMETH - FILTER METHOD: (0) DETREND (1) DETREND AND FLAG OUTLIERS (2) FLAG OUTLIERS (3) NO MODIFICATION.
ISPEC - FFT SPECTRAL ANALYSIS: (0) NO ANALYSIS (1) ZERO-MEAN
ANALYSIS (2) DIRECT ANALYSIS.

NEXTIN - NUMBER OF SUCCESSIVE FILTER WINDOWS TO BE PROCESSED
DURING THIS RUN BEGINNING WITH WINDOW NUMBER "INTRVL".

IBTBS - FINAL TAPE OUTPUT COORDINATES: (0) FORMATTED TOPOCENTRIC
(1) FORMATTED/BINARY FIT/MAGSAT (2) SAME AS 1, PLUS
FORMATTED DESIRED SATELLITE.

SIGMLT - OUTLIER MULTIPLICATION FACTOR FOR TREND RESIDUAL SIGMA.

NFLAGK - DATA QUALITY FLAG RETENTION CODE FOR FILTER: EACH DIGIT
INDICATES FLAG TO BE RETAINED FOR TREND FITTING.

IOWIOF - UNIT IOW INTERVALS FOR FINAL PROCESSING: (0) INTRVL ONLY
(1) INTRVL AND PRECEEDING (2) ALL.

IOFI - OUTPUT DATA FLAG FOR UNITS IOF AND IOB: (0) DATA WILL BE
APPENDED (1) DATA WILL BE FIRST.

IODI - OUTPUT DATA FLAG FOR UNIT IOD: (0) DATA WILL BE APPENDED
(1) DATA WILL BE FIRST.

IOWIO - OUTPUT DATA FLAG FOR UNIT IOW: (0) DATA WILL BE APPENDED
(1) DATA WILL BE FIRST.

NAMELIST OUTLIM -

=========

DXOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC X
COMPONENT (NT).

DYOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Y
COMPONENT (NT).

DZOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Z
COMPONENT (NT).

DBOL - MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC B
COMPONENT (NT).

XWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT X COMPONENT.

YWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT Y COMPONENT.

ZWINDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT Z COMPONENT.

BWNDO - MAGNETIC LATITUDE TOLERANCE FOR FIT/MAGSAT B COMPONENT.

ABVLAT - FILTER GEOCENTRIC LATITUDE TOLERANCE FOR ALL COMPONENTS.

TRNLAT - GEODETIC LATITUDE ABOVE WHICH SATELLITE VELOCITY
DIRECTION IS INDETERMINABLE.

ITMGAP - TIME-GAP TOLERANCE INCREMENT FOR DATA (SECONDS).

============================================================
CHARACTER*80 TITLE

A-7
INTEGER H(3)
DIMENSION EU(3),CA(3,3),QI(3),QF(3),CF(3),RF(3,3),RC(3,3),NN(3)
DIMENSION NT(3),KA(3),ITERMX(3),LGRMAX(3),EPS(3),KO(3),SIG(3,500)
DIMENSION EKNOTS(3,500),FREQ(3,500)
NAMELIST /IDFILE/ IST1,IST2,IST3,IST4,IOR,IOF,IOD,IOB,ISC1,
  IS2,ISC3
NAMELIST /FIELD/ JJ,MM,NMX,NEXT,IOC,F,IDST,DST,LL
NAMELIST /BSPLIN/ H,NN,NT,KA,ITERMX,LGRMAX,EPS,KO,IOBS
NAMELIST /TRFORM/ EU,QI,QF,CF,CA,RF,RC
NAMELIST /CONTRL/ IMODE,IFORM,NDATAR,INPUTF,IARC,IYRBEG,IDYBEG,
  ISCBEG,IDYREND,ISCEND,ORBINC,ERAD,IEPDAY,
  INCREM,INTRVL,IMETH,ISPEC,NEXTIN,IBTBS,SIGMLT,
  NFLAGK,IOHIOF,IOFIST,IODIST,IOWIST
NAMELIST /OUTLIM/ DXOL,DYOL,DZOL,DBOL,XNINDO,YNINDO,ZHINDO,BHINDO,
  ABVLAT,TRNLAT,ITMGAP
COMMON /STFILE/ ISTI,ISTZ,ISTS,IST4
COMMON /MDFILE/ IOR,IOF,IOD,IOB,IOFIST,IODIST,IOWIST
COMMON /SCFILE/ ISCI,ISC2,ISC3
COMMON /ARCLIM/ IARC,IYRBEG,IDYBEG,ISCBEG,IDYREND,ISCEND,
  IFORM,NDATAR,INPUTF
COMMON /IFIELD/ JJ,MM,NMX,NEXT,IOC,F,IDST,DST,LL
COMMON /SPLINE/ H,NN,NT,KA,ITERMX,LGRMAX,EPS,KO,SIG,EKNOTS,FREQ
COMMON /COTRAN/ EU,CA,QI,QF,CF,RF,RC
COMMON /EPHEMS/ ORBINC,ERAD,IEPDAY,INCREM,INTRVL
COMMON /FILTMAP/ IMETH,ISPEC,IBTBS,SIGMLT,NFLAGK
COMMON /LIMITS/ DXOL,DYOL,DZOL,DBOL,XHINDO,YHINDO,ZHINDO,BHINDO,
  ABVLAT,TRNLAT,ITMGAP
DATA IOBS /22/
READ(5,100) TITLE
100 FORMAT(A80)
READ(5,CONTRL)
READ(5,IOFILE)
READ(5,BSPLIN)
READ(5,OUTLIM)
IF((IMODE.EQ.0).OR.(IMODE.EQ.3)) READ(5,FIELDP)
IF(IMODE.NE.3) READ(5,TRFORM)
WRITE(6,101)
101 FORMAT(1X,'GSFC SATELLITE MAGNETIC DATA PRE-
  PROCESSING PROGRAM ***'/IX,'SATELLITE ORBIT INCLINATION ANGLE (DEGREES)'/1X,
  'MEAN EARTH RADIUS (KM)'/1X,'FITTER REFERENCE DAY NUMBER'/1X,'FILTER WINDOW
WRITE(6,102) IMODE
102 FORMAT(1X,'PROGRAM OPERATION MODE --> IMODE = ',I1,') RA
  W TO FINAL FIT TAPE TOTAL PROCESSING'/45X,'(1) FILTER TO FINAL FIT
  TAPE PROCESSING'/45X,'(2) FILTER PROCESSING ONLY'/45X,'(3) RAM-TO-
  TAPE PROCESSING'/1X,'EARLY FORMAT -- 2 SAMPLES/SECOND'/39X,'LATTER FORMAT -- 20 SAMPLES/S
  ECOND'/1X,'F'
WRITE(6,104) TITLE
104 FORMAT(1X,'TITLE --> ',A80/)
WRITE(6,105) ORBINC,ERAD,IEPDAY,INCREM,INTRVL,IMETH,ISPEC,NEXTIN,
  IBTBS,SIGMLT,NDATAR,INPUTF
105 FORMAT(1X,'CONTROL AND EPHEMERIS INFORMATION>/1X,'ORBINC = ',F7.
  2,' SATELLITE ORBIT INCLINATION ANGLE (DEGREES)/1X,'ERAD
  = ',F7.2,' MEAN EARTH RADIUS (KM)/1X,'IEPDAY = ',I7,' F
  FILTER REFERENCE DAY NUMBER>/1X,'INCREM = ',I7,' FILTER WINDOW
LENGTH (SECONDS)'/1X,'INTRVL = ',I7,' --> FILTER WINDOW NUMBER F
FROM BEGINNING OF REFERENCE DAY'/1X,'IMETH = ',I7,' --> FILTER ME
THOD: (0) DETREND (1) DETREND AND FLAG OUTLIERS (2) FLAG OUTLI
ERS (3) NO MODIFICATION'/1X,'ISPEC = ',I7,' --> FFT SPECTRAL ANA
LYSIS: (0) NO ANALYSIS (1) ZERO-MEAN ANALYSIS (2) DIRECT ANALYS
IS'/1X,'NEXTIN = ',I7,' --> NUMBER OF SUCCESSIVE FILTER WINDOWS T
O BE PROCESSED DURING THIS RUN BEGINNING WITH WINDOW NUMBER INTRVL
'/1X,'IBTBS = ',I7,' --> FINAL TAPE OUTPUT COORDINATES: (0) FMT
TOPOCENTRIC (1) FMT/BIN FIT/MAGSAT (2) SAME AS 1, PLUS FMT DESI
RED'/1X,'SIGMLT = ',F7.5,' --> OUTLIER MULTIPLICATION FACTOR FOR
TREND RESIDUAL SIGMA'/1X,'NDATAR = ',I7,' --> NUMBER OF DATA RECO
RDS PROCESSED AFTER EPHEMERIS RECORD'/1X,'INPUTF = ',I7,' --> NUM
BER OF INPUT FILES TO BE PROCESSED)
WRITE(6,106) NFLAGE,IOFIST,IODIST,IONIST,IONIOF
106 FORMAT(1X,'NFLAGK = ',I7,' --> DATA QUALITY FLAG RETENTION CODE F
OR FILTER: EACH DIGIT INDICATES FLAG TO BE RETAINED FOR TREND FIT
ITION'/1X,'IOFIST = ',I7,' --> OUTPUT DATA FLAG FOR UNITS IOF AND
TIOB: (0) DATA WILL BE APPENDED (1) DATA WILL BE FIRST'/1X,'IODIS
RT = ',I7,' --> OUTPUT DATA FLAG FOR UNIT IOD: (0) DATA WILL BE A
PPENDED (1) DATA WILL BE FIRST'/1X,'IODIST = ',I7,' --> OUTPUT D
ATA FLAG FOR UNIT IOW: (0) DATA WILL BE APPENDED (1) DATA WILL B
E FIRST'/1X,'IONIOF = ',I7,' --> UNIT IOW INTERVALS FOR FINAL PRO
CESSING: (0) INTRVL ONLY (1) INTRVL AND PRECEDING (2) ALL'/1X)
WRITE(6,107) IARC
107 FORMAT(1X,'<SATELLITE ARC PROCESSING INFORMATION>',//1X,'IARC =
*,I5,' --> ARC PROCESSING LENGTH: (0) ENTIRE ARC (1) ARC SEGMENT
T BETWEEN BEGINNING AND ENDING TIMES ONLY'))
IF(IARC.EQ.0) WRITE(6,108)
108 FORMAT(/)
IF(IARC.EQ.1) WRITE(6,109) IYRBEG, IDYBEG,ISCBEG, IYREND, IDYEND,
ISCBEND
109 FORMAT(1X,'IYRBEG = ',I5,' --> BEGINNING TIME YEAR SINCE I900'/1X,
*IDYBEG = ',I5,' --> BEGINNING TIME DAY NUMBER'/1X,'ISCBEG = ',I
5,' --> BEGINNING TIME SECONDS'/1X,'IYREND = ',I5,' --> ENDING T
IME YEAR SINCE I900'/1X,'IDYEND = ',I5,' --> ENDING TIME DAY NUM
BER'/1X,'ISCBEND = ',I5,' --> ENDING TIME SECONDS'/)//
WRITE(6,110) ISTI,IST2,IST3,IST4,IOR, IOH,IOF,IOD, IOB,ISCI,ISC2,
ISC3
110 FORMAT(1X,'<INPUT/OUTPUT FILE INFORMATION>',//1X,'ISTI = ',I2,' -->
INPUT UNIT FOR ORIGINAL RAW DATA TAPE(S) IN STEPS'/1X,'IST2 = ',I
2,' --> INPUT UNIT IN STEP2, OUTPUT UNIT IN STEP1, MAGNETIC FIELD
D IN FIT/MAGSAT COORDINATES'/1X,'IST3 = ',I2,' --> INPUT UNIT IN
STEP3, OUTPUT UNIT IN STEP2, VELOCITY DIRECTIONS AND PADDED TIME-G
APS'/1X,'IST4 = ',I2,' --> INPUT UNIT IN STEP4, OUTPUT UNIT IN ST
EP3, MAGNETIC FIELD AND RESIDUALS IN TOPOCENTRIC COORDINATES'/1X,
IOR = ',I2,' --> FILTER INPUT UNIT, SAME AS IST4 IN OPERATION MO
DE 0 AND 3'/1X,'IOH = ',I2,' --> FILTER OUTPUT UNIT, INPUT UNIT
IN STEP5'/1X,'IOF = ',I2,' --> OUTPUT UNIT IN STEP5, FORMATTED M
AGNETIC FIELD IN FIT/MAGSAT OR TOPOCENTRIC COORDINATES DEPENDING ON
IN IBTBS VALUE'/1X,'IOD = ',I2,' --> OUTPUT UNIT IN STEP5, FORMAT
TED MAGNETIC FIELD IN DESIRED SPACECRAFT COORDINATES'/1X,'IOB = '
I2,' --> OUTPUT UNIT IN STEP5, BINARY MAGNETIC FIELD IN PROGRAM
FIT FORMAT'/1X,'ISC1 = ',I2,' --> FILTER SCRATCH UNIT'/1X,'ISC2 = '
I2,' --> FILTER SCRATCH UNIT'/1X,'ISC3 = ',I2,' --> SCRATCH U
NIT USED IN SUBPROGRAM DPINFO TO STORE VARIOUS DATA PARAMETERS'//)
WRITE(6,111) IOBS
111 FORMAT(1X,'<TREND-FIT INPUT FILE NUMBER>',//1X,'IOBS = ',I2,' -->
INPUT UNIT IN FILTER, CONTAINS KNOTS, A PRIORI FREQUENCIES, AND OB
SERVATION SIGMAS FOR EACH FIELD COMPONENT'/)
WRITE(6,112) DXOL,DYOL,DZOL,DBOL,XWINDO,YWINDO,ZWINDO,BWINDO,
WABLAT,TRNLAT,ITMGAP

112 FORMAT(IX,'<OUTLIER LIMIT INFORMATION>'//IX,'DXOL = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC X COMPONENT (NT)'/IX,'DYOL = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Y COMPONENT (NT)'/IX,'DZOL = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Z COMPONENT (NT)'/IX,'DXINDO = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC X COMPONENT (NT)'/IX,'DYINDO = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Y COMPONENT (NT)'/IX,'DZINDO = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC Z COMPONENT (NT)'/IX,'DXBOL = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC B COMPONENT (NT)'/IX,'DYBOL = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC B COMPONENT (NT)'/IX,'DXWINDO = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC W COMPONENT (NT)'/IX,'DWINDO = ',F8.2,' --> MAGNITUDE TOLERANCE FOR RESIDUAL TOPOCENTRIC W COMPONENT (NT)'/IX,'ABVLAT = ',F8.2,' --> FILTER GEOCENTRIC LATITUDE TOLERANCE FOR ALL COMPONENTS'/IX,'TRNLAT = ',F8.2,' --> GEODETIC LATITUDE ABOVE WHICH SATELLITE VELOCITY DIRECTION IS INDETERMINABLE'/IX,'ITMGAP = ',I8,' --> TIME-GAP TOLERANCE INCREMENT FOR DATA (SECONDS)'/)

IF((IMODE.EQ.0).OR.(IMODE.EQ.3)) WRITE(6,113) JJ,MM,NMX,NEXT,IOCF, WIDST,DST,LL

113 FORMAT(IX,'<INPUT MAGNETIC FIELD PARAMETERS>'//IX,'JJ = ',I7,' --> FID INPUT POSITION COORDINATES= (0) GEODETIC (1) GEOCENTRIC'//IX,'MM = ',I7,' --> FID EQU. RADIUS AND RCP. FLATTENING= (0)'//IX,'DEFAULT AE = 6378.16 KM, FLAT = 298.25 (1) INPUT VALUES'/IX,'NMX = ',I7,' --> MAXIMUM DEGREE OF FID MODEL EVALUATION'/IX,'NEXT = ',I7,' --> EXTERNAL FIELD MODEL: (0) DO NOT EVALUATE (1) EVALUATE'/IX,'QI = ',I7,' --> INPUT UNIT IN FID FOR COMPUTED MAGNETIC FIELD MODEL'/IX,'E = ',F12.5,' QI = ',F12.5,' --> GSFC NOMINAL BIAS CORRECTIONS IN ORIGIN TOPOCENTRIC COORDINATES'/IX,'QF = ',F12.5,' CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'QF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CF = ',F12.5,' --> FIT CALIBRATION SLOPE ADJUSTMENT MATRIX'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIAS ADJUSTMENTS (NT)'/IX,'CA = ',F12.5,' --> CALIBRATION MATRIX IN ORIGINAL SATELLITE COORDINATE SYSTEM'/IX,'CF = ',F12.5,' --> FIT MAGNETOMETER BIA

10  NBD=NBD+1
10  READ(IOBS,115,END=20) (EKNOTS(IK,NBD),FREQ(IK,NBD),SIG(IK,NBD), *IK=1,5)
115 FORMAT(3(F7.2,F7.4,F-7.3))
GO TO 10
20 IF((IMODE.EQ.1).OR.(IMODE.EQ.2)) GO TO 30
   CALL STEP1
   CALL STEP2
   CALL STEP3
30  DO 40 INTADD=I,NEXTIN
   CALL STEP4(w60,_50)
40  IF((IMODE.EQ.2).OR.(IMODE.EQ.3)) GO TO 50
   CALL STEP5
   IOFIST=0
   IOD1ST=0
A-10
SUBROUTINE STEPI

C SUBROUTINE TO READ ORIGINAL SATELLITE MAGNETIC DATA TAPE AND TRANSFORM
C RAW MAGNETOMETER COUNTS TO MAGNETIC FIELD VALUES IN THE SPACECRAFT
C COORDINATE SYSTEM, AND ALSO PROCESS EPHEMERIS INFORMATION
C
C DATA DESCRIPTION FOR UNIT ISTI INPUT TAPE(S)
C
C IYR = YEAR - 1900
C IDAY = DAY NUMBER (JAN FIRST = 1)
C IDTIME = TIME OF EPHEMERIS RECORD (SEC U.T.)
C IALT = ALTITUDE (NAUTICAL MILES)
C GLAT = GEOGRAPHIC LATITUDE
C GLON = GEOGRAPHIC LONGITUDE
C GMMLAT = CORRECTED GEOMAGNETIC LATITUDE
C GMMLON = CORRECTED GEOMAGNETIC LONGITUDE
C XMLT = CORRECTED GEOMAGNETIC LOCAL TIME
C NS = NUMBER OF DATA RECORDS FOLLOWING EPHEMERIS RECORD
C IDSEC = TIME OF DATA RECORD (SEC U.T.)
C JD = RAW MAGNETOMETER COUNTS:
C OLD TAPE FORMAT --> 2 SAMPLES/SECOND, 3 AXES/SAMPLE
C NEW TAPE FORMAT --> 20 SAMPLES/SECOND, 3 AXES/SAMPLE
C
CHARACTER*24 FMT

DIMENSION JD(3,20),EU(3),CA(3,3),QI(3),QF(3),CF(3),RF(3,3),RC(3,3)
REAL*8 TBEG, TEND, TCUR, DAYDIV
COMMON /STFILE/ IST1, IST2, IST3, IST4
COMMON /ARCLIM/ IARC, IYRBEG, IDYBEG, ISCBEG, IYREND, IDYEND, ISCEND,
*              IFORM, NDATAR, INPUTF
COMMON /COTRAN/ EU, CA, QI, QF, CF, RF, RC

C COMPUTE BEGIN (TBEG) AND END (TEND) YEAR TIME OF SATELLITE ARC TO BE
C PROCESSED, ACCOUNTING FOR LEAP YEARS
C
DAYDIV=365.DO
IF(MOD(IYRBEG,4).EQ.0) DAYDIV=366.DO
TBEG=DBLE(IYRBEG)+(DBLE(IDYBEG)+(DBLE(ISCBEG)/86400.DO))/DAYDIV
DAYDIV=365.DO
IF(MOD(IYREND,4).EQ.0) DAYDIV=366.DO
TEND=DBLE(IYREND)+(DBLE(IDYEND)+(DBLE(ISCEND)/86400.DO))/DAYDIV

C DETERMINE THE FORMAT (FMT) OF THE INPUT TAPE(S):
C IF IFORM = 0, USE OLD TAPE FORMAT --> 2 SAMPLES/SECOND, 3 AXES/SAMPLE
C IF IFORM = 1, USE NEW TAPE FORMAT --> 20 SAMPLES/SECOND, 3 AXES/SAMPLE
C
IF(IFORM.EQ.0) FMT='(416,52X,5F10.2,I_)
IF(IFORM.EQ.1) FMT='(2I4,I6,I4,5F10.0,3X,I4)'

C COUNTER DEFINITIONS:
C
C NFREAD COUNTS NUMBER OF INPUT FILES THAT HAVE BEEN READ ON UNIT IST1
C NEPHEM COUNTS NUMBER OF EMPHEMERIS RECORDS READ
C NDRECT COUNTS TOTAL NUMBER OF DATA RECORDS READ
C NDRECP COUNTS NUMBER OF DATA RECORDS ELIGIBLE FOR PROCESSING
C NDRECA COUNTS NUMBER OF DATA RECORDS ACTUALLY PROCESSED WITHIN ARC
C
NFREAD=1
NEPHEM=0
NDRECT=0
NDRECP=0
NDRECA=0

C READ INPUT DATA FROM AN INPUTF NUMBER OF ORIGINAL TAPES ON UNIT IST1
C
10 READ(IST1,FMT,END=30) IYR, IDAY, IETIME, IALT, GLAT, GLON, GMLAT, GMLON,
* XMALT, NS
   NEPHEM=NEPHEM+1
C
C COMPUTE CURRENT YEAR TIME (TCUR) FOR THIS DATA POINT, ACCOUNTING FOR
C LEAP YEARS
C
DAYDIV=365.DO
IF(MOD(IYR,4).EQ.0) DAYDIV=366.DO
TCUR=DBLE(IYR)+(DBLE(IDAY)+(DBLE(IETIME)/86400.DO))/DAYDIV
C
C CONVERT ALTITUDE FROM NAUTICAL MILES TO KM
C
ALT=REAL(IALT)*1.853
C
C READ RAW MAGNETOMETER DATA FOR EACH TIME INCREMENT IN SPACECRAFT
C COORDINATES. IF IFORM = 0 OR 1, THEN USE OLD OR NEW TAPE FORMAT,
C RESPECTIVELY
C
IUTFLG=0
DO 20 I=1, NS
IF(IFORM.EQ.0) READ(IST1,100,END=30) IDSEC,((JD(MM,NN),NN=1,20),
*MM=1,3)
IF(IFORM.EQ.1) READ(IST1,101,END=30) IDSEC,((JD(MM,NN),MM=1,3),
*NN=1,2)
100 FORMAT(I6,4X,20I16/10X,20I16/10X,20I16/10X)
101 FORMAT(I6,5X,316,5I16,_X,316,_X)

C PROCESS FIRST NDATAR DATA RECORD AFTER EMPHEMERIS RECORD ONLY IF
C UNIVERSAL TIME OF FIRST DATA RECORD AND EPHEMERIS RECORD MATCH, THAT
C IS, IUTFLG = 0
C
NDRECT=NDRECT+1
IF(I.GT.NDATOR) GO TO 20
IF((I.EQ.1).AND.(IETIME.NE.IDSEC)) IUTFLG=1
IF(IUTFLG.EQ.1) GO TO 20
NDRECP=NDRECP+1

C IF IARC = 0, THEN PROCESS ENTIRE SATELLITE ARC TAPE
C IF IARC = 1, THEN PROCESS SATELLITE ARC BETWEEN TBEG AND TEND ONLY
C IF((IARC.EQ.1).AND.((TCUR.LT.TBEG).OR.(TCUR.GT.TEND))) GO TO 20
NDRECA=NDRECA+1

C TRANSFORM RAW SATELLITE MAGNETOMETER COUNTS INTO MAGNETIC FIELD
C COMPONENTS IN FIT (MAGSAT) SPACECRAFT COORDINATES BY PERFORMING:
C
BS=RE*CF*(RF*(CA*M+QI)-QF)

WHERE BS = MAGNETIC FIELD COMPONENTS IN FIT SPACECRAFT COORDINATES
RE = EULER ANGLE ADJUSTMENT MATRIX IN 1-3-2 ROTATION SYSTEM
CF = FIT CALIBRATION SLOPE ADJUSTMENT MATRIX
RF = ROTATION MATRIX FROM M TO BS COORDINATE SYSTEM
CA = CALIBRATION MATRIX IN ORIGINAL SPACECRAFT COORDINATES
M = RAW MAGNETOMETER COUNTS IN ORIGINAL SPACECRAFT COORDINATES
QI = GSFC NOMINAL BIAS CORRECTIONS
QF = FIT MAGNETOMETER BIAS ADJUSTMENTS

BS = (BX,BY,BZ) WHERE BX, BY, AND BZ ARE THE FIT/MAGSAT SPACECRAFT
COMPONENTS (CROSS-TRACK,RADIAL,ALONG-TRACK)

M = (XM,YM,ZM) WHERE XM, YM, AND ZM ARE THE ORIGINAL SPACECRAFT
MAGNETOMETER COMPONENTS

XM=JD(1,1)
YM=JD(2,1)
ZM=JD(3,1)

C PERFORM: P=CA*M+QI
PX=CA(1,1)*XM+CA(1,2)*YM+CA(1,3)*ZM+QI(1)
PY=CA(2,1)*XM+CA(2,2)*YM+CA(2,3)*ZM+QI(2)
PZ=CA(3,1)*XM+CA(3,2)*YM+CA(3,3)*ZM+QI(3)

C PERFORM: S=RF*P
SX=RF(1,1)*PX+RF(1,2)*PY+RF(1,3)*PZ
SY=RF(2,1)*PX+RF(2,2)*PY+RF(2,3)*PZ
SZ=RF(3,1)*PX+RF(3,2)*PY+RF(3,3)*PZ

C PERFORM: W=CF*(S-QF)
WX=(SX-QF(I))/CF(1)
WY=(SY-QF(Z))/CF(2)
WZ=(SZ-QF(S))/CF(3)

C PERFORM: BS=RE_N
C
CALL EULER(WX,WY,WZ,BX,BY,BZ)
C
WRITE EPHEMERIS AND MAGNETIC FIELD INFORMATION TO STORAGE UNIT IST2
WRITE(IST2,102) IYR,IDAY,IDSEC,ALT,GLAT,GLON,GMLAT,GMLON,BX,BY,BZ
102 FORMAT(I2,14,I6,5F7.2,3F8.1)
CONTINUE
GO TO 10

C FILE NUMBER NFREAD ON UNIT IST1 HAS JUST BEEN READ, COMPARE CURRENT
C NUMBER OF FILES READ (NFREAD) WITH TOTAL NUMBER OF FILES TO BE READ
C (INPUTF). IF ALL INPUT FILES HAVE BEEN READ, THEN RETURN TO FILTER.
C IF ADDITIONAL INPUT FILES HAVE NOT BEEN READ, THEN READ NEXT FILE
C
30 IF(NFREAD.EQ.INPUTF) GO TO 40
C
C RECORD NUMBER OF NEXT FILE TO BE READ
C
NFREAD=NFREAD+1
GO TO 10
C
DETERMINE TOTAL NUMBER OF RECORDS (NTOTR) READ ON UNIT IST1
C
40 NTOTR=NEPHEM+NDRECT
C
PRINT INPUT AND OUTPUT DATA SET INFORMATION FOR STEP1
C
WRITE(6,103) IST1,NTOTR,NEPHEM,NDRECT,NDRECP,NDRECA,IST2,NDRECA
103 FORMAT('1',***PRE-FILTER PROCESSING***'/1
**X,'INPUT DATA TYPE: RAW MAGNETOMETER COUNTS ON UNIT ',I2
**X,'TOTAL RECORDS READ = ',I5/'X,'NUMBER OF EMPHEMERIS RECORDS
**X=READ = ',I5/'X,'NUMBER OF DATA RECORDS READ = ',I5/'X,'NUMBER OF
**XDATA RECORDS ACCEPTED FOR PROCESSING = ',I5/'X,'NUMBER OF DATA RE
**XCORDS PROCESSED IN ARC SEGMENT = ',I5/'X,'OUTPUT DATA TYPE: MAGN
**XETIC FIELD COMPONENTS IN FIT/MAGSAT COORDINATES ON UNIT ',I2/'X,'*
**XTOTAL RECORDS WRITTEN = ',I5/'
RETURN
END

SUBROUTINE EULER(WX,WY,WZ,BX,BY,BZ)

C SUBROUTINE TO PERFORM EULER ANGLE ADJUSTMENT ON TEMPORARY W VECTOR
C WITH FULL ROTATION MATRIX: RE=RI*R3*R2 CORRESPONDING TO ROTATIONS
C ABOUT EULER ANGLES EU(1), EU(3), AND EU(2), RESPECTIVELY
C
DIMENSION EU(3),CA(3,3),QI(3),QF(3),CF(3),RF(3,3),RC(3,3)
COMMON /COTRAN/ EU,CA,QI,QF,CF,RF,RC
C
Determine degrees-to-radians conversion
C
DTR=3.1415926530/180.0
C
A-14
C ADJUST SIGNS OF ANGLES SUPPLIED BY PROGRAM FIT AND CONVERT TO RADIANS
C
EU1=-EU(1)*DTR
EU2=-EU(2)*DTR
EU3=EU(3)*DTR
C
C DETERMINE NEEDED TRIGONOMETRIC FUNCTIONS OF THE EULER ANGLES
C
CE1=COS(EU1)
SE1=SIN(EU1)
CE2=COS(EU2)
SE2=SIN(EU2)
CE3=COS(EU3)
SE3=SIN(EU3)
C
C PERFORM: BS=RE.NEW
C
BX=WX*(CE1*CE3)+WY*(CE1*SE3*CE2+SE1*SE2)+WZ*(CE1*SE3*SE2-SE1*CE2)
BY=WX*(-SE3)+WY*(CE3*CE2)+WZ*(CE3*SE2)
BZ=WX*(SE1*CE3)+WY*(SE1*SE3*CE2-CE1*SE2)+WZ*(SE1*SE3*SE2+CE1*CE2)
RETURN
END
SUBROUTINE STEP2
C
C SUBROUTINE TO LOCATE AND PAD TIME GAPS IN THE DATA, AND DETERMINE THE
C DIRECTION OF THE SPACECRAFT VELOCITY VECTOR
C
REAL*8 TIME,TIMED
COMMON /STFILE/ IST1,IST2,IST3,IST4
COMMON /LIMITS/ DXOL,DYOL,DZOL,DXOL,XYINDO,XYINDO,ZWINDO,BWINDO,
* ABVLAT,TRNLAT,ITMGAP
DATA IPASS /1/, MDIRO /-1/
C
C COUNTER DEFINITIONS:
C
C NTGAP COUNTS NUMBER OF PADDED TIME-GAP VALUES APPENDED TO OUTPUT DATA
C NDASC COUNTS NUMBER OF ASCENDING POINTS
C NDDSC COUNTS NUMBER OF DESCENDING POINTS
C NDTRN COUNTS NUMBER OF TURNING POINTS
C
NTGAP=0
NDASC=0
NDDSC=0
NDTRN=0
C
C INITIALLY REWIND STORAGE UNIT IST2 CREATED IN SUBROUTINE STEP1
C
REWIND IST2
READ(IST2,100) IYR,Iday,ITIM0
C
C CALL CLTIME WHEN FIRST POINT OF NEW PASS SEGMENT IS ENCOUNTERED
C
10 CALL CLTIME(GLAT0,TIMED)
20 READ(IST2,100,END=99) IYR,Iday,ITIM0,ALT,GLAT,GLON,GMLAT,GMLON,
* BX,BY,BZ
100 FORMAT(I2,I4,I6,5F7.2,3F8.1)
C
C AFTER READING NEXT DATA POINT ON UNIT IST2, DETERMINE ITS UNIVERSAL
C TIME AND COMPARE WITH UNIVERSAL TIME OF PREVIOUS POINT. IF TIME
C DIFFERENCE IS GREATER THAN ITMGAP SECONDS, THEN TIME GAP HAS OCCURRED
C AND NEW PASS SEGMENT IS INITIALIZED BY CLTIME

C
TIME=DBLE(IDAY)+DBLE(IETIME)/86400.DO
IF(TIME-TIMEO.GT.(DBLE(ITMGAP)+0.5DO)/86400.DO) GO TO 10
TIMEO=TIME

C CALCULATE DELTA LATITUDE OF PRESENT POINT. IF A TIME GAP PRESENTLY
C EXISTS BETWEEN THE PRESENT AND PREVIOUS POINT, THEN USE A FORWARD
C DIFFERENCE BETWEEN THE PRESENT AND FOLLOWING DATA POINT (CALculated
C IN CLTIME), OTHERWISE, USE A BACKWARD DIFFERENCE. IF DELTA LATITUDE
C IS NON-NEGATIVE, THEN SATELLITE IS CONSIDERED ASCENDING, IF NEGATIVE,
C THEN DESCENDING. IF LATITUDE OF PRESENT POINT IS ABOVE +TRNLAT OR
C BELOW -TRNLAT DEGREES LATITUDE, THEN VELOCITY DIRECTION CANNOT BE
C ACCURATELY DETERMINED AND SATELLITE IS CONSIDERED TO BE TURNING (IDIR)
C
DELAT=GLAT-GLATO
IF(DELAT.GE.0.0) IDIR=1
IF(DELAT.GE.0.0) MDIR=1
IF(DELAT.LT.0.0) IDIR=-1
IF(DELAT.LT.0.0) MDIR=-1
IF(ABS(GLAT).GE.TRNLAT) IDIR=0

C IF SATELLITE DIRECTION CHANGES FROM DESCENDING TO ASCENDING (MDIR),
C THEN NEW PASS HAS BEGUN. CALL PASDEN TO PROCESS PRESENT DATA POINT
C WITHIN PROPER PASS
C
IF((MDIR0.EQ.-1).AND.(MDIR.EQ.1)) IPASS=IPASS+1
CALL PASDEN(GLAT,ALT,IPASS,MDIR)

C CHECK FOR TIME GAPS BETWEEN PRESENT AND PREVIOUS POINT THAT OCCUR ON
C SAME DAY
C
IF(IDAY.EQ.IDAYO) THEN
ITIME=IETIME-IETIMO

C IF A TIME GAP OF GREATER THAN ITMGAP SECONDS IS FOUND, THEN DETERMINE
C NUMBER OF ITMGAP SECOND PADS NEEDED AND WRITE THEM OUT TO UNIT IST3 AT
C PROPER TIME INTERVALS. INOTE = 2 INDICATES A PADDED TIME GAP VALUE.
C
IF(ITIME.GT.ITMGAP) THEN
INOTE=2
IT=ITIME/ITMGAP-1
DO 30 I=IT,ITMGAP+1
NTGAP=NTGAP+1
ITIME=IETIMO+I*ITMGAP
30 WRITE(IST3,101) IYR,IDAY,ITIME,INOTE
END IF

C CHECK FOR TIME GAPS BETWEEN PRESENT AND PREVIOUS POINT THAT OCCUR ON
C DIFFERENT DAYS
C
ELSE
ITIME=86400-86400+IETIMO
C IF A TIME GAP OF GREATER THAN ITMGAP SECONDS IS FOUND, THEN DETERMINE
C NUMBER OF ITMGAP SECOND PADS NEEDED AND WRITE THEM OUT TO UNIT IST3 AT
C PROPER TIME INTERVALS. INOTE = 2 INDICATES A PADDED TIME GAP VALUE.
C
IF(ITIME.GT.ITMGAP) THEN
INOTE=2

A-16
IT=ITIME/ITMGAP-1
IDAYC=IDAY0
DO 40 I=1,IT
NTGAP=NTGAP+1
ITIME=IETIMO+I*ITMGAP
IF(ITIME.GE.86400) IDAYC=IDAY
IF(ITIME.GE.86400) ITIME=ITIME-86400
40 WRITE(IYR,IDAYC,ITIME,INOTE)
END IF
END IF
C C RESET DATA QUALITY FLAG INOTE = 0 INDICATING NO CONSTRAINTS ON DATA
C INOTE=0
C IF VELOCITY DIRECTION IS INDETERMINABLE (IDIR = 0), THEN SET INOTE = 7
C IF(IDIR.EQ.0) INOTE=7
C C WRITE OUT PRESENT DATA POINT EPHemeris, MAGNETIC FIELD, AND VELOCITY C VECTOR DIRECTION INFORMATION
C IF(IDIR.EQ.1) NDASC=NDASC+1
IF(IDIR.EQ.-1) NDDSC=NDDSC+1
IF(IDIR.EQ.0) NDTRN=NDTRN+1
WRITE(IYR,IDAYIETIME,ALT,GLAT,GLON,GMLAT,GMLON,BX,BY,
*BZ,IDIR,INOTE
C C INITIALIZATION FOR PROCESSING NEXT DATA POINT. SET PRESENT DATA POINT C PARAMETERS TO PREVIOUS DATA POINT PARAMETERS
C MDIRO=MDIR
GLATO=GLAT
IDAYO=IDAY
IETIMO=IETIME
GO TO 20
C C END OF FILE ON UNIT IST2, CALL PASDEN AT PASEND ENTRY POINT TO C COMPLETE DATA DISTRIBUTION PLOTS
C 99 CALL PASEND
C C DETERMINE TOTAL NUMBER OF RECORDS (NTOTR) READ ON UNIT IST2 C DETERMINE TOTAL NUMBER OF RECORDS (NTOTW) WRITTEN ON UNIT IST3 C NTOTR=NDASC+NDDSC+NDTRN
NTOTW=NTOTR+NTGAP
C C PRINT INPUT AND OUTPUT DATA SET INFORMATION FOR STEP2 C WRITE(6,103) IYR,IDAY,ALT,GLAT,GLON,GMLAT,GMLON,BX,BY,
*IDIR,INOTE
103 FORMAT(1X,'<STEP2 PROCESSING>'//1X,'INPUT DATA TYPE: MAGNETIC F
*XIELD COMPONENTS IN FIT/MAGSAT COORDINATES ON UNIT *.I2/'//3X,'TOTAL
*XRECORDS READ = ',I5//'X,'OUTPUT DATA TYPE: SAME AS INPUT WITH VEL
*XITY DIRECTIONS AND PADDED TIME-GAP INFORMATION APPENDED ON UNIT *
*.I2//'X,'TOTAL RECORDS WRITTEN = ',I5//'X,'NUMBER OF ASCENDING PO
*XNTS = ',I5//'X,'NUMBER OF DESCENDING POINTS = ',I5//'X,'NUMBER OF T
*XURNING POINTS = ',I5//'X,'NUMBER OF TIME-GAP POINTS = ',I5//'X)
101 FORMAT(I2,I4,I6,64X,I5)
102 FORMAT(I2,I4,I6,5F7.2,3F8.1,2I5)
SUBROUTINE CLTIME(GLAT0,TIME0)

