N64-29364
(ACCESSION NUMBER)
22
(PAGES)
CR-58293
(NASA CR OR TMX OR AD NUMBER)
(THRU)
1
(CODE)
20
(CATEGORY)
FACILITY FORM 602

72p

COMPUTER CODES FOR THE EVALUATION OF SPACE RADIATION HAZARDS

VOL. 2. VAN ALLEN RADIATION

D2-90418-2

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER

OTS PRICE
XEROX $ 2.60 ph
MICROFILM $

Contract NAS 9-1393

By
Wing Y. Wong

Aero-Space Division
THE BOEING COMPANY
Seattle, Washington

TABLE OF CONTENTS

	PAGE
GENERAL INFORMATION	4
PURPOSE	4
ASSUMPTIONS	6
LIMITATIONS	6
RECOMMENDATIONS	7
PROCEDURE	8
NOMENCLATURE	8
METHOD	8
RESULTS	16
INPUT PREPARATION AND OUTPUT DESCRIPTION	17
INPUT DATA PREPARATION	17
OUTPUT DATA	18
PROGRAMMING INFORMATION	20
FLOW CHARTS	22
PROGRAM LISTING	24
SAMPLE INPUT DATA	60
SAMPLE PROBLEM	62
APPENDIX A	68
APPENDIX B	70
REFERENCES	71

LIST OF TABLES

TABLE		PAGE
1.	Nomenclature	9

LIST OF FIGURES

FIGURE		PAGE
1	The method and derivation used to generate trajectory points for the circular orbits.	13

GENERAL INFORMATION

PURPOSE

The Van Allen Radiation Belt is comprised of highly energetic protons and electrons. Space vehicles passing through the belt are subjected to radiation hazards. In order to evaluate the radiation hazard to components and personnel, it is necessary to measure the total flux of energetic protons and electrons along the path of the vehicle.

29364

This program is an all FORTRAN code which computes the flux and the total time-integrated fluxes of protons and electrons for vehicle trajectories along the Van Allen Radiation Belt. For study purposes, this belt is differentiated into four separate belts:

1. High energy proton belt, for protons with energy greater than 20 Mev.
2. Low energy proton belt, for protons with energy less than 20 Mev.
3. AFSWC threat electron belt, which is a theoretical belt.
4. Electron belt, as it existed on October 20, 1962.

These belts are represented by spatial flux maps which are read in as inputs to the program. These maps were constructed from empirical measurements received from satellites. Flux values are given as functions of B (magnetic flux density in gauss) and of L (McIlwain parameter in units of earth radii). A description of, and the reasons for, using the B and L coordinate system, are given in Ref. 1.

Author

Three types of inputs are acceptable. The program will accept trajectory points, one at a time, or it can generate trajectory points for circular orbits,

Hohman transfer orbits, or elliptic orbits. The inputs required for each type are given as follows:

1. Input of trajectory points:
   a. Time (sec)
   b. Latitude (deg)
   c. Longitude (deg)
   d. Altitude (km)
2. Input for generation of circular orbits:
   a. Initial latitude (deg)
   b. Initial longitude (deg)
   c. Orbital inclination (deg)
   d. Altitude (km)
   e. Time interval (sec)
   f. Final time (sec)
   g. Starting time (sec)
3. Input for generation of Hohman transfer orbit and/or elliptic orbits:
   a. Initial latitude (deg)
   b. Initial longitude (deg)
   c. Orbital inclination (deg)
   d. Perigee or initial altitude (km)
   e. Apogee or final altitude (km)
   f. Interval between points in the ephemeris (deg)
   g. Total number of trajectory points (n.d.)

At each point of calculation, the program prints out the input quantities (or generated trajectory points), the corresponding B and L coordinates, the individual values of flux, the product of flux and time increment, and the total integrated fluxes for each of the flux belts.

ASSUMPTIONS

1. Spherical earth is assumed.

2. The flux maps used are not functions of the energy spectrum, and the flux values are assumed to be omnidirectional.

3. The proton and electron belts are assumed to be adequately represented by flux maps which can be interpolated for specified values of B and L.

LIMITATIONS

1. Since a spherical earth is assumed, orbit precession is not taken into account.

2. The flux maps may not be too reliable because of various limitations of empirical measurements.

3. Interpolation errors may result because of sparcity of flux entries in the flux maps. This can be rectified by expanding the size of the array.