C DETERMINES THE TIME AND DELTA LATITUDE OF THE PRESENT RECORD
C THIS ROUTINE IS CALLED FOR INITIAL STARTS AND WHEN TIME GAPS ARE
C ENCOUNTERED IN THE DATA
C
REAL*8 TIME0
COMMON /STFILE/ IST1,IST2,IST3,IST4
BACKSPACE IST2
READ(IST2,100,END=99) IYR,IDAY0,IETIMO,ALT,GLAT0
READ(IST2,100,END=99) IYR,IDAY,IETIM,ALT,GLAT
100 FORMAT(I2,I4,IG,SFT.2,3FS.1)
BACKSPACE IST2
BACKSPACE IST2
C DETERMINE DELTA LATITUDE BY FORWARD DIFFERENCE, THEN ADJUST PRESENT
C DATA POINT LATITUDE GLAT0 SO THAT PROPER DELTA SIGN WILL BE DETERMINED
C IN SUBROUTINE STEP2
C
DELAT=GLAT-GLAT0
IF(DELAT.GE.0.0) GLAT0=GLAT0-1.0
IF(DELAT.LT.0.0) GLAT0=GLAT0+1.0
TIME0=DBLE(IDAY0)+DBLE(IETIMO)/86400.DO
RETURN
99 WRITE(6,101)
101 FORMAT(/1X,'** END OF FILE IN SUBROUTINE CLTIME **')
STOP
END

SUBROUTINE PASDEP(ALAT,ALT,IPASS,MDIR)

C THIS SUBROUTINE PLOTS THE DISTRIBUTION OF DATA POINTS BY PASS, AND
C ALSO CALCULATES AVERAGE ALTITUDE AND NUMBER OF POINTS PER PASS
C
CHARACTER*1 P1(73) /73*/ , STAR /***/ , BLANK / /
LOGICAL FIRST /.TRUE./ , PRINT
DATA ALTSUM /0.0/, NUM /0/, ICNT /0/
C ON FIRST CALL SETUP THE PLOT HEADING
C
IF(FIRST) THEN
  FIRST=.FALSE.
  IPOLD=IPASS
  WRITE(6,100)
10  WRITE(6,101)
END IF
10 IF(IPOLD.EQ.IPASS) THEN
C IF PRESENT DATA POINT BELONGS TO CURRENT PASS THEN CALCULATE
C RELATIVE POSITION IN P1 ARRAY (5 POINTS/ARRAY ELEMENT) DEPENDING UPON
C VELOCITY DIRECTION. ALSO CONTINUE POINT COUNT AND ALTITUDE SUMMATION
C
IF(MDIR.EQ.1) THEN
  LAT=INT((ALAT+92.5)/5.0)+1
ELSE
  LAT=INT((92.5-ALAT)/5.0)+37
END IF
P1(LAT)=STAR
NUM=NUM+1
ALTSUM=ALTSUM+ALT
PRINT=.TRUE.
ELSE
C
C IF PRESENT DATA POINT BELONGS TO A SUCCEEDING PASS THEN CALCULATE
C AVERAGE ALTITUDE FOR LAST PASS AND PRINT LAST PASS INFORMATION
C
AVGALT=ALTSUM/REAL(NUM)
WRITE(6,102) IPOLD,P1,NUM,AVGALT
PRINT=.FALSE.
ICNT=ICNT+1
C
C IF MORE THAN 50 PASSES HAVE BEEN PRINTED ON ONE PAGE, THEN SKIP PAGE
C
IF(MOD(ICNT,50).EQ.0) THEN
WRITE(6,101)
WRITE(6,100)
WRITE(6,101)
END IF
C
C CLEAR P1 ARRAY AND RESET VARIABLES TO BEGIN PROCESSING NEW PASS
C
DO 20 I=1,73
20 P1(I)=BLANK
NUM=0
ALTSUM=0.0
IPOLD=IPASS
GO TO 10
END IF
C
C ENTRY POINT AFTER LAST PASS ON DATA TAPE, PRINT LAST PASS INFORMATION
C
ENTRY PASEND
IF(NUM.NE.0) AVGALT=ALTSUM/REAL(NUM)
IF(PRINT) WRITE(6,102) IPOLD,P1,NUM,AVGALT
IF(PRINT) WRITE(6,101)
RETURN
100 FORMAT('I','<SATELLITE-PASS DENSITY DISTRIBUTIONS>'//_2X,'TIME ---
   *',10X,'(5 DEGREES PER *)')
101 FORMAT('/X,'PASS# -9 -7 -6 -4 -3 -1 0 1 3 4 6 7 9 7 6
   * 4 3 1 0 -1 -3 -4 -6 -7 -9 << LAT #POINTS AVG ALT!/1
   *2X,'0 5 0 5 0 5 0 5 0 5 0 5 0 5 0 5 0 5 0 5 0
   *
   *5 0 5 0'//)
102 FORMAT(1X,15.6X,73A1,18X,14.2X,F9.2)
END
SUBROUTINE STEP3
C
SUBROUTINE TO TRANSFORM MAGNETIC FIELD MEASUREMENTS FROM SPACECRAFT
C TO TOPOCENTRIC COORDINATE SYSTEM, COMPUTE FIELD VALUES FROM INPUT
C MODEL, AND DETERMINE FIELD RESIDUALS (OBSERVED MINUS COMPUTED). FLAG
C DATA POINTS WHOSE RESIDUALS ARE GREATER THAN A SPECIFIED TOLERANCE
C
REAL*8 COSLAT,SINALP,COSALP,SINDEL,SADCL,CAMSD,DTR
COMMON /STFILE/ IST1,IST2,IST3,IST4
COMMON /IFIELD/ JJ,MM,NMX,NEXT,IOCF,IDST,DST
COMMON /EPHEMS/ ORBINC,ERAD,IEPDAY,INCREM,INTRVL
COMMON /LIMITS/ DXOL,DYOL,DZOL, DBOL, XWINDO,YWINDO,ZWINDO,BWINDO, ABVLAT,TRNLAT,ITMGAP
C
A-19
C CALCULATE DEGREES-TO-RADIANS CONVERSION
C
DTR=3.1415926535D0/180.D0
C
C COUNTER DEFINITIONS:
C
C NTOTR COUNTS TOTAL RECORDS READ ON UNIT IST3
C NTGAP COUNTS PADDED TIME-GAP POINTS NOT TRANSFORMED TO TOPOCENTRIC
C NDTRN COUNTS SATELLITE TURNING POINTS NOT TRANSFORMED TO TOPOCENTRIC
C NOUTX COUNTS NUMBER OF TOPOCENTRIC X GROSS-OUTLIERS
C NOUTY COUNTS NUMBER OF TOPOCENTRIC Y GROSS-OUTLIERS
C NOUTZ COUNTS NUMBER OF TOPOCENTRIC Z GROSS-OUTLIERS
C NOUTB COUNTS NUMBER OF TOPOCENTRIC B GROSS-OUTLIERS
C NTOTN COUNTS TOTAL RECORDS WRITTEN ON UNIT IST4
C
NTOTR=0
NTGAP=0
NDTRN=0
NOUTX=0
NOUTY=0
NOUTZ=0
NOUTB=0
NTOTN=0
C
C DETERMINE NEGATIVE COMPLEMENT ALPHA OF ORBIT INCLINATION ANGLE ORBINC
C
ALPHA=ORBINC-90.0
C
C TRANSFER IFIELD COMMON PARAMETERS TO ARGUMENT LIST FOR SUBROUTINE FID
C
J1=IOCF
J2=JJ
J3=MM
J4=NEXT
J5=IDST
J6=NMX
J7=LL
P1=DST
C
C REWIND STORAGE UNIT IST3 AND BEGIN TO PROCESS DATA
C
REWIND IST3
10 READ(IYR, IDAY, IETIME, ALT, GLAT, GLON, GMLAT, GMLON, *BX, BY, BZ, IDIR, INOTE)
100 FORMAT(I2, I4, I6, 5F7.2, 3F8.1, 2I5)
NTOTR=NTOTR+1
C
C IF DATA POINT IS A PADDED TIME-GAP VALUE, THEN SKIP PROCESSING
C
IF(INOTE.EQ.2) GO TO 20
C
C COMPUTE GEOCENTRIC LATITUDE AND RADIUS
C
CALL GEOCEN(GLAT, GCLAT, ALT, CALT)
C
C TRANSFORM SPACECRAFT FIELD VECTOR INTO TOPOCENTRIC MAGNETIC FIELD
C VECTOR BY PERFORMING:
C
BT=TS*BS

A-20
WHERE BT = FIELD COMPONENTS IN CARTESIAN TOPOCENTRIC COORDINATES
TS = ROTATION MATRIX FROM SPACECRAFT TO TOPOCENTRIC COORDINATES
BS = MAGNETIC FIELD COMPONENTS IN FIT SPACECRAFT COORDINATES

MATRIX TS HAS THE FOLLOWING FORM:

\[
\begin{pmatrix}
\sin(\alpha)/\cos(\text{GCLAT}) & 0 & \cos(\alpha)\sin(\delta) \\
\cos(\alpha)\sin(\delta) & 0 & -\sin(\alpha)/\cos(\text{GCLAT}) \\
0 & 1 & 0
\end{pmatrix}
\]

WHERE \(\alpha\) = NEGATIVE COMPLEMENT OF ORBIT INCLINATION
GCLAT = GEOCENTRIC LATITUDE
\(\delta\) = \(\arccos(\tan(\text{GCLAT})\tan(\alpha))\)

**BT** = (TX, TY, TZ) WHERE TX, TY, AND TZ ARE THE CONVENTIONAL TOPOCENTRIC
COMPONENTS, THAT IS, (-BTHETA, BPHI, -BRHO)

**CALCULATE SCALAR FIELD VALUE IN TOPOCENTRIC COORDINATES**

\(BB = \sqrt{BX^2 + BY^2 + BZ^2}\)

**IF VELOCITY DIRECTION CANNOT BE DETERMINED, THEN SKIP PROCESSING**

**IF** (IDIR.EQ.0) **GO TO 30**

**DETERMINE NEEDED TRIGONOMETRIC FUNCTIONS OF GCLAT, ALPHA, AND DELTA**

\(\cos(\text{LAT}) = \text{DCOS(DBLE(GCLAT)*DTR)}\)
\(\sin(\alpha) = \text{DSIN(DBLE(\alpha)*DTR)}\)
\(\cos(\alpha) = \text{DCOS(DBLE(\alpha)*DTR)}\)
\(\sin(\delta) = \text{DSIN(DACOS(DTAN(DBLE(GCLAT)*DTR)*DTAN(DBLE(\alpha)*DTR))))}\)
\(\cos(\delta) = \text{SINALP}/\cos(\text{LAT})\)
\(\sin(\delta) = \text{DSIN(DSIN(DBLE(GCLAT)*DTR)*DTAN(DBLE(\alpha)*DTR)))}\)
\(\cos(\delta) = \text{DCOS(\alpha)*\sin(\delta)\sin(\delta)}\)

**IF** (IDIR.EQ.-1) **GO TO 40**

**PERFORM TRANSFORMATION IF SATELLITE IS ASCENDING**

\(TX = BX * \cos(\delta) + BY * \sin(\delta)\)
\(TY = -BX * \sin(\delta) - BY * \cos(\delta)\)

**GO TO 50**

**PERFORM TRANSFORMATION IF SATELLITE IS DESCENDING**

\(TX = BX * \cos(\delta) - BY * \sin(\delta)\)
\(TY = -BX * \sin(\delta) + BY * \cos(\delta)\)

**50 TZ = BY**

**CALCULATE SCALAR FIELD VALUE IN SPACECRAFT COORDINATES**

\(TB = \sqrt{TX^2 + TY^2 + TZ^2}\)

**DETERMINE TIME IN YEARS FOR CURRENT DATA POINT FOR INPUT TO FID**

\(TM = 1900.0 + \text{REAL(IYR)} + (\text{REAL(IDAY}) + (\text{REAL(IETIME})/86400.0))/365.0\)

**DETERMINE THE COMPUTED FIELD VALUE FOR THIS POINT AT TIME TM USING THE**
**MODEL THAT IS INPUT ON UNIT IOCF**

CALL FID(J1, J2, J3, J4, J5, GCLAT, GLON, CALT, TM, P1, J6, J7, CX, CY, CZ, CB)

A-21
C CALCULATE RESIDUAL MAGNETIC FIELD VALUES

DX=TX-CX
DY=TY-CY
DZ=TZ-CZ
DB=TB-CB

C FLAG POINTS WHOSE RESIDUAL VALUES ARE GREATER THAN SPECIFIED VALUES FOR ANY PARTICULAR COMPONENT, USING A FLAG OF INOTE = 1. WRITE MAGNETIC FIELD AND EPHEMERIS INFORMATION TO UNIT IST4

IF((ABS(DX).GT.DXOL)) NOUTX=NOUTX+1
IF((ABS(DY).GT.DYOL)) NOUTY=NOUTY+1
IF((ABS(DZ).GT.DZOL)) NOUTZ=NOUTZ+1
IF((ABS(DB).GT.DBOL)) NOUTB=NOUTB+1
NTOTW=NTOTW+1
WRITE(IST4,101) IYR,IDAY,IE TIME,GLAT,GCLAT,GLON,GM LON,ALT,*CALT,BX,BY,BZ,BB,TX,TY,TZ,TB,DX,DY,DZ,DB,CX,CY,CZ,CB,IDIR,INOTE 101 FORMAT(12,14,16,7F7.2,4F8.1,12F8.1,2I5)
GO TO 10

C IF PADDED TIME-GAP VALUES ARE ENCOUNTERED, THEN WRITE INFORMATION TO UNIT IST4 USING A FLAG OF INOTE = 2

20 NTGAP=NTGAP+1
WRITE(IST4,102) IYR,IDAY,IE TIME,INOTE 102 FORMAT(12,14,16,214X,12F8.1,2I5)
GO TO 10

C IF VELOCITY DIRECTION CANNOT BE DETERMINED FOR THIS DATA POINT, THEN WRITE SPACECRAFT FIELD VECTOR COMPONENTS ONLY TO UNIT IST4

30 NDTRN=NDTRN+1
WRITE(IST4,103) IYR,IDAY,IE TIME,GLAT,GCLAT,GLON,GM LON,ALT,*CALT,BX,BY,BZ,BB,IDIR,INOTE 103 FORMAT(12,14,16,7F7.2,4F8.1,12B8.1,2I5)
GO TO 10

C DETERMINE POINT TOTAL OMITTED (NOMIT) FROM TOPOCENTRIC TRANSFORMATION
C DETERMINE POINT TOTAL FLAGGED (NFLAG) AS GROSS-OUTLIERS

60 NOMIT=NTGAP+NDTRN
NFLAG=NOUTX+NOUTY+NOUTZ+NOUTB

C PRINT INPUT AND OUTPUT DATA SET INFORMATION FOR STEP3
C
WRITE(6,104) IST3,NTOTR,NOMIT,NTGAP,NDTRN,IST4,NTOTW,NFLAG,NOUTX,*NOUTY,NOUTZ,NOUTB
104 FORMAT(/1X, '<STEP 3 PROCESSING>'//1X,'<INPUT DATA TYPE: FIT/MAGSAT
 * FIELD COMPONENTS WITH APPENDED VELOCITY DIRECTION/TIME-GAP INFORMATION ON UNIT ','//1X,'TOTAL RECORDS READ = ','//1X,'TOTAL RECORDS OMITTED FROM TOPOCENTRIC TRANSFORMATION = ','//1X,'TIME-GAP OM ISSIONS = ','//1X,'SATellite TURNING POINT OMISSIONS = ','//1X,'0
 *PUT DATA TYPE: MAGNETIC FIELD AND RESIDUALS IN TOPOCENTRIC COORDINATES ON UNIT ','//1X,'TOTAL RECORDS WRITTEN = ','//1X,'TOTAL GROSS-OUTLIERS = ','//1X,'TOPOCENTRIC X OUTLIERS = ','//1X,'TOPOCENTRIC Y OUTLIERS = ','//1X,'TOPOCENTRIC Z OUTLIERS = ','//1X,'TOP A-22
SUBROUTINE GEOCEN(GLAT, GCLAT, ALT, CALT)

C
C CONVERT GEODETIC LATITUDE (GLAT) AND ALTITUDE (ALT) TO GEOCENTRIC
C LATITUDE (GCLAT) AND RADIUS (CALT)
C
IMPLICIT REAL*8(A-H, O-Z)
REAL*4 GLAT, GCLAT, ALT, CALT
DTR=3.141592653D0/180.0D0

C A = EQUATORIAL RADIUS, E = OPTICAL FLATTENING, BOA = RATIO OF POLAR TO
C EQUATORIAL RADII, AN = EAST-WEST RADIUS OF CURVATURE
C
A=6378.16D0
E=1.0D0/298.25D0
BOA=1.0D0-E
AN=A/DSQRT(DCOS(GLAT*2DTR)**2+(BOA*DSIN(GLAT*DTR))**2)

C CALCULATE GCLAT AND CALT USING PYTHAGOREAN RELATIONSHIPS
C
H=ALT
top=(BOA*2*AN+H)*DSIN(GLAT*DTR)
bot=(AN+H)*DCOS(GLAT*DTR)
GCLAT=DATAN2(TOP,BOT)/DTR
CALT=DSQRT(BOT**2+TOP**2)
RETURN
END

SUBROUTINE TRANSF(PHIR, ALAMR, IDIR, RH, ANORM, VH)

C
C SUBROUTINE TO CREATE TRANSFORMATION MATRIX BETWEEN SPACECRAFT AND
C GEOCENTRIC COORDINATE SYSTEMS
C
C ORBINC = ANGLE OF ORBIT INCLINATION
C RH = SATELLITE POSITION VECTOR IN (X, Y, Z) COORDINATES
C ANORM = ORBIT NORMAL VECTOR IN (X, Y, Z) COORDINATES
C VH = SATELLITE VELOCITY VECTOR IN (X, Y, Z) COORDINATES
C PHIR = GEOCENTRIC LATITUDE OF POSITION VECTOR
C PHIN = GEOCENTRIC LATITUDE OF NORMAL VECTOR
C ALAMR = LONGITUDE OF POSITION VECTOR
C ALAMN = LONGITUDE OF NORMAL VECTOR
C IDIR = VELOCITY VECTOR DIRECTION: +1 --> ASCENDING
C 0 --> TURN AROUND
C -1 --> DESCENDING
C
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION RH(3), ANORM(3), VH(3)
REAL*4 ORBINC, ERAD
COMMON /EPHEMS/ ORBINC, ERAD, IEPDAY, INCREM, INTRVL
DTR=3.141592654D0/180.0D0

C PHIN IS THE COMPLEMENT OF THE ANGLE OF INCLINATION
C
PHIN=90.0D0-DBL(ORBINC)

C INITIALIZE POSITION, NORMAL, AND VELOCITY VECTORS FOR NEXT MATRIX
C
DO 10 I=1,3
RH(I)=0.0D0
10 CONTINUE

A-23
ANORM(I)=0.DO
10 VH(I)=0.DO
C
C IF SATELLITE IS TURNING THEN ORBIT NORMAL CANNOT BE DETERMINED
C
IF(IDIR.NE.0) GO TO 20
WRITE(6,100) PHIR,ALAMR
100 FORMAT(/1X,'ATTENTION: CANNOT FIND ORBIT NORMAL VECTOR FOR TURNING POINT AT LATITUDE: ',F7.2,' LONGITUDE: ',F7.2)
RETURN
C
C CALCULATE LONGITUDE DIFFERENCE BETWEEN POSITION AND NORMAL VECTORS
C DETERMINE WHICH QUADRANT NORMAL LONGITUDE LIES IN
C
20 ANGLE=DARCOS(-DTAN(PHIR*DTR)*DTAN(PHIN*DTR))/DTR
   IF(IDIR.EQ.1) ALAMN=ALAMR-ANGLE
   IF(IDIR.EQ.-1) ALAMN=ALAMR+ANGLE
C
C TRANSFORMATION FROM SPHERICAL TO CARTESIAN COORDINATES FOR POSITION
C AND NORMAL VECTORS
C
RH(1)=DCOS(PHIR*DTR)*DCOS(ALAMR*DTR)
RH(2)=DCOS(PHIR*DTR)*DSIN(ALAMR*DTR)
RH(3)=DSIN(PHIR*DTR)
ANORM(1)=DCOS(PHIN*DTR)*DCOS(ALAMN*DTR)
ANORM(2)=DCOS(PHIN*DTR)*DSIN(ALAMN*DTR)
ANORM(3)=DSIN(PHIN*DTR)
C
C DETERMINE VELOCITY VECTOR FROM CROSS PRODUCT OF POSITION AND NORMAL
C VECTORS: (ANORM X RH) = VH
C
VH(1)=ANORM(2)*RH(3)-RH(2)*ANORM(3)
VH(2)=ANORM(3)*RH(1)-RH(3)*ANORM(1)
VH(3)=ANORM(1)*RH(2)-RH(1)*ANORM(2)
RETURN
END
SUBROUTINE FID (IU,J,MM,NEXT,IDST,DLAT,DLONG,Q1,TM,DST,NMX,L,X,Y,Z,F)
C
C FID INPUT PARAMETERS:
C
C J.EQ.0 INPUTS LATITUDE AND ALTITUDE (KM) RELATIVE TO
C ELLIPSOID (GEODETIC COORDINATES). OUTPUT FIELD
C COMPONENTS NORTH, EAST, VERTICAL IN GEODETIC
C COORDINATES
C
J.NE.0 INPUTS LATITUDE AND LONGITUDE IN GEOCENTRIC
C COORDINATES AND GEOCENTRIC RADIUS (KM). OUTPUT FIELD
C COMPONENTS NORTH, EAST, VERTICAL IN SPHERICAL
C COORDINATES
C
MM.EQ.0 USE DEFAULT VALUES AE=6378.16, FLAT=298.25
C MM.NE.0 INPUT VALUES FOR AE AND FLAT ON FIRST CALL TO FID
C
NEXT.EQ.0 DO NOT EVALUATE EXTERNAL FIELD MODEL, DO NOT READ
C NEXT.NE.0 EVALUATE EXTERNAL FIELD MODEL, READ INPUT VALUES FOR
C EXTERNAL FIELD PARAMETERS WHEN L IS GREATER THAN 0
C
A-24
IDST .EQ. 0  DO NOT EVALUATE INDUCED COEFFICIENTS
IDST .EQ. 1  EVALUATE INDUCED COEFFICIENTS

DLAT  GEODE蒂C LATITUDE IN DEGREES WHEN J = 0
      GEOCENTRIC LATITUDE IN DEGREES WHEN J = 1

DLONG  LONGITUDE IN DEGREES

Q1  GEODE蒂C ALTITUDE (KM) WHEN J = 0
      GEOCENTRIC RADIUS (KM) WHEN J = 1

NMX  MAXIMUM DEGREE OF MODEL EVALUATION

DST  DST VALUE

NMX  MAXIMUM DEGREE AND ORDER OF CONSTANT FIELD TERMS
NMAXT  MAXIMUM DEGREE AND ORDER OF FIRST ORDER TIME TERMS
NMAXTT  MAXIMUM DEGREE AND ORDER OF SECOND ORDER TIME TERMS
NMAXTTT  MAXIMUM DEGREE AND ORDER OF THIRD ORDER TIME TERMS

K .EQ. 0  FIELD MODEL COEFFICIENTS SCHMIDT NORMALIZED
K .NE. 0  FIELD MODEL COEFFICIENTS GAUSS NORMALIZED

TZERO  EPOCH TIME FOR FIELD MODEL COEFFICIENTS

TM  TIME OF PARTICULAR FIELD EVALUATION

ABAR  MEAN RADIUS USED IN FIELD MODEL POTENTIAL EXPANSION
       (DEFAULT = 6371.2)

MODEXT .EQ. 0  NO EXTERNAL FIELD SOLVED WITH MODEL
MODEXT .NE. 0  EXTERNAL FIELD SOLVED WITH MODEL

MODIND .EQ. 0  NO INDUCED COEFFS SOLVED WITH MODEL
MODIND .NE. 0  INDUCED COEFFS SOLVED WITH MODEL

L .EQ. 0  EVALUATE FIELD
L .GT. 0  READ IN FIELD MODEL AND EVALUATE FIELD
L .LT. 0  EVALUATE FIELD AT OLD TIME

EQUIVALENCE (SHMIT(1,1),TG(1,1))
COMMON /COEFFS/TG(31,31)
COMMON /INDUCE/IIDST,ALFA1,ALFA2,ALFA3,ALFA4,DSTT
COMMON /FLDCOM/ST,CT,SPH,CPH,R,NMAX,BT,BP,BR,B,
&ABAR,E1,E2,E3,NEXTF,Q(5,5)
DIMENSION G(31,31),GT(31,31),SHMIT(31,31),AID(33)
DIMENSION GTT(8,8),GTT(31,31)
DATA IFRST/0/
DATA AE,FLAT/6378.16,298.25/
DATA TLAST/0.0/
DATA TABAR/6371.2/
IF(IFRST) 110,100,110

EQUATORIAL EARTH RADIUS AND FLATTENING FACTOR
USED IN GEODE蒂C-GEOCENTRIC COORDINATES.
THE MODEL ITSELF IS INDEPENDENT OF THOSE PARAMETERS
100 IF(MM.NE.0) READ(IU,101) AE, FLAT
101 FORMAT(1X,2F6.1)
   WRITE(6,112)
112 FORMAT(*'INPUT MAGNETIC FIELD MODEL INFORMATION'>)
   WRITE(6,109) AE, FLAT
109 FORMAT(*'/5X,'CONSTANTS USED : '/,22X,'EQUATORIAL EARTH RADIUS ',
   &F8.3,22X,'EARTH RECIPROCAL FLATTENING ',F6.1//)
   IFRST=1
   FLAT=1. -1./FLAT
   E1=0.
   E2=0.
   E3=0.
   ALFA1=0.
   ALFA2=0.
   ALFA3=0.
   ALFA4=0.
   A2=AE**2
   A4=AE**4
   B2=(AE*FLAT)**2
   A2B2=A2*(1.-FLAT**2)
   A4B4=A4*(1.-FLAT**4)
110 IF (L) 19,1,2
1 IF (TM-TLAST) 17,19,17
2 READ (IU,3) NMAX, NMAXT, NMAXTT, NMXTTTT, MODEXT, K, TZERO, ABAR, MODIND,
   &(AID(I),I=1,13)
3 FORMAT(421,212,2F6.1,I2,12A4,A2)
   IF(ABAR.EQ.0.) ABAR=TABAR
   READ(IU,103) (AID(I),I=I,14,33)
103 FORMAT(20A4)
   L=0
   WRITE (6,104) (AID(I),I=I,13)
104 FORMAT (25X,12A4,2X/5X,20A4//)
   WRITE(6,105) NHAX, NMAXT, NMAXTT, NMXTTTT, MODEXT, K, TZERO, ABAR, NEXT
105 FORMAT(5X,'EXTERNAL FIELD SOLVED WITH MODEL ( 0-NO; .GT.0-DEGREE)',I2,/
   5X,'NORMALIZATION ( K=0-SCHMIDT ; K.NE.0-GAUSS)' ,I2,/
   5X,'FIELD MODEL EPOCH ',F6.1,/
   5X,'FIELD MODEL MEAN RADIUS ',F6.1,/
   5X,'EVALUATE EXTERNAL FIELD TO DEGREE',I2//)
   MAXN=0
   TEMP=0.
5 READ (IU,6) N,M,GNM, HNM, GTNM, HTNM, GTTNM, HTTNM
6 FORMAT (2I3,6F11.4)
C N=NL+1
C M=ML+1
IF (N.LE.0) GOTO7
   MAXN=(MAX0(N,MAXN))
   G(N,M)=GNM
   GT(N,M)=GTNM
   GTT(N,M)=GTTNM
   TEMP=AMAX1(TEMP,ABS(GTNM))
IF (M.EQ.1) GOTO5
   G(M-1,N)=HNM
   GT(M-1,N)=HTNM
   GTT(M-1,N)=HTTNM
   GO TO 5
7 IF(NMXTTT.EQ.0) GO TO 107
106 READ(IU,6) N,M, GTTNNM, HTTNM
IF(N.EQ.0) GO TO 107
IF(N.GT.8) STOP 106
GTTT(N,M)=GTTTNM
IF(M.EQ.1) GO TO 106
GTTT(M-1,N)=GTTTNM
GO TO 106
107 CONTINUE
C
READ EXTERNAL FIELD

IF(MODEXT.NE.0) THEN
30 READ(IU,6) N,M,QNM,SNM
IF(N.LE.0) GO TO 31
Q(N,M) = QNM
IF(M.EQ.1) GO TO 30
Q(M-1,N) = SNM
GO TO 30
END IF
31 CONTINUE
IF(MODIND.NE.0.AND.IDST.NE.0) READ(IU,102) ALFA1, ALFA2, ALFA3, ALFA4

102 FORMAT(6X,4F11.4)
WRITE6(6,8)
8 FORMAT(6HO N M, 6X, 1HG, 10X, 1HH, 9X, 2HG, 9X, 2HHT, 8X, 3HGTT, 
. 8X, 3HMTT, 7X, 4HGTTT, 7X, 4HHTT//)
DO 12 N=2, MAXN
DO 12 M=I,N
MI=M-1
IF (M.EQ.1) GOTO10
IF(N.GT.MNXTTT) WRITE(6,9) N,M,G(N,M),G(MI,N), 
.GT(N,M), GT(MI,N), GTT(N,M), GTT(MI,N)
IF(N.LE.MNXTTT) WRITE(6,9) N,M,G(N,M),G(MI,N), 
.GT(N,M), GT(MI,N), GTT(N,M), GTT(MI,N)
9 FORMAT(2I3,8F11.4)
GO TO 12
10 CONTINUE
IF(N.GT.NMXTTT) WRITE(6,11) N,M,G(N,M),GT(N,M), 
&GT(N,M), GTT(N,M)
IF(N.LE.NMXTTT) WRITE(6,11) N,M,G(N,M),GT(N,M), 
&GT(N,M), GTT(N,M)
12 CONTINUE
IF(MODEXT.NE.0) THEN
WRITE6(6,10B)
DO 32 N=2, MODEXT
DO 32 M=1,N
IF(M.EQ.1) SNM = 0.0
IF(M.NE.1) SNM = Q(M-1,N)
WRITE6(6,6) N,M,Q(N,M),SNM
32 CONTINUE
END IF
IF(IDST.NE.0) WRITE(6,111) ALFA1, ALFA2, ALFA3, ALFA4
111 FORMAT(/5X, 'INDUCED COEFFS', 1H1)
108 FORMAT(/5X, 'HEXTFILD', //)
13 FORMAT (1H1)
IF (TEMP.EQ.0.) L=-1
14 IF (K.NE.0) GOTO17
SHMIT(1,1)=-1.
DO 15 N=2, MAXN
SHMIT(N,1)=SHMIT(N-1,1)*FLOAT(2*N-3)/FLOAT(N-1)
SHMIT(1,N)=0.
JJ=2
DO 15 M=2,N

A-27
C300  FORMAT(' FID SCHMIT')
DO 16 N=2,MAXN
DO 16 M=1,N
G(N,M)=G(N,M)*SHMIT(N,M)
GT(N,M)=GT(N,M)*SHMIT(N,M)
GTT(N,M)=GTT(N,M)*SHMIT(N,M)
IF(NMXTTT.GT.0.AND.N.LE.8)GTTT(N,M)=GTTT(N,M)*SHMIT(N,M)
IF (M.EQ.1) GOTO16
G(M-1,N)=G(M-1,N)*SHMIT(M-1,N)
GT(M-1,N)=GT(M-1,N)*SHMIT(M-1,N)
GTT(M-1,N)=GTT(M-1,N)*SHMIT(M-1,N)
IF(NMXTTT.GT.0.AND.N.LE.8)GTTT(M-1,N)=GTTT(M-1,N)*SHMIT(M-1,N)
16  CONTINUE
IF(MODEXT .NE. 0) THEN
   DO 33 N = 2,MDDEXT
   DO 33 M = 1,N
      Q(N,M) = Q(N,M)*SHMIT(N,M)
      IF(M .EQ. 1) GO TO 33
      Q(M-1,N) = Q(M-1,N)*SHMIT(M-1,N)
   CONTINUE
END IF
C310  FORMAT(' FID COEF')
T=TM-TZERO
DO 18 N=1,MAXN
DO 18 M=1,N
TGX=0.
THX=0.
IF(M.EQ.1) GO TO 270
IF(N.GT.NMXTTT) GO TO 210
TGX=GTTT(N,M)*T
THX=GTTT(M-1,N)*T
210  IF(N.GT.NMAXTT) GO TO 220
TGX=(TGX + GTTT(N,M))*T
THX=(THX + GTTT(M-1,N))*T
220  IF(N.GT.NMAXT) GO TO 230
TGX=(TGX + GTT(N,M))*T
THX=(THX+GT(N,M))*T
230  TGX= TGX+G(N,M)
THX=THX+G(M-1,N)
TG(N,M)=TGX
TG(M-1,N)=THX
GO TO 18
270  CONTINUE
IF(N.GT.NMXTTT) GO TO 240
   TGX=GTTT(N,M)*T
240  IF(N.GT.NMAXTT) GO TO 250
TGX=(TGX+GTT(N,M))*T
250  IF(N.GT.NMAXT) GO TO 260
TGX=(TGX+GT(N,M))*T
260  TGX= TGX+G(N,M)
TG(N,M)=TGX
18  CONTINUE
TLAST=TM
19  DLATR=DLAT/57.295795D0
SINLA=SIN(DLATR)
RLONG=DLONG/57.295795D0
CPH=COS(RLONG)
SPH=SIN(RLONG)
      IF (J.EQ.0) GOTO 20
C
      Q1 IS GEOCENTRIC RADIUS WHEN J=1
C
      R=Q1
      CT=SINLA
      GO TO 21
20
      SINLA2=SINLAM**2
C
      Q1 IS GEODETIC ALTITUDE WHEN J=0
C
      ALT=Q1
C
      COSLA2=1.-SINLA2
      DEN2=A2-A2B2*SINLA2
      DEN=SQRT(DEN2)
      FAC=((Q1*DEN)+A2)/((Q1*DEN)+B2)**2
      CT=SQRT((Q1/SQRT(FAC*COSLA2+Q1**2))**2
      R=SQRT(Q1**2+(A4-A4B4**2)/DEN2)
C
      WRITE(6,330) DLAT,DLONG,R,TM
330
      FORMAT('FID ',_F12._)
NMAX=MINO(NMX,MAXN)
NEXTF=NEXT
      DSTT=DST
      IIDST=IDST
      CALL MAGF
      Y=BP
      F=B
      IF (J) 22,23,22
22
      X=-BT
      Z=-BR
      RETURN
C
TRANSFORMS FIELD TO GEODETIC DIRECTIONS
23
      SIND=SINLA*ST-SQRT(COSLA2)*CT
      COSD=SQRT(1.0-SIND**2)
      X=-BT*COSD-BR*SIND
      Z=BT*SIND-BR*COSD
      RETURN
END
C
SUBROUTINE MAGF
DIMENSION P(31,31),DP(31,31),CONST(31,31),SP(31),CP(31),FN(31),
               FM(31),DXDQ(31,31),DXDS(31,31),DYDQ(31,31),DYDS(31,31),
               DZDQ(31,31),DZDS(31,31)
COMMON /INDUCE/ IDST,ALFAI,ALFAZ,ALFA3,ALFA4,DST
COMMON /COEFFS/ G(51,31)
COMMON /FCORE/ BRC,BTC,BPC,BC,BNEXT
COMMON /FLDCOM/ ST,CT,SPH,CPH,R,NMAX,BT,BP,BR,B,ABAR,E1,E2,E3,
               NEXT,Q(31,31)
DATA NCORE/I_/
      IF (P(1,1).EQ.1.0) GO TO 3
1
      P(1,1)=1.
      DP(1,1)=0.
      SP(1)=0.
      CP(1)=1.
      DO 2 N=2,NMAX
      FN(N)=N
      3
DO 2 M=1,N
FM(M)=M-1
2 CON(N,N)=FLOAT((N-2)*X2-(M-1)*X2)/FLOAT((2*N-3)*(2*N-5))
3 SP(2)=SPH
CP(2)=CPH
DO 4 M=3,NMAX
S(P(M))=SP(2)*CP(M-1)+CP(2)*SP(M-1)
4 CON(N,M)=FLOAT((N-2)*X2-(M-1)*X2)/FLOAT((2*N-5)*(2*N-5))
$P(2)=SPH$
CP(2)=CPH
DO _ M=3,NMAX
$P(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)$
AOR=ABAR/R
AR=AOR*X2
BTC=0.0
BPC=0.0
BRC=0.0
BT=0
BP=0
BR=0.
IF(IDST.EQ.0) GO TO 12
GBAR=G(2,1)
G(2,1)=GBAR + ALFA1*DIST
E1BAR=E1
E2BAR=E2
E3BAR=E3
E1=E1 + ALFA2*DIST
E2=E2 + ALFA3*DIST
E3=E3 + ALFA4*DIST
E1BAR = Q(2,1)
E2BAR = Q(2,2)
E3BAR = Q(1,2)
Q(2,1) = Q(2,1) + ALFA2*DIST
Q(2,2) = Q(2,2) + ALFA3*DIST
Q(1,2) = Q(1,2) + ALFA4*DIST
12 CONTINUE
DO 8 N=2,NMAX
AR=AOR*XAR
DO 8 M=1,N
IF (N-M) 6,5,6
5 P(N,N)=ST*X(N-1,N-1)
DP(N,N)=ST*XDP(N-1,N-1)+CT*X(N-1,N-1)
GO TO 7
6 P(N,M)=CT*X(N-1,M)-CON(N,M)*X(N-2,M)
C
NOTE: CON(2,1)=0
C
DP(N,M)=CT*XDP(N-1,M)-ST*X(N-1,M)-CON(N,M)*DP(N-2,M)
7 PAR=P(N,M)*XAR
IF (M.EQ.1) GO TO 9
TEMP=G(N,M)*CP(M)+G(M-1,N)*SP(M)
BP=BP+(G(N,M)*CP(M)-G(M-1,N)*CP(M))*FM(M)*PAR
GO TO 10
9 TEMP=G(N,M)*CP(M)
10 BT=BT+TEMP*DP(N,M)*XAR
BR=BR-TEMP*FN(N)*PAR
IF(N.GT.NCORE) GO TO 8
BTC=BT
BRC=BR
BPC=BP
8 CONTINUE
BP=BP/ST
BPC=BPC/ST
BNEXT=SQRT(BT*BT + BP*BP + BR*BR)

IF(NEXT.GT.0) THEN

CCC

MONO = 2
SIND = 0.0
COSD = 1.0
CX = -BT
CY = BP
CZ = -BR

C

IF(EXTFLD.EQ.0) GO TO 14

ROA = 1.0/AOR
RB = (ROA)**(2*(MONO-2)+1)
ROA2 = ROA**ROA

DO 11 N = MONO,NEXT
FNC = N-1
RB = RB*ROA2

DO 11 M = 1,N
FMC = M-1
P(N,M) = P(N,M)*RB
DP(N,M) = DP(N,M)*RB
TEMP = -FNC*P(N,M)*SIND - DP(N,M)*COSD
DXDQ(N,M) = TEMP*CP(M)
DXDS(N,M) = TEMP*SP(M)
TEMP = FMCP(N,M)/ST
DYDQ(N,M) = -TEMP*SP(M)
DYDS(N,M) = TEMP*CP(M)
TEMP = -FNC*P(N,M)*COSD + DP(N,M)*SIND
DZDQ(N,M) = TEMP*CP(M)
DZDS(N,M) = TEMP*SP(M)

IF(M.EQ.1) THEN

CX = CX + Q(N,M)*DXDQ(N,M)
CY = CY + Q(N,M)*DYDQ(N,M)
CZ = CZ + Q(N,M)*DZDQ(N,M)

ELSE

CX = CX + Q(N,M)*DXDQ(N,M) + Q(M-1,N)*DXDS(N,M)
CY = CY + Q(N,M)*DYDQ(N,M) + Q(M-1,N)*DYDS(N,M)
CZ = CZ + Q(N,M)*DZDQ(N,M) + Q(M-1,N)*DZDS(N,M)

END IF

11 CONTINUE

BRC = BRC + (-CZ - BR)
BPC = BPC + ( CY - BP)
BTC = BTC + (-CX - BT)
BT = -CX
BP = CY
BR = -CZ

CCC

END IF

B = SQRT(BT*BT + BP*BP + BR*BR)

**** B(14 - 30) ****

BC = SQRT(BTC*BTC + BRC*BRC + BPC*BPC)
BTC = BT - BTC
BPC = BP - BPC
BRC = BR - BRC
BC = B - BC

IF(IDST.EQ.0) RETURN

IF(ABS(DST).LT.1.E-4.AND.DST.NE.0.) WRITE(6,999)ST,CT,SPH,CPH,R,DST,E1

999 FORMAT(10X,5F10.3,5X,2E20.12)

C
E1 = E1BAR

C
E2 = E2BAR
C       E3=E3BAR
Q(2,1) = E1BAR
Q(2,2) = E2BAR
Q(1,2) = E3BAR
G(2,1)=GBAR
RETURN
END
SUBROUTINE EXTFLD
COMMON/FLDCOM/ST,CT,SPH,CPH,R,NMAX,BT,BP,BR,B,ABAR,E1,E2,E3
 COMMON/FCORE/BRC,BTC,BPC
T1=E1*CPH+E3*SPH
T2=E1*ST-T1*CT
T1=E1*CT+T1*ST
BR=BR-T1
BP=BP+E2*SPH-E3*CPH
BT=BT+T2
BRC=BRC - T1
BPC=BPC ÷ E2*SPH - E3*CPH
BTC=BTC + T2
RETURN
END

SUBROUTINE STEP4(w,w)
C
C SUBROUTINE TO FIT A TREND TO MAGNETIC FIELD RESIDUALS (OBSERVED MINUS
C COMPUTED) WITH A B- SPLINE AND/OR FOURIER WAVEFORMS, WITH THE OPTION OF
C FLAGGING POINTS WHOSE TREND RESIDUALS LIE OUTSIDE A GIVEN TOLERANCE
C LEVEL AND/OR DETRENDING THE ORIGINAL DATA
C
CHARACTER*1 CLABEL(3)
INTEGER H(3)
DIMENSION NN(3),NT(3),KA(3),ITERMX(3),LGRMAX(3),EPS(3),KEEPDQ(8)
DIMENSION KO(3),EKNOTS(3,500),FREQ(3,500),SIG(3,500),ICLASS(3,8)
REAL*8 BSPLX(500),BSPLY(500),V(5,500),COEF(500),D(13000)
REAL*8 GSIG(5,500),EKN(500),FRQ(500),SIGCOM(500),RESID(500)
REAL*8 TS,TF,NTRMS,AKNOT
COMMON /STFILE/ IST1,IST2,IST3,IST4
COMMON /MDFILE/ IOR,IO4,IOF,IOD,IOB,IO1ST,IO1ST,IOW1ST,IOWIOF
COMMON /SCFILE/ ISC1,ISC2,ISC3
COMMON /SPLINE/ H,NN,NT,KA,ITERMX,LGRMAX,EPS,KO,SIG,EKNOTS,FREQ
COMMON /EPHEMS/ ORBINC,ERAD,IEPDAY,INCREM,INTRVL
COMMON /FILTOP/ IMETH,ISPEC,IBTBS,SIGMLT,NFLAGK
COMMON /LIMITS/ DXOL,DYOL,DZOL,DXOL,YWINDO,ZWINDO,BWINDO,BWINDO,
                 ABVLAT,TRNLAT,ITMGAP
C
COMMON REGION BSHARE COMMUNICATES BSPLYN SUBPROGRAM INFORMATION TO
C DTREND FOR INTERPOLATION PURPOSES AND SPECT FOR SPECTRAL ANALYSES
C
COMMON /BSHARE/ TS,TF,EKN,FRQ,BSPLX,BSPLY,SIGCOM,V,COEF,D,WTRMS,
                 *       GSIG,RESID
DATA CLABEL /"X","Y","Z"/
C
C INITIALIZE ARRAY ICLASS FOR CLASSIFICATION COUNTS IN THIS INTERVAL
C
DO 1 INTROW=1,3
DO 1 INTCOL=1,8
1 ICLASS(INTROW,INTCOL)=0
C
C REWIND FILTER INPUT UNIT IOR, IF OPERATION MODES 0 OR 3 ARE USED, THEN
C IOR = IST4
C
RECORD IOR

C SET STEP4 SCRATCH UNITS TO ISC1 AND ISC2. OUTPUT SCRATCH UNIT ISC3
C STORES DATA QUALITY INFORMATION TO BE PLOTTED
C
IWS=ISC1
IRS=ISC2
IWP=ISC3

C REWIND THE STEP4 SCRATCH UNITS
C
REWIND IWS
REWIND IRS
REWIND IWP

C DECODE THE DATA QUALITY RETENTION CODE NFLAGK FOR THE FILTER. STORE
C EACH DIGIT OF THE CODE IN ARRAY KEEPDQ INDICATING DATA FLAG NUMBERS
C TO BE USED IN TREND FITTING. NFKEEP COUNTS NUMBER OF FLAGS TO BE
C RETAINED. IKEEP6 RECORDS RETENTION STATUS FOR INOTE = 6 DATA:
C
C IKEEP6 = 0 --> INOTE = 6 DATA WILL BE OMITTED
C IKEEP6 = 1 --> INOTE = 6 DATA WILL BE RETAINED
C
IKEEP=0
NFKEEP=0

C STORE RIGHT-MOST DIGIT IN NUMK AND THEN REDUCE NFLAGK
C
5 NUMK=MOD(NFLAGK,10)
IF(NUMK.EQ.6) IKEEP6=1
NFLAGK=NFLAGK/10
NFKEEP=NFKEEP+1

C PLACE NUMK IN ELEMENT NUMBER NFKEEP OF ARRAY KEEPDQ
C
KEEPDQ(NFKEEP)=NUMK

C IF NFLAGK HAS BEEN COMPLETELY DECODED, THEN EXIT THIS PROCESS
C
IF(NFLAGK.EQ.0) GO TO 10
GO TO 5

C COUNTER DEFINITIONS:
C
C NREAD IS TOTAL NUMBER OF POINTS READ BY THE FILTER
C I IS CURRENT NUMBER OF DATA POINTS FOUND WITHIN TIME INTERVAL OF
C INTEREST
C J IS CURRENT NUMBER OF DATA POINTS READ THROUGH THE END OF THE
C INTERVAL OF INTEREST
C K IS CURRENT NUMBER OF DATA POINTS WHICH WILL BE USED IN THE TREND
C FITTING PROCESS
C L IS CURRENT NUMBER OF DATA POINTS WHICH WILL BE FILTERED
C
10 NREAD=0
I=0
J=0
K=0
L=0

C BEGIN FILTERING INPUT DATA SET FROM UNIT IOR
C 15 J=J+1
20 READ(IOR,100,END=50) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
*ALT,CAI,BC,XY,BZ,BX,HY,HZ,HB,TX,TY,TZ,DX,SY,DY,DZ,DB,CX,CY,CZ,
*CB,DIRY,NOTE
100 FORMAT(I2,I4,I6,7F7.2,20F8.1,2I5)
NREAD=NREAD+I
C
C SETUP FOR DATA QUALITY CLASSIFICATION COUNTER ICLASS:
C
ICLASS(1,II) --> STATUS ON TOTAL IOR DATA SET AVAILABLE TO FILTER
ICLASS(2,II) --> STATUS ON ACTUAL INPUT DATA SET BEING FILTERED
ICLASS(3,II) --> STATUS ON ACTUAL DATA SET USED IN THE TREND FIT
C
C UPDATE QUALITY CLASSIFICATION COUNTS OF DATA SETS AS FOLLOWS:
C
ICLASS(1,1) --> COUNTS NOTE = 0
ICLASS(1,2) --> COUNTS NOTE = 1
ICLASS(1,3) --> COUNTS NOTE = 2
ICLASS(1,4) --> COUNTS NOTE = 3
ICLASS(1,5) --> COUNTS NOTE = 4
ICLASS(1,6) --> COUNTS NOTE = 5
ICLASS(1,7) --> COUNTS NOTE = 6
ICLASS(1,8) --> COUNTS NOTE = 7 (IDIR = 0)
C
C UPDATE QUALITY CLASSIFICATION COUNTS FOR ENTIRE UNIT IOR DATA SET
C
ICLASS(1,NOTE+1)=ICLASS(1,NOTE+1)+1
C
C DETERMINE RELATIVE TIME OF DATA POINT (ICTIME) WITH RESPECT TO
C BEGINNING OF EPOCH DAY (IEPDAY), THEN DETERMINE ITS TIME INTERVAL (NI)
C WITH RESPECT TO INTERVAL WIDTH (INCREM). IF CURRENT DAY (IDAY) IS
C EARLIER THAN EPOCH DAY, THEN REJECT CURRENT POINT IMMEDIATELY
C
IF(IDAY-IEPDAY.LT.0) GO TO 15
ICTIME=(IDAY-IEPDAY)*86400+IETIME
NI=INT(ICTIME/INCREM)+1
C
C COMPARE NI WITH TIME INTERVAL OF INTEREST (INTRVL)
C
IF(NI-INTRVL) 15,25,20
25 I=I+1
C
C IF NI MATCHES INTRVL, THEN EVALUATE POINT WITH RESPECT TO DATA QUALITY
C FLAGS DEFINED BELOW:
C
C NOTE = 0 --> NO LIMITATIONS OR CONSTRAINTS ON DATA
C NOTE = 1 --> GROSS-OUTLIER WITH RESPECT TO OBSERVED - COMPUTED FIELD
C NOTE = 2 --> PADDED TIME-GAP VALUE
C NOTE = 3 --> B-SPLINE FIT-OUTLIER
C NOTE = 4 --> FOURIER FIT-OUTLIER
C NOTE = 5 --> COMBINATION B-SPLINE/FOURIER FIT-OUTLIER
C NOTE = 6 --> GEOCENTRIC LATITUDE OUTLIES TOLERANCE LEVEL ABVLAT
C NOTE = 7 --> SATELLITE VELOCITY VECTOR DIRECTION IS INDETERMINABLE
C
C UPDATE QUALITY CLASSIFICATION COUNTS FOR THIS INTERVAL OF INTEREST
C
ICLASS(2,NOTE+1)=ICLASS(2,NOTE+1)+1
C
C WRITE INFORMATION FOR ALL POINTS IN THIS INTERVAL TO SCRATCH UNIT IWP
A-34
WRITE(IWP) GCLAT,IDIR,INOTE,I
C IF INOTE = 6, THEN BYPASS THIS CHECK AND EVALUATE AT NEXT CHECK
C IF(INOTE.EQ.6) GO TO 35
C CHECK DATA QUALITY FLAG INOTE AGAINST THE NFKEEP FLAGS TO BE RETAINED:
C IF INOTE MATCHES AN ELEMENT OF ARRAY KEEPDQ, THEN ACCEPT DATA POINT
C IF INOTE DOES NOT MATCH AN ELEMENT OF KEEPDQ, THEN REJECT DATA POINT
C DO 30 ICHECK=I,NFKEEP
  30 IF(INOTE.EQ.KEEPDQ(ICHECK)) GO TO 35
  GO TO 15
  35 L=L+1
C IF CURRENT POINT PASSES PREVIOUS EVALUATION, THEN IT WILL BE FILTERED
C CHECK IF GEOCENTRIC LATITUDE LIES WITHIN THE TOLERANCE LEVEL. FLAG
C POINTS WITH INOTE = 6 IF THE FOLLOWING CONDITIONS EXIST:
C GCLAT > +ABVLAT --> POLAR DATA WITH GEOCENTRIC LATITUDE ABOVE +ABVLAT
C GCLAT < -ABVLAT --> POLAR DATA WITH GEOCENTRIC LATITUDE BELOW -ABVLAT
C IF(ABS(GCLAT).GT.ABVLAT) INOTE=6
C WRITE OUT ON SCRATCH UNIT IWS INFORMATION WHICH MAY BE LATER MODIFIED.
C THIS INCLUDES POINTS THAT HAVE BEEN FLAGGED DUE TO GEOCENTRIC LATITUDE
C CONSTRAINTS, WHICH MAY BE EXCLUDED FROM THE TREND FIT
C WRITE(IWS) TX,TY,TZ, TB, INOTE, DX, DY, DZ, DB, I
C CHECK DATA QUALITY FLAG INOTE AGAINST THE NFKEEP FLAGS TO BE RETAINED:
C DO 40 ICHECK=I,NFKEEP
  40 IF(INOTE.EQ.KEEPDQ(ICHECK)) GO TO 45
  GO TO 15
C IF CURRENT POINT PASSES PREVIOUS EVALUATION, THEN IT WILL BE USED IN
C THE TREND FIT
C 45 K=K+1
C UPDATE QUALITY CLASSIFICATION COUNTS FOR DATA USED IN TREND FIT
C ICLASS(3,INOTE+I)=ICLASS(3,INOTE+I)+1
C STORE CURRENT ABSCISSA IN ARRAY BSPLX FOR INPUT TO BSPLYN SUBPROGRAM
C BSPLX(K)=DBLE(I)
C DETERMINE THE LOWER (TS) AND UPPER (TF) LIMIT OF THE ABSCISSA INTERVAL
C IF(K.EQ.1) TS=BSPLX(K)
C IF(K.EQ.1) TF=TS
C IF(BSPLX(K).GT.TF) TF=BSPLX(K)
  GO TO 15
C WHEN ALL DATA IS READ FROM UNIT IOR, THEN REWIND IOR FOR MODIFICATION
C ALSO SWITCH STORAGE INPUT AND OUTPUT UNITS
C 50 REWIND IOR
   CALL SWITCH(IWS,IRS)
C
C CALCULATE BEGINNING AND ENDING TIME IN DAYS AND SECONDS OF ARC
C SEGMENT TO BE FILTERED, THEN PRINT HEADING
C
   IBADD=(INTRVL-1)*INCREM
   IEADD=IBADD+INCREM
C
C CALCULATE BEGINNING (IBDY) AND ENDING (IEDY) DAYS
C
   IBDY=IEPDAY+INT(IBADD/86400)
   IEDY=IEPDAY+INT(IEADD/86400)
C
C CALCULATE BEGINNING (IBSC) AND ENDING (IESC) SECONDS
C
   IBSC=MOD(IBADD,86400)
   IESC=MOD(IEADD,86400)
   WRITE(6,101) IBDY,IBSC,IEDY,IESC
   101 FORMAT('I1','**'*200'**','/*1x','**STEP 4 FILTER ARC SEGMENT FROM
   **MM: ',I3,' DAYS ',I5,' SECS TO ',I3,' DAYS ',I5,' SECS **'
   '*/*1x','**'*200'**')
C
C PRINT QUALITY CLASSIFICATION STATUS OF INPUT DATA SET ON UNIT IOR
C
   WRITE(6,102)
   102 FORMAT(/1x,'<CLASSIFICATION OF INPUT DATA AVAILABLE FOR FILTERING
   **>'')
   WRITE(6,103) (ICLASS(I,ICL),ICL=1,8),I
   103 FORMAT(/6x,'FLG0',4x,'COUNT',27x,'DESCRIPTION'/1x,'INOTE = 0',4x,
   *I5,' --> NO LIMITATIONS OR CONSTRAINTS'/1x,'INOTE = 1',4x,I5,' -->
   *GROSS-OUTLIER WITH RESPECT TO OBSERVED - COMPUTED FIELD'/1x,'IN
   *OTE = 2',4x,I5,' --> PADDED TIME-GAP VALUE'/1x,'INOTE = 3',4x,I5,
   *' --> B-SPLINE FIT-OUTLIER'/1x,'INOTE = 4',4x,I5,' --> FOURIER F
   *IT-OUTLIER'/1x,'INOTE = 5',4x,I5,' --> COMBINATION B-SPLINE/FOURI
   *ER FIT-OUTLIER'/1x,'INOTE = 6',4x,I5,' --> GEOCENTRIC LATITUDE LI
   *ES OUTSIDE TOLERANCE LEVEL'/1x,'INOTE = 7',4x,I5,' --> SATELLITE
   *VELOCITY VECTOR DIRECTION IS INDETERMINABLE'/1x,'TOTAL ===== ',I
   *5,' RECORDS (EACH RECORD HAS 4 COMPONENTS: X, Y, Z, AND B)'/*)
C
C IF NO POINTS ARE FOUND WITHIN THE INTERVAL OF INTEREST, THEN TERMINATE
C
   IF(I.EQ.0) WRITE(6,104) INTRVL
   104 FORMAT(/1x,'**'*300'**','/1x,'**ATTENTION: NO POINTS WERE FOUND WITHIN INTERVAL
   **NUMBER: ',I3,' **')
   IF(I.EQ.0) RETURN 2
C
C PRINT QUALITY CLASSIFICATION STATUS FOR THIS INTERVAL OF INTEREST
C
   WRITE(6,105)
   105 FORMAT(/1x,'<FILTER INPUT DATA CLASSIFICATION>''
   WRITE(6,103) (ICLASS(2,ICL),ICL=1,8),I
C
C PRINT QUALITY CLASSIFICATION STATUS FOR DATA SET USED IN TREND FIT
C
   WRITE(6,106)
106 FORMAT(//1X,'<CLASSIFICATION OF DATA USED IN TREND FIT>')
WRITE(6,103) (ICLASS(3,ICL),ICL=1,8),K

C PLOT VARIOUS DATA PARAMETERS
C
CALL DPINFO(IWP,I)
C
C INDEPENDENTLY FILTER THE 3 TOPOCENTRIC FIELD COMPONENTS: IF LOOP = 1, THEN X; IF LOOP = 2, THEN Y; IF LOOP = 3, THEN Z
C
DO 55 LOOP=1,3
WRITE(6,107) CLABEL(LOOP)
107 FORMAT('I','<FILTER TOPOCENTRIC ','AI,',' COMPONENT>'//)

C READ FIELD COMPONENTS FROM IRS, STORE PROPER ORDINATE IN ARRAY BSPLY
C FOR INPUT TO BSPLYN SUBPROGRAM, THEN REWIND IRS FOR NEXT COMPONENT
C
IV=0
DO 60 II=1,L
READ(IRS) TX,TY,TZ,TB,INOTE,DX,DY,DZ,DB,III

C IF GEOCENTRIC LATITUDE OF CURRENT POINT LIES OUTSIDE TOLERANCE LEVEL
C (INOTE = 6), THEN CHECK ITS TREND FIT RETENTION STATUS
C
IF((INOTE.EQ.6).AND.(IKEEP6.EQ.0)) GO TO 60
IV=IV+1
IF(LOOP.EQ.1) BSPLY(IV)=DBLE(DX)
IF(LOOP.EQ.2) BSPLY(IV)=DBLE(DY)
IF(LOOP.EQ.3) BSPLY(IV)=DBLE(DZ)
60 CONTINUE
REWIND IRS
C
C TRANSFER PROPER INTERNAL KNOT INFORMATION TO ONE-DIMENSIONAL ARRAY
C EKN FOR INPUT TO BSPLYN
C
KNOTF=0
NKNOT=0
DO 65 II=1,H(LOOP)
AKNOT=DBLE(EKNOTS(LOOP,II))

C CHECK IF KNOT NUMBER II FOR THIS COMPONENT LIES WITHIN TIME DOMAIN
C (BETWEEN TS AND TF) OF THIS INTERVAL. IF IT DOES NOT, THEN OMIT THIS
C KNOT AND SET KNOTF = 1
C
IF((AKNOT.LE.TS).OR.(AKNOT.GE.TF)) KNOTF=1
IF((AKNOT.LE.TS).OR.(AKNOT.GE.TF)) GO TO 65
C
C IF KNOT LIES WITHIN TIME DOMAIN, THEN STORE IT IN THE NKNOT POSITION
C OF ARRAY EKN
C
NKNOT=NKNOT+1
EKN(NKNOT)=AKNOT
65 CONTINUE

C IF KNOT SET HAS BEEN REDUCED (KNOTF = 1), THEN PRINT INDICATION
C
IF(KNOTF.EQ.1) WRITE(6,108) NKNOT
108 FORMAT(1X,'*** ATTENTION: KNOT SET HAS BEEN REDUCED TO ',I2,' KNOTS TO CONFORM WITH DATA TIME CONSTRAINTS ***'/)
C
C Transfer proper a priori frequency and observation sigma information
C to one-dimensional arrays FRQ and SIGCOM, respectively, for BSPLYN
C
C DO 70 II=1,NT(LOOP)
70 FRQ(II)=DBLE(FREQ(LOOP,II))
   DO 75 II=1,L
75 SIGCOM(II)=DBLE(SIG(LOOP,II))
C
C Fit the residual data with a B-spline and/or Fourier waveforms using
C the BSPLYN subprogram
C
C CALL BSPLYN(TS,TF,NN(LOOP),NKNOT,NT(LOOP),0,0,0,0,2,40,
*KA(LOOP),ITERMX(LOOP),LGRMAX(LOOP),EPS(LOOP),K,KO(LOOP),EKN,FRQ,
*BSPLX,BSPLY,SIGCOM,V,COEF,D,WTRMS,GSIG,RESID,0.DO)
C
C Option: Perform spectral analysis on trend fit of magnetic component
C
C IF(ISPEC.NE.0) CALL SPECT(LOOP,K,NKNOT)
C
C Option: Flag points whose trend residuals fall outside tolerance level
C
C IF((IMETH.NE.0).AND.(IMETH.NE.3)) CALL OUTLIE(RESID,K,L,LOOP,IRS,
*IMS)
C
C Option: Detrend input magnetic field components
C
55 IF((IMETH.NE.2).AND.(IMETH.NE.3)) CALL DTREND(LOOP,K,L,IRS,IMS,
*NKNOT)
C
C Write out filtered data set to unit IOW
C
C CALL MODIFY(I,J,L,IRS,IMS)
RETURN 1
END
SUBROUTINE SPECT(LOOP,K,NKNOT)
C
C Subroutine to perform spectral analysis on trend fits of the magnetic
C field components in the frequency domain using a mixed-radix fast
C Fourier transform. Analysis may be done directly or with zero-mean
C adjustment
C
C LOGICAL IXFMT(7),IYFMT(7)
INTEGER H(3)
DIMENSION NN(3),NT(3),KA(3),ITERMX(3),LGRMAX(3),EPS(3),KO(3)
DIMENSION EKNOTS(3,500),FREQ(3,500),SIG(3,500),AMP(500),PHI(500)
DIMENSION POWER(500),PERIOD(500)
REAL BSPLY(500),BSPLX(500),V(5,500),COEF(500),D(13000)
REAL GSIG(500),EKN(500),FRQ(500),SIGCOM(500),RESID(500)
REAL Q(5,500),TS,TF,WTRMS,AREAL(500),AIMAG(500)
COMMON /FILTOP/ IMETH,ISPEC,IBTBS,SIGMLT,NFLAGK
COMMON /SPLINE/ H,NN,NT,KA,ITERMX,LGRMAX,EPS,KO,SIG,EKNOTS,FREQ
COMMON /BSHARE/ TS,TF,EKN,FRQ,BSPLY,BSPLX,SIGCOM,V,COEF,D,WTRMS,
*GSIG,RESID
C
C Set degrees-to-radians conversion
C
DTR=3.141592653D0/180.DO
C
C Determine total number of input data values (NTOTL), assuming a time
C increment of ITMGAP seconds, over the time segment from TS to TF fit
C BY THE TREND
C
ITS=NINT(TS)
ITF=NINT(TF)
NTOTL=ITF-ITS+1
C
C SET FLAG FOR ODD (IEVEN = 0) OR EVEN (IEVEN = 1) NUMBER OF DATA POINTS
C
IEVEN=0
IF(MOD(NTOTL,2).EQ.0) IEVEN=1
C
C DETERMINE HALF-INTERVAL (IHALF) OF SYMMETRIC DATA INTERVAL (NTOTL)
C
IHALF=NTOTL/2+1
C
C GENERATE AN ARRAY (AREAL) CONTAINING REAL COMPONENTS OF THE DATA AT
C EQUALLY SPACED TIME INTERVALS OF ITMGAP SECONDS OVER THE TIME SEGMENT
C FROM ITS TO ITF. IV IS CURRENT ELEMENT OF AREAL TO BE ASSIGNED AND IB
C IS CURRENT ELEMENT OF BSPLX AND COUNT OF NEXT TREND FIT VALUE TO BE
C ASSIGNED TO AREAL
C
IV=0
IB=I
SUM=0.0
DO 10 I=ITS,ITF
IV=IV+1
C
C SET IMAGINARY COMPONENT OF INPUT DATA (AIMAG) TO ZERO
C
AIMAG(IV)=0.DO
C
C DETERMINE WHETHER TREND FIT VALUE EXISTS AT CURRENT RELATIVE TIME I
C
IF(I.EQ.NINT(BSPLX(IB))) GO TO 20
C
C IF CURRENT RELATIVE TIME VALUE I DOES NOT MATCH TIME VALUE OF NEXT
C TREND FIT POINT, THEN CALL SUBPROGRAM BSPLYN TO INTERPOLATE A TREND
C FIT VALUE AT TIME I, THEN ASSIGN THIS VALUE Q(1,1) TO CURRENT ELEMENT
C OF AREAL
C
XINTRP=DBLE(I)
CALL BSPLYN(TS,TF,NN(LOOP),NKNOT,NT(LOOP),0,0,1,0,2,1,40,
*KA(LOOP),ITERMX(LOOP),LGMAX(LOOP),EPS(LOOP),K,KD(LOOP),EKN,FRQ,
*BSPLX,BSPLY,SIGCOM,Q,COEF,D,WTRMS,GSIG,RESID,XINTRP)
AREAL(IV)=Q(1,1)
GO TO 10
C
C IF CURRENT RELATIVE TIME VALUE I MATCHES TIME VALUE OF NEXT TREND FIT
C POINT (STORED IN ELEMENT IB OF BSPLX), THEN ASSIGN TREND FIT VALUE OF
C THAT POINT V(1,IB) TO CURRENT ELEMENT OF AREAL
C
20 AREAL(IV)=V(1,IB)
IB=IB+1
C
C SUM THE REAL COMPONENTS OF THE DATA
C
10 SUM=SUM+REAL(AREAL(IV))
C
C IF ISPEC = 1, DETERMINE DATA MEAN AND SUBTRACT FROM REAL COMPONENTS
C IF ISPEC = 2, DO NOT DETERMINE OR SUBTRACT DATA MEAN
IF(ISPEC.EQ.2) GO TO 30
AMEAN=SUM/REAL(NTOTL)
DO 40 IM=1,NTOTL
  40 AREAL(IM)=AREAL(IM)-AMEAN
C
C COMPUTE COMPLEX FOURIER TRANSFORM OF AN NTOTL NUMBER OF REAL, EQUALLY
C SPACED DATA COMPONENTS IN PLACE USING A MIXED-RADIX FAST FOURIER
C TRANSFORM. SUBPROGRAM FFT RETURNS REAL AND IMAGINARY COMPONENTS OF THE
C RESULTING FOURIER COEFFICIENTS IN AREAL AND AIMAG, RESPECTIVELY
C
  30 CALL FFT(AREAL,AIMAG,NTOTL,NTOTL,NTOTL,1)
C
C PRINT SPECTRAL ANALYSIS HEADINGS
C
  WRITE(6,100)
  100 FORMAT(/1X,'<SPECTRAL ANALYSIS OF TREND FIT>')
  IF(ISPEC.EQ.1) WRITE(6,101)
  101 FORMAT(/1X,'** THIS IS A ZERO-MEAN ANALYSIS **')
  IF(ISPEC.EQ.2) WRITE(6,102)
  102 FORMAT(/1X,'** THIS IS A DIRECT ANALYSIS **')
  WRITE(6,103)
  103 FORMAT(/1X,'NUM',8X,'PERIOD',16X,'AMPLITUDE',20X,'PHASE',20X,'POWER
          XR>')
C
C CALCULATE VARIOUS SPECTRA ONLY IN POSITIVE FREQUENCY DOMAIN DUE TO
C SYMMETRY CONSIDERATIONS
C
  DO 50 IK=2,IHALF
     IKM1=IK-1
  
C CALCULATE POWER SPECTRUM
C
     POWER(IKM1)=REAL(2.DO*(AREAL(IK)**2+AIMAG(IK)**2))
C
C CALCULATE PHASE SPECTRUM
C
     PHI(IKM1)=REAL(DATANZ(AIMAG(IK),AREAL(IK))/DTR)
C
C CALCULATE AMPLITUDE SPECTRUM
C
     AMP(IKM1)=REAL(DSQRT(AREAL(IK)**2+AIMAG(IK)**2))
C
C IF EVEN NUMBER OF DATA POINTS WERE ANALYZED, THEN HIGHEST FREQUENCY
C AMPLITUDE IS EQUALLY DISTRIBUTED OVER ITS CORRESPONDING POSITIVE AND
C NEGATIVE FREQUENCIES
C
     IF((IK.EQ.IHALF).AND.(IEVEN.EQ.1)) AMP(IKM1)=AMP(IKM1)/2.0
C
C CALCULATE PERIODS CORRESPONDING TO FOURIER FREQUENCIES
C
     PERIOD(IKM1)=REAL(NTOTL)/REAL(IK-1)
C
C PRINT FREQUENCY NUMBER, CORRESPONDING PERIOD, AND VARIOUS SPECTRA
C
  50 WRITE(6,104) IKM1,PERIOD(IKM1),AMP(IKM1),PHI(IKM1),POWER(IKM1)
  104 FORMAT(1X,I3,4X,F10.5,3F25.5)
C
C PLOT VARIOUS SPECTRA
LTOTL=IMHALF-1

C INITIALIZE PRINTER PLOTTING
C
CALL PLOTST(00001,1)

C DETERMINE MINIMUM AND MAXIMUM VALUES FOR PERIOD
C
CALL MAXMIN(PERIOD,LTOTL,XMIN,XMAX)

C DETERMINE PLOTTING FORMAT FOR PERIOD
C
CALL FORMAT(XMIN,XMAX,IXFMT)

C IF NPLT = 1, PLOT AMPLITUDE SPECTRUM ON LOG VERSUS LOG GRID
C IF NPLT = 2, PLOT PHASE SPECTRUM ON LINEAR VERSUS LOG GRID
C IF NPLT = 3, PLOT POWER SPECTRUM ON LOG VERSUS LOG GRID
C
DO 60 NPLT=1,3

C DETERMINE MINIMUM AND MAXIMUM VALUES FOR AMPLITUDE, PHASE, AND POWER
C DETERMINE FORMAT OF ORDINATE FOR PHASE PLOT
C
IF(NPLT.EQ.1) CALL MAXMIN(AMP,LTOTL,YMIN,YMAX)
IF(NPLT.EQ.2) CALL GRDNUM(PHI,LTOTL,YMIN,YMAX,KINT,IYFMT)
IF(NPLT.EQ.3) CALL MAXMIN(POWER,LTOTL,YMIN,YMAX)

C DETERMINE FORMAT OF ORDINATE FOR AMPLITUDE AND POWER PLOTS
C
IF(NPLT.NE.2) CALL FORMAT(YMIN,YMAX,IYFMT)

C DEFINE CARTESIAN OBJECT SPACE FOR PLOTS
C
CALL SETGRD(11.0,12.0,123.0,65.0,1)

C IF PLOTTING:
C
C AMPLITUDE --> OVERLAY CARTESIAN LOG-LOG GRID WITH TICK MARKS
C PHASE --> OVERLAY CARTESIAN SEMI-LOG GRID WITH TICK MARKS
C POWER --> OVERLAY CARTESIAN LOG-LOG GRID WITH TICK MARKS
C
IF(NPLT.EQ.2) CALL OGRID(XMIN,XMAX,9,1XFMT,2,YMIN,YMAX,KINT,IYFMT,
  $2,1)
IF(NPLT.NE.2) CALL OGRID(XMIN,XMAX,9,1XFMT,2,YMIN,YMAX,9,IYFMT,
  $2,3)

C PLOT AMPLITUDE, PHASE, AND POWER SPECTRA
C
IF(NPLT.EQ.1) CALL PLOT(PERIOD,AMP,LTOTL,' ')
IF(NPLT.EQ.2) CALL PLOT(PERIOD PHI,LTOTL,' ')
IF(NPLT.EQ.3) CALL PLOT(PERIOD,POWER,LTOTL,' ')

C PRINT HEADING
C
IF(NPLT.EQ.1) CALL HORLIN('AMPLITUDE SPECTRUM (LOG VS. LOG)',32,
  $66.0,67.0,0,0)
IF(NPLT.EQ.2) CALL HORLIN('PHASE SPECTRUM (LINEAR VS. LOG)',31,
  $66.0,67.0,0,0)
IF(NPLT.EQ.3) CALL HORLIN('POWER SPECTRUM (LOG VS. LOG)',28,
  $66.0,67.0,0,0)
SUBROUTINE FFT(A,B,NTOT,N,NSPAN,ISN)
C MULTIVARIATE COMPLEX FOURIER TRANSFORM, COMPUTED IN PLACE
C USING MIXED-RADIX FAST FOURIER TRANSFORM ALGORITHM.
C BY R. C. SINGLETON, STANFORD RESEARCH INSTITUTE, OCT. 1968
C ARRAYS A AND B ORIGINALLY HOLD THE REAL AND IMAGINARY
C COMPONENTS OF THE DATA, AND RETURN THE REAL AND
C IMAGINARY COMPONENTS OF THE RESULTING FOURIER COEFFICIENTS.
C MULTIVARIATE DATA IS INDEXED ACCORDING TO THE FORTRAN
C ARRAY ELEMENT SUCCESSOR FUNCTION, WITHOUT LIMIT
C ON THE NUMBER OF IMPLIED MULTIPLE SUBSCRIPTS.
C THE SUBROUTINE IS CALLED ONCE FOR EACH VARIATE.
C THE CALLS FOR A MULTIVARIATE TRANSFORM MAY BE IN ANY ORDER.
C NTOT IS THE TOTAL NUMBER OF COMPLEX DATA VALUES.
C N IS THE DIMENSION OF THE CURRENT VARIABLE.
C NSPAN/N IS THE SPACING OF CONSECUTIVE DATA VALUES
C WHILE INDEXING THE CURRENT VARIABLE.
C THE SIGN OF ISN DETERMINES THE SIGN OF THE COMPLEX
C EXPONENTIAL, AND THE MAGNITUDE OF ISN IS NORMALLY ONE.
C A TRI-VARIATE TRANSFORM WITH A(N1,N2,N3), B(N1,N2,N3)
C IS COMPUTED BY
C CALL FFT(A,B,N1*N2*N3,N1,N1,1)
C CALL FFT(A,B,N1*N2*N3,N2,N2,1)
C CALL FFT(A,B,N1*N2*N3,N3,N1*N2*N3,1)
C FOR A SINGLE-VARIATE TRANSFORM,
C NTOT = N = NSPAN = (NUMBER OF COMPLEX DATA VALUES), E.G.
C CALL FFT(A,B,N,N,1)
C THE DATA MAY ALTERNATIVELY BE STORED IN A SINGLE COMPLEX
C ARRAY A, THEN THE MAGNITUDE OF ISN CHANGED TO TWO TO
C GIVE THE CORRECT INDEXING INCREMENT AND A(2) USED TO
C PASS THE INITIAL ADDRESS FOR THE SEQUENCE OF IMAGINARY
C VALUES, E.G.
C CALL FFT(A,A(2),NTOT,N,NSPAN,2)
C ARRAYS AT(MAXF), CK(MAXF), BT(MAXF), SK(MAXF), AND NP(MAXP)
C ARE USED FOR TEMPORARY STORAGE. IF THE AVAILABLE STORAGE
C IS INSUFFICIENT, THE PROGRAM IS TERMINATED BY A STOP.
C *** TO CONVERT PROGRAM TO DOUBLE PRECISION, REMOVE C FROM
C FOLLOWING STATEMENTS
DOUBLE PRECISION AJP,BJP,AKP,BKP,C1,C2,C3,S1,S2,S3,CD,SD,CK,SK
DOUBLE PRECISION S72,C72,S72Z,RAD,RADF,ZERO,HALF,ONE,TWO,FIVE
MAXF MUST BE .GE. THE MAXIMUM PRIME FACTOR OF N.
MAXP MUST BE .GT. THE NUMBER OF PRIME FACTORS OF N.
IN ADDITION, IF THE SQUARE-FREE PORTION K OF N HAS TWO OR
MORE PRIME FACTORS, THEN MAXP MUST BE .GE. K-1.
DIMENSION A(1),B(1)
C ARRAY STORAGE IN NFAC FOR A MAXIMUM OF 20 FACTORS OF N.
C IF N HAS MORE THAN ONE SQUARE-FREE FACTOR, THE PRODUCT OF THE
C SQUARE-FREE FACTORS MUST BE .LE. 502
DIMENSION NFAC(20),NP(501)
C ARRAY STORAGE FOR MAXIMUM PRIME FACTOR OF 501
DIMENSION AT(501),CK(501),BT(501),SK(501)
EQUIVALENCE (I,II)
SIN(AA)=DSIN(AA)
COS(AA)=DCOS(AA)
FLOAT(I)=DFLOAT(I)
C THE FOLLOWING TWO CONSTANTS SHOULD AGREE WITH THE ARRAY DIMENSIONS.
MAXF=501
MAXP=501
IF(N.LT.2)RETURN
INC=ISN
RAD=6.28318530717958647692528676655900576
S72=RAD/5.0
C72=COS(S72)
S72=SIN(S72)
C120=SQRT(3)/2
C120=0.86602540378443864676372317075293618
IF(ISN.GE.0)GO TO 10
S72=-S72
C120=-C120
RAD=-RAD
INC=-INC
10 NT=INC*NTOT
KS=INC*NSPAN
KSPAN=KS
NN=NT-INC
JC=KS/N
RADF=RAD*FLOAT(JC)*0.5
I=0
JF=0
C DETERMINE THE FACTORS OF N
M=0
K=N
GO TO 20
15 M=M+1
NFAC(M)=4
K=K/16
20 IF(K-(K/16)*16 .EQ. 0) GO TO 15
J=3
JJ=9
GO TO 30
25 M=M+1
NFAC(M)=J
K=K/JJ
30 IF(MOD(K,JJ) .EQ. 0) GO TO 25
J=J+2
JJ=J*2
IF(JJ .LE. K) GO TO 30
IF(K .GT. 4) GO TO 40
KT=M
NFAC(M+1)=K
IF(K .NE.1) M=M+1
GO TO 80
40 IF(K-(K/4)*4 .NE. 0) GO TO 50
M=M+1
NFAC(M)=2
K=K/4
50 KT=M
J=2
60 IF(MOD(K,J) .NE. 0) GO TO 70
M=M+1
NFAC(M)=J
K=K/J