4. Errors may be introduced if the integration time interval is too large, causing some areas of the belt to be bypassed. This error is due, in part, to the process of interpolation. A simple solution would be to decrease the integration time interval, or alternatively, to employ a more sophisticated interpolation scheme.

RECOMMENDATIONS

1. To avoid inaccuracies due to the methods of integration and interpolations used, it is suggested that some compromise be made involving the integration time interval and the use of a more refined interpolation scheme. Machine time is a governing factor, however.

2. It may be worthwhile to refine the flux maps to have the flux values dependent on the energy spectrum, so that total fluxes can be obtained as functions of the energy spectrum.

PROCEDURE

NOMENCLATURE

The nomenclature is presented in Table 1.

METHOD

When the trajectory input is given point by point, then the method is straightforward. The points are converted to B and L coordinates from which a table look-up and linear interpolation are performed to obtain the flux values. The product of the flux and the time interval are added to the previous product sum to form the total time-integrated flux. An explanation of the B and L conversion is given in Ref. 1. The interpolation scheme is given in Appendix A.

When circular orbits are to be generated, the program will calculate the trajectory points. The latitude and longitude for each trajectory point are calculated from the equations given in Ref. 2. The following explains the method in more detail:

1. Generation of circular orbits.

To generate trajectory points for the circular orbits, the method and derivation used are presented in Fig. 1. In Fig. 1 the initial heading, β_1, is solved from the following equation:

$$\sin \beta_1 = \frac{\cos i}{\cos \psi_1}$$

The trajectory points for the circular orbits are calculated in the following manner:

TABLE 1. Nomenclature

MATH SYMBOL	PROGRAM SYMBOL	DEFINITION	UNITS
h	ALT	Altitude	km
h_1	ALT1	Perigee or initial altitude	km
h_2	ALT2	Apogee or final altitude	km
h	ALTI	Altitude array	km
	ALTNM	Altitude in nautical miles	n. m.
B	BB	Magnetic flux density	gauss
B_{max}	BBMAX	Maximum B value of flux map	gauss
B_{min}	BBMIN	Minimum B value of flux map	gauss
Δt	DT	Trajectory time interval	sec
	DTIME	Time between trajectory points	sec
	DFXELG	Electron flux $L>1.5$	electrons/cm^2
	DFXELL	Electron flux $L\leq 1.5$	electrons/cm^2
	DFXETG	Threat electron flux $B\geq BC1$	electrons/cm^2
	DFXETL	Threat electron flux $B<BC1$	electrons/cm^2
	DFXPHE	High energy proton flux	protons/cm^2
	DFXPLE	Low energy proton flux	protons/cm^2
	DV	Interval between points in the ephemeris	deg.
	ELEB	Electron map B values	gauss
	ELEFX	Electron map flux values	electrons/cm^2-sec
	ELEL	Electron map L values	earth radii

TABLE 1. Nomenclature (continued)

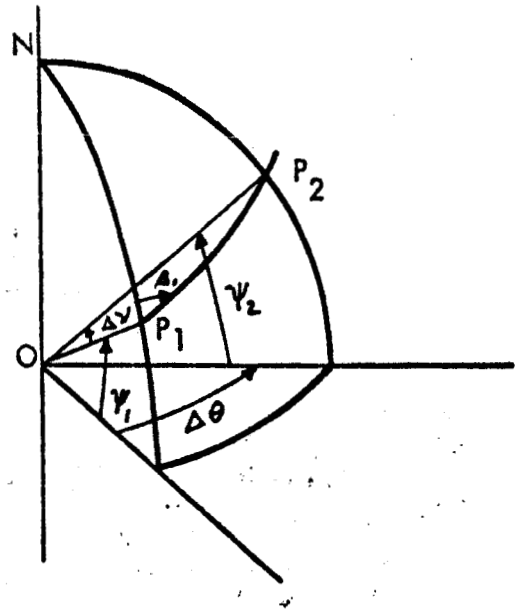
MATH SYMBOL	PROGRAM SYMBOL	DEFINITION	UNITS
	ETHB	AFSWC threat electrons B values	gauss
	ETHB1	AFSWC threat electrons BC1 values	gauss
	ETHBC	AFSWC threat electrons BC values	gauss
	ETHFX	AFSWC threat electrons flux values	electrons/cm^2-sec
	ETHL	AFSWC threat electrons L values	earth radii
	ETHL1	AFSWC threat electrons LC1 values	earth radii
	ETHLC	AFSWC threat