70 J=((J+1)/2)*2+1
   IF(J .LE. K) GO TO 60
80 IF(KT .EQ. 0) GO TO 100
   J=KT
90 M=M+1
NFAC(M)=NFAC(J)
   J=J-1
   IF(J .NE. 0) GO TO 90
C COMPUTE FOURIER TRANSFORM
100 SD=RADF/FLOAT(KSPAN)
    CD=2.0*SIN(SD)**2
    SD=SIN(SD+SD)
    KK=1
    I=I+1
    IF(NFAC(I) .NE. 2) GO TO 400
C TRANSFORM FOR FACTOR OF 2 (INCLUDING ROTATION FACTOR)
    KSPAN=KSPAN/2
    K1=KSPAN+2
210 K2=KK+KSPAN
    AK=A(K2)
    BK=B(K2)
    A(K2)=A(KK)-AK
    B(K2)=B(KK)-BK
    A(KK)=A(KK)+AK
    B(KK)=B(KK)+BK
    KK=K2+KSPAN
    IF(KK .LE. NN) GO TO 210
    KK=KK-NN
    IF(KK .LE. JC) GO TO 210
    IF(KK .GT. KSPAN) GO TO 800
220 CI=1.0-CD
    S1=SD
230 K2=KK+KSPAN
    AK=A(KK)-A(K2)
    BK=B(KK)-B(K2)
    A(KK)=A(KK)+A(K2)
    B(KK)=B(KK)+B(K2)
    A(K2)=CI*AK-S1*BK
    B(K2)=S1*AK+CI*BK
    KK=K2+KSPAN
    IF(KK .LT. NT) GO TO 230
    K2=KK-NT
    CI=-CI
    KK=K1-K2
    IF(KK .LT. K2) GO TO 230
    AK=CI-(CD*CI+SD*S1)
    S1=(SD*CI-CD*S1)+S1
C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
C
    CI=AK
    CI=0.5/(AK**2+S1**2)+0.5
    S1=CI*S1
    CI=CI*AK
    KK=KK+JC
    IF(KK .LT. K2) GO TO 230
    K1=K1+INC+INC
    KK=(K1-KSPAN)/2+JC
IF(KK .LE. JC+JC) GO TO 220
GO TO 100
C TRANSFORM FOR FACTOR OF 3 (OPTIONAL CODE)
   320 K1=KK+KSPAN
   K2=K1+KSPAN
   AK=A(KK)
   BK=B(KK)
   AJ=A(K1)+A(K2)
   BJ=B(K1)+B(K2)
   A(KK)=AK+AJ
   B(KK)=BK+BJ
   AK=-0.5*AJ+AK
   BK=-0.5*BJ+BK
   AJ=(A(K1)-A(K2))*S120
   BJ=(B(K1)-B(K2))*S120
   A(K1)=AK-BJ
   B(K1)=BK+AJ
   A(K2)=AK+BJ
   B(K2)=BK-AJ
   KK=K2+KSPAN
   IF(KK .LT. NN) GO TO 320
   KK=KK-NN
   IF(KK .LE. KSPAN) GO TO 320
   GO TO 700
C TRANSFORM FOR FACTOR OF 4
   400 IF(NFAC(I) .NE. 4) GO TO 600
   KSPNN=KSPAN
   KSPAN=KSPAN/4
   410 C1=1.0
      S1=0.
   420 K1=KK+KSPAN
   K2=K1+KSPAN
   K3=K2+KSPAN
   AKP=A(KK)+A(K2)
   AKM=A(KK)-A(K2)
   AJP=A(K1)+A(K3)
   AJM=A(K1)-A(K3)
   A(KK)=AKP+AJP
   AJP=AKP-AJP
   BKP=B(KK)+B(K2)
   BKM=B(KK)-B(K2)
   BJP=B(K1)+B(K3)
   BJM=B(K1)-B(K3)
   B(KK)=BKP+BJP
   BJP=BKP-BJP
   IF(ISN .LT. 0) GO TO 450
   AKP=AKM-BJM
   AKM=AKM+BJM
   BKP=BKM+AJM
   BKM=BKM-AJM
   IF(S1 .EQ. 0.0) GO TO 460
   430 A(K1)=AKP*C1-BKP*S1
      B(K1)=AKP*S1+BKP*C1
   A(K2)=AJP*C2-BJP*S2
   B(K2)=AJP*S2+BJP*C2
   A(K3)=AKM*C3-BKM*S3
   B(K3)=AKM*S3+BKM*C3
   KK=K3+KSPAN
   IF(KK .LE. NT) GO TO 420
   440 C2=C1-(CDXCL+SDXSL)
S1=(SD+C1-CD*S1)+S1
C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
C
C C1=C2
C1=0.5/(C2*C2+S1*S1)+0.5
S1=C1*S1
C1=C1*C2
C2=C1*C2-S1*S1
S2=2.0*C1*S1
S3=C2*S1+S2*C1
KK=KK-NT+JC
IF(KK .LE. KSPAN) GO TO 420
KK=KK-KSPAN+INC
IF(KK .LE. JC) GO TO 410
IF(KSPAN .EQ. JC) GO TO 800
GO TO 100

450 AKP=AKM+B JM
AKM=AKM-B JM
BKP=BKM-AJM
BKM=BKM+AJM
IF(S1 .NE. 0.0) GO TO 430

460 A(K1)=AKP
B(K1)=BKP
A(K2)=AJ P
B(K2)=BJP
A(K3)=AKM
B(K3)=BKM
KK=K3+KSPAN
IF(KK .LE. NT) GO TO 420
GO TO 440

C TRANSFORM FOR FACTOR OF 5 (OPTIONAL CODE)

510 C2=-C72*C2-S72*C2
S2=2.0*C72*S72
520 K1=KK+KSPAN
K2=K1+KSPAN
K3=K2+KSPAN
K4=K3+KSPAN
AKP=A(K1)+A(K4)
AKM=A(K1)-A(K4)
BKP=B(K1)+B(K4)
BKM=B(K1)-B(K4)
AJP=A(K2)+A(K3)
AJM=A(K2)-A(K3)
BJP=B(K2)+B(K3)
BJM=B(K2)-B(K3)
AA=A(KK)
BB=B(KK)
A(KK)=AA+AKP+AJP
B(KK)=BB+BKP+BJP
AK=AKP*C72+AJP*C2+AA
BK=BKP*C72+BJP*C2+BB
AJ=AKM*S72+AJM*S2
BJ=BKM*S72+BJM*S2
A(K1)=AK-BJ
A(K4)=AK+BJ
B(K1)=BK+AJ
B(K4)=BK-AJ
AK=AKP*C2+AJP*C72+AA
BK=BKP*C2+BJP*C72+BB

A-46
\[ \begin{align*}
AJ &= AKM_S2 - AJM_S72 \\
BJ &= BKM_S2 - BJM_S72 \\
A(K2) &= AK - BJ \\
A(K3) &= AK + BJ \\
B(K2) &= BK + AJ \\
B(K3) &= BK - AJ \\
KK &= K4 + KSPAN \\
\text{IF}(KK \text{ LT. } NN) \text{ GO TO } 520 \\
KK &= KK - NN \\
\text{IF}(KK \text{ LE. } KSPAN) \text{ GO TO } 520 \\
\text{GO TO } 700
\end{align*} \]

C TRANSFORM FOR ODD FACTORS

\begin{align*}
600 \quad K &= \text{NFAC}(I) \\
\quad & \quad \text{KSPNN} = KSPAN \\
\quad & \quad \text{KSPAN} = KSPAN / K \\
\quad & \quad \text{IF}(K \text{ EQ. 3}) \text{ GO TO } 320 \\
\quad & \quad \text{IF}(K \text{ EQ. 5}) \text{ GO TO } 510 \\
\quad & \quad \text{IF}(K \text{ EQ. JF}) \text{ GO TO } 640 \\
\quad & \quad \text{JF} = K \\
S1 &= \text{RAD} / \text{FLOAT}(K) \\
C1 &= \text{COS}(S1) \\
S1 &= \text{SIN}(S1) \\
\quad & \quad \text{IF}(JF \text{ GT. MAXF}) \text{ GO TO } 998 \\
CK(JF) &= 1.0 \\
SK(JF) &= 0.0 \\
J &= 1 \\
\quad & \quad 630 \quad CK(J) = CK(K) \times C1 + SK(K) \times S1 \\
\quad & \quad SK(J) = CK(K) \times S1 - SK(K) \times C1 \\
\quad & \quad K = K-1 \\
\quad & \quad CK(K) = CK(J) \\
\quad & \quad SK(K) = -SK(J) \\
\quad & \quad J = J+1 \\
\quad & \quad \text{IF}(J \text{ LT. K}) \text{ GO TO } 630 \\
\quad & \quad 640 \quad K1 = KK \\
\quad & \quad K2 = KK + KSPAN \\
\quad & \quad AA = A(KK) \\
\quad & \quad BB = B(KK) \\
\quad & \quad AK = AA \\
\quad & \quad BK = BB \\
\quad & \quad J = 1 \\
\quad & \quad K1 = K1 + KSPAN \\
\quad & \quad 650 \quad K2 = K2 - KSPAN \\
\quad & \quad J = J+1 \\
\quad & \quad AT(J) = A(K1) + A(K2) \\
\quad & \quad AK = AT(J) + AK \\
\quad & \quad BT(J) = B(K1) + B(K2) \\
\quad & \quad BK = BT(J) + BK \\
\quad & \quad J = J+1 \\
\quad & \quad AT(J) = A(K1) - A(K2) \\
\quad & \quad BT(J) = B(K1) - B(K2) \\
\quad & \quad K1 = K1 + KSPAN \\
\quad & \quad \text{IF}(K1 \text{ LT. K2}) \text{ GO TO } 650 \\
\quad & \quad A(KK) = AK \\
\quad & \quad B(KK) = BK \\
\quad & \quad K1 = KK \\
\quad & \quad K2 = KK + KSPAN \\
\quad & \quad J = 1 \\
\quad & \quad 660 \quad K1 = K1 + KSPAN \\
\quad & \quad K2 = K2 - KSPAN \\
\quad & \quad JJ = J
\end{align*} \]
AK=AA
BK=BB
AJ=0.0
BJ=0.0
K=1

670 K=K+1
AK=AT(K)×CK(JJ)+AK
BK=BT(K)×CK(JJ)+BK
K=K+1
AJ=AT(K)×SK(JJ)+AJ
BJ=BT(K)×SK(JJ)+BJ
JJ=JJ+J
IF(JJ .GT. JF) JJ=JJ-JF
IF(K .LT. JF) GO TO 670
K=JF-J
A(K1)=AK-BJ
B(K1)=BK+AJ
A(K2)=AK+BJ
B(K2)=BK-AJ
J=J+1
IF(J .LT. K) GO TO 660
KK=KK+KSPAN
IF(KK .LE. NN) GO TO 640
KK=KK-KSPAN

C MULTIPLY BY ROTATION FACTOR (EXCEPT FOR FACTORS OF 2 AND 4)
700 IF(I .EQ. M) GO TO 800
KK=JC+1

710 C2=1.0-CD
S1=SD
720 C1=C2
S2=S1
KK=KK+KSPAN

730 AK=A(KK)
A(KK)=C2×AK-S2×B(KK)
B(KK)=S2×AK+C2×B(KK)
KK=KK+KSPAN
IF(KK .LE. NT) GO TO 730
AK=S1×S2
S2=S1×C2+C1×S2
C2=C1×C2-AK
KK=KK-NT+KSPAN
IF(KK .LE. KSPAN) GO TO 730
C2=C1-(CD×C1+SD×S1)
S1=S1+(SD×C1-CD×S1)

C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC USED, THEY MAY
C BE DELETED.
C1=0.5/(C2×K2+S1×K2)+0.5
S1=C1×S1
C2=C1×C2
KK=KK-KSPAN+JC
IF(KK .LE. KSPAN) GO TO 720
KK=KK-KSPAN+JC+INC
IF(KK .LE. JC+JC) GO TO 710
GO TO 100

C PERMUTE THE RESULTS TO NORMAL ORDER---DONE IN TWO STAGES
C PERMUTATION FOR SQUARE FACTORS OF N
800 NP(1)=KS
IF(KT .EQ. 0) GO TO 890
K=KT+KT+1
IF(M .LT. K) K=K-1
J=1
NP(K+1)=JC

810 NP(J+1)=NP(J)/NFAC(J)
NP(K)=NP(K+1)/NFAC(J)
J=J+1
K=K-1
IF(J .LT. K) GO TO 810
K3=NP(K+1)
KSPAN=NP(2)
KK=JC+1
K2=KSPAN+1
J=1
IF(N .NE. NTOT) GO TO 850

C PERMUTATION FOR SINGLE-VARIATE TRANSFORM (OPTIONAL CODE)

820 AK=A(KK)
A(KK)=A(K2)
A(K2)=AK
BK=B(KK)
B(KK)=B(K2)
B(K2)=BK
KK=KK+INC
K2=KSPAN+K2
IF(K2 .LT. KS) GO TO 820

830 K2=K2-NP(J)
J=J+1
K2=NP(J+1)+K2
IF(K2 .GT. NP(J)) GO TO 830
J=1

840 IF(KK .LT. K2) GO TO 820
KK=KK+INC
K2=KSPAN+K2
IF(K2 .LT. KS) GO TO 840
IF(KK .LT. KS) GO TO 830
JC=K3
GO TO 890

C PERMUTATION FOR MULTIVARIATE TRANSFORM

850 K=KK+JC
860 AK=A(KK)
A(KK)=A(K2)
A(K2)=AK
BK=B(KK)
B(KK)=B(K2)
B(K2)=BK
KK=KK+INC
K2=K2+INC
IF(KK .LT. K) GO TO 860
KK=KK+KS-JC
K2=K2+KS-JC
IF(KK .LT. NT) GO TO 850
K2=K2-NT+KSPAN
KK=KK-NT+JC
IF(K2 .LT. KS) GO TO 850

870 K2=K2-NP(J)
J=J+1
K2=NP(J+1)+K2
IF(K2 .GT. NP(J)) GO TO 870
J=1
880 IF(KK .LT. K2) GO TO 850

A-49
KK = KK + JC
K2 = KSPAN + K2
IF(K2 .LT. KS) GO TO 880
IF(KK .LT. KS) GO TO 870
JC = K3

890 IF(2*KT+1 .GE. M) RETURN
KSPN = NP(KT+1)

C PERMUTATION FOR SQUARE-FREE FACTORS OF N
J = M - KT
NP(J+1) = 1
900 NP(J) = NP(J) * NP(J+1)
J = J - 1
IF(J .NE. KT) GO TO 900
KT = KT + 1
NN = NP(KT) - 1
IF(NN .GT. MAXF) GO TO 998
JJ = 0
J = 0
GO TO 906

902 JJ = JJ - K2
K2 = KK
K = K + 1
KK = NP(K)
904 JJ = KK + JJ
IF(JJ .GE. K2) GO TO 902
NP(J) = JJ
906 K2 = NP(KT)
K = KT + 1
KK = NP(K)
J = J + 1
IF(J .LE. NN) GO TO 904

C DETERMINE THE PERMUTATION CYCLES OF LENGTH GREATER THAN 1
J = 0
GO TO 914

910 K = KK
KK = NP(K)
NP(K) = -KK
IF(KK .NE. J) GO TO 910
K3 = KK

914 J = J + 1
KK = NP(J)
IF(KK .LT. 0) GO TO 914
IF(KK .NE. J) GO TO 910
NP(J) = -J
IF(J .NE. NN) GO TO 914
MAXF = INCWMAXF

C REORDER A AND B, FOLLOWING THE PERMUTATION CYCLES
GO TO 950

924 J = J - 1
IF(NP(J) .LT. 0) GO TO 924
JJ = JC

926 KSPAN = JJ
IF(JJ .GT. MAXF) KSPAN = MAXF
JJ = JJ - KSPAN
K = NP(J)
KK = JC * K + I * JJ
K1 = KK + KSPAN
K2 = 0

928 K2 = K2 + 1
AT(K2) = A(K1)
BT(K2)=B(K1)
K1=K1-INC
IF(K1 .NE. KK) GO TO 928
932 K1=KK+KSPAN
K2=K1-JC(K+NP(K))
K=-NP(K)
936 A(K1)=A(K2)
B(K1)=B(K2)
K1=K1-INC
K2=K2-INC
IF(K1 .NE. KK) GO TO 936
KK=K2
IF(K .NE. J) GO TO 932
K1=KK+KSPAN
K2=0
940 K2=K2+1
A(K1)=AT(K2)
B(K1)=BT(K2)
K1=K1-INC
IF(K1 .NE. KK) GO TO 940
IF(JJ .NE. 0) GO TO 926
950 J=K3+I
NT=NT-KSPNN
II=NT-INC+I
IF(NT .GE. 0) GO TO 926
RETURN
C ERROR FINISH, INSUFFICIENT ARRAY STORAGE
998 ISN=0
PRINT 999
999 FORMAT(44H0ARRAY BOUNDS EXCEEDED WITHIN SUBROUTINE FFT)
RETURN
SUBROUTINE REALTR(A,B,N,ISN)
C IF ISN=1, THIS SUBROUTINE COMPLETES THE FOURIER TRANSFORM
C OF 2N REAL DATA VALUES, WHERE THE ORIGINAL DATA VALUES ARE
C STORED ALTERNATELY IN ARRAYS A AND B, AND ARE FIRST
C TRANSFORMED BY A COMPLEX FOURIER TRANSFORM OF DIMENSION N.
C THE COSINE COEFFICIENTS ARE IN A(1),A(2),...A(N+1) AND
C THE SINE COEFFICIENTS ARE IN B(1),B(2),...B(N+1).
C A TYPICAL CALLING SEQUENCE IS
C CALL FFT(A,B,N,N,1)
C CALL REALTR(A,B,N,1)
C THE RESULTS SHOULD BE MULTIPLIED BY 0.5/N TO GIVE THE
C USUAL SCALING OF COEFFICIENTS.
C IF ISN=-1, THE INVERSE TRANSFORMATION IS DONE, THE FIRST STEP
C IN EVALUATING A REAL FOURIER SERIES.
C A TYPICAL CALLING SEQUENCE IS
C CALL REALTR(A,B,N,-1)
C CALL FFT(A,B,N,N,-1)
C THE RESULTS SHOULD BE MULTIPLIED BY 0.5 TO GIVE THE USUAL
C SCALING, AND THE TIME DOMAIN RESULTS ALTERNATE IN ARRAYS A
C AND B, I.E. A(1),B(1),A(2),B(2),...A(N),B(N).
C THE DATA MAY ALTERNATIVELY BE STORED IN A SINGLE COMPLEX
C ARRAY A, THEN THE MAGNITUDE OF ISN CHANGED TO TWO TO
C GIVE THE CORRECT INDEXING INCREMENT AND A(2) USED TO
C PASS THE INITIAL ADDRESS FOR THE SEQUENCE OF IMAGINARY
C VALUES, E.G.
C CALL FFT(A,A(2),N,N,2)
C CALL REALTR(A,A(2),N,2)
IN THIS CASE, THE COSINE AND SINE COEFFICIENTS ALTERNATE IN A.

BY R.C. SINGLETON, STANFORD RESEARCH INSTITUTE, OCT. 1968

*** TO CONVERT PROGRAM TO DOUBLE PRECISION, REMOVE C FROM

FOLLOWING STATEMENTS

REAL IM

DOUBLE PRECISION A,B,AA,BB,AB,BA,BI,AR,SD,SN,CN,FN,PI

DIMENSION A(1),B(1)

SIN(AA)=DSIN(AA)
FLOAT(I)=DFLOAT(I)
INC=IABS(ISN)
NK=N*INC+2
NH=NK/2

SD=3.14159265358979323846264338327950288/FLOAT(2**N)
CD=2.0*SIN(SD)**2
SN=0.0

IF(ISN.LT.0)GO TO 30
CN=1.0
A(NK-1)=A(1)
B(NK-1)=B(1)

10 DO 20 J=1,NH,INC
    K=NK-J
    AA=A(J)+A(K)
    AB=A(J)-A(K)
    BA=B(J)+B(K)
    BB=B(J)-B(K)
    RE=CN*BA+SN*AB
    IM=SN*BA-CN*AB
    B(K)=IM-BB
    B(J)=IM+BB
    A(K)=AA-RE
    A(J)=AA+RE
    AA=CN-(CD-CN+SD*SN)
    SN=(SD-CN-CD*SN)+SN

C THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C ERROR. IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE

20 CN=AA
    CN=0.5/(AA**2+SN**2)+0.5
    SN=CN*SN
20 CN=CN*AA
RETURN
30 CN=-1.0
SD=-SD
GO TO 10

END

SUBROUTINE FASTFT(N,H,ISN)

COMPUTES FAST FOURIER TRANSFORM OF 2**N POINTS

N = NUMBER OF POINTS
H = COMPLEX ARRAY OF DATA TO BE TRANSFORMED
ISN = 1 FOR DIRECT TRANSFORM, 0 FOR INVERSE TRANSFORM

DIMENSION M(20)
COMPLEX H(N),WK,A,B
VA=6.2831853070/FLOAT(N)
IF(ISN.GT.0)VA=-VA
LOG=1
K=N
1 K=K/2
M(LOG)=K
IF(K.EQ.1)GO TO 2

A-52
LOG=LOG+1
GO TO 1
2 K=0
DO 5 L=1,LOG
NB=2**(L-1)
LB=N/NB
LBH=LB/2
K=0
DO 5 IB=1,NB
V=VAX*FLOAT(K)
WK=CMPLX(COS(V),SIN(V))
IS=LB*(IB-1)
DO 5 I=1,LBH
J=IS+I
JH=J+LBH
A=H(J)
B=H(JH)*WK
H(JH)=A-B
3 H(J)=A+B
DO 6 I=2,LOG
IF(K.LT.M(I))GO TO 5
4 K=K-M(I)
5 K=K+M(I)
K=0
DO 8 J=1,N
IF(K.LT.J)GO TO 6
A=H(J)
H(J)=H(K+I)
H(K+I)=A
6 DO 7 I=1,LOG
IF(K.LT.M(I))GO TO 8
7 K=K-M(I)
8 K=K+M(I)
IF(ISN.GT.O)RETURN
A=CMPLX(1./FLOAT(N),0.)
DO 9 I=1,N
9 H(I)=H(I)*A
RETURN
END
SUBROUTINE OUTLIE(RESID,K,L,LOOP,IRS,INS)
C
C SUBROUTINE TO COMPUTE STATISTICS ON B-SPLINE AND/OR FOURIER TREND FIT
C RESIDUAL VECTOR, WHERE K IS THE VECTOR LENGTH. THEN FLAG POINTS WHOSE
C TREND RESIDUALS LIE OUTSIDE A SPECIFIED MULTIPLE (SIGMLT) OF THE TEND
C FIT RESIDUAL SIGMA (RSIGMA)
C
INTEGER H(3)
DIMENSION NN(3),NT(3),KA(3),ITERMX(3),LGRMAX(3),EPS(3),KO(3)
DIMENSION EKNOTS(3,500),FREQ(3,500),SIG(3,500)
REAL*8 RESID(500)
COMMON /FILT/ IMETH,ISPEC,IBTBS,SIGMLT,NFLAGK
COMMON /SPLINE/ H,NN,NT,KA,ITERMX,LGRMAX,EPS,KO,SIG,EKNOTS,FREQ
C
C CALCULATE TREND RESIDUAL MEAN
C
RMEAN=0.0
DO 10 IB=1,K
10 RMEAN=RMEAN+RESID(IB)
RMEAN=RMEAN/REAL(K)
C
A-53
C CALCULATE TREND RESIDUAL SIGMA
C
RSIGMA=0.0
DO 20 IB=1,K
20 RSIGMA=RSIGMA+(RESID(IB)-RMEAN)**2
RSIGMA=SQRT(RSIGMA/REAL(K-1))
C
C CALCULATE THE TOLERANCE LEVEL FOR TREND RESIDUALS AND PRINT HEADING
C
TOLER=RSIGMA*SIGMLT
WRITE(6,100) TOLER
100 FORMAT(//1X,'<RESIDUALS FOUND OUTSIDE SIGMA TOLERANCE LEVEL OF: ',
      *F15.5,' >'/1X,'NUM',6X,'TIME',14X,'RESIDUAL',6X,'FLAG'/)
C
C BEGIN PROCESSING RESIDUALS AND READING DATA QUALITY INFORMATION ON
C SCRATCH INPUT UNIT IRS
C
ID=0
KALT=0
DO 30 IB=1,L
   READ(IRS) TX,TY,TZ,TB,INOTE,DX,DY,DZ,DB,I
C
C IF GEOCENTRIC LATITUDE OF CURRENT POINT LIES OUTSIDE TOLERANCE LEVEL
C (INOTE = 6), THEN DO NOT INVOLVE POINT IN RESIDUAL OUTLIER CHECK
C
   IF(INOTE.EQ.6) GO TO 30
C
C ID IS ELEMENT NUMBER OF CURRENT RESIDUAL TO BE CHECKED
C
   ID=ID+1
C
C IF MAGNITUDE OF TREND RESIDUAL LIES OUTSIDE THE TOLERANCE LEVEL, RESET
C THE DATA QUALITY FLAG INOTE USING THE FOLLOWING CRITERIA:
C
   IF(ABS(RESID(ID)).LE.TOLER) GO TO 30
C
C DATA POINT HAS BEEN FOUND OUTSIDE THE TOLERANCE LEVEL. PRINT THE
C RESIDUAL, ITS SEQUENTIAL OUTLIER NUMBER (KALT), ITS TIME (I), AND
C THE ASSIGNED DATA QUALITY FLAG (INOTE)
C
   KALT=KALT+1
   RTIME=REAL(I)
   IF(NN(LOOP).GT.0) INOTE=3
   IF(NT(LOOP).GT.0) INOTE=4
   IF((NN(LOOP).GT.0).AND.(NT(LOOP).GT.0)) INOTE=5
   WRITE(6,101) KALT,RTIME,RESID(ID),INOTE
101 FORMAT(1X,IS,ZX,FB.3,TX, F15.5,6X,I_)
C
C WRITE DATA QUALITY INFORMATION BACK OUT TO SCRATCH UNIT IWS
C
30 WRITE(IWS) TX,TY,TZ,TB,INOTE,DX,DY,DZ,DB,I
C
C SWITCH SCRATCH INPUT AND OUTPUT UNITS FOR NEXT DATA MODIFICATION
C
   CALL SWITCH(IWS,IRS)
   RETURN

A-54
SUBROUTINE DTREND(LOOP,K,L,IRS,IHS,NKNOT)

C SUBROUTINE TO DETREND THE OBSERVED GEOCENTRIC MAGNETIC DATA, THAT IS,
C SUBTRACT THE TREND FIT OF THE RESIDUALS (DX,DY,DZ)

C INTEGER H(3)
DIMENSION NN(3),NT(3),KA(3),ITERMX(3),LGRMAX(3),EPS(3),KO(3)
DIMENSION EKNOTS(3,500),FREQ(3,500),SIG(3,500)
REAL BSPLX(500),BSPLY(500),V(5,500),COEF(500),D(15000)
REAL GSIG(5,500),EKN(500),FRQ(500),SIGCOM(500),RESID(500)

COMMON /SPLINE/ H,NN,NT,KA,ITERMX, LGRMAX,EPS,KO,SIG,EKNOTS,FREQ
COMMON /BSHARE/ TS,TF,EKN,FRQ,BSPLX,BSPLY,SIGCOM,V,COEF,D,WTRMS,
* GSIG,RESID

C BEGIN MODIFYING OBSERVED MAGNETIC FIELD, WHICH IS READ IN ON SCRATCH
C INPUT UNIT IRS. NUMOUT COUNTS NUMBER OF GEOCENTRIC LATITUDE OUTLIERS
C
NUMOUT=0
KB=0
DO 10 IB=1,L
READ(IRS) TX,TY,TZ,TB,INOTE,DX,DY,DZ,DB,I

C DETERMINE IF POINT NUMBER IB WAS USED IN TREND FITTING
C
IF(INOTE.EQ.6) GO TO 20

C IF POINT NUMBER IB WAS USED IN TREND FITTING, THEN KB RECORDS THE
C POSITION OF ITS COMPUTED TREND VALUE IN ARRAY V
C
KB=KB+1
TREND=REAL(V(I,KB))
GO TO 30

C IF POINT NUMBER IB WAS NOT USED IN TREND FITTING, THEN ITS GEOCENTRIC
C LATITUDE LIES OUTSIDE THE TOLERANCE LEVEL (INOTE = 6), SO CALL BSPLYN
C SUBPROGRAM USING INTERPOLATION MODE:
C
C INTERPOLATION ABDISSA SUPPLIED --> TIME I
C INTERPOLATION ORDINATE RETURNED --> Q(1,1)
C
C PRINT TREND FIT INTERPOLATION HEADING
C
20 NUMOUT=NUMOUT+1
IF(NUMOUT.EQ.1) WRITE(6,100)
100 FORMAT(/IX,'<GEOCENTRIC LATITUDE OUTLIER INTERPOLATION INFORMATION
*H>'/IX,'NUM','6X,'TIME',7X,'COMPONENT VALUE'/)
XINTRP=DBLE(I)
CALL BSPLYN(TS,TF,NN(LOOP),NKNOT,NT(LOOP),0,0,0,1,0,1,490,
*KA(LOOP),ITERMX(LOOP),LGRMAX(LOOP),EPS(LOOP),KO(LOOP),EKN,FRQ,
*BSPLX,BSPLY,SIGCOM,V,COEF,D,WTRMS,GSIG,RESID,XINTRP)
TREND=REAL(Q(1,1))

C PRINT TREND FIT INTERPOLATION ABDISSA, ORDINATE, AND OUTLIER NUMBER
C
WRITE(6,101) NUMOUT,XINTRP,TREND
101 FORMAT(1X,I5,2X,F8.3,2X,F20.10)

C DETREND ONE COMPONENT OF THE OBSERVED AND RESIDUAL MAGNETIC FIELD

A-55
C DEPENDING UPON THE VALUE OF LOOP
C
30 IF(LOOP.EQ.1) TX=TX-TREND
   IF(LOOP.EQ.1) DX=DX-TREND
   IF(LOOP.EQ.2) TY=TY-TREND
   IF(LOOP.EQ.2) DY=DY-TREND
   IF(LOOP.EQ.3) TZ=TZ-TREND
   IF(LOOP.EQ.3) DZ=DZ-TREND
C
C IF ALL 3 VECTOR COMPONENTS HAVE BEEN DETRENDED, THEN CALCULATE
C DETRENDED SCALAR VALUES
C
   IF(LOOP.EQ.3) TB=SQRT(TX**TX+TY**TY+TZ**TZ)
   IF(LOOP.EQ.3) DB=SQRT(DX**DX+DY**DY+DZ**DZ)
C
C WRITE MODIFIED MAGNETIC FIELD BACK OUT TO SCRATCH UNIT IWS
C
   10 WRITE(IWS) TX,TY,TZ,TB,INOTE,DX,DY,DZ, DB, I
C
C SWITCH SCRATCH INPUT AND OUTPUT UNITS FOR NEXT DATA MODIFICATION
C
   CALL SWITCH(IWS,IRS)
   RETURN
END
SUBROUTINE MODIFY(I,J,L,IRS,INS)
C
C SUBROUTINE TO WRITE MODIFIED DATA SET TO UNIT IOW WHICH HAS BEEN
C OUTPUT BY THE FILTER FOR THIS TIME INTERVAL OF INTEREST
C
DIMENSION ICLASS(2,8)
COMMON /MDFILE/ IOR, IOW, IOF, IOD, IOB, IOFI, IODI, IOWI, IOWIO
C
C INITIALIZE ARRAY ICLASS FOR CLASSIFICATION COUNTS IN THIS INTERVAL
C
   DO 1 INTROW=1,2
       DO 11 NTCOL=1,8
           ICLASS(INTROW,INTCOL)=0
C
C BEGIN FILTER OUTPUT PROCEDURES FOR THIS INTERVAL, REWIND UNIT IOW
C
C REWIND IOW
C
C SETUP FOR DATA QUALITY CLASSIFICATION COUNTER ICLASS:
C
C ICLASS(1,II) --> STATUS ON TOTAL DATA SET EXISTING ON UNIT IOW
C ICLASS(2,II) --> STATUS ON FILTERED OUTPUT DATA SET FOR THIS INTERVAL
C
C COUNTER DEFINITIONS:
C
C NRiow COUNTS TOTAL RECORDS EXISTING ON UNIT IOW
C
NRiow=0
C
C CHECK IF CURRENTLY GENERATED FILTER OUTPUT WILL BE FIRST DATA
C (IOWI=1) OR APPENDED DATA (IOWI=0) ON UNIT IOW.
C IF APPENDED, THEN POSITION FILE MARKER AFTER LAST EXISTING RECORD
C
   IF(IOWI.EQ.1) GO TO 15
   5 READ(IOW,200,END=10) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
   * ALT,CALT,BX,BY,BZ,BB,HX,HY,HZ,HB,TX,TY,TZ,TB,DX,DY,DZ,DB,CX,CY,CZ,
C
C UPDATE QUALITY CLASSIFICATION COUNTS OF DATA SET CURRENTLY ON UNIT IOW
C
ICLASS(1,INOTE+1)=ICLASS(1,INOTE+1)+1
GO TO 5
10 BACKSPACE IOW
C
C COUNTER DEFINITIONS:
C
C II IS CURRENT NUMBER OF POINTS READ ON UNIT IOR
C JJ IS CURRENT NUMBER OF POINTS FOUND IN TIME INTERVAL OF INTEREST
C LL IS CURRENT NUMBER OF MODIFIED POINTS
C
15 II=0
JJ=0
LL=0
C
C BEGIN READING INPUT DATA SET ON UNIT IOR
C
20 READ(IOR,200,END=35) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
*ALT,CALT,BX,BY,BZ,BB,HX,HY,HZ,HB,TX,TY,TZ,TB,DY,DZ,DB,CX,CY,CZ,
*CB,IDIR,INOTE
II=II+1
C
C CHECK IF CURRENT POINT IS WITHIN TIME INTERVAL OF INTEREST:
C INTERVAL RANGES FROM RECORD NUMBER (J-I) TO (J-I) ON UNIT IOR
C
IF(II.LT.J-I) GO TO 20
IF(II.GT.J-I) GO TO 35
C
C CURRENT POINT LIES WITHIN THE TIME INTERVAL OF INTEREST. CHECK IF ALL
C MODIFIED DATA POINTS (L) HAVE BEEN WRITTEN OUT, IF SO, THEN WRITE OUT
C DUPLICATE RECORD
C
IF(LL.GE.L) GO TO 30
JJ=JJ+1
C
C IF ALL MODIFIED POINTS HAVE NOT BEEN WRITTEN OUT, THEN READ NEXT SET
C OF MODIFICATION INFORMATION ON SCRATCH UNIT IRS
C
READ(IRS) PX,PY,PZ,PB,MNOTE,QX,QY,QZ,QB,III
IF(JJ.LT.III) GO TO 25
C
C IF CURRENT POINT IN INTERVAL (JJ) MATCHES CURRENT POINT TO BE MODIFIED
C (III), THEN WRITE OUT THE MODIFIED POINT AND RECORD MODIFICATION TOTAL
C
LL=LL+1
WRITE(IOW,200) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,ALT,
*CALT,BX,BY,BZ,BB,HX,HY,HZ,HB,PX,PY,PZ,PB,QX,QY,QZ,QB,CX,CY,CZ,CB,
*IDIR,MNOTE
C
C UPDATE QUALITY CLASSIFICATION COUNTS OF DATA SET FOR FILTER OUTPUT
C
ICLASS(2,MNOTE+1)=ICLASS(2,MNOTE+1)+1
GO TO 20
C
C IF CURRENT POINT IN INTERVAL DOES NOT MATCH CURRENT POINT TO BE
  MODIFIED, THEN WRITE OUT DUPLICATE RECORD AND BACKSPACE UNIT IRS TO
  RETAIN CURRENT MODIFICATION INFORMATION
C 25 BACKSPACE IRS
C UPDATE QUALITY CLASSIFICATION COUNTS OF DATA SET FOR FILTER OUTPUT
C 30 ICLASS(2,INOTE+1)=ICLASS(2,INOTE+1)+1
C WRITE OUT DUPLICATE RECORDS TO UNIT IOW
C WRITE(IOW,200) IYR,IDAY,TIME,GLAT,GCLAT,GLON,GMLAT,GMLON,ALT,
  *CALT,BX,BY,BZ,BH,TX,TY,TZ,DX,DY,DZ,DX,CY,CZ,CB,
  *IDIR,INOTE
  GO TO 20
C ADJUST CLASSIFICATION COUNTS FOR TOTAL UNIT IOW DATA SET DUE TO NEWLY
  APPENDED FILTER DATA
C 35 NRIOH=NRIOW+I
  DO 40 IADD=I,B
  40 ICLASS(1,IADD)=ICLASS(1,IADD)+ICLASS(2,IADD)
C PRINT QUALITY CLASSIFICATION STATUS OF DATA SET OUTPUT FROM THE FILTER
C WRITE(6,201)
  201 FORMAT(/1X,'<FILTER OUTPUT DATA CLASSIFICATION>'),
    WRITE(IO,202) (ICLASS(2,ICL),ICL=1,8),I
  202 FORMAT(/6X,'FLAG',4X,'COUNT',27X,'DESCRIPTION'/1X,'INOTE = 0',4X,
    *I5,' --> NO LIMITATIONS OR CONSTRAINTS'/1X,'INOTE = 1',4X,I5,' -->
    *GROSS-OUTLIER WITH RESPECT TO OBSERVED - COMPUTED FIELD'/1X,'IN
    *OTE = 2',4X,I5,' --> PADDED TIME-GAP VALUE'/1X,'INOTE = 3',4X,I5,
    *' --> B-SPLINE FIT-OUTLIER'/1X,'INOTE = 4',4X,I5,' --> FOURIER F
    *IT-OUTLIER'/1X,'INOTE = 5',4X,I5,' --> COMBINATION B-SPLINE/FOURI
    *ER FIT-OUTLIER'/1X,'INOTE = 6',4X,I5,' --> GEOCENTRIC LATITUDE LI
    *ES OUTSIDE TOLERANCE LEVEL'/1X,'INOTE = 7',4X,I5,' --> SATELLITE
    *VELLOCITY VECTOR DIRECTION IS INDETERMINABLE'/1X,'TOTAL ===> ',I
    *S,' RECORDS (EACH RECORD HAS 4 COMPONENTS: X, Y, Z, AND B)'/)
C PRINT QUALITY CLASSIFICATION STATUS OF TOTAL DATA SET ON UNIT IOW
C WRITE(6,203) IOW
  203 FORMAT(/1X,'<TOTAL FILTERED OUTPUT DATA CLASSIFICATION EXISTING ON
    *N UNIT ',I2,'>'),
    WRITE(IO,202) (ICLASS(1,ICL),ICL=1,8),NRIOW
    RETURN
END
SUBROUTINE DPINFO(IWP,NTOTL)
C SUBROUTINE TO PLOT VARIOUS DATA PARAMETERS: TIME/LATITUDE POSITION,
  VELOCITY VECTOR DIRECTION, AND TIME-GAP/OUTLIER INFORMATION
C CHARACTER*1 SYMBOL(8)
LOGICAL INUM
DIMENSION X(500),Y(500)
COMMON /EPHEMS/ ORBINC,ERAD,IEPDAY,INCREM,INTRVL
COMMON /LIMITS/ DXOL,DYOL,DZOL,DBOL,XWINDO,YWINDO,ZWINDO,BWINDO,
  * ABVLAT,TRNLAT,ITMGAP
C DEFINE SYMBOLS USED IN PLOTTING
C
DATA SYMBOL /*+','-','*','O','G','B','F','C'*/
C
C INITIALIZE PRINTER PLOTTING, DEFINE CARTESIAN OBJECT SPACE, AND
C OVERLAY CARTESIAN LINEAR GRID WITH TICK MARKS
C
XMAX=REAL(NTOTL)
CALL PLOTST(00001,1)
CALL SETGRD(11.0,12.0,123.0,62.0,1)
CALL OGRID(1.0,XMAX,8,'I3'),1,-90.0,90.0,18,'I3'),2,0)
C
C PLOT THE LATITUDE TOLERANCE WINDOW DEFINED BY +ABVLAT AND -ABVLAT
C USING THE SYMBOL --> =
C
DO 10 L=1,2
   DO 20 K=1,NTOTL
      IF(L.EQ.1) X(K)=REAL(K)
      IF(L.EQ.1) Y(K)=ABVLAT
20   IF(L.EQ.2) Y(K)=-Y(K)
10   CALL PLOT(X,Y,NTOTL,'=')
C
C PLOT 8 DATA POINT PARAMETERS, ONE AT A TIME
C
DO 30 IDQUAL=1,8
C
C REWIND SCRATCH UNIT IWP, WHICH CONTAINS PARAMETER INFORMATION
C
REWIND IWP
IK=0
C
C BEGIN READING PARAMETER INFORMATION FOR ALL NTOTL POINTS ON UNIT IWP
C
DO 40 K=1,NTOTL
READ(IWP) GCLAT,IDIR,INOTE,I
C
C CHECKING SEQUENCE
C
IF IDQUAL = 1, CHECK FOR ASCENDING POINTS
C IF IDQUAL = 2, CHECK FOR DESCENDING POINTS
C IF IDQUAL = 3, CHECK FOR TURN-AROUND POINTS
C IF IDQUAL = 4, CHECK FOR GROSS-OUTLIERS
C IF IDQUAL = 5, CHECK FOR PADDED TIME-GAP POINTS
C IF IDQUAL = 6, CHECK FOR B-SPLINE FIT-OUTLIERS
C IF IDQUAL = 7, CHECK FOR FOURIER FIT-OUTLIERS
C IF IDQUAL = 8, CHECK FOR B-SPLINE/FOURIER FIT-OUTLIERS
C
IF((IDQUAL.EQ.1).AND.(IDIR.GT.0)) GO TO 50
IF((IDQUAL.EQ.2).AND.(IDIR.LT.0)) GO TO 50
IF((IDQUAL.EQ.3).AND.(IDIR.EQ.0)) GO TO 50
IF((IDQUAL.EQ.4).AND.(INOTE.EQ.1)) GO TO 50
IF((IDQUAL.EQ.5).AND.(INOTE.EQ.2)) GO TO 50
IF((IDQUAL.EQ.6).AND.(INOTE.EQ.3)) GO TO 50
IF((IDQUAL.EQ.7).AND.(INOTE.EQ.4)) GO TO 50
IF((IDQUAL.EQ.8).AND.(INOTE.EQ.5)) GO TO 50
GO TO 40
C
C IF PARTICULAR DATA QUALITY IS CURRENTLY FOUND, THEN STORE POINT TIME
C IN ARRAY X, POINT LATITUDE IN ARRAY Y, AND RECORD TOTAL POINTS HAVING
C THIS QUALITY
C 50 IK=IK+1
   X(IK)=REAL(I)
   Y(IK)=GCLAT
40 CONTINUE
C PLOT POINTS HAVING CURRENT DATA QUALITY WITH FOLLOWING SYMBOLS:
C ASCENDING --> +, DESCENDING --> -, TURNING --> *, GROSS OUTLIER --> O,
C TIME-GAP --> G, B-SPLINE FIT OUTLIER --> B, FOURIER FIT OUTLIER --> F,
C COMBINATION B-SPLINE/FOURIER FIT OUTLIER --> C
C 30 IF(IK.NE.0) CALL PLOT(X,Y,IK,SYMBOL(IDQUAL))
C PRINT HEADING AND LEGEND
C CALL HORLIN('DATA QUALITY INFORMATION FOR INTERVAL: ',39,66.0,
   *67.0,0,0)
   CALL EDIT(INTRVL,'I2',INUM,NNUM,IBL)
   CALL HORLIN(INUM,2.66.0,67.0,40.0)
   CALL HORLIN('ASCENDING --> + DESCENDING --> - TURNING --> * GRO
   *SS OUTLIER --> O TIME-GAP --> G B-SPLINE FIT OUTLIER --> B',113,
   *66.0,65.0,0,0)
   CALL HORLIN('FOURIER FIT OUTLIER --> F COMBINATION B-SPLINE/FOURI
   *ER FIT OUTLIER --> C LATITUDE TOLERANCE RANGE --> = ',113,
   *66.0,64.0,0,0)
C LABEL PLOT AXES
C CALL HORLIN('TIME',4.66.0,8.0,0,0)
   CALL VERLIN('LATITUDE',8.5.0,37.0,0,0)
C TERMINATE PLOTTING SEQUENCE
C CALL ENDP LT
   RETURN
   END
   SUBROUTINE SWITCH(IWS,IRS)
C SWITCH SCRATCH INPUT AND OUTPUT UNITS BETWEEN UNITS ISC1 AND ISC2. IWS
C AND IRS ARE CURRENT OUTPUT AND INPUT UNITS, RESPECTIVELY
C AUX=IWS
   IWS=IRS
   IRS=AUX
C REMIND THE NEW IRS UNIT FOR NEXT READING AND THE NEW IWS UNIT FOR NEXT
C WRITING
C REWIND IRS
   REWIND IWS
   RETURN
   END
   SUBROUTINE STEP5
C SUBROUTINE TO WRITE OUT FINAL MODIFIED SATELLITE MAGNETIC FIELD DATA
C THREE VERSIONS OF FINAL DATA TAPES ARE AVAILABLE INDICATED BY IBTBS:
C (1) IF IBTBS = 0: WRITE OUT TO UNIT IOF:
   A) EPHEMERIS INFORMATION
(2) IF IBTBS = 1: WRITE OUT TO UNIT IOF:
A) EPHEMERIS INFORMATION
B) FIT/MAGSAT OBSERVED FIELD
C) FIT/MAGSAT RESIDUAL FIELD
D) TOPOCENTRIC OBSERVED FIELD
E) TOPOCENTRIC RESIDUAL FIELD
F) TOPOCENTRIC COMPUTED FIELD
G) DATA QUALITY INFORMATION
H) GEOMAGNETIC LATITUDE OUTLIER INFORMATION

WRITE OUT TO BINARY UNIT IOB IN FIT FORMAT:
A) EPHEMERIS INFORMATION
B) DATA QUALITY INFORMATION
C) FIT/MAGSAT OBSERVED FIELD
D) GEOMAGNETIC LATITUDE OUTLIER INFORMATION

(3) IF IBTBS = 2: SAME AS OPTION (2), BUT AN ADDITIONAL DATA TAPE, 
ANALOGOUS TO TAPE WRITTEN TO UNIT IOF, WILL BE 
WRITTEN OUT TO UNIT IOD IN A DESIRED SPACECRAFT 
COORDINATE SYSTEM

DIMENSION EU(3),CA(3,3),QI(3),QF(3),CF(3),RF(3,3),RC(3,3) 
DIMENSION A(28,100),IA(28,100),KCLASS(4,8),IFS(4) 
DIMENSION NOUTX(8),NOUTY(8),NOUTZ(8),NOUTB(8),NRCOUT(8) 
DIMENSION NOLDX(8),NOLDY(8),NOLDZ(8),NOLDB(8),NRCOLD(8) 
EQUIVALENCE (A(1,1),IA(1,1))
COMMON /MDFILE/ IOR, IOH, IOF, IOD, IOB, IOFIST, IODIST, IOH1ST, IOWIOF 
COMMON /COTRAN/ EU, CA, QI, QF, CF, RF, RC 
COMMON /EPHEMS/ ORBINC, ERAD, IEPDAY, INCREMENT, INTRVL 
COMMON /FILTOP/ IMETH, ISPEC, IBTBS, SIGMLT, NFLAGK 
COMMON /LIMITS/ DXOL, DYOL, DZOL, DBOL, XWINDO, YWINDO, ZWINDO, BWINDO, 
X ABVLAT, TRNLAT, ITMGAP 
DATA IFS /22,23,24,25/

C INITIALIZE MAGNETIC LATITUDE OUTLIER COUNTER ARRAYS IN THIS INTERVAL

DO 1 INTCOL=1,8
NOUTX(INTCOL)=0
NOUTY(INTCOL)=0
NOUTZ(INTCOL)=0
NOUTB(INTCOL)=0
NRCOUT(INTCOL)=0
NOLDX(INTCOL)=0
NOLDY(INTCOL)=0
NOLDZ(INTCOL)=0
NOLDB(INTCOL)=0
NRCOLD(INTCOL)=0

C INITIALIZE ARRAY KCLASS FOR CLASSIFICATION COUNTS IN THIS INTERVAL

DO 1 INTROW=1,4
1 KCLASS(INTROW,INTCOL)=0

C GENERATE VECTOR (IFS) WHICH PERMUTES MAGNETIC LATITUDE TOLERANCE FLAGS 
FROM FIT/MAGSAT TO DESIRED SPACECRAFT COORDINATES FOR OUTPUT TAPE
IFS(1)=NINT(RC(1,1))*22+NINT(RC(1,2))*23+NINT(RC(1,3))*24
IFS(2)=NINT(RC(2,1))*22+NINT(RC(2,2))*23+NINT(RC(2,3))*24
IFS(3)=NINT(RC(3,1))*22+NINT(RC(3,2))*23+NINT(RC(3,3))*24

C PRINT HEADING FOR STEP5 POST-FILTER PROCESSING

WRITE(6,200)
200 FORMAT('1','*'*** POST-FILTER PROCESSING ***'/1X,'**
**'*** POST-FILTER PROCESSING ***'/1X,'**

C BEGIN FINAL DATA MODIFICATION AND OUTPUT:

C REWIND INPUT UNITS --> IOW REMIND OUTPUT UNITS --> IOF
C
C REWIND IOW
REWIND IOF
REWIND IOD
REWIND IOB

C SETUP FOR DATA QUALITY CLASSIFICATION COUNTER KCLASS:

C KCLASS(1,II) --> STATUS ON UNIT IOF FILTER OUTPUT FOR THIS INTERVAL
C KCLASS(2,II) --> STATUS ON UNIT IOB FILTER OUTPUT FOR THIS INTERVAL
C KCLASS(3,II) --> STATUS OF ENTIRE DATA SETS ON UNITS IOF AND IOB
C KCLASS(4,II) --> STATUS OF ENTIRE DATA SET ON UNIT IOD

C COUNTER DEFINITIONS:

C NTOPO COUNTS TOTAL RECORDS READ FROM UNIT IOW AND WRITTEN TO UNIT IOF
C IN TOPOCENTRIC COORDINATES FOR THIS INTERVAL
C NTOTR COUNTS TOTAL RECORDS READ FROM UNIT IOW AND WRITTEN TO UNIT IOF
C IN FIT/ MAGSAT COORDINATES FOR THIS INTERVAL
C NSAT COUNTS TOTAL RECORDS WRITTEN TO UNIT IOD IN DESIRED SPACECRAFT
C COORDINATES FOR THIS INTERVAL
C NFIT COUNTS TOTAL NON-ZERO PADDED RECORDS WRITTEN TO UNIT IOB FOR
C THIS INTERVAL
C NBLK COUNTS TOTAL 100-RECORD BLOCKS WRITTEN IN BINARY FOR FIT INPUT
C ON UNIT IOB
C NRIOF COUNTS TOTAL RECORDS EXISTING ON UNIT IOW
C NRIOD COUNTS TOTAL RECORDS EXISTING ON UNIT IOD
C NBIOB COUNTS TOTAL NON-ZERO PADDED RECORDS EXISTING ON UNIT IOB

C NSTART STORES POSITION OF FIRST ZERO-PADDED RECORD OF LAST 100-RECORD
C BINARY BLOCK EXISTING ON UNIT IOB PRIOR TO THIS RUN. THIS INFORMATION
C IS USED WHEN APPENDING DATA (SEE BELOW)

C NSTART=1

A-62
C C FINAL DATA OUTPUT VERSION OPTION DEPENDING ON IBTBS
C
   IF(IBTBS.NE.0) GO TO 30
C C IF IBTBS = 0, CHECK IF CURRENTLY GENERATED OUTPUT DATA WILL BE FIRST
C DATA (IOFIST = 1) OR APPENDED DATA (IOFIST = 0) ON UNIT IOF.
C C IF APPENDED, THEN POSITION FILE MARKER AFTER LAST EXISTING RECORD
C
   IF(IOFIST.EQ.1) GO TO 15
   5 READ(IOF,201,END=10) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
      *ALT,CALT,tx,ty,tz,Db,dx,dy,dz,Db,CY,CZ,CB,IDIR,INOTE
   201 FORMAT(I2,I4,16,7F7.2,6X,12F8.1,2I5)
   NRIOF=NRIOF+1
C C UPDATE QUALITY CLASSIFICATION COUNTS OF TOTAL TOPOCENTRIC OUTPUT DATA
C SET PRESENTLY RESIDING ON UNIT IOF
C
   KCLASS(3,INOTE+I)=KCLASS(3,INOTE+I)+1
   GO TO 5
10 BACKSPACE IOF
C C IF IBTBS = 0, WRITE OUT TOPOCENTRIC FORMAT TAPE. NEW UNIT IOF TAPE
C HAS IDENTICAL INFORMATION AS INPUT UNIT IOW TAPE, EXCEPT UNMODIFIED
C FIT/MAGSAT MAGNETIC FIELD COMPONENTS ARE OMITTED
C
   15 READ(IOW,202,END=20) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
      *ALT,CALT,BX,BY,BZ,BB,HX,HY,HZ,HB,tx,ty,tz,Db,dx,dy,dz,Db,CX,CY,CZ,
      *CB,IDIR,INOTE
   202 FORMAT(I2,I4,16,7F7.2,ZOFB.1,215)
C C DETERMINE UNIT IOW TIME INTERVALS TO BE PROCESSED DURING THIS STEP:
C C IF IOHIOF = 0 --> PROCESS INTRVL ONLY
C C IF IOHIOF = 1 --> PROCESS INTRVL AND PRECEEDING INTERVALS
C C IF IOHIOF = 2 --> PROCESS ALL INTERVALS
C C IF IOHIOF = 0 --> IF CURRENT DAY (IDAY) IS EARLIER THAN EPOCH DAY
C (IEPDAY), THEN REJECT POINT
C
   IF((IOHIOF.EQ.0).AND.(IDAY.LT.IEPDAY)) GO TO 15
C C DETERMINE RELATIVE TIME OF DATA POINT (ICTIME) WITH RESPECT TO
C BEGINNING OF EPOCH DAY (IEPDAY), THEN DETERMINE ITS TIME INTERVAL (NI)
C WITH RESPECT TO INTERVAL WIDTH (INCREM).
C
   ICTIME=(IDAY-IEPDAY)*86400+IETIME
   NI=INT(ICTIME/INCREM)+1
C C IF IOHIOF = 0 --> IF CURRENT TIME INTERVAL IS LESS THAN INTERVAL OF
C INTEREST, THEN REJECT POINT
C
   IF((IOHIOF.EQ.0).AND.(NI.LT.INTRVL)) GO TO 15
C C IF IOHIOF = 0 OR 1 --> IF CURRENT TIME INTERVAL IS GREATER THAN
C INTERVAL OF INTEREST, THEN REJECT POINT
C
   IF((IOHIOF.LE.1).AND.(NI.GT.INTRVL)) GO TO 20
C C BEGIN COUNT OF DATA ACCEPTED FROM UNIT IOW

A-63
C      NTOPO=NTOPO+1
C UPDATE QUALITY CLASSIFICATION COUNTS OF TOPOCENTRIC OUTPUT DATA SET
C
   KCLASS(1,INOTE+1)=KCLASS(1,INOTE+1)+1
C
C WRITE TOPOCENTRIC DATA SET OUT TO UNIT IOF
C
   WRITE(IOF,201) IYR,IDAY, IETIME, GLAT, GCLAT, GLON, GMLAT, GMLON, ALT,
           * CALT, TX, TY, TZ, DB, DX, DY, DZ, DB, CX, CY, CZ, CB, IDIR, INOTE
   GO TO 15
C
C CALCULATE TOTAL NUMBER OF COMPONENTS (NCOMPF) OUTPUT BY THE FILTER
C EXCLUDING PADDED TIME-GAP RECORDS
C
   20 NCOMPF=4*(NTOPO-KCLASS(1,3))
C
C END PROCESSING OF FINAL TOPOCENTRIC FORMAT TAPE. PRINT CLASSIFICATION
C COUNTS FOR THIS TAPE
C
   WRITE(6,203) IOF
   203 FORMAT(/1X,'<STEP5 TOPOCENTRIC FORMATTED OUTPUT DATA CLASSIFICATI
   * ON UNIT ',I2,">')
   WRITE(6,204) (KCLASS(1,KCL),KCL=1,8), NTOPO, NCOMPF

   204 FORMAT(/6X,'FLAG',4X,'COUNT',27X,'DESCRIPTION'/1X,'INOTE = 0',4X,
           *15,' --> NO LIMITATIONS OR CONSTRAINTS'/1X,'INOTE = 1',4X,15,' -->
           * B-SPLINE OUTLIER WITH RESPECT TO OBSERVED - COMPUTED FIELD'/1X,
           * INOTE = 2',4X,15,' --> PADDED TIME-GAP VALUE'/1X,'INOTE = 3',4X,15,
           * --> FOURIER FIT-OUTLIER'/1X,'INOTE = 4',4X,15,' --> COMBINATION B-SPLINE/FOURI
           * FIT-OUTLIER'/1X,'INOTE = 5',4X,15,' --> GEOCENTRIC LATITUDE LI
           * ES OUTSIDE TOLERANCE LEVEL'/1X,'INOTE = 7',4X,15,' --> SATELLITE
           * VELOCITY VECTOR DIRECTION IS INDETERMINABLE'/1X,'TOTAL ====>
           ',I
           *5,' RECORDS'/1X,'TOTAL ====> ',I5,' COMPONENTS'/)
C
C ADJUST UNIT IOF CLASSIFICATION COUNTS DUE TO NEWLY APPENDED DATA
C
   NRIOF=NRIOF+NTOPO
   DO 25 IADD=1,8
   25 KCLASS(3,IADD)=KCLASS(3,IADD)+KCLASS(I,IADD)
C
C CALCULATE TOTAL NUMBER OF COMPONENTS (NCOMPT) EXISTING ON UNIT IOF
C EXCLUDING PADDED TIME-GAP RECORDS
C
   NCOMPT=4*(NTOPO-KCLASS(3,3))
C
C PRINT STATUS OF ENTIRE OUTPUT DATA SET EXISTING ON UNIT IOF
C
   WRITE(6,205) IOF
   205 FORMAT(/1X,'<TOTAL TOPOCENTRIC FORMATTED OUTPUT DATA CLASSIFICATI
   * ON EXISTING ON UNIT ',I2,">')
   WRITE(6,204) (KCLASS(3,KCL),KCL=1,8), NRIOF, NCOMPT
   RETURN
C
C IF IBTBS = 1 OR 2, CHECK IF CURRENTLY GENERATED OUTPUT DATA WILL BE
C FIRST DATA (IOF1ST = 1) OR APPENDED DATA (IOF1ST = 0) ON UNITS IOF AND
C IDB, AND FIRST DATA (IOD1ST = 1) OR APPENDED DATA (IOD1ST = 0) ON UNIT
C IOD. IF APPENDED, THEN POSITION FILE MARKER AFTER LAST EXISTING RECORD

A-64
30 IF(IOFIST.EQ.1) GO TO 70
C
C POSITION FILE MARKER FOR APPENDING DATA ON UNIT IOF
C
40 READ(IOF,206,END=45) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
    *ALT,CALT,PX,PY,PZ,PB,HX,HY,HZ,HB,TX,TY,TZ,TB,DX,DY,DZ,DB,CX,CY,CZ,
    *CB,IDIR,INOTE,IAX,IAY,IAZ,IAB
206 FORMAT(I2,I4,I6,7F7.2,20F8.1,2I5,4I2)
NRIOF=NRIOF+1
C
C UPDATE QUALITY CLASSIFICATION COUNTS OF TOTAL FIT/MAGSAT OUTPUT DATA
C SET PRESENTLY RESIDING ON UNIT IOF
C
    KCLASS(3,INOTE+1)=KCLASS(3,INOTE+1)+1
C
C IF CURRENT POINT IS A Padded TIME-GAP VALUE (INOTE = 2), THEN DO NOT
C UPDATE MAGNETIC LATITUDE OUTLIER COUNTS
C
    IF(INOTE.EQ.2) GO TO 40
C
C UPDATE MAGNETIC LATITUDE OUTLIER RECORD COUNTS
C
    IF((IAX.EQ.0).OR.(IAY.EQ.0).OR.(IAZ.EQ.0).OR.(IAB.EQ.0))
        *NRCDLD(INOTE+1)=NRCDLD(INOTE+1)+1
C
OUTLIER COUNTER DEFINITIONS FOR INDIVIDUAL DATA QUALITY FLAGS:
C
    NOLDX-B FOR ENTIRE UNIT IOF DATA SET IS ANALOGOUS TO NOUTX-B FOR
C CURRENT FILTER OUTPUT DATA SET (SEE DESCRIPTION BELOW)
C
C TALLY MAGNETIC LATITUDE OUTLIER COMPONENTS EXISTING ON UNIT IOF
C
    IF(IAX.EQ.0) NOLDX(INOTE+1)=NOLDX(INOTE+1)+1
    IF(IAY.EQ.0) NOLDY(INOTE+1)=NOLDY(INOTE+1)+1
    IF(IAZ.EQ.0) NOLDZ(INOTE+1)=NOLDZ(INOTE+1)+1
    IF(IAB.EQ.0) NOLDB(INOTE+1)=NOLDB(INOTE+1)+1
GO TO 40
45 BACKSPACE IOF
C
C POSITION FILE MARKER FOR APPENDING DATA ON UNIT IOB
C
50 READ(IOB,END=60) A
C
C CHECK THE MODIFIED JULIAN DAY STORED IN FIRST ELEMENT OF EACH RECORD
C OR COLUMN OF THIS IOO-RECORD BLOCK (SEE FIT INPUT FORMAT BELOW):
C
C IF MOD JUL DAY IS NOT ZERO --> DATA EXISTS ON THE RECORD
C IF MOD JUL DAY IS ZERO    --> NO DATA EXISTS ON THE RECORD (PADDED)
C
C TOTAL NON-PADDED RECORDS: NBIOB
C CURRENT RECORD CHECKED: NBI0B
C PROCESSING ORDER: FROM RECORD NUMBER 100 --> 1 SO THAT
C FULL-RECORD CHECK TIME IS MINIMIZED
C
    NSTART=101
55 NSTART=NSTART-1
C
C IF ENTIRE BLOCK IS PADDED (NSTART = 0), THEN SET APPEND POSITION AT
C THE FIRST RECORD, THERE OVERWRITING THE ENTIRE PADDED BLOCK
C
IF(NSTART.EQ.0) GO TO 65
C
C IF THE FIRST NSTART RECORDS ARE NOT ZERO, THEN SET APPEND POSITION AT
C NEXT RECORD AFTER THESE, THUS OVERWRITING THE PADDED PORTION
C
IF(IA(1,NSTART).NE.0) GO TO 65
GO TO 55
C
C ENTRY POSITION IF END OF FILE MARK IS ENCOUNTERED, TREAT THE SAME AS
C IF ENTIRE PADDED BLOCK HAS BEEN ENCOUNTERED
C
60 NSTART=0
C
C UPDATE COUNT OF NON-ZERO RECORDS EXISTING ON UNIT IOB
C
65 NBIOB=NBIOB+NSTART
C
C IF A FULL NON-ZERO 100-RECORD BLOCK WAS ENCOUNTERED, THEN READ AND
C CHECK NEXT BLOCK UNTIL A BLOCK IS FOUND WITH PADDED VALUES OR AN END
C OF FILE MARK IS ENCOUNTERED
C
IF(NSTART.EQ.100) GO TO 50
C
C ADJUST NSTART FROM LAST NON-ZERO RECORD POSITION TO APPEND POSITION
C
NSTART=NSTART+1
C
C SET UNIT IOB FILE POSITION MARKER TO REWRITE LAST DATA BLOCK
C
BACKSPACE IOB
70 IF((IBTBS.NE.2).OR.(IODIST.EQ.I)) GO TO 85
C
C POSITION FILE MARKER FOR APPENDING DATA ON UNIT IOD
C
75 READ(IOD,206,END=80) IYR,IDAY, IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
*WALT,CALT,UX,UY, UZ,UB,WX, WY,WZ, WB,TX, TY, TZ, TB,DX, DY, DZ, DB,CX, CY,CZ,
*CB,IDIR,INOTE,IAX, IAY,IAZ, IAB
NRIOD=NRIOD+1
C
C UPDATE QUALITY CLASSIFICATION COUNTS OF TOTAL DESIRED SPACECRAFT
C OUTPUT DATA SET PRESENTLY RESIDING ON UNIT IOD
C
KCLASS(_,INOTE+I)=KCLASS(_,INOTE+I)+I
GO TO 75
80 BACKSPACE IOD
C
C IF IBTBS = 1 OR 2, WRITE OUT FIT/MAGSAT FORMAT TAPES AND IF IBTBS = 2,
C WRITE OUT DESIRED SPACECRAFT FORMAT TAPE. BINARY OUTPUT IS WRITTEN TO
C UNIT IOB IN BLOCKS OF 100 RECORDS.
C
C INITIALIZE COLUMN NUMBER NSTART THROUGH NUMBER 100 OF BLOCK STORAGE
C ARRAYS A AND IA FOR GENERATION OF NEXT DATA BLOCK
C
85 DO 90 II=NSTART,100
 DO 90 JJ=1,28
 IA(JJ,II)=0
 90 A(JJ,II)=0.0
C
C PROCESSING FOR 100-RECORD DATA BLOCKS IN THIS RUN:
C
FIRST BLOCK --> FIRST 101 - NSTART RECORDS FROM INPUT UNIT IOW
NSTART - 1 RECORDS ALREADY EXIST FROM UNIT IOB
SUBSEQUENT BLOCKS --> NEXT 100 RECORDS FROM INPUT UNIT IOW

DO 95 II=NSTART,100
100 READ(IOW,202,END=115) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,
ALT,CALT,BX,BY,BZ,OX,OY,OZ,OB,TX,TY,TZ,TB,DX,DY,DZ,DB,CX,CY,CZ,
CB,IDIR,INOTE

DETERMINE UNIT IOW TIME INTERVALS TO BE PROCESSED DURING THIS STEP:

IF IOWIOF = 0 --> PROCESS INTRVL ONLY
IF IOWIOF = 1 --> PROCESS INTRVL AND PRECEEDING INTERVALS
IF IOWIOF = 2 --> PROCESS ALL INTERVALS
IF IOWIOF = 0 --> IF CURRENT DAY (IDAY) IS EARLIER THAN EPOCH DAY
   (IEPDAY), THEN REJECT POINT
IF((IOWIOF.EQ.0).AND.(IDAY.LT.IEPDAY)) GO TO 100

DETERMINE RELATIVE TIME OF DATA POINT (ICTIME) WITH RESPECT TO
BEGINNING OF EPOCH DAY (IEPDAY), THEN DETERMINE ITS TIME INTERVAL (NI)
WITH RESPECT TO INTERVAL WIDTH (INCREM).

ICTIME=(IDAY-IEPDAY)*86400+IETIME
NI=INT(ICTIME/INCREM)+1
IF((IOWIOF.EQ.0).AND.(NI.LT.INTRVL)) GO TO 100
IF((IOWIOF.EQ.0).AND.(NI.LT.INTRVL)) GO TO 100
IF(IOWIOF=0 OR 1 --> IF CURRENT TIME INTERVAL IS GREATER THAN
CURRENT INTERVAL OF INTEREST, THEN REJECT POINT
IF((IOWIOF.LE.1).AND.(NI.LT.INTRVL)) GO TO 115

BEGIN COUNT OF DATA ACCEPTED FROM UNIT IOW
NTOTR=NTOTR+1

UPDATE QUALITY CLASSIFICATION COUNTS OF IOW INPUT AND IOF OUTPUT DATA
KCLASS(1,INOTE+I)=KCLASS(1,INOTE+I)+1

PERFORM TRANSFORMATION FROM TOPOCENTRIC TO FIT/MAGSAT TO THE DESIRED
SPACECRAFT COORDINATE SYSTEMS IN THE FOLLOWING ORDER:
FOR IDIR = -1 OR 1: TOPOCENTRIC FIT/MAGSAT DESIRED
(1) OBSERVED COMPONENTS: (TX,TY,TZ) --> (PX,PY,PZ) --> (UX,UY,UZ)
(2) RESIDUAL COMPONENTS: (DX,DY,DZ) --> (HX,HY,HZ) --> (WX,NY,NZ)

FOR IDIR = 0: FIT/MAGSAT DESIRED
(1) OBSERVED COMPONENTS: (BX,BY,BZ) --> (PX,PY,PZ) --> (UX,UY,UZ)
(2) RESIDUAL COMPONENTS: (0,0,0) --> (0,0,0) --> (0,0,0)

IF(IDIR.NE.0) CALL BTTOBS(GCLAT,IDIR,TX,TY,TZ,PX,PY,PZ,PB,
* UX, UY, UZ, UB*

IF(IDIR.EQ.0) CALL BTTOBS(GCLAT, IDIR, BX, BY, BZ, PX, PY, PZ, PB, UX, UY, UZ, UB)
* CALL BTTOBS(GCLAT, IDIR, DX, DY, DZ, HX, HY, HZ, HB, WX, WY, WZ, WB)

C IF CURRENT DATA POINT IS A TIME-GAP PADDED VALUE (INOTE = 2), THEN:

C OMIT FROM --> BINARY UNIT IOB FINAL OUTPUT TAPE
C INCLUDE IN --> FORMATTED UNIT IOF FINAL OUTPUT TAPE

C IF(INOTE.EQ.2) GO TO 105

C UPDATE QUALITY CLASSIFICATION COUNTS OF UNIT IOB FIT/MAGSAT OUTPUT

C KCLASS(2, INOTE+1)=KCLASS(2, INOTE+1)+1
NFIT=NFIT+1

C STORE CURRENT DATA POINT INFORMATION, RECORD II OF CURRENT 100 RECORD
C BLOCK, IN COLUMN II OF STORAGE ARRAYS A AND IA ACCORDING TO THE FIT
C INPUT FORMAT:

C IA(1,II)  = MODIFIED JULIAN DAY
C IA(2,II)  = MILLISECONDS OF DAY
C A(3,II)   = NOT USED
C A(4,II)   = FRACTION OF DAY
C A(5,II)   = TIME IN YEARS FROM 1900
C A(6,II)   = GEOCENTRIC LATITUDE
C A(7,II)   = LONGITUDE
C A(8,II)   = NOT USED
C A(9,II)   = NOT USED
C A(10,II)  = NOT USED
C A(11,II)  = SATELLITE X-AXIS COMPONENT IN FIT/MAGSAT COORDINATES
C A(12,II)  = SATELLITE Y-AXIS COMPONENT IN FIT/MAGSAT COORDINATES
C A(13,II)  = SATELLITE Z-AXIS COMPONENT IN FIT/MAGSAT COORDINATES
C A(14,II)  = SCALAR INTENSITY
C IA(15,II) = GEOCENTRIC ALTITUDE (METERS) ABOVE ERAD (KM)
C A(16,II)  = NOT USED
C A(17,II)  = NOT USED
C IA(18,II) = DATA QUALITY CLASSIFICATION FLAG (INOTE)
C IA(19,II) = 0
C IA(20,II) = SATELLITE VELOCITY VECTOR DIRECTION (IDIR)
C IA(21,II) = 0
C IA(22,II) = MAGNETIC LATITUDE OUTLIER FLAG FOR SATELLITE X-AXIS
C IA(23,II) = MAGNETIC LATITUDE OUTLIER FLAG FOR SATELLITE Y-AXIS
C IA(24,II) = MAGNETIC LATITUDE OUTLIER FLAG FOR SATELLITE Z-AXIS
C IA(25,II) = MAGNETIC LATITUDE OUTLIER FLAG FOR SCALAR INTENSITY
C A(26,II)  = NOT USED
C A(27,II)  = NOT USED
C A(28,II)  = NOT USED

C ASSIGN ARRAYS A AND IA NOW FOR CURRENT RECORD II

C IA(1,II)=IDAY
IA(2,II)=IETIME*1000
A(4, II)=REAL(IETIME)/86400.0
A(5,II)=REAL(IYR)+(REAL(IDAY)+A(4,II))/365.0
A(6,II)=GCLAT
A(7,II)=GLON
A(11,II)=PX
A(12,II)=PY
A(13,II)=PZ
A(14,II)=PB
IA(15,II)=INT((CALT-ERAD)*1000.0)
IA(18,II)=INOTE
IA(20,II)=IDIR

C CHECK MAGNETIC LATITUDE AGAINST GIVEN MAGNETIC LATITUDE TOLERANCE
C WINDOW FOR EACH VECTOR AND SCALAR COMPONENT USING THE FOLLOWING FLAGS:
C
C IF OUTSIDE WINDOW --> IA = 0
C IF INSIDE WINDOW --> IA = 2

C OUTLIER COUNTER DEFINITIONS FOR INDIVIDUAL DATA QUALITY FLAGS:
C
C NOUTX(I) COUNTS TOTAL FIT/MAGSAT X MAGNETIC LATITUDE OUTLIERS
C NOUTY(I) COUNTS TOTAL FIT/MAGSAT Y MAGNETIC LATITUDE OUTLIERS
C NOUTZ(I) COUNTS TOTAL FIT/MAGSAT Z MAGNETIC LATITUDE OUTLIERS
C NOUTB(I) COUNTS TOTAL FIT/MAGSAT B MAGNETIC LATITUDE OUTLIERS
C NRCDUT(I) COUNTS TOTAL RECORDS WHICH HAVE AT LEAST ONE COMPONENT
C OUTSIDE THE MAGNETIC LATITUDE TOLERANCE LEVEL
C
C (WHERE I = 1-8 CORRESPONDS TO INOTE = 0-7)
C
C AGMLAT=ABS(GMLAT)
C
C ASSIGN MAGNETIC OUTLIER FLAGS
C
C IF(AGMLAT.LE.XWINDO) IA(22,II)=2
IF(AGMLAT.LE.YWINDO) IA(23,II)=2
IF(AGMLAT.LE.ZWINDO) IA(24,II)=2
IF(AGMLAT.LE.BWINDO) IA(25,II)=2

C UPDATE MAGNETIC LATITUDE OUTLIER RECORD COUNTS
C
C IF((AGMLAT.GT.XWINDO).OR.(AGMLAT.GT.YWINDO).OR.(AGMLAT.GT.ZWINDO)
* OR.(AGMLAT.GT.BWINDO)) NRCDOUT(INOTE+1)=NRCDOUT(INOTE+1)+1

C UPDATE MAGNETIC LATITUDE OUTLIER COMPONENT COUNTS
C
C IF(AGMLAT.GT.XWINDO) NOUTX(INOTE+I)=NOUTX(INOTE+I)+1
IF(AGMLAT.GT.YWINDO) NOUTY(INOTE+I)=NOUTY(INOTE+I)+1
IF(AGMLAT.GT.ZWINDO) NOUTZ(INOTE+I)=NOUTZ(INOTE+I)+1
IF(AGMLAT.GT.BWINDO) NOUTB(INOTE+I)=NOUTB(INOTE+I)+1

C ENTRY POINT HERE IF CURRENT POINT IS PADDED TIME-GAP VALUE
C
105 IF(IBTBS.NE.2) GO TO 110
C
C IF IBTBS = 2, THEN WRITE CURRENT DATA POINT INFORMATION TO UNIT IOD
C IN THE DESIRED SPACECRAFT COORDINATES
C
NSAT=NSAT+1
WRITE(IOD,206) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,ALT,
CALT,UX,UY,UZ,UB,MX,MY,MZ,MB,TX,TY,TZ,TB,DX,DY,DZ,DB,CX,CY,CZ,CB,
IDIR,INOTE,(IA(IFS(KK),II),KK=I,q)

C WRITE CURRENT DATA POINT INFORMATION TO UNIT IOF IN THE FIT/MAGSAT
C COORDINATES, INCLUDING DATA FLAGS FOR THE INDIVIDUAL COMPONENTS
C
110 WRITE(IOF,206) IYR,IDAY,IETIME,GLAT,GCLAT,GLON,GMLAT,GMLON,ALT,
C IF CURRENT POINT IS PADDED TIME-GAP VALUE, THEN READ NEXT RECORD ON
C UNIT IOH, BUT DO NOT PROGRESS TO NEXT RECORD OF DATA BLOCK A
C
IF(INOTE.EQ.2) GO TO 100
C
C IF CURRENT POINT IS NOT A TIME-GAP VALUE, THEN PROGRESS TO NEXT
C RECORD OF DATA BLOCK A
C
95 CONTINUE
C
WRITE FULL 100 RECORD DATA BLOCK TO BINARY UNIT IOB
C
NBLK=NBLK+1
WRITE(IOB) A
IF(NBLK.GT.1) GO TO 85
C
Determine number of non-zero (NOZERO) and padded-zero (NPAD) records
C that occurred on last 100-record block of unit IOB prior to this run
C
NPAD=IO1-NSTART
NOZERO=NSTART-1
C
SET RECORD APPEND POSITION TO NSTART = 1 FOR BLOCKS SUBSEQUENT TO THE
C FIRST BLOCK SO THAT A FULL 100 RECORDS MAY BE WRITTEN TO THEM
C
NSTART=1
GO TO 85
C
WRITE FINAL PARTIAL 100 RECORD DATA BLOCK TO BINARY UNIT IOB
C
115 NBLK=NBLK+1
WRITE(IOB) A
C
CALCULATE TOTAL NUMBER OF COMPONENTS (NCOMPF) OUTPUT BY THE FILTER
C EXCLUDING PADDED TIME-GAP RECORDS
C
NCOMPF=4*(NTOTR-KCLASS(I,3))
C
PRINT QUALITY CLASSIFICATION STATUS OF UNIT IOF FIT/MAGSAT OUTPUT
C
WRITE(6,207) IOF
207 FORMAT(//1X,'<STEP5 FIT/MAGSAT FORMATTED OUTPUT DATA CLASSIFICATIO
XN ON UNIT ',I2,',''>')
WRITE(6,204) (KCLASS(1,KCL),KCL=I,8),NTOTR,NCOMPF
C
PRINT QUALITY CLASSIFICATION STATUS OF UNIT IOB FIT/MAGSAT OUTPUT
C
WRITE(6,208) IOB
208 FORMAT(//1X,'<STEP5 FIT/MAGSAT BINARY OUTPUT DATA CLASSIFICATION O
XN UNIT ',I2,',''>')
WRITE(6,204) (KCLASS(2,KCL),KCL=I,8),NFIT,NCOMPF
C
PRINT WRITTEN RECORD TOTALS FOR EACH OUTPUT DATA SET TYPE
C
WRITE(6,209) NTOTR,IOF,NSAT,IOD,NFIT,IOB,NBLK,IOB
209 FORMAT(//1X,'OUTPUT RECORD SUMMARY:'//1X,'TOTAL ==> ',I5,' FORM
*ATTED FIT/MAGSAT RECORDS WRITTEN TO UNIT ',I2//1X,'TOTAL ==> ',
*I5,' FORMATTED DESIRED RECORDS WRITTEN TO UNIT ',I2//1X,' TOTAL ===
* => ',I5,' NON-ZERO PADDED RECORDS WRITTEN TO UNIT ',I2//1X,' TOTAL
* ====> ',I5,' BINARY 100-RECORD BLOCKS WRITTEN TO UNIT ',I2//)

C PRINT NUMBER OF PADDED RECORDS OVERWRITTEN (BY DATA GENERATED IN THIS
C RUN) AND NUMBER OF RECORDS ALREADY EXISTING ON FIRST 100-RECORD DATA
C BLOCK TRANSMITTED TO UNIT IOB DURING THIS RUN

WRITE(6,210) NOZERO,NPAD
210 FORMAT(20X,' TOTAL ===> ',I5,' PREVIOUSLY EXISTING RECORDS INCORP
*ORATED IN FIRST 100-RECORD DATA BLOCK'//20X,' TOTAL ===> ',I5,' R
*ECORDS GENERATED DURING THIS INTERVAL INCORPORATED IN FIRST 100-RE
*CORD DATA BLOCK')

C PRINT MAGNETIC LATITUDE OUTLIER HEADING

WRITE(6,211)
211 FORMAT(//1X,' MAGNETIC LATITUDE OUTLIER BREAKDOWN BY FLAGS:'//6X,'F
*LAG',10X,' X OUTLIERS',10X,' Y OUTLIERS',10X,' Z OUTLIERS',4X,' B OUTLIE
*RS',6X,' COMPONENTS',4X,' RECORDS')

C MAGNETIC LATITUDE OUTLIER COUNTER DEFINITIONS:

C NXO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE X OUTLIERS
C NYO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE Y OUTLIERS
C NZO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE Z OUTLIERS
C NBO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE B OUTLIERS
C NCO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE OUTLIER COMPONENTS
C NFOUT COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE OUTLIERS FOR
C PARTICULAR DATA QUALITY FLAG
C NRCTOT COUNTS TOTAL NUMBER OF RECORDS CONTAINING MAGNETIC LATITUDE
C OUTLIERS

NXO=0
NYO=0
NZO=0
NBO=0
NRCTOT=0

C PRINT MAGNETIC LATITUDE OUTLIER COUNTERS PER EACH DATA QUALITY FLAG

DO 120 IN=1,8
NF=IN-1
NXO=NXO+NOUTX(IN)
NYO=NYO+NOUTY(IN)
NZO=NZO+NOUTZ(IN)
NBO=NBO+NOUTB(IN)
NRCTOT=NRCTOT+NRCOUT(IN)
NFOUT=NOUTX(IN)+NOUTY(IN)+NOUTZ(IN)+NOUTB(IN)
120 WRITE(6,212) NF,NOUTX(IN),NOUTY(IN),NOUTZ(IN),NOUTB(IN),NFOUT,
*XRCOUT(IN)
212 FORMAT(1X,' INOTE = ',I1,' --> ',4I14,' --> ',5X,I5,6X,I5)

C NCO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE OUTLIER COMPONENTS

NCO=NXO+NYO+NZO+NBO

C PRINT MAGNETIC LATITUDE OUTLIER COUNTERS PER EACH COMPONENT

WRITE(6,213) NXO,NYO,NZO,NBO,NCO,NRCTOT
213 FORMAT(/1X,'TOTAL  --> ',4I14,' ==> ',5X,I5,6X,I5///)
C
C BEGIN PROCESSING STATUS ON ENTIRE DATA SETS EXISTING ON UNITS IOF, IOD, AND IOB
C
C ADJUST OUTPUT UNIT CLASSIFICATION COUNTS DUE TO NEWLY APPENDED DATA
C
NRIOF=NRIOF+NTOTR
NRIOD=NRIOD+NSAT
DO 125 IADD=1,8
   KCLASS(3,IADD)=KCLASS(3,IADD)+KCLASS(1,IADD)
125 KCLASS(4,IADD)=KCLASS(4,IADD)+KCLASS(1,IADD)
C
C CALCULATE TOTAL NUMBER OF COMPONENTS (NCOMPT) EXISTING ON UNIT IOF
C EXCLUDING PADDED TIME-GAP RECORDS
C
NCOMPT=NRIOF-KCLASS(3,3)
C
C PRINT STATUS OF ENTIRE OUTPUT DATA SET EXISTING ON UNIT IOF
C
WRITE(6,214) IOF
214 FORMAT(/1X,'TOTAL FIT/MAGSAT FORMATTED OUTPUT DATA CLASSIFICATION
*EXISTING ON UNIT ',II,','>'
   WRITE(6,204) (KCLASS(3,KCL),KCL=I,8),NRIOF,NCOMPT
C
C CALCULATE TOTAL NUMBER OF 100-RECORD BLOCKS EXISTING ON UNIT IOB
C AFTER APPENDING NEW DATA AND ELIMINATING PADDED ZEROS
C
NBIOB=NBIOB+NFIT
NBLK=INT(NBIOB/100)
IF(MOD(NBIOB,100).GT.0) NBLK=NBLK+1
C
C ADJUST UNIT IOB OUTPUT DATA QUALITY CLASSIFICATION STATUS BY OMITTING
C PADDED TIME-GAP VALUE COUNTS STORED IN KCLASS(3,3)
C
KCLASS(3,3)=0
C
C PRINT STATUS OF ENTIRE OUTPUT DATA SET EXISTING ON UNIT IOB
C
WRITE(6,215) IOB
215 FORMAT(/1X,'TOTAL FIT/MAGSAT BINARY OUTPUT DATA CLASSIFICATION E
*XISTING ON UNIT ',II,','>'
   WRITE(6,204) (KCLASS(3,KCL),KCL=I,8),NBIOB,NCOMPT
C
C PRINT WRITTEN RECORD TOTALS FOR EACH OUTPUT DATA SET TYPE
C
WRITE(6,209) NRIOF,IOF,NRIOD,IOD,NBIOB,IOB,NBLK,IOB
C
C PRINT MAGNETIC LATITUDE OUTLIER HEADING
C
WRITE(6,211)
C
C MAGNETIC LATITUDE OUTLIER COUNTER DEFINITIONS:
C
NXO-NRCTOT ARE ANALOGOUS FOR THIS TOTAL OUTPUT STATUS OF UNIT IOF
C (SEE DESCRIPTION ABOVE) THESE ARE CUMULATIVE SUMS FROM THE PRESENT
C FILTER OUTPUT COUNTS AND THE COUNTS MADE ON DATA WHICH EXISTED PRIOR
C TO THIS RUN ON UNIT IOF
C
C PRINT MAGNETIC LATITUDE OUTLIER COUNTS PER EACH DATA QUALITY FLAG

A-72
DO 130 IN=I,B
  NF=IN-1
  NRCTOT=NRCTOT+NRCold(IN)
  NRCold(IN)=NRCold(IN)+NRColdOut(IN)
  NXO=NXO+NOLDX(IN)
  NYO=NYO+NOLDY(IN)
  NZO=NZ0+NOLDZ(IN)
  NBO=NBO+NOLDB(IN)
  NOLDX(IN)=NOLDX(IN)+NOUTX(IN)
  NOLDY(IN)=NOLDY(IN)+NOUTY(IN)
  NOLDZ(IN)=NOLDZ(IN)+NOUTZ(IN)
  NOLDB(IN)=NOLDB(IN)+NOUTB(IN)
  NFOUT=NOLDX(IN)+NOLDY(IN)+NOLDZ(IN)+NOLDB(IN)
130 WRITE(6,212) NF,NOLDX(IN),NOLDY(IN),NOLDZ(IN),NOLDB(IN),NFOUT,
  *NRCold(IN)

C NCO COUNTS TOTAL NUMBER OF MAGNETIC LATITUDE OUTLIER COMPONENTS
C NCO=NXO+NYO+NZO+NBO

C PRINT MAGNETIC LATITUDE OUTLIER COUNTS PER EACH COMPONENT
C WRITE(6,213) NXO,NYO,NZO,NBO,NCO,NRCTOT

C PRINT STATUS OF ENTIRE OUTPUT DATA SET EXISTING ON UNIT IOD ONLY IF
C DESIRED SPACECRAFT OUTPUT WAS PRODUCED DURING THIS RUN (IBTBS = 2)
C IF(IBTBS.NE.2) RETURN

C CALCULATE TOTAL NUMBER OF COMPONENTS (NCOMPD) EXISTING ON UNIT IOD
C EXCLUDING PADDED TIME-GAP RECORDS
C NCOMPD=_W(NRIOD-KCLASS(q,S))

C PRINT STATUS OF ENTIRE OUTPUT DATA SET EXISTING ON UNIT IOD
C WRITE(6,216) IOD
216 FORMAT(//1X,'<TOTAL DESIRED SPACECRAFT FORMATTED OUTPUT DATA CLASS
  XIFICATION EXISTING ON UNIT ',I2,'>')
  WRITE(6,204) (KCLASS(4,KCL),KCL=1,8),NRIOD,NCOMPD
RETURN
END

SUBROUTINE BTOBS(GCLAT, IDIR, EX,
  EY, EZ, FX, FY, FZ, FB, SX,
  SY, SZ, SB)

C SUBROUTINE TO TRANSFORM MAGNETIC FIELD COMPONENTS FROM TOPOCENTRIC
C TO FIT/MAGSAT SPACECRAFT-FIXED BY PERFORMING:
C
  BS=RC*ST*BT

C WHERE BS = FIELD COMPONENTS IN DESIRED SPACECRAFT COORDINATES
C RC = ROTATION MATRIX FROM FIT/MAGSAT TO BS COORDINATES
C ST = ROTATION MATRIX FROM GEOCENTRIC TO FIT/MAGSAT COORDINATES
C BT = FIELD COMPONENTS IN CARTESIAN TOPOCENTRIC COORDINATES

C MATRIX ST = TS' HAS THE FOLLOWING FORM:
C
  ST = ( SIN(ALPHA)/COS(GCLAT)  #COS(ALPHA)*SIN(Delta)  0 )
      ( 0 0 1 )
C
( #COS(ALPHA)*SIN(Delta) -SIN(ALPHA)/COS(GCLAT) 0 )
C
WHERE TS' = INVERSE OF MATRIX TS = TRANSPOSE OF MATRIX TS
C
ALPHA = NEGATIVE COMPLEMENT OF ORBIT INCLINATION
C
GCLAT = GEOCENTRIC LATITUDE
C
DELTA = ARCOS(TAN(GCLAT)*TAN(ALPHA))
C
0 = + FOR ASCENDING AND - FOR DESCENDING SATELLITE DATA
C
BS = (SX,SY,SZ) WHERE SX, SY, AND SZ ARE THE DESIRED SPACECRAFT
C
COMPONENTS
C
BT = (EX,EY,EZ) WHERE EX, EY, AND EZ ARE THE CONVENTIONAL TOPOCENTRIC
C
COMPONENTS, THAT IS, (-BTHETA, BPHI, -BRHO)
C
REAL*8 COSLAT,SINALP,COSALP,SINDEL,SADCL,CAMSD,DTR
DIMENSION EU(3),CA(3,3),QI(3),QF(3),CF(3),RF(3,3),RC(3,3)
COMMON /EPHEMS/ ORBINC,ERAD,IEPDAY,INCREM,INTRVL
COMMON /COTRAN/ EU,CA,QI,QF,CF,RF,RC
C
CALCULATE DEGREES-TO-RADIANS CONVERSION
C
DTR=3.14159265358/180.D0
C
C IF SATELLITE VELOCITY DIRECTION IS INDETERMINABLE (IDIR = 0), THEN
C NO TOPOCENTRIC COMPONENTS HAVE BEEN CALCULATED. SIMPLY USE ORIGINAL,
C UNMODIFIED COMPONENTS IN FIT/MAGSAT COORDINATES DETERMINED IN STEP1
C AS THE OUTPUT FIELD COMPONENTS IN STEP5 BY PERFORMING:
C
FS=BT
C
BT = (EX,EY,EZ) WHERE EX, EY, AND EZ ARE THE ORIGINAL, UNMODIFIED
C
FIT/MAGSAT COMPONENTS
C
IF(IDIR.NE.0) GO TO 10
FX=EX
FY=EY
FZ=EZ
GO TO 20
C
SATELLITE VELOCITY DIRECTION HAS BEEN DETERMINED AND TOPOCENTRIC FIELD
C COMPONENTS HAVE BEEN GENERATED FOR THIS DATA POINT
C
PERFORM: FS=ST_BT
C
FS = (FX,FY,FZ) WHERE FX, FY, AND FZ ARE THE FIT/MAGSAT SPACECRAFT
C COMPONENTS, WHICH ARE PASSED BACK TO STEP5 FOR FURTHER USE
C
DETERMINE NEGATIVE COMPLEMENT ALPHA OF ORBIT INCLINATION ANGLE ORBINC
C
10 ALPHA=ORBINC-90.0
C
DETERMINE NEEDED TRIGONOMETRIC FUNCTIONS OF GCLAT, ALPHA, AND DELTA
C
COSLAT=DCOS(DBLE(GCLAT)*DTR)
SINALP=DSIN(DBLE(ALPHA)*DTR)
COSALP=DCOS(DBLE(ALPHA)*DTR)
SINDEL=DSIN(DACOS(DTAN(DBLE(GCLAT)*DTR)*DTAN(DBLE(ALPHA)*DTR)))
SADCL=SINALP/COSLAT
CAMSD=COSALP*SINDEL
IF(IDIR.EQ.-1) GO TO 30

A-74
C PERFORM TRANSFORMATION IF SATELLITE IS ASCENDING
C
    FX=EX*SADCL+EY*CAMSD
    FZ=EX*CAMSD-EY*SADCL
    GO TO 40
C
C PERFORM TRANSFORMATION IF SATELLITE IS DESCENDING
C
    30 FX=EX*SADCL-EY*CAMSD
    FZ=-EX*CAMSD-EY*SADCL
    40 FY=EZ
C
C PERFORM: BS=RC*FS
C
    20 SX=RC(1,1)*FX+RC(1,2)*FY+RC(1,3)*FZ
    SY=RC(2,1)*FX+RC(2,2)*FY+RC(2,3)*FZ
    SZ=RC(3,1)*FX+RC(3,2)*FY+RC(3,3)*FZ
C
C COMPUTE SCALAR FIELD VALUES IN BOTH FIT/MAGSAT AND DESIRED SPACECRAFT
C COORDINATES
C
    FB=SQR(DFX+FY*FY+FZ*FZ)
    SB=SQR(SDLX+SY*SY+SZ*SZ)
    RETURN
END
SUBROUTINE BSPLYN(TS,TF,N,H,T,ICOV,ICOR,NDCOVM,INTERP,NDERV,ISHOW,*IPRINT,INTV,KNTADJ,ITERMX,LRMAX,EPS,NOBS,KO,EKNOTS,FREQ,X,S,SIG,*V,COEF,D,WTRMS,GSIG,RESID,XINTRP)

IMPLICIT REAL*8(A-H,O-Z)

DIMENSION X(500),S(500),COEF(500),KSKIP(500),ELAM(500),D(13000)
DIMENSION V(5,500),EKNOTS(500),SIG(500),GSIG(5,500),RESID(500)
DIMENSION FREQ(500)

INTEGER H,H2N,T
LDV=1
LDC=0
LCR=0
NSHOW=0
NPRINT=0
MH=H

IF(N.EQ.0) H=0
IF(N.LT.1) GO TO 38
NM1=N-1

274 FORMAT(1X,'*** ATTENTION: NDERV MUST NOT EXCEED ',I2,' ***')
STOP

38 DO 1 I=1,N
ELAM(I)=TS
1 ELAM(N+H+I)=TF
DO 50 II=I,H
50 ELAM(II+N)=EKNOTS(II)
DO 800 NEX=I,H
800 KSKIP(NEX)=0

801 KNUM=MOD(KS,IDIV)
KS=KS/IDIV
IF(KNUM.EQ.0) IDIV=IDIV*10
IF(KNUM.NE.0) KSKIP(KNUM)=N
IF(KS.EQ.0) GO TO 800
GO TO 801

802 NDERVP=NDERV+1
NDCVMP=NDCOVM+1
IPARM=N+H
H2N=IPARM+N
NPARM=IPARM+2*T
IF((IPARM.EQ.0).AND.(T.NE.0)) NPARM=NPARM+1
NPP=NPARM+2
NPL=N+1
IF(NDERV.GE.NDCOVM) GO TO 25
WRITE(6,330) NDERV

330 FORMAT(1X,'*** ATTENTION: NDCOVM MUST NOT EXCEED ',I2,' ***')
STOP

25 IF(NPARM.LE.NOBS) GO TO 26
WRITE(6,331) NPARM,NOBS

331 FORMAT(1X,'*** ATTENTION: NUMBER OF PARAMETERS ',I5,' EXCEEDS AMOUNT OF DATA ',I5,' ***')
STOP

26 IF(INTERP.EQ.1) GO TO 470
IF((ISHOW.EQ.0).OR.(ISHOW.EQ.3)) GO TO 710
WRITE(6,23) TS,N,ICOV,NDCOVM,NDERV,IPRINT,ITERMX,NOBS,TF,MH,*ICOR,INTERP,ISHOW,KNTADJ,LRMAX,EPS,KO,T,INTV

23 FORMAT(1X,'B-SPLINE OUTPUT .......'/1X,'PARAMETERS:'/1X,'TS = ',F15.8,' N = ',I3,' ICOV = ',I3,' NDCOVM = ',I3,' NDERV = ',I3)
F15.8, ' H = ',I3, ' ICOR = ',I3, ' INTERP = ',I3, ' ISHOW = '
X, I15, ' T = ',I3, ' INTV = ',I3//IX,'RAW DATA ....'//5X,'OBS',
XI1X, 'X VALUE',19X,'F(X)',13X,'SIGMA'/)
DO 69 II=I,NOBS
69 WRITE(6,678) II,X(II),S(II),SIG(II)
678 FORMAT(1X, I5,3X, F15.8,3X,F20.10,3X, F15.8)
IF(T.NE.O) WRITE(6,604)
604 FORMAT(/3X,'NUM',7X,'A PRIORI FIT FREQUENCY'//)
DO 33 II=I,T
33 WRITE(6,626) II,FREQ(II)
626 FORMAT(1X,I5,14X,F15.8)
IF(H2N.NE.O) WRITE(6,677)
677 FORMAT(/3X,'NUM',7X,'ORIGINAL KNOT POSITION'//)
DO 66 II=I,H2N
66 WRITE(6,676) II,ELAM(II)
676 FORMAT(1X,I5,14X,F15.8)
IF(IPARM.EQ.O) GO TO 700
CALL CALBSP(ELAM(NP1),NP1,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
WCOEF,D,NPARM,NOPP,H2N,FTRMS,ISING,NP1,NSHOW,NPRINT,INTERP,GSIG,
*RESID,XINTRP,FREQ,T,TS,INTV)
WRITE(6,400) NTRMS
400 FORMAT(/1X,'HEIGHTED RMS
OF FIT = ',F20.10)
710 IF(IPARM.EQ.O) GO TO 700
IF(KNTADJ.EQ.O) GO TO 700
DO 10 KITER=1,ITERMX
DIFMAX=O.DO
DO 20 IUV=NP1,IPARM
IUVEC=IPARM-IUV+NP1
IF(KSKIP(IUVEC-N).EQ.1) GO TO 20
VALMIN=ELAM(IUVEC-1)
VALMAX=ELAM(IUVEC+I)
BPT=ELAM(IUVEC)
IF(IUVEC.NE.NP1) GO TO 110
VALMIN=VALMIN+EPS
CALL CALBSP(VALMIN,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
WCOEF,D,NPARM,NOPP,H2N,FVMIN,ISING,NP1,NSHOW,NPRINT,INTERP,GSIG,
*RESID,XINTRP,FREQ,T,TS,INTV)
IF(ISING.EQ.1) GO TO 110
100 IF(IUVEC.NE.IPARM) GO TO 120
130 VALMAX=VALMAX-EPS
CALL CALBSP(VALMAX,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
WCOEF,D,NPARM,NOPP,H2N,FVMAX,ISING,NP1,NSHOW,NPRINT,INTERP,GSIG,
*RESID,XINTRP,FREQ,T,TS,INTV)
IF(ISING.EQ.1) GO TO 130
120 DPAST=BPT
CALL CALBSP(BPT,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
WCOEF,D,NPARM,NOPP,H2N,FBPT,ISING,NP1,NSHOW,NPRINT,INTERP,GSIG,
*RESID,XINTRP,FREQ,T,TS,INTV)
DOLD=FBPT
STEP=10.DO*EPS
140 IF(VALHI.LT.VALMAX) GO TO 150
CPT=VALMAX
IF(IUVEC.EQ.IPARM) GO TO 145
CALL CALBSP(CPT,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
WCOEF,D,NPARM,NOPP,H2N,FCPT,ISING,NP1,NSHOW,NPRINT,INTERP,GSIG,
*RESID,XINTRP,FREQ,T,TS,INTV)
GO TO 160

A-77
5 FCPT=FVMAX
GO TO 160

150 CALL CALBSPL(VALHI, IUVEC, ELAM, N, H, X, S, SIG, NOBS, V, LDV, LCV, LDC, LCR,
* COEF, D, NPARM, NOPP, H2N, DNEW, ISING, NP1, NSHOW, NPRINT, INTERP, GSIG,
* RESID, XINTRP, FREQ, T, TS, INTV)
IF(DNEW.GT.DOLD) GO TO 170
DOLD=DNEW
STEP=STEP+STEP
GO TO 140

170 CPT=VALHI
FCPT=DNEW

160 DOLD=FBPT
STEP=10.D0*EPS

180 VALLO=BPT-STEP
IF(VALL0.GT.VALMIN) GO TO 190
APT=VALMIN
IF(IUVEC.EQ.NP1) GO TO 195
CALL CALBSPL(APT, IUVEC, ELAM, N, H, X, S, SIG, NOBS, V, LDV, LCV, LDC, LCR,
* COEF, D, NPARM, NOPP, H2N, FAPT, ISING, NP1, NSHOW, NPRINT, INTERP, GSIG,
* RESID, XINTRP, FREQ, T, TS, INTV)
GO TO 200

195 FAPT=FVMIN
GO TO 200

190 CALL CALBSPL(VALL0, IUVEC, ELAM, N, H, X, S, SIG, NOBS, V, LDV, LCV, LDC, LCR,
* COEF, D, NPARM, NOPP, H2N, DNEW, ISING, NP1, NSHOW, NPRINT, INTERP, GSIG,
* RESID, XINTRP, FREQ, T, TS, INTV)
IF(DNEW.GT.DOLD) GO TO 210
DOLD=NEW
STEP=STEP+STEP
GO TO 180

210 APT=VALLO
FAPT=DNEW

200 CALL LAGRAN(APT, BPT, CPT, DPT, ELAM, IUVEC, N, H, X, S, SIG, NOBS, V, LDV, LCV,
* LDC, LCR, COEF, D, NPARM, H2N, EPS, LGRMAX, NOPP, NP1, FAPT, FBPT, FCPT, FDPT,
* KITER, ISHOW, GSIG, RESID, XINTRP, FREQ, T, TS, INTV)
DIFLAM=DABS(DPAST-DPT)
ELAM(IUVEC)=DPT
IF(DIFLAM.GT.DIFMAX) DIFMAX=DIFLAM

20 CONTINUE
10 IF(DIFMAX.LE.EPS) GO TO 30
IF((ISHON.EQ.1).OR.(ISHON.EQ.2)) WRITE(6,119) ITERMX
119 FORMAT(/1X,'ADJUSTED KNOT POSITIONS ARE BEST FOR MAXIMUM ITERATION
* NUMBER OF ',I2)
GO TO 55

30 IF((ISHON.EQ.1).OR.(ISHON.EQ.2)) WRITE(6,555) KITER
555 FORMAT(/1X,'ADJUSTED KNOT POSITIONS CONVERGED AFTER ',I2, ' ITERATIONS')

55 IF((ISHON.EQ.0).OR.(ISHON.EQ.3)) GO TO 700
WRITE(6,122)
122 FORMAT(/3X,'NUM',7X,'ADJUSTED KNOT POSITION'//)
DO 72 II=1,H2N
72 WRITE(6,644) II, ELAM(II)

644 FORMAT(1X,15,1X,F15.8)
CALL CALBSPL(ELAM(NP1), NP1, ELAM, N, H, X, S, SIG, NOBS, V, LDV, LCV, LDC, LCR,
* COEF, D, NPARM, NOPP, H2N, WTRMS, ISING, NP1, NSHOW, NPRINT, INTERP, GSIG,
* RESID, XINTRP, FREQ, T, TS, INTV)
WRITE(6,600) WTRMS

700 CALL CALBSPL(ELAM(NP1), NP1, ELAM, N, H, X, S, SIG, NOBS, V, NDERVP, ICOV,
* NDCVMP, ICOV, COEF, D, NPARM, NOPP, H2N, FLAST, ISING, NP1, ISHOW, IPRINT,
* INTERP, GSIG, RESID, XINTRP, FREQ, T, TS, INTV)

A-78
DO 40 II=1,H
40 EKNOTS(II)=ELAM(II+N)
RETURN
470 IF((ISHOW.EQ.1).OR.(ISHOW.EQ.3)) WRITE(6,520) XINTRP
520 FORMAT(/1X,'** B-SPLINE INTERPOLATION: X = ',F15.8,' **/
CALL CALBSP(ELAM(NP1),NP1,ELAM,N,H,X,S,SIG,NOBS,V,NDERVP,ICOV,
  *NDCVMPI,ICOR,COEF,D,NPARM,NOPP,H2N,FLAST,ISING,NP1,ISHOW,IPRINT,
  *INTERP,GSIG,RESID,XINTRP,FREQ,T,TS,INTV)
RETURN
END
SUBROUTINE LAGRAN(CAPT,BPT,CPT,DPT,ELAM,IUVEC,N,H,X,S,SIG,NOBS,V,
  *LDV,LCV,LDC,LCR,COEF,D,NPARM,H2N,EPS,LGRLM,X,NOBS,FBPT,
  *FLPT,FDPT,KITER,ISHON,GSIG,RESID,XINTRP,FREQ,T,TS,INTV)
IMPLICIT
REAL_A(H,O-Z)
DIMENSION
X(500),S(500),COEF(500),ELAM(500),V(500),SIG(500)
DIMENSION
D(15000),GSIG(5,500),RESID(500),FREQ(500)
INTEGER
H,H2N,T
NSHON=0
NPRINT=0
INTERP=0
DO 1 ITER=1,LGRLM
  DENOM=(BPT-CPT)*FAPT+(CPT-APT)*FBPT+(APT-BPT)*FCPT
  IF(DENOM.LT.0.D0) GO TO 50
  IF(FAPT.LT.FCPT) GO TO 10
  APT=BPT
  FAPT=FBPT
  BPT=(APT+CPT)/2.D0
  DOLD=BPT
  IF(DABS(CPT-BPT).GT.EPS) GO TO 2
  DPT=BPT
  RETURN
  2 CALL CALBSP(BPT,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
    *COEF,D,NPARM,H2N,FBPT,ISING,NP1,NSHON,NPRINT,INTERP,GSIG,
    *RESID,XINTRP,FREQ,T,TS,INTV)
  GO TO 1
  10 CPT=BPT
    FCPT=FBPT
    BPT=(APT+CPT)/2.D0
    DOLD=BPT
    IF(DABS(BPT-APT).GT.EPS) GO TO 3
    DPT=BPT
    RETURN
  3 CALL CALBSP(BPT,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
    *COEF,D,NPARM,H2N,FBPT,ISING,NP1,NSHON,NPRINT,INTERP,GSIG,
    *RESID,XINTRP,FREQ,T,TS,INTV)
    GO TO 1
  50 DPT=0.5D0*((BPT**2-CPT**2)*FAPT+(CPT**2-APT**2)*FBPT+(APT**2-
    *BPT**2)*FCPT)/DENOM
    IF(ITER.EQ.1) GO TO 4
    IF(DABS(DOLD-DPT).LE.EPS) RETURN
  4 DOLD=DPT
    CALL CALBSP(DPT,IUVEC,ELAM,N,H,X,S,SIG,NOBS,V,LDV,LCV,LDC,LCR,
      *COEF,D,NPARM,H2N,FDPT,ISING,NP1,NSHON,NPRINT,INTERP,GSIG,
      *RESID,XINTRP,FREQ,T,TS,INTV)
    IF(CAPT.LE.DPT.AND.DPT.LE.BPT) GO TO 5
    IF(BPT.LE.DPT.AND.DPT.LE.CPT) GO TO 6
    IF(DPT.LT.APT) GO TO 7
    IF(DPT.GT.CPT) GO TO 8
    5 IF(FDPT.LE.FBPT) GO TO 9
    APT=DPT
FAPT = FDPT
GO TO 1

9 CPT = BPT
FCPT = FBPT
BPT = DPT
FBPT = FDPT
GO TO 1

IF (FDPT .LE. FBPT) GO TO 12
CPT = DPT
FCPT = FDPT
GO TO 1

12 APT = BPT
FAPT = FBPT
BPT = DPT
FBPT = FDPT
GO TO 1

7 DPT = APT
FDPT = FAPT
RETURN

8 DPT = CPT
FDPT = FCPT
RETURN

1 CONTINUE
IF ((ISHON .EQ. 1).OR. (ISHOH .EQ. 2)) WRITE (6, 100) LGRMAX, IUVEC, KITER
100 FORMAT (// 1X, 'WARNING: LAGRANGIAN INTERPOLATION DID NOT CONVERGE WITHIN ', I2, ' STEPS FOR KNOT NUMBER ', I2, ' AT ITERATION ', I2)
RETURN
END

SUBROUTINE CALBSP (PNT, IUVEC, ELAM, N, H, X, S, NOBS, V, NDERVP, ICOV,
* NDCVMPI, ICOR, COEF, D, NPARM, NPP, H2N, MTRMS, ISING, NP1, ISHOW, IPRINT,
* INTERP, GSIG, RESID, XINTRP, FREQ, T, TS, INTV)
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION F(500), X(500), S(500), COEF(500), RESID(500), IST(500)
DIMENSION ELAM(500), DIAG(500), CSUM(500), FCT(500), GSIG(5, 500)
DIMENSION V(5, 500), SIG(500), XARRAY(6, 500, 20), D(13000), G(13000)
DIMENSION FREQ(500)
INTEGER H, H2N, T
INDX(NROW, NCOL, NDIM) = (NROW*(NDIM+NDIM+5-NROW))/2+NCOL-NDIM-2
IPARM = H+N
JPARM = NPARM-IPARM
ELAM(IUVEC) = PNT
IF (INTERP .EQ. 1) GO TO 560
MAXD = (NPARM*(NPARM+1))/2+2*NPARM
DO 9 II = 1, MAXD
   D(II) = 0.0 DO
   DO 10 II = 1, NPARM
   F(II) = 0.0 DO
   DO 100 NOB = 1, NOBS
      WT = 1.0 DO
      IF (SIG(NOB) .NE. 0.0) WT = 1.0/SIG(NOB)
      IF (N .NE. 0) CALL CALMTX (X(NOB), H2N, N, IPARM, ELAM, I, NDERVP, XARRAY,
         NOB)
      IST(NOB) = I
      DO 110 ILOP = 1, N
         110 F(I+ILOP-1) = XARRAY(1, NOB, N-ILOP+1)*WT
      IF (T .NE. 0) CALL CALTRG (T, TS, N, NOB, X(NOB), FREQ, XARRAY, NDERVP)
      IF ((IPARM .EQ. 0) .AND. (T .NE. 0)) XARRAY (1, NOB, NPARM) = 1.0 DO
      DO 123 ILOP = 1, JPARM
         123 F(IPARM+ILOP) = XARRAY(1, NOB, N+ILOP)*WT
F(NPARM+1)=S(NDB)wNT
CALL CALNOR(D,F,NPARM)

120 F(I+ILOP-1)=0.DO
DO 121 ILOP=1,N
121 F(IPARM+ILOP)=0.DO
DO 100 F(NPARM+1)=0.DO

DO 176 I=1,NPARM
CSUM(I)=0.DO
DO 176 J=1,NPARM
NTOT=I+J
NRON=MIN0(I,J)
NCOL=NTOT-NROW
K=INDX(NROW, NCOL, NPARM)
176 CSUM(I)=CSUM(I)+D(K)
IF(N.EQ.0) IUVEC=2
CALL CALINV(NPARM,NOPP,D,DIAG,NPI,IUVEC,ISING)

885 DIAG(I)=DIAG(I)+D(K)*CSUM(J)
IF(I.LE.IPARM) WRITE(6,990) I,COEF(I),DIAG(I)

990 FORMAT(1X,I5,2(3X,F20.10))
DO 890 ID=I,NDERVP
V(ID, NOB)=FCT(ID)
CALL CALVAR(NOBS,IST,XARRAY,D,N,NPARM,G,ICOV,NDCVMP,ICOR,GSIG,
*ISHOW,INTERP,JPARM)
IF((ISHOW.EQ.1).OR.(ISHOW.EQ.2)) WRITE(6,951)
951 FORMAT(//IX,'B-SPLINE FITS ..... '//3X,'OBS',BX,'X VALUE',15X,'SO(X )',15X,'S1(X)',15X,'S2(X)',15X,'S3(X)...'//)
RMEAN=0.DO
RSS=0.DO
DO 819 NOB=1,NOBS
RESID(NOB)=S(NOB)-V(1,NOB)
IF(NOB.EQ.1) RESMIN=RESID(NOB)
IF(NOB.EQ.1) RESMAX=RESID(NOB)
IF(RESID(NOB).LT.RESMIN) RESMIN=RESID(NOB)
IF(RESID(NOB).GT.RESMAX) RESMAX=RESID(NOB)
IF((ISHOW.EQ.1).OR.(ISHOW.EQ.2)) WRITE(6,995) NOB,X(NOB),V(1,NOB),
*RESID(NOB),V(ID,NOB),ID=2,NDERVP
995 FORMAT(1X,I5,F15.8,5F20.10)
RMEAN=RMEAN+RESID(NOB)
WT=1.DO
IF(SIG(NOB).NE.0.DO) WT=1.DO/SIG(NOB)

819 RSS=RSS+(RESID(NOB)*WT)**2
RESINC=(RESMAX-RESMIN)/REAL(INTV)
RMEAN=RMEAN/NOBS
WTRMS=DSQRT(RSS/NOBS)
IF((ISHOW.EQ.1).OR.(ISHOW.EQ.2)) CALL CALSTA(WTRMS,RMEAN,RESID,*NOBS,RESMAX,RESMIN,RSTDV)
IF(IPRINT.GT.0) CALL BSPLT(IPRINT,X,S,V,GSIG,NOBS,NDCVMP,NDERV, *ELAM,H,N,RESID,INTV,RESMIN,RESINC,RMEAN,RSTDV)
RETURN

360 NOB=1
IF(N.NE.0) CALL CALMTX(XINTRP,H2N,N,IPARM,ELAM,I,NDERVP,XARRAY,*NOB)
IF(T.NE.0) CALL CALTRG(T,TS,N,NOB,XINTRP,FREQ,XARRAY,NDERV)
CALL CALCOF(FCT,COEF,XARRAY,N,NDERVP,1,NOB,NPARM,IPARM)
IST(NOB)=I
CALL CALVAR(NOB,IST,XARRAY,D,N,NPARM,G,ICOV,NDCVMP,ICOR,GSIG,*ISHOW,INTERP,IPARM)
DO 893 ID=I,NDERVP
V(ID,1)=FCT(ID)
IF((ISHON.EQ.O).OR.(ISHON.EQ.2)) GO TO 893
GVAR=GSIG(ID,1)**2
ID-1,V(ID,1),GSIS(ID, I),GVAR
491 FORMAT(1X,'SI,II,t(X) VALUE = ',FZO.IO,3X, mSIGMA = ',F20.10,3X,'VARIANCE = ',F20.10)
IF(ID.GT.NDCVMP) NRITE(6,_91)
492 FORMAT(1X,'S',II,'(X) VALUE = ',F20.10)
893 CONTINUE
RETURN
END

SUBROUTINE CALMTX(X,H2N,N,NPARM,ELAM,I,NDERVP,ARRAY,NOB)
IMPLICIT REALW8(A-H,O-Z)
DIMENSION ELAMC500),ARRAYC6,BOO,ZO),ARNNJJ(6)
INTEGER HZN
DATA ARNNJJ/6WO.DO/
INDXX(NRON,NCOL,NDIM)=(NRONW(NDIM+NDIM+Z-NRON))/2+NCOL-NDIM
NPARMI=NPARM+I
CALL BSERCH(N,NPARMI,ILAM,X,ELAM, I)
DO 747 ID=1,NDERVP
V(ID,1)=FCT(ID)
IF((ISHOW.EQ.0).OR.(ISHOW.EQ.2)) GO TO 747
GVAR=GSIG(ID,1)**2
IF(ID.LE.NDCVMP) WRITE(6,491) ID-1,V(ID,1),GSIG(ID,1),GVAR
491 FORMAT(1X,'S',II,'(X) VALUE = ',FZO.IO,3X, mSIGMA = ',F20.10,3X,'VARIANCE = ',F20.10)
IF(ID.GT.NDCVMP) WRITE(6,492) ID-1,V(ID,1)
492 FORMAT(1X,'S',II,'(X) VALUE = ',F20.10)
747 CONTINUE
RETURN
END
IF(ID.EQ.1) GO TO 35
IF(JJ.NE.1) ARIJM1=ARRAY(ID-1,NOB,INDJ1)
IF(NN.NE.JJ) ARIJEQ=ARRAY(ID-1,NOB,INDJEQ)
35 ARNNJJ(ID)=(X-ELAM(IILAM-NN))*ARIJM1+(ELAM(IILAM)-X)*ARIJEQ+(DFLOAT( *ID-1)*(ARIJM1-ARIJEQ))
210 IF(DABS(ARNNJJ(ID)).LE.1.D-25) ARNNJJ(ID)=0.DO
DO 177 ID=1,NDERVP
IF(JJ.EQ.1) ARRAY(ID,NOB,INDJEQ+1)=ARNNJJ(ID)/DEL
177 IF(JJ.NE.1) ARRAY(ID,NOB,INDJM1)=ARNNJJ(ID)/DEL
150 CONTINUE
100 IILAM=IILAM+1
200 RETURN
END
SUBROUTINE CALCOF(FCT,COEF,ARRAY,N,NDERVP,I,NOB,NPARM,IPARM)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION FCT(1),COEF(1),ARRAY(6,500,20)
DO 200 ID=I,NDERVP
200 FCT(ID)=0.DO
DO 300 K=I,N
DO 300 ID=I,NDERVP
300 FCT(ID)=FCT(ID)+COEF(I+K-1)*ARRAY(ID,NOB,N-K+1)
JPARM=NPARM-IPARM
IF(JPARM.EQ.0) RETURN
DO 400 L=1,JPARM
DO 400 ID=I,NDERVP
400 FCT(ID)=FCT(ID)+COEF(IPARM+L)*ARRAY(ID,NOB,N+L)
RETURN
END
SUBROUTINE CALNOR(D,F,NPARM)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION D(1),F(1)
NOP=NPARM+I
K=1
DO 27 I=1,NPARM
FNT=F(I)
NLENG=NOP-I+I
DO 28 J=I,NLENG
28 D(K+J-1)=D(K+J-I)+FNT*F(I+J-1)
27 K=K+NLENG+1
RETURN
END
SUBROUTINE CALINV(LL,MM,A,R,NPI,IUVEC,ISING)
DOUBLE PRECISION A(1),DPIV,DSUM,A2,R(I)
IDIGL=0
LTROW=1
IF(LL.LT.1)GO TO 900
LLL=LL-1
K1=0
LM=MM-LL
IND=LM
DO 90 K=1,LLL
IND=IND+LM
KPIV=IND+1
LEND=K-1
TOL=A(KPIV)
DO 80 I=K,LLL
IND=IND+1
DSUM=0.DO
IF(LEND)30,30,10
80 LANF=K
LIND=I-K
DO 20 L=I,LEND
DSUM=DSUM+A(LANF)*A(LANF+LIND)
20 LAF=LANF+MM-L
30 DSUM=A(IND)-DSUM
IF(I.NE.K)GO TO 70
IF(DSUM)>900,900,40
40 IDIG=ALOG10(TOL/SNGL(DSUM))-0.5
IF(IDIG.LE.IDIGL)GO TO 60
IDIGL=IDIG
LTOW=I
60 DPIV=DSQRT(DSUM)
A1=(1.DO/DPIV)
A2=(1.DO-DBLE(A1)*DPIV)/DPIV
A(IND)=DPIV
R(K)=DPIV
GO TO 80
70 A(IND)=A2*DSUM+DBLE(A1)*DSUM
80 CONTINUE
90 CONTINUE
DO 152 K=I,LL
DPIV=A(KPIV)
A1=(1.DO/DPIV)
A2=(1.DO-DBLE(A1)*DPIV)/DPIV
A(KPIV)=A2+DBLE(A1)
R(K)+K+1=A(KPIV)
LEND=K-1
IF(LEND)130,130,110
110 DO 120 L=I,LEND
IND=KPIV+L
120 A(IND)=-A2*A(IND)+DBLE(A1)*A(IND)
130 IF(K.EQ.LL)GO TO 152
IND=KPIV
KPIV=KPIV-LM-I-K
LANF=IND
DO 151 I=K,LL1
LANF=LANF-LM-I
DSUM=A(LANF)
A(LANF)=A2*DSUM+DBLE(A1)*DSUM
IF(LEND)151,151,140
140 DO 150 L=I,LEND
LIND=LANF+L
150 A(LIND)=A(LIND)+DSUM*A(IND+L)
151 CONTINUE
152 CONTINUE
DO 180 K=I,LL
LIND=KPIV-1
LANF=KPIV
DO 170 I=K,LL
DSUM=0.DO
DO 160 L=KPIV,IND
LIND=LIND+1
160 DSUM=DSUM+A(L)*A(LIND)
A(KPIV)=DSUM
LIND=LIND+LM
170 KPIV=KPIV+1
KPIV=KPIV+LM
180 IND=IND+MM-K
ISING=0
RETURN
A-84
900 IF((IUVEC.EQ.NP1).OR.(IUVEC.EQ.LL)) GO TO 700
  IDIGL=-1
  LTROW=I
  WRITE(6,920) LTROW
920 FORMAT(5X,'*** INVERSION FAILED AT ROW ',I3,' ***')
  STOP 13
700 ISING=1
  RETURN
END
SUBROUTINE CALVAR(INUM,IST,ARRAY,D,N,NPARM,G,ICOV,NDCVMP,ICOR,
  *GSIG,ISHOW,INTERP,JPARM)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION G(13000),IST(I),ARRAY(6,500,20),D(1),GSIG(5,500)
  INDX(NROW,NCOL,NDIM)=(NROW*(NDIM+NDIM+5-NROW))/2+NCOL-NDIM-2
  INDX(NROW,NCOL,NDIM)=(NROW*(NDIM+NDIM+1-NROW))/2+NCOL-NDIM
  MPARM=N+JPARM
  DO 10 LO=1,NDCVMP
     IF((INTERP.EQ.0).AND.(ISHOW.NE.0).AND.(ISHOW.NE.3)) WRITE(6,327)
     *L0=1
327 FORMAT(/1X,'S',II,'(X) COVARIANCE MATRIX ..... CORRELATION MAT
     *RIX ..... '//4X,'I',3X,'J',ZOX,'COV(I,J)',IOX,'SIGMAS''IOX' wCOR(I,J)
     //)
  KOUNT=0
  DO 11 LI=1,INUM
     IF(NPARM.NE.JPARM) ISTLMI=IST(LI)-1
     LIM=INUM
     IF(ICOV.EQ.0) LIM=L1
     DO 11 L2=L1,LIM
        KOUNT=KOUNT+1
        G(KOUNT)=0.DO
        DO 12 JJ=1,MPARM
           IF(JJ.LE.N) JJPOS=ISTLMI+JJ
           IF(JJ.GT.N) JJPOS=NPARM-MPARM+JJ
           IF(JJ.LE.N) ARR2=ARRAY(LO,L2,N-JJ+I)
           IF(JJ.GT.N) ARR2=ARRAY(LO,L2,JJ)
           SUM=SUM+ARR2*D(K)
        12 SUM=SUM+ARR1*D(K)
        G(KOUNT)=G(KOUNT)+ARR2*SUM
        KOWNT=0
        DO 13 IRS=1,INUM
           LIM=INUM
           IF(ICOV.EQ.0) LIM=IRS
           KI=INDXX(IRS,IRS,INUM)
           IF(ICOV.EQ.0) KI=IRS
           GIPIV=G(KI)
           DO 13 ICS=IRS,LIM
              KJ=INDXX(ICS,ICS,INUM)
              IF(ICOV.EQ.0) KJ=IRS
              GJPIV=G(KJ)
              KIJ=INDXX(IRS,ICS,INUM)

A-85
IF(ICOV.EQ.0) KIJ=IRS
GIJ=G(KIJ)
IF(IRS.EQ.ICS) KOWNT=KOWNT+1
IF(IRS.EQ.ICS) GSIG(LO,KOWNT)=DSQRT(GIJ)
IF(INTERP.EQ.1) RETURN
IF(ICOR.EQ.1) GCOR=GIJ/DSQRT(GIPIV*GPIV)
IF((ISHOW.EQ.0).OR.(ISHOW.EQ.3)) GO TO 13
IF(ICOR.EQ.0) GO TO 27
IF(IRS.EQ.ICS) WRITE(6,224) IRS,ICS,GIJ,GSIG(LO,KOWNT),GCOR
224 FORMAT(1X,2I4,3X,F15.8,5X,F11.8,3X,F15.8)
IF(IRS.NE.ICS) WRITE(6,225) IRS,ICS,GIJ,GCOR
225 FORMAT(1X,2I4,3X,F15.8,19X,F15.8)
GO TO 13
27 IF(IRS.EQ.ICS) WRITE(6,226) IRS,ICS,GIJ,GSIG(LO,KOWNT)
226 FORMAT(1X,2I4,3X,F15.8,5X,F11.8)
IF(IRS.NE.ICS) WRITE(6,227) IRS,ICS,GIJ
227 FORMAT(1X,2I4,3X,F15.8)
13 CONTINUE
10 CONTINUE
RETURN
END

SUBROUTINE BSEARCH(N,NPARM1,ILAM,X,ELAM,I)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ELAM(1)
IBEG=N
IEND=NPARM1
30 MID=(IBEG+IEND)/2
IF(IEND-IBEG.LE.I) GO TO 40
IF(X.GE.ELAM(MID)) IBEG=MID
IF(X.LT.ELAM(MID)) IEND=MID
GO TO 30
40 ILAM=MID+I
I=ILAM-N
RETURN
END

SUBROUTINE BSPLOT(IFLAG,X,S,V,GSIG,NOBS,NDCVMP,NDERVP,ELAM,H,N,*RESID,INTV,RESMIN,RESINC,RMEAN,RSTDV)
IMPLICIT REAL*4(A-H,O-Z)
CHARACTER*1 SYMBOL(5)
LOGICAL*1 IXFMT(7),IYFMT(7),DVNUM
REAL*8 X(500),S(500),V(5,500),GSIG(5,500),ELAM(500),RESID(500)
REAL*8 RESMIN,RESINC,RMEAN,RSTDV
DIMENSION XS(500),SS(500),VS(500),GSIGS(500),ENOTX(100),ENOTY(100)
DIMENSION RESIDS(500),EK(5,100),KK(5)
INTEGER H
EXTERNAL GAUSS
DATA SYMBOL /'S','D','T','Q','V'/
IF(IFLAG.EQ.1) CALL PLOTST(00001,1)
IF(IFLAG.EQ.2) CALL PLOTST(02000,4)
IF(IFLAG.EQ.3) CALL PLOTST(02001,4)
IF(IFLAG.EQ.1) PHORX=66.0
IF(IFLAG.EQ.1) PHORY=63.0
IF(IFLAG.GT.1) PHORX=4.5
IF(IFLAG.GT.1) PHORY=4.8
DO 5 II=1,NOBS
XS(II)=X(II)
5 SS(II)=S(II)
CALL GRDNUM(XS,NOBS,XMIN,XMAX,LINT,IXFMT)
CALL NOTPOS(IFLAG,N,H,XMAX,XMIN,ELAM,LINT,KK,EK)
DO 10 II=1,NDERVP
DO 20 JJ=I,NOBS
20 VS(JJ)=V(II,JJ)
IF(IFLAG.EQ.1) CALL SETGRID(11.0,10.0,123.0,60.0,1)
IF(IFLAG.GT.1) CALL SETGRID(1.0,1.0,9.0,4.0,4)
IF(II.EQ.1) GO TO 25
CALL GRDNUM(VS,NOBS,PMIN,PMAX,MINT,IYFMT)
GO TO 26
25 CALL MAXMIN(SS,NOBS,YMIN1,YMAX1)
CALL MAXMIN(VS,NOBS,YMIN2,YMAX2)
YMIN=YMIN1
IF(YMIN2.LT.YMIN1) YMIN=YMIN2
YMAX=YMAX1
IF(YMAX2.GT.YMAX1) YMAX=YMAX2
CALL PTYNUM(YMIN,YMAX,PMIN,PMAX,MINT)
CALL FORMAT(PMIN,PMAX,IYFMT)
26 CALL OGRID(XMIN,XMAX,LINT,IXFMT,1,PMIN,PMAX,MINT,IYFMT,2,0)
IF(II.EQ.1) CALL PLOT(XS,SS,NOBS,'X')
DO 59 JL=1,LTYPE
NKNT=KK(JL)
DO 44 IK=1,NKNT
ENOTX(IK)=EK(JL,IK)
ENOTY(IK)=PMIN
44 IF(KK(JL).GT.0) CALL PLOT(ENOTX,ENOTY,NKNT,SYMBOL(JL))
IIM1=II-1
CALL EDIT(IIM1,'II'),DVNUM,NNUM,IBL)
CALL HORLIN('B-SPLINE FIT: DERIVATIVE = ',27,PHORX,PHORY,0,0)
CALL HORLIN(DVNUM,1,PHORX,PHORY,28,0)
10 CALL FRMADV
DO 30 II=1,NDCVMP
DO 40 JJ=1,NOBS
40 GSIGS(JJ)=GSIG(II,JJ)
IF(IFLAG.EQ.1) CALL SETGRID(11.0,10.0,123.0,60.0,1)
IF(IFLAG.GT.1) CALL SETGRID(1.0,1.0,9.0,4.0,4)
CALL GRDNUM(GSIGS,NOBS,YMIN3,YMAX3,KINT,IYFMT)
CALL OGRID(XMIN,XMAX,LINT,IXFMT,1,YMIN3,YMAX3,KINT,IYFMT,2,0)
CALL PLOT(XS,GSIGS,NOBS,'')
DO 75 JL=1,LTYPE
NKNT=KK(JL)
DO 74 IK=1,NKNT
ENOTX(IK)=EK(JL,IK)
ENOTY(IK)=YMIN3
74 IF(KK(JL).GT.0) CALL PLOT(ENOTX,ENOTY,NKNT,SYMBOL(JL))
IIM1=II-1
CALL EDIT(IIM1,'II'),DVNUM,NNUM,IBL)
CALL HORLIN('SIGMA PER OBSERVATION: DERIVATIVE = ',36,PHORX,PHORY,0,0)
CALL HORLIN(DVNUM,1,PHORX,PHORY,37,0)
30 CALL FRMADV
DO 50 JJ=1,NOBS
50 RESIDS(JJ)=RESID(JJ)
IF(IFLAG.EQ.1) CALL SETGRID(11.0,10.0,123.0,60.0,1)
IF(IFLAG.GT.1) CALL SETGRID(1.0,1.0,9.0,4.0,4)
CALL GRDNUM(RESIDS,NOBS,YMIN4,YMAX4,LINT,IYFMT)
CALL OGRID(XMIN,XMAX,LINT,IXFMT,1,YMIN4,YMAX4,LINT,IYFMT,2,0)
CALL PLOT(XS,RESIDS,NOBS,'')
DO 17 JL=1,LTYPE
NKNT=KK(JL)
DO 84 IK=1,NKNT
ENOTX(IK)=EK(JL,IK)
17 IF(KK(JL).GT.0) CALL PLOT(ENOTX,ENOTY,NKNT,SYMBOL(JL))
IIM1=II-1
CALL EDIT(IIM1,'II'),DVNUM,NNUM,IBL)
CALL HORLIN('RESIDUALS: DERIVATIVE = ',30,PHORX,PHORY,0,0)
CALL HORLIN(DVNUM,1,PHORX,PHORY,30,0)
10 CALL FRMADV
ENOTY(IK) = YMIN

IF(KK(JL).GT.O) CALL PLOT(ENOTX,ENOTY,NKNT,SYMBOL(JL))
CALL FRMADV
INTVP1 = INTV+1
ELM = RESMIN
DO 60 II = I, INTVP1
   XS(II) = ELM
   SS(II) = 0.0
60  ELM = ELM + RESINC
RMAX = 0.0D0
DO 70 II = I, NOBS
   DO 70 JJ = I, INTV
      IF((RESIDS(II).GE.XS(JJ)).AND.(RESIDS(II).LE.XS(JJ+1)))
       SS(JJ+1) = SS(JJ+1) + I.0
      IF(SS(JJ+1).GT.RMAX)
       RMAX = SS(JJ+1)
70   DO 80 II = I, NOBS
       ENOTX(II) = (XS(II) + XS(II + 1))/Z.0
       ENOTY(II) = GAUSS(RMEAN, RSTDV, XS(II), XS(II + 1), NOBS)
      IF(ENOTY(II).GT.RMAX)
       RMAX = ENOTY(II)
80   MRX = NINT(RMAX)
      IF(IFLAG.EQ.1) CALL SETGRD(11.0, 10.0, 123.0, 60.0, 1)
      IF(IFLAG.GT.1) CALL SETGRD(1.0, 1.0, 9.0, 4.0, _)
      CALL OGRID(XS(1), XS(INTVP1), INTV, _)
      CALL VERHST(XS, SS, INTVP1)
      CALL PLOT(ENOTX, ENOTY, INTV, _)
      CALL HORLIN('RESIDUAL DISTRIBUTION (NORMAL=X)', 32, PHORX, PHORY, 0.0)
      CALL ENDPLT
RETURN
END

SUBROUTINE NOTPOS(IFLAG, N, NH, XMAX, XMIN, ELAM, LTYPEP, KK, EK)
REAL*8 ELAM(500)
DIMENSION EK(5, 100), KK(5)
DO 10 II = 1, 5
10  KK(II) = 0
   LTYPEP = 1
   IF(IFLAG.EQ.1) TINC = (XMAX - XMIN)/112.0
   IF(IFLAG.GT.1) TINC = (XMAX - XMIN)/80.0
   II = 0
20  II = II + 1
   IF(II.GT.NH) GO TO 50
   SUMK = REAL(ELAM(II + N))
   JJ = 0
30  JJ = JJ + 1
   IF(II + JJ.GT.NH) GO TO 40
   IF(ABS(REAL(ELAM(II + N) - ELAM(II + N + JJ))).GT.TINC) GO TO 40
   SUMK = SUMK + REAL(ELAM(II + N + JJ))
   GO TO 30
40  KK(JJ) = KK(JJ) + 1
   IF(LTYPEP.LT.JJ) LTYPEP = JJ
   EK(JJ, KK(JJ)) = SUMK/REAL(JJ)
   II = II + JJ - 1
   GO TO 20
50  RETURN
END

SUBROUTINE CALTRG(NTRIG, TS, N, NOB, XOB, FREQ, XARRAY, NDERVP)
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION FREQ(500), XARRAY(6, 500, 20)
DATA TNOPI / 6.283185307179579/
OBS = XOB - TS
IPARTL=N
DO 10 IFRQ=1,NTRIG
OMEGA=TWOPITAL1*FREQ(IFRQ)
THETA=OMEGA*NOBS
DO 20 I=1,NDERVP
IF(MOD(I,2).EQ.0) GO TO 30
XARRAY(I,NOB,IPARTL+1)=OMEGA*(I-1)*COS(THETA)
XARRAY(I,NOB,IPARTL+2)=OMEGA*(I-1)*SIN(THETA)
GO TO 40
30 XARRAY(I,NOB,IPARTL+1)=OMEGA*(I-1)*SIN(THETA)
XARRAY(I,NOB,IPARTL+2)=OMEGA*(I-1)*COS(THETA)
GO TO 20
40 IREM=MOD(I,4)
GO TO (20,50,60),IREM
XARRAY(I,NOB,IPARTL+2)=-XARRAY(I,NOB,IPARTL+2)
GO TO 20
50 XARRAY(I,NOB,IPARTL+1)=-XARRAY(I,NOB,IPARTL+1)
GO TO 20
60 XARRAY(I,NOB,IPARTL+1)=-XARRAY(I,NOB,IPARTL+1)
XARRAY(I,NOB,IPARTL+2)=-XARRAY(I,NOB,IPARTL+2)
20 CONTINUE
10 IPARTL=IPARTL+2
RETURN
END
SUBROUTINE CALSATWTRMS,RMEAN,RESID,NOBS,RESMAX,RESMIN,RSTDV
IMPLICIT REAL*(A-H,O-Z)
DIMENSION RESID(500)
RSKEW=0.0D0
RKURT=0.0D0
DO 10 MEXP=2,4
RMOM=0.0D0
DO 20 II=1,NOBS
RMOM=RMOM+(RESID(II)-RMEAN)**MEXP
IF(MEXP.NE.2) GO TO 30
RSTDV=SQRT(RMOM/(NOBS-1))
IF(RSTDV.EQ.0.0D0) GO TO 50
30 IF(MEXP.NE.3) GO TO 50
RSKEW=RMOM/(RSTDV**3*NOBS)
40 IF(MEXP.NE.4) GO TO 10
RKURT=RMOM/(RSTDV**4*NOBS)
10 CONTINUE
50 WRITE(6,100) RESMAX,WTRMS,RSKEW,RMEAN,RSTDV,RKURT,RESMIN
100 FORMAT(//1X,'RESIDUAL STATISTICS..... //1X,'MAXIMUM = ',F20.10,10*
X,'WEIGHTED RMS = ',F20.10,10,'SKEWNESS = ',F20.10/1X,'AVERAGE = 
X',F20.10,10,'STANDARD DEV = ',F20.10,10,'KURTOSIS = ',F20.10/1X, 
X'MINIMUM = ',F20.10)
RETURN
END
FUNCTION GAUSS(RMEAN,RSTDV,X1,X2,NOBS)
IMPLICIT REAL*S (A-H,O-Z)
REAL*8 RMEAN,RSTDV
X3=(X1-RMEAN)/RSTDV/SQRT(2.0)
X4=(X2-RMEAN)/RSTDV/SQRT(2.0)
GAUSS=REAL(NOBS)*0.5*(ERF(X4)-ERF(X3))
RETURN
END
PROGRAM BINSIFT: ............

//XRJRRSFT  JOB (F8002,X22,15), 'BINSIFT', TIME=(1,0), CLASS=O, 00010005
// MSGCLASS=X 00020000
// * JCL = XRJRR.DMSP.PROGRAMS(BINSIFT) 00030005
//STEP1 EXEC FORTRAN,FVPXREF=XREF 00040000
//SYSIN DD * 00050000
00060000

C THIS PROGRAM TAKES DMSP DATA, SORTED INTO EQUAL AREA BINS, C00070000
C AND WEEDS THE NUMBER OF POINTS IN THAT BIN DOWN TO A SPECIFIED C00080000
C NUMBER OF POINTS. A PREVIOUS DELETION OF POINTS WAS MADE ON AN C00090000
C ORBIT-BY-ORBIT BASIS BY T.J. SABAKA'S DMSP PROCESSING PROGRAM. C00100000
C DST VALUES HAVE BEEN ADDED TO THE DATA BY 'DSTADD', INTO SLOT #17. C00110000
C BIN NUMBERS ARE IN SLOT #16, AND THE 'INOTE' FLAG IN SLOT #18. C00120000
C FOR THIS PROGRAM, AN INOTE=0 MEANS GOOD DATA, AND INOTE NE. 0 C00130000
C MEANS BAD DATA. C00140000
C
C THIS PROGRAM CONTINUES THE REJECTION PROCESS. IT FIRST REJECTS C00150000
C POINTS ACCORDING TO DST(RANGE -20 TO +5) AND THEN RANDOMLY REJECTS C00160000
C POINTS IN A BIN UNTIL THE GOAL IS MET. C00170000
C
C THE GOAL IS 9 POINTS PER BIN FOR DIPOLAR LATITUDE GT. 30 DEGREES, C00180000
C AND 3 POINTS PER BIN FOR DIPOLAR LT. 30 DEGREES, FOR DMSP DATA ONLYC00190000
C
C=====================================================================
C00200000
C00210000
C00220000

REAL*8 CENTER, BLKSIZ, CLAT, CLON, DIPLAT
DIMENSION TEMP(28,500),ITEMP(28,500),RA(28),IA(28),OA(28), 00230000
KTEMP(500), CENTER(500,2)
& EQUIVALENCE (TEMP(I,I),ITEMP(I,I)) 00240000
& EQUIVALENCE (RA(1),IA(1)) 00250000
DATA BLKSIZ/10.0/, ISEED/123456/ 00260000
00270000
00280000
00290000
DATA 123456/10.0/, ISEED/123456/
00300000
00310000
00320000
C CALL ZONE AND MIDDLE TO FIND CENTER OF EACH 10X10 BIN. 00330000
CALL ZONE(BLKSIZ) 00340000
CALL MIDDLE(CENTER) 00350000
00360000
00370000
C INITIALIZE RANDOM NUMBER GENERATOR. 00380000
CALL RANDU(ISEED,IY,YOUT) 00390000

C COUNTER VARIABLES:
C NBIN : BIN # 00400000
C IBIN : # OF POINTS IN A BIN 00410000
C IGOAL : # OF POINTS IN BIN AFTER WEEDING PROCESS 00420000
C IBAD : # OF BAD POINTS PER BIN FROM PREVIOUS WEEDING. 00430000
C IGOOD : # OF GOOD POINTS PER BIN. CHANGES W/ WEEDING. 00440000
C IDST : # OF POINTS PER BIN REJECTED BECAUSE OF DST. 00450000
C IRAN : RANDOM INTEGER BETWEEN 1 AND IGOOD 00460000
C K2AP : NUMBER OF POINTS REJECTED IN RANDOM POINT 00470000
C REJECTION OPTION. 00480000
C ITEMP(18,I) : INOTE REJECTION FLAG. INOTE=0 MEANS GOOD DATA. 00490000
C
A-90
NBIN = 1
IBIN = 0

2 READ(10,END=6) RA
   IF(IA(1) .EQ. 0) GO TO 2

3 IBLKNO = IA(16)
   IF(IBLKNO .NE. NBIN) GO TO 6
   IBIN = IBIN+1
   DO 4 J=1,28
4 TEMP(J,IBIN) = RA(J)
   GO TO 2

6 IF(IBIN .EQ. 0) THEN
   WRITE(6,600) NBIN
   NBIN = NBIN+1
   C FOR A BLKSIZE OF 10.0, THE MAXIMUM # OF BINS EQUALS 426
   IF(NBIN .GT. 426) GO TO 44
   ENDIF

   CLAT = CENTER(NBIN,1)
   CLON = CENTER(NBIN,2)
   C CALCULATE DIPOLE LATITUDE OF CENTER OF BIN NBIN.
   CALL DIPOLE(CLAT,CLON,DIPLAT)
   WRITE(6,611) NBIN,CLAT,CLON,DIPLAT
   611 FORMAT(/,8X,I3o' ,CLAT,CLON,DIPLAT: ',3(F7.2,2X))
   SET IGOAL ACCORDING TO DIP LATITUDE.
   IF( DABS(DIPLAT) .LE. 30.0 ) THEN
      IGOAL = 3
   ELSE
      IGOAL = 9
   ENDIF

   C FIND IBAD, INITIAL IGOOD
   IBAD = 0
   IGOOD = 0
   DO 10 I=1,IBIN
      IF(ITEMP(18,I) .EQ. 0) THEN
         IGOOD = IGOOD+1
      ELSE
         IBAD = IBAD + 1
      ENDIF
   10 CONTINUE

   IF( IGOOD .LE. IGOAL ) THEN
      WRITE(6,601) NBIN,IBIN,IGOOD,IBAD,IGOAL
      GO TO 33
   ENDIF
   C ALL BINS WHICH GET TO THIS STAGE STILL HAVE TOO MANY GOOD POINTS.
   IDST = 0
   DO 15 I=1,IBIN
      IF(ITEMP(18,I) .NE. 0) GO TO 15
C (NEXT IF BLOCK IS FOR GOOD POINTS ONLY)
IF (IGOOD .GT. IGOAL) THEN
  IF( ITEMP(17, I) .LT. -20 .OR. ITEMP(17, I) .GT. 5 ) THEN
    ITEMP(18, I) = 7
    IGOOD = IGOOD - 1
    IDST = IDST + 1
    GO TO 15
  ENDIF
ELSE
  WRITE(6, 602) NBIN, IBIN, IGOOD, IBAD, IDST, IGOAL
  GO TO 33
ENDIF

15 CONTINUE

C AT THIS POINT, BIN STILL HAS TOO MANY GOOD POINTS, EVEN AFTER IBAD,
C DST REJECTIONS.
C NOW RANDOMLY REJECT POINTS IN THE BIN UNTIL IGOAL HAS BEEN REACHED.

  K = 0
  DO 20 I = 1, IBIN
    IF( ITEMP(18, I) .EQ. 0 ) THEN
      K = K + 1
      KTEMP(K) = I
    ENDIF
  20 CONTINUE

  KZAP = 0
  21 IF (IGOOD .EQ. IGOAL) GO TO 25

C GENERATE IRAN (BETWEEN 1 AND IGOOD)
  IX = IY
  CALL RANDU(IX, IY, ZOUT)
  IRAN = INT(ZOUT*IGOOD)
  IF(IRAN .EQ. 0) IRAN = I
  KZAP = KZAP + 1

C ADJUST KTEMP ACCORDING TO IRAN.
  IF( IRAN .EQ. IGOAL ) THEN
    IGOOD = IGOOD - 1
    GO TO 21
  ELSE
    DO 23 K = IRAN, IGOOD - 1
      KTEMP(K) = KTEMP(K + 1)
      IGOOD = IGOOD - 1
    23 CONTINUE
    GO TO 21
  ENDIF

C FIRST SET ALL POINT FLAGS TO 'BAD', THEN SET POINTS IN KTEMP TO
C GOOD.
  25 DO 27 I = 1, IBIN
  27 ITEMP(18, I) = 7
  28 DO 27 I = 1, IGOAL
        IQ = KTEMP(I)
    28 CONTINUE
writes 33 DO 35 I=1, IBIN
DO 34 J=1, 28
34 OA(J) = TEMP(J, I)
35 WRITE(11) OA
NBIN = NBIN+1
IF(NBIN .GT. 426) GO TO 46
IBIN = 0
GO TO 3
44 STOP
C------------- FORMATS -----------------C
600 FORMAT(/,2X,'************* BIN NUMBER ',I3,' HAS ZERO POINTS')
601 FORMAT(2X,'@@@@@@@@ BIN # ',I3,' ALREADY AT IGOAL.',
@ '/10X, # POINTS IN BIN: ',I3, '/',10X,
@ '# GOOD POINTS IN BIN: ',I3, '/',10X, '# BAD POINTS IN BIN: ',I3,
@ '/15X, 'IGOAL : ',I3)
602 FORMAT(2X,'$$$$$$$$$ BIN # ',I3,' REDUCED TO IGOAL USING DST',
@ '/10X, # POINTS IN BIN: ',I3, '/',10X,
@ '# GOOD POINTS IN BIN: ',I3, '/',10X, '# BAD POINTS IN BIN: ',I3,
@ '/15X, 'IGOAL : ',I3)
603 FORMAT(2X,'ZZZZZZZZI BIN # ',I3,' REDUCED TO IGOAL USING IKAP',
@ '/10X, # POINTS IN BIN: ',I3, '/',10X,
@ '# GOOD POINTS IN BIN: ',I3, '/',10X, '# BAD POINTS IN BIN: ',I3,
@ '/15X, 'IGOAL : ',I3)
END
SUBROUTINE DIPOLE(DLAT,DLON,DIPLAT)
C THIS ROUTINE CALCULATES THE DIPOLE LATITUDE OF A POSITION, GIVEN
C ITS LAT AND LONG (DEGREES). A GEOCENTRIC EARTH IS ASSUMED.
C DRA CONVERTS DEGREES TO RADIANS. THETA0 IS THE CO-LATITUDE OF
C THE GEOMAGNETIC POLE, PHIO THE LONGITUDE OF THE POLE (IN DEGREES).
IMPLICIT REAL*8 (A-H,O-Z)
DATA DRA/.0174532925208DOJ,PHIO/289.2/
C
THETA0 = 11.12*DRA
TCOS0 = DCOS(THETA0)
TSINO = DSIN(THETA0)
C
COMPUTE DIPLAT, ABDIP
COLAT = DRA*(90.0-DLAT)
DELLON = DRA*(DLON-PHI0)
Q = TCOSO*DCOS(COLAT) + TSINO*DSIN(COLAT)*DCOS(DELLON)
DIPLAT = 90.0 - (DCOS(Q))/DRA
C
RETURN
END
SUBROUTINE ZONE(DELT)
IMPLICIT REAL*8(A-H,O-Z)

DIMENSION RLAT(180),RSQ(180),AREA(180)
COMMON J,N(180),M,T1,PHITOP(180),DLAM(180),NROW(180),PHIBAR(180)

C SPECIFY INITIAL DELT = DEL(ALAT) = C
C DEL(LAMBD) = 10 DEGREES C
C DELT=10.DO C
C C
C SPECIFY POINT (ALAT,ALONG) C
C ALAT(-90,+90) ALONG(0,360) C
C C

DRCONV=3.14159265D0/180.D0

J = (90.DO/DELT)
NPOLBK = J * .25
IF(NPOLBK.GT.3)NPOLBK=3
DELL = 0
DO 10 K=I,J
   DELL = DELL + DELT
10    PHITOP(K) = DELL

LAST=4
DO 100 ITER=1,LAST
   PHIBAR(1) = PHITOP(1)/2.DO
   RLAT(1) = PHITOP(1)
   DO 20 K=2,J
      PHIBAR(K) = (PHITOP(K) + PHITOP(K-1))/2.DO
   20   RLAT(K) = PHITOP(K) - PHITOP(K-1)
   DO 30 K=1,J
      N(K) = 360.DO/DELT * DCOS(PHIBAR(K)*DRCONV)+.5
      DO 50 K=1,NPOLBK
         KJ = J+1-K
      50   N(KJ) = 4*(2*K-1)
   30   DO 60 K=1,J
      DLAM(K)=360.DO/N(K)
      RSQ(K)=DLAM(K)*DCOS(PHIBAR(K)*DRCONV)/RLAT(K)
   60   DO 90 K=1,J
      WRITE(6,120)K,N(K),PHITOP(K),PHIBAR(K),DLAM(K),RSQ(K),AREA(K)
   90   C FORMAT(1X,'K=',I3,2X,'N=',I3,2X,'PHITOP=',FS.2,2Z,'PHIBAR=',F5.2,2X,'DLAM=',F5.2,2X,'RSQ=',F9.1)
C 120 FORMAT(6,120)K,N(K),PHITOP(K),PHIBAR(K),DLAM(K),RSQ(K),AREA(K)
C 120 FORMAT(1X,'K=',I3,2X,'N=',I3,2X,'PHITOP=',F5.2,2X,'PHIBAR=',F5.2,2X,'DLAM=',F5.2,2X,'RSQ=',F9.1)
C 120 FORMAT(1X,'AREA(SQKM)=',F9.1)
   IF(ITER.EQ.LAST)GO TO 100
   CALL NEWTON(J,K,N,PHITOP(K))
   DO 70 K=1,J
      CALL NEWTON(J,K,N,PHITOP(K))
   70 CONTINUE
   CONTINUE
   M=0
SUBROUTINE NEWTON

IMPLICIT REAL *8(A-H,O-Z)
DIMENSION N(180)
DRCONV=3.14159265D0/180.D0

COMPUTE AREA FACTORS
SUML=0.00
SUMLL=0.00
DO 10 L=1,K
  10   SUML=SUML+N(L)
DO 20 LL=1,J
  20   SUMLL=SUMLL+N(LL)
FACTOR=SUML/SUMLL
ALAT=ALAT*DRCONV
PHI0=ALAT
DO 100 L=1,5
  100  DERIV=D COS(PHI0)
       FP=(DSIN(PHI0)-FACTOR)/DERIV
       EPS=FP/PHI0
       IF(DABS(EPS).LT.1.D-5)GO TO 200
       ALAT=PHI0-FP
  200  CONTINUE
ALAT=ALAT/DRCONV
RETURN
END

SUBROUTINE MIDDLE(CENTER)

THIS SUBROUTINE CALCULATES THE LATITUDE AND LONGITUDE OF THE CENTER OF EACH BIN. THE COORDINATES ARE THEN STORED IN THE ARRAY 'CENTER.'
FOR EXAMPLE, CENTER(4,1) AND CENTER(4,2) WOULD BE THE LATITUDE AND LONGITUDE (RESPECTIVELY) OF THE CENTER OF THE FOURTH BIN.
IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 RLAT,RLON,CSIZE
CHARACTER*1 BIN(5)
COMMON J,N(180),M,T1,PHITOP(180),DLAM(180),NROW(180),PHIBAR(180)
DIMENSION CENTER(500,2),NN(500)

C NN NUMBERS THE BINS CONSECUTIVELY.
C (N IS THE NUMBER OF BINS IN EACH ROW.)

NN(1) = 0
DO 6 I = 2,J+1
6   NN(I) = NN(I-1) + N(I-1)

CALCULATE CENTERS FOR NORTHERN HEMISPHERE

DO 9 K = 2,J+1
DO 8 I = NN(K-1)+1, NN(K)
   CENTER(I,1) = PHIBAR(K-1)
   IF(I.EQ.1) GO TO 7
   CENTER(I,2) = CENTER(I-1,2) + DLAM(K-1)
7   CENTER(I,2) = DLAM(1)/2.D0
8   IF(CENTER(I,2).GT.360.) CENTER(I,2) = DLAM(K-1)/2.D0
9   CONTINUE

CALCULATE CENTERS FOR SOUTHERN HEMISPHERE

DO 10 I = NN(J+1)+1, 2*NN(J+1)
   CENTER(I,1) = -CENTER(I-NN(J+1),1)
   CENTER(I,2) = CENTER(I-NN(J+1),2)
10  CONTINUE
RETURN
END

SUBROUTINE RANDU(IX,IY,YFL)

SUBROUTINE RANDU

PURPOSE
COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS BETWEEN 0 AND 1.0 AND RANDOM INTEGERS BETWEEN ZERO AND 2**31. EACH ENTRY USES AS INPUT AN INTEGER RANDOM NUMBER AND PRODUCES A NEW INTEGER AND REAL RANDOM NUMBER.

USAGE
CALL RANDU(IX,IY,YFL)

DESCRIPTION OF PARAMETERS
IX - FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD INTEGER NUMBER WITH NINE OR LESS DIGITS. AFTER THE FIRST ENTRY, IX SHOULD BE THE PREVIOUS VALUE OF IY COMPUTED BY THIS SUBROUTINE.
IY - A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR THE NEXT ENTRY TO THIS SUBROUTINE. THE RANGE OF THIS NUMBER IS BETWEEN ZERO AND 2**31.
YFL - THE RESULTANT UNIFORMLY DISTRIBUTED, FLOATING POINT, RANDOM NUMBER IN THE RANGE 0 TO 1.0

REMARKS
THIS SUBROUTINE IS SPECIFIC TO SYSTEM/360 AND WILL PRODUCE 2**29 TERMS BEFORE REPEATING. THE REFERENCE BELOW DISCUSSES SEEDS (65539 HERE), RUN PROBLEMS, AND PROBLEMS CONCERNING RANDOM DIGITS USING THIS GENERATION SCHEME. MACLAREN AND MARSAGLIA, JACM 12, P. 83-89, DISCUSS CONGRUENTIAL GENERATION METHODS AND TESTS. THE USE OF TWO GENERATORS OF THE RANDU TYPE, ONE FILLING A TABLE AND ONE PICKING FROM THE TABLE, IS OF BENEFIT IN SOME CASES. 65549 HAS BEEN SUGGESTED AS A SEED WHICH HAS BETTER STATISTICAL PROPERTIES FOR HIGH ORDER BITS OF THE GENERATED DEVIATE. SEEDS SHOULD BE CHOSEN IN ACCORDANCE WITH THE DISCUSSION GIVEN IN THE REFERENCE BELOW. ALSO, IT SHOULD BE NOTED THAT IF FLOATING POINT RANDOM NUMBERS ARE DESIRED, AS ARE AVAILABLE FROM RANDU, THE RANDOM CHARACTERISTICS OF THE FLOATING POINT DEVIATES ARE MODIFIED AND IN FACT THESE DEVIATES HAVE HIGH PROBABILITY OF HAVING A TRAILING LOW ORDER ZERO BIT IN THEIR FRACTIONAL PART.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD
POWER RESIDUE METHOD DISCUSSED IN IBM MANUAL C20-8011, RANDOM NUMBER GENERATION AND TESTING.

IY=IX*65539
IF(IY)5,6,6
5 IY=IY+2147483647+I
6 YFL=IY
   YFL=YFL*.4656613E-9
RETURN
END

// EXEC LINKGO, REGION.GO=500K
// GO.FT10F001 DD DSN=XRJRR.NOV2385.DST1,DISP=SHR
// GO.FT11F001 DD DSN=XRJRR.NOV2385.S1PT.DATA,
// UNIT=SYSDA,SPACE=(TRK,(10,5),RLSE),VOL=SER=SACC05,
// DISP=(NEW,CATLG),DCB=(RECFM=VBS,LRECL=116,BLKSIZE=11604)
// EXEC NOTIFYTS
11 9 0 0 0 01984.06371.2 0
<table>
<thead>
<tr>
<th>COEFFICIENTS</th>
<th>CAL84FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>XRDSC.DMSP.CALBB.TEST4.TEST.DATA</td>
<td></td>
</tr>
<tr>
<td>2 1-29878.1992 0.0000 26.9879 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>2 2 -1924.0500 5526.5508 7.9558 -19.3154 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>3 1 -2063.3401 0.0000 -16.6929 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>3 2 3044.3201 -2183.8701 4.2478 -13.6396 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>3 3 1682.8701 -291.6460 5.0440 -22.9796 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>4 1 1279.3401 0.0000 -0.5587 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>2 2 -2200.8101 -317.4509 -5.0723 4.5528 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>4 3 1250.1299 282.9050 3.0011 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>4 4 831.3350 -289.1660 -0.3731 -9.2377 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>5 1 943.0530 0.0000 1.3461 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>5 2 776.3311 230.8580 -1.4823 4.6653 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>5 3 370.7820 -248.3420 -6.7795 2.0878 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>5 4 -424.3979 64.1152 -1.3651 2.8099 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>5 5 174.5670 -294.2991 -6.0780 0.7172 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 1 -211.9340 0.0000 1.4847 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 2 358.8789 45.6665 0.4091 -0.1262 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 3 252.2410 145.8200 -2.2093 -0.9964 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 4 -90.4987 -152.3840 4.0607 -0.4410 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 5 -162.3880 -77.5140 -0.1193 0.0529 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>6 6 -48.5517 97.0991 0.1276 1.2475 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 1 50.2750 0.0000 0.5828 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 2 65.8066 -14.4218 0.0736 0.0835 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 3 48.4155 88.5492 1.6242 -1.1229 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 4 -186.4770 71.0999 1.4083 0.1306 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 5 1.9859 -47.6321 -0.4003 -1.1404 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 6 15.7450 20.6672 0.0817 -0.1775 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>7 7 -103.6940 20.6672 1.0083 0.8616 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 1 75.1637 0.0000 0.8018 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 2 -62.4921 -83.4985 -0.8234 -0.2392 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 3 2.8062 -24.7745 0.3449 0.6610 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 4 23.7248 -4.3455 0.7469 0.2003 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 5 -4.9795 20.8105 1.8643 1.1319 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 6 1.1965 21.6843 0.1407 0.9747 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 7 10.5049 -23.1920 0.0207 -0.0463 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>8 8 -2.1680 -5.2178 0.1217 1.1180 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 1 20.3340 0.0000 0.4621 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 2 5.2416 6.0690 -0.3242 -0.1839 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 3 1.0119 -18.4504 0.3637 -0.2367 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 4 -9.5814 6.2423 0.3523 0.5092 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 5 -10.2597 -23.2842 -0.8320 -0.2703 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 6 3.3773 6.9580 -0.2127 -0.5421 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 7 3.8130 14.4615 0.2749 -0.4055 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 8 4.6053 -15.2854 -0.3537 -0.5202 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>9 9 -2.7086 -11.8510 -0.3326 0.6775 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 1 5.4469 0.0000 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 2 10.3427 -20.8446 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 3 1.5372 15.3630 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 4 -12.3475 8.9692 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 5 9.4340 -5.3201 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 6 -3.4223 -6.3449 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 7 -1.1907 8.9932 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td>10 8 6.6870 9.6466 0.0000 0.0000 0.0000 0.0000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.5169</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>10</td>
<td>-5.0012</td>
</tr>
<tr>
<td>10</td>
<td>-3.4339</td>
</tr>
<tr>
<td>11</td>
<td>-3.9929</td>
</tr>
<tr>
<td>11</td>
<td>2.2212</td>
</tr>
<tr>
<td>11</td>
<td>-5.4240</td>
</tr>
<tr>
<td>11</td>
<td>-1.9861</td>
</tr>
<tr>
<td>11</td>
<td>4.5759</td>
</tr>
<tr>
<td>11</td>
<td>3.1589</td>
</tr>
<tr>
<td>11</td>
<td>0.9086</td>
</tr>
<tr>
<td>11</td>
<td>1.9800</td>
</tr>
<tr>
<td>11</td>
<td>2.7993</td>
</tr>
<tr>
<td>11</td>
<td>-0.2744</td>
</tr>
</tbody>
</table>
0 0
0 0
PROGRAM DSTADD ..........

//XRJRRDST JOB (F8002,X22,20),DSTADD,TIME=(2,00),CLASS=O,NOTIFY=XRJRR,
// MSGCLASS=X
/*JOBPARAM LINES=30
/* PROGRAM TO ADD DST VALUES TO DMSP DATA. ALSO ADDS BIN NUMBERS.
/* XRJRR.DMSP.PROGRAMS(DSTADD)
// EXEC FORTRAN,PARM='XREF'
//SYSIN DD *

DIMENSION A(28,100),IA(28,100),AA(28,5000),IAA(28,5000),
@ IDST(24),IDEL(13),AOUT(28)
EQUIVALENCE(A,IA)
EQUIVALENCE(AA,IAA)

C THIS PROGRAM READS IN DMSP DATA, IN "FIT" FORMAT, AS OUTPUT FROM
C T.J. SABAKA'S DMSP PROCESSING PROGRAM. IT ADDS DST VALUES TO THE
C DATA, AND ALSO ADDS BIN NUMBERS TO THE DATA. IT ALSO DELETES
C SPECIFIED DATES AND HOURS FROM THE DATA, WHICH HAVE BEEN CHECKED TO
C HAVE HIGH KP INDICES OR EXCEPTIONALLY LARGE DST INDICES.
C FINALLY, THE PROGRAM Sorts THE DATA BY BIN NUMBER.
C
C ***NOTE: DATA IS INPUT IN A FORMAT WHICH HAS 100 DATA VALUES PER
C LOGICAL RECORD. IT IS OUTPUT ONE VALUE PER LREC.
C DATA WILL EVENTUALLY BE OUTPUT 100 POINTS PER RECORD,
C WITH A LATER PROGRAM.
C **NOTE#2: THIS VERSION OF THE PROGRAM Sorts THE DATA WITH THE
C "NEW" METHOD, POSSIBLY INEFFICIENT.

DRCONV = 3.14159265/180.0

READ INITIAL LINE OF DATA FROM DST TAPE.
READ(9,101) IYR,IDAY,IDST
I01 FORMAT(2X,I2,I3,2X,2414)

C READ LINE OF DATA FROM TAPE 10.
C KCOUNT COUNTS FROM 1 TO THE ENTIRE DATA SET.
C NWRITE COUNTS THE NUMBER OF RECORDS WRITTEN OUT.
C I COUNTS FROM 1 TO 100
C KP COUNTS THE NUMBER OF RECORDS DELETED BECAUSE OF KP INDEX.
C
C CALL ZONE TO SET UP EQUAL AREA BLOCKS OF (BLKSIZ) SIZE.
CALL ZONE(BLKSIZ)
KCOUNT = 0
NWRITE = 0
KP = 0
2     READ(10,END=55) A
     KCOUNT = KCOUNT + 1
     I = 1
C ACCEPT DATA ONLY WHICH DO NOT HAVE NEGATIVE DATA QUALITY FLAGS.
   4     IF(IA(18,I) .NE. 0 ) THEN
          I = I+1
          IF( I .GT. 100 ) GO TO 2
          GO TO 4
       ENDIF
   IF(IA(1,1) .EQ. 0) THEN
          I = I+1
          IF( I .GT. 100 ) GO TO 2
          GO TO 4
       ENDIF
JYR = IFIX(A(5,I))
JDAY = IA(1,I)
STIME = FLOAT(IA(2,I))/3.60E6
JHR = INT(STIME) + 1
C DELETE DATA WHICH HAVE BAD KP OR DST INDICES(GIVEN IN DATA STMT).
JTEST = ( (JHR-I) / 3 + 1 )*1000 + JDAY
       IF( NDEL .EQ. 0 ) GO TO 7
       DO 5 J=I,NDEL
          IF(JTEST .EQ. IDEL(J)) THEN
             I = I+1
             KP = KP+1
             IF(I .GT. 100) GO TO 2
             GO TO 4
          ENDIF
      5 CONTINUE
C TEST WHETHER JYR EQUALS IYR
   7     IF(JYR .EQ. IYR) THEN
          GO TO 9
       ENDIF
C JYR DOESN'T EQUAL IYR, SO READ ANOTHER LINE OF DST TAPE
   8     READ(9,101) IYR,IDAY,IDST
          GO TO 7
C
9 CONTINUE
C NOW THAT JYR EQUALS IYR, TEST WHETHER JDAY EQUALS IDAY
10     IF(JDAY .EQ. IDAY) THEN
          GO TO 14
       ENDIF
C JDAY DOESN'T EQUAL IDAY, SO READ ANOTHER LINE OF DST TAPE.
12     READ(9,101) IYR,IDAY,IDST
          GO TO 10
14 CONTINUE
C NOW JYR AND JDAY ARE CORRECT. PULL OFF THE CORRECT HOURLY DST.
JDST = IDST(JHR)
NWRITE=NWRITE+1
C NOW FIND THE CORRECT BIN NUMBER FOR THE DATA.
CALL INDX TO DETERMINE BLOCK NUMBER
DLAT = A(6,I)
DLON = A(7,I)
CALL INDX(DLAT,DLON,IBLKNO)
WRITE(6,607) IBLKNO,NWRITE,IA(1,I),IA(2,I),DLAT,DLON
607 FORMAT(3X,'***',2X,IS,2X,IS,2X,I3,2X,19,SX,2(F5.1,1X))
IA(16,I)=IBLKNO
IA(17,I)=JDST

C PUT INFORMATION INTO OUTPUT ARRAY.
C PUT BIN # INTO SLOT 16, DST VALUE INTO SLOT 17.
C DO THIS THE "NEW" WAY, IN WHICH IT WILL BE PRE-SORTED BY BIN
C NUMBER AS IT IS BEING PUT INTO THE ARRAY.
IF(NWRITE .EQ. 1) THEN
    DO 17 JJ=1,28
    17   AA(JJ,1) = A(JJ,I)
        I = I+1
        IF( I .GT. 100 ) GO TO 2
        GO TO 4
ENDIF

K = 1
20 KBLKNO = IAA(16,K)
IF(KBLKNO .LE. IBLKNO) THEN
    K = K+1
    IF(K .GT. NWRITE) GO TO 25
    GO TO 20
ENDIF

DO 22 J=NWRITE,K,-1
DO 22 JJ=1,28
22   AA(JJ,J+1) = AA(JJ,J)
        DO 24 JJ=1,28
24    AA(JJ,K) = A(JJ,I)
        GO TO 27

25 DO 26 JJ=1,28
26   AA(JJ,K)=A(JJ,I)

27 CONTINUE
   I = I+1
   IF( I .GT. 100 ) GO TO 2
   GO TO 4

C IF THE PROGRAM REACHES THIS NEXT LINE, THEN ALL DATA POINTS HAVE BEEN
C READ IN AND SORTED. NOW WRITE OUT THE DATA.
55 WRITE(6,605) NWRITE,KCOUNT,KP

      DO 58 J=1,NWRITE
      DO 57 JJ=1,28
57    AOUT(JJ) = AA(JJ,J)
58    WRITE(11) AOUT

C ******** FORMATS ******** C
SUBROUTINE ZONE(DEL1)
IMPLICIT REAL *8(A-H,O-Z)
DIMENSION N(180), PHIBAR(180), PHITOP(180), DLAM(180), NPOI(180), RSQ(180), AREA(180)
COMMON /ZONE1/ J,N,M, PHITOP, DLAM
DRCONV=3.14159265D0/180.D0
J=(90.D0/DEL1)
NPOLBK=J*0.25 IF(NPOLBK.GT.3) NPOLBK=3
DELL=0 DO 10 K=1,J DELL=DELL+DEL1
10 PHITOP(K)=DELL LAST=4 DO 100 ITER=1,LAST
PHIBAR(1)=PHITOP(1)/2.D0
12 DO 20 R_1(T)=PHITOP(1)
20 PHIBAR(K)=(PHITOP(K)+PHITOP(K-1))/2.D0 DO 30 K=1,J
30 N(K)=360.D0/DEL1*DCOS(PHIBAR(K)*DRCONV)+0.5 DO 50 K=I,NPOLBK KJ=J+I-K
50 N(KJ)=4*(2*K-1) DO 60 K=1,J DLAM(K)=360.D0/N(K)
60 RSQ(K)=DLAM(K)*DCOS(PHIBAR(K)*DRCONV)/RLAT(K) IF(ITER.EQ.LAST) GO TO 100 DO 70 K=1,J CALL NEWTON(J,K,N,PHITOP(K)) 70 CONTINUE
100 CONTINUE M=0 DO 110 KK=1,J
110 M=M+N(KK) RETURN
END

SUBROUTINE NEWTON(J,K,N,ALAT)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION N(180)
DRCONV=3.14159265D0/180.D0
SUML=0.00 SUMLL=0.00 DO 10 L=1,K 10 SUML=SUML+N(L) DO 20 LL=1,J

A-103
20 SUMLL=SUMLL+LL
FACTOR=SUMLL/SUMLL
ALAT=ALAT*DRCONV
PHI0=ALAT
DO 100 L=1,5
DERIV=DCOS(PHI0)
FF=(DSIN(PHI0)-FACTOR)/DERIV
EPS=FF/PHI0
IF(DABS(EPS).LT.1.0D-5)GO TO 200
ALAT=PHI0-FF
100 PHI0=ALAT
200 CONTINUE
ALAT=ALAT/DRCONV
RETURN
END

SUBROUTINE INDX(ALAT,ALONG,IBLNO)
IMPLICIT REAL *8(A-H,O-Z)
DIMENSION N(180),PHITOP(180),DLAM(180)
COMMON /ZONE1/ J,N,M,PHITOP,DLAM
IF(ALONG .LT. 0.D0) ALONG = ALONG + 360.D0
APHI=DABS(ALAT)
DO 10 IS=1,J
I=IS
IF(PHITOP(IS).GE.APHI) GO TO 300
10 CONTINUE
NTOT=0
L=(ALONG/DLAM(I))+1
IF(I.EQ.1) GO TO 30
DO 20 JJ=2,1
NTOT=NTOT+N(JJ-1)
20 CONTINUE
30 CONTINUE
IBLNO=NTOT+L
IF(ALAT.LT.0) IBLNO=IBLNO+M
RETURN
END

C
// EXEC LINKGO,REGION.GO=3000K
//GO.FT9F001 DD DSN=XRJRR.DST81,DISP=SHR
//GO.FT10F001 DD DSN=XRSHS.NOV2385.STEP5.OUTBIN,DISP=SHR
//* WRITE OUT TO TAPE 11
//GO.FT11F001 DD DSN=XRJRE.NOV2385.DSTI,UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=116,BLFSIZE=11604),SPACE=(TRK,(10,5),RLSE),
// VOL=SER=SACC03,DISP=(NEW,CATLG)
// EXEC NOTIFYTS
PROGRAM EUTRANS  

//XRJREUT JOB (F8002,X2,25), EUTRANS, TIME=(5,0), CLASS=E, NOTIFY=XRJRR.
// MSGCLASS=X
//* INPUT: DMSP DATA IN FIT FORMAT, SPACECRAFT COORDINATES (UNIT 10)
//* INPUT#2: 3 EULER ANGLES, 3 BIASES (DATA STATEMENT).
//* OUTPUT: DMSP DATA TRANSFORMED AND CORRECTED (UNIT11).
//*JOPPARM LINES=10
// EXEC FORTRAN, PARM='XREF'
//SYSIN DD *
   DIMENSION A(28,100)
   DIMENSION AA(28)
   REAL*8 EU1,EU2,EU3,DRCONV,SL1,SL2,SL3
C
C SET EULER ANGLES, BIASES, SLOPES FOR CORRECTIONS.
   DATA EU1/0.00119751D0/, EU2/-0.002591042D0/, EU3/-0.0032734508D0/
   DATA BS1/4.797/, BS2/-5051/, BS3/0.6497/
   DATA SL1/0.99993646D0/, SL2/0.99960327D0/, SL3/1.0011901D0/
C
C DRCONV = 3.14159265/180.0
C
C CALL ROUTINE TO CALCULATE TSM (EULER TRANSFORMATION) MATRIX FROM
C INPUT EULER ANGLES EU1,EU2,EU3. THE OUTPUT ELEMENTS OF THIS
C MATRIX ARE STORED IN COMMON FOR USE IN SUBROUTINE APPLY.
C
C CALL EULER(EU1,EU2,EU3)
C
C READ DATA FROM UNIT#10.
  1 READ(10,END=22) A
      I=1
C
C PULL OFF A VALUE FROM A, PUT INTO AA FOR PROCESSING.
  2 DO 4 J=1,28
      4 AA(J) = A(J,i)
C
C WRITE(6,611) AA(11),AA(12),AA(13)
C 611 FORMAT(IX,'BEFORE APPLY: ',3(F15.5,2X))
C
IF( AA(11) .EQ. 0.0 ) GO TO 6
C APPLY SLOPES, BIASES AND EULER ANGLE CORRECTIONS TO AA.
C CALL APPLY(AA,BS1,BS2,BS3,SL1,SL2,SL3)
C
C WRITE(6,612) AA(11),AA(12),AA(13)
C 612 FORMAT(2X,'AFTER APPLY: ',3(F15.5,2X),/)  
C
C PUT AA BACK INTO A
  6 DO 8 J=11,13
      8 A(J,i) = AA(J)
C
C WRITE OUT A IF NECESSARY.
C IF(I .EQ. 100) THEN
C   WRITE(11) A
C   NWRITE=NWRITE+1
C   GO TO 1
C ENDF
C INCREMENT I, GO TO 2
   I = I+1
   GO TO 2
C
22 WRITE(6,601) NWRITE
601 FORMAT(/,5X,'A TOTAL OF ',I4,' RECORDS READ IN, WRITTEN OUT')
   STOP
END
C*****************************************************************************
MATH 1
C
SUBROUTINE APPLY(A,BSX,BSY,BSZ,SLX,SLY,SLZ)
C
THIS ROUTINE APPLIES THE EULER ROTATION MATRIX CALCULATED IN
ROUTINE "EULER" TO DATA IN ARRAY A. FOR THIS VERSION OF APPLY,
A IS ASSUMED TO ALREADY BE IN SPACECRAFT COORDINATES.
A IS CORRECTED WITH BIASES AND SLOPES, THEN
CORRECTED FOR EULER ANGLES BY TRANSFORMING WITH
ROTATION MATRIX TSM (IN COMMON BLOCK).
C
INPUT: A, X,Y, AND Z BIASES, X,Y,Z SLOPES, TSM MATRIX.
C
OUTPUT: A (CORRECTED)
C
REAL*8 TSM11,TSM12,TSM13,TSM21,TSM22,TSM23,TSM31,TSM32,TSM33,
@ SLX,SLY,SLZ
DIMENSION A(28)
COMMON /TSM/ TSM11,TSM12,TSM13,TSM21,TSM22,TSM23,TSM31,TSM32,
@ TSM33
C
B1X = A(11)
B1Y = A(12)
B1Z = A(13)
C
APPLY SLOPES AND BIASES.
   B2X = (1.0/SLX)*( B1X - BSX )
   B2Y = (1.0/SLY)*( B1Y - BSY )
   B2Z = (1.0/SLZ)*( B1Z - BSZ )
C
APPLY EULER ANGLE ROTATIONS TO B2X,B2Y,B2Z TO GET CORRECTED A.
   A(11) = TSM11*B2X + TSM12*B2Y + TSM13*B2Z
   A(13) = TSM31*B2X + TSM32*B2Y + TSM33*B2Z
C
RETURN
END
C
SUBROUTINE EULER(EU1,EU2,EU3)
C
THIS ROUTINE CALCULATES THE NINE ELEMENTS OF TRANSFORMATION
MATRIX TSM. GIVEN EULER ANGLES EU1,EU2,EU3. OUTPUT IS STORED
IN COMMON.
REAL*8 EU1,EU2,EU3,TCOS1,TCOS2,TCOS3,TSS1,TSN2,TSS3,
@ TSM11,TSM12,TSM13,TSM21,TSM22,TSM23,TSM31,TSM32,TSM33
REAL*8 PI, DRA

C COMMON/TSM/ TSM11, TSM12, TSM13, TSM21, TSM22, TSM23, TSM31, TSM32, TSM33
C
C DATA PI/3.14159265/
DRA = PI/180.0
C
C CONVERT DEGREES TO RADIANS.
EU1 = EU1*DRA
EU2 = EU2*DRA
EU3 = EU3*DRA
TCOS1 = DCOS(EU1)
TCOS2 = DCOS(EU2)
TCOS3 = DCOS(EU3)
TSIN1 = DSIN(EU1)
TSIN2 = DSIN(EU2)
TSIN3 = DSIN(EU3)
C
TSM11 = TCOS1*TCOS3
TSM12 = TCOS1*TSIN3*TCOS2 + TSIN1*TSIN2
TSM13 = (-1.0)*TCOS1*TSIN3*TSIN2 + TSIN1*TCOS2
TSM21 = (-1.0)*TSIN3
TSM22 = TCOS3*TCOS2
TSM23 = (-1.0)*TSIN1*TCOS3
TSM31 = (-1.0)*TSIN1*TSIN3*TCOS2 + TCOS1*TSIN2
TSM32 = (-1.0)*TSIN1*TSIN3*TSIN2*TSIN3 + TSIN1*TCOS2
TSM33 = TSIN1*TSIN2*TSIN3 + TCOS1*TCOS2
C
RETURN
END
C
// EXEC LINKGO,REGION.GO=500K
//GO.FT10F001 DD DSN=XRJRR.DMSP.FITPRP,DISP=SHR
// TAPE 11 IS OUTPUT
//GO.FT11F001 DD DSN=XRJRR.GARP,UNIT=SYSDA,DISP=(NEW,CATLG),
00006000
// DCB=(RECFM=VBS,LRECL=11204,BLKSIZE=22412),SPACE=(TRK,(20,20),RLSE),
00006100
// VOL=SER=SACC07
// EXEC NOTIFYTS

A-107
PROGRAM FITPREP ........

//XRJRP JOB (F8002,X22,30),FITRP,TIME=(0,30),CLASS=0,NOTIFY=XEJRR.
// MSGCLASS=X
/*JOBPARM LINES=10
// EXEC FORTRAN,PARM='XREF'

//SYSIN DD *
  DIMENSION A(28),IA(28)
  DIMENSION AA(28,100)
  EQUIVALENCE(A,IA)

C THIS PROGRAM READS IN DMSP DATA, WHICH HAS BEEN FLAGGED
C AND SIFTED BY PROGRAMS DSTADD AND BINSIFT. FOURTEEN DATA SETS
C ARE READ IN. THESE ARE MERGED INTO ONE DATA
C SET WHICH HAS GOOD POINTS ONLY AND WHICH CONTAINS 100 DATA POINTS
C PER LOGICAL RECORD. THE FINAL LOGICAL RECORD IS PADDED OUT TO 100
C BY ADDING ZEROS.
C
C INPUT DATA IS ON UNITS #11 - 24, AND OUTPUT ON UNIT #26.
C
C==============================================C

I = 1
NWRITE = 0
1 ITOT = 0
IGOOD = 0
K = 10+I
2 READ(K,END=I8) A
ITOT = ITOT+1
INOTE = IA(18). AN INOTE .NE. ZERO IS A BAD POINT.
IF( IA(18) .NE. 0 ) GO TO 2
IGOOD = IGOOD+1
ITEMP = ITEMP+1

C TRANSFER INFORMATION FROM A TO AA (100 POINTS PER LRECL).
DO 4 J=I,28
4 AA(J,ITEMP) = A(J)

IF( ITEMP .EQ. 100 ) THEN
  ITEMP = 0
  WRITE(26) AA
  NWRITE = NWRITE+1
  GO TO 2
ELSE
  GO TO 2
ENDIF

18 WRITE(6,601) I,ITOT,IGOOD,NWRITE
IF( I .NE. 14 ) THEN
  I = I+1
  GO TO 1
ELSE
  GO TO 22
ENDIF
C IF THE PROGRAM REACHES THIS NEXT LINE, THEN ALL DATA POINTS HAVE BEEN READ IN AND MOST OF THEM HAVE BEEN WRITTEN OUT. FILL AA OUT TO 100 WITH ZEROS IF NECESSARY AND WRITE OUT AA THE LAST TIME.

22 IF(ITEMP .LE. 10) THEN
   GO TO 25
ELSE
   DO 24 J=ITEMP+1,100
   DO 24 I=1,28
24   AA(I,J)=0
   WRITE(26) AA
   NWRITE = NWRITE+1
   WRITE(6,605)
   GO TO 25
ENDIF

C 25 WRITE(6,606) NWRITE

C ******** FORMATS ********
C 601 FORMAT(/,5X,'ZONE # ',I2,/,5X,'TOTAL POINTS: ',I5,5X,
     & '# GOOD POINTS: ',I5,9X,'# RECORDS WRITTEN OUT: ',I4)
C 605 FORMAT(/,2X,'***** LAST RECORD HAS SOME ZEROED POINTS *****')
C 606 FORMAT(/,5X,'TOTAL RECORDS WRITTEN OUT: ',I5)

C STOP
END
PROGRAM XYZTRANS ..........

//XJRRTRA JOB(F8002,X22,30),XYZTRANS,TIME=(2,0),CLASS=E,NOTIFY=XJRRA.
// MSGCLASS=X

//* INPUT: DMSP DATA IN FIT FORMAT, SPACECRAFT COORDINATES (UNIT 10)
//* OUTPUT: DMSP DATA TRANSFORMED TO GEOCENTRIC COORDINATES (11).
//*JOBPARM LINES=10
// EXEC FORTRAN,PARM='XREF'
//SYSIN DD *

DIMENSION A(28,100)
DIMENSION AA(28),IA(28)
EQUIVALENCE(AA,IA)
REAL*8 DLAT,DLON,DRA,RH(3),VH(3),ANORM(3),EFX,EFY,EFZ

C
DRA = 3.14159265/180.0
ICOUNT = 0
IGOOD = 0
NWRITE=0

C READ DATA FROM UNIT#10.
1 READ(10,END=22) A
I=1

C PULL OFF A VALUE FROM A, PUT INTO AA FOR PROCESSING.
2 DO 4 J=1,28
4 AA(J) = A(J,I)
ICOUNT = ICOUNT+1

C IDIR IS THE SATELLITE DIRECTION (+ OR -)
IDIR = IA(20)

C DATA QUALITY TEST
IF( IDIR.EQ.0 .OR. IA(18).NE.0 .OR. IA(1).EQ.0 ) THEN
   GO TO 6
ENDIF
IGOOD = IGOOD + 1

DLAT = AA(6)
DLON = AA(7)
IF(DLON.GT. 180.0) DLON = DLON-360.0

C CALL TRANSF WITH IDIR, LAT AND LON INFORMATION, TO GET OUT
C RH,VH AND ANORM COMPONENTS. USE THESE TO CALCULATE THE TGS
C MATRIX, WHICH TRANSFORMS SPACECRAFT COORDINATES TO EARTH-FIXED.
   CALL TRANSF(DLAT,DLON,RH,ANORM,VH,IDIR)

C XI,YI,ZI ARE SPACECRAFT COORDINATES.
XI = AA(11)
YI = AA(12)
ZI = AA(13)

C TRANSFORM COORDINATES FROM SPACECRAFT TO EARTH-FIXED.
EFX = -ANORM(1)*XI - RH(1)*YI + VH(1)*ZI
EFY = -ANORM(2)*XI - RH(2)*YI + VH(2)*ZI
EFZ = -ANORM(3)*XI - RH(3)*YI + VH(3)*ZI

*
TRANSFORM EARTH-FIXED COORDINATES BACK TO GEOCENTRIC X,Y,Z.
C PUT X INTO AA(11), Y INTO AA(12), Z INTO AA(13).

\[
\begin{align*}
DLAT &= DLAT \times DRA \\
DLON &= DLON \times DRA \\
AA(11) &= -\sin(DLAT) \times \cos(DLON) \times EFX - \sin(DLAT) \times \sin(DLON) \times EFY + \cos(DLAT) \times EFZ \\
AA(12) &= -\sin(DLON) \times EFX + \cos(DLON) \times EFY \\
AA(13) &= -\cos(DLAT) \times \cos(DLON) \times EFX - \cos(DLAT) \times \sin(DLON) \times EFY - \sin(DLAT) \times EFZ
\end{align*}
\]

RE-ASSIGN FLAGS
ITEMP = IA(22)
IA(22) = IA(24)
IA(24) = IA(23)
IA(23) = TEMP

PUT AA INTO A
6 DO 8 J=1,28
8 A(J,I) = AA(J)

WRITE OUT A IF NECESSARY.
IF(I .EQ. 100) THEN
WRITE(11) A
NWRITE=NWRITE+1
ENDIF

INCREMENT I, GO TO 2
I = I+1
IF(I .GT. 100) GO TO 1
GO TO 2

WRITE(6,601) NWRITE
601 FORMAT(//,5X,'A TOTAL OF ',I4,' RECORDS WRITTEN OUT')
WRITE(6,602) ICOUNT,IGOOD
602 FORMAT(//,5X,'TOTAL POINTS INPUT:',I5,', WITH',I5,' GOOD POINTS')

STOP
END

SUBROUTINE TRANSF(PHIR,ALAMR,RH,ANORM,VH,IDIR)
C RH=3 COMPS OF POSITION OF SATELLITE IN (X,Y,Z) COORDS
C ANORM=3 COMPS OF ORBIT NORMAL IN (X,Y,Z) COORDS
C VH=RH CROSS ANORM, VELOCITY UNIT VECTOR
C PHIR,PHIN=GEOCENTRIC LAT OF POSITION,NORMAL
C ALAMR,ALAMN=LONG OF POSITION,NORMAL
C IDIR=+I,0,-I SATELLITE ASCENDING,TURNING AROUND,DESCENDING
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION RH(3),ANORM(3),VH(3)
DATA PI/3.141592654D0/
DTR=PI/180.0D0
C PHIN IS COMPUTED BY KNOWING THE ORBIT INCLINATION, I=98.26 FOR
DATA PHIN/-8.74D0/
C CHECK THAT IDIR IS NOT ZERO.
IF(IDIR .EQ. 0) WRITE(6,601)
601 FORMAT(//'**       IDIR = ZERO. STOP EXECUTION.**)')
IF(IDIR .EQ. 0) STOP
DO 1 I=1,3
  RH(I)=0.D0
  ANORM(I)=0.D0
  VH(I)=0.D0
1 CONTINUE
IF(IDIR.EQ.0) WRITE(6,100) PHIR,ALAMR
100 FORMAT(1HO,'CANNOT FIND NORMAL FOR TURNING POINT AT',2F10.2)
IF(IDIR.EQ.0) RETURN
ANGLE=DARCOS(-DTAN(PHIR*DTR)*DTAN(PHIN*DTR))/DTR
IF(IDIR.EQ. 1) ALAMN=ALAMR-ANGLE
IF(IDIR.EQ.-1) ALAMN=ALAMR+ANGLE
RH(1)=DCOS(PHIR*DTR)*DCOS(ALAMR*DTR)
RH(2)=DCOS(PHIR*DTR)*DSIN(ALAMR*DTR)
RH(3)=DSIN(PHIR*DTR)
ANORM(1)=DCOS(PHIN*DTR)*DCOS(ALAMN*DTR)
ANORM(2)=DCOS(PHIN*DTR)*DSIN(ALAMN*DTR)
ANORM(3)=DSIN(PHIN*DTR)
VH(I)=-R/4*(2)*ANORM(3)+ANORM(2)*RH(3)
VH(2)=-R(3)*ANORM(1)+ANORM(3)*RH(1)
VH(3)=-R(1)*ANORM(2)+ANORM(1)*RH(2)
RETURN
END
// EXEC LINKGO,REGION.GO=500K
//GO.FT10F001 DD DSN=XRJRR.GARP,DISP=SHR
//* TAPE 11 IS OUTPUT
//GO.FT11F001 DD DSN=XRJRR.DMSP.FITXYZ,UNIT=SYSDA,DISP=SHR
//* DBC=(RECFM=VBS,LRECL=11204,BLKSZE=22412),SPACE=(TRK,(20,20),RLSE),
//* VOL=SER=SACC03
// EXEC NOTIFYTS

A-112
The DMSP F-7 satellite was an operational Air Force meteorological satellite which carried a magnetometer for geophysical measurements. The magnetometer was located within the body of the spacecraft in the presence of large spacecraft fields. In addition to stray magnetic fields, the data have inherent position and time inaccuracies. Algorithms were developed to identify and remove time varying magnetic field noise from the data. These algorithms are embodied in an automated procedure which fits a smooth curve through the data and them identifies outliers and which filters the predominant Fourier components of noise from the data. Techniques developed for Magsat were then modified and used to attempt determination of the spacecraft fields, of any rotation between the magnetometer axes and the spacecraft axes, and of any scale changes within the magnetometer itself. Software setup and usage are documented and program listings are included in the Appendix. The initial and resulting data are archived on magnetic cartridge and the formats are documented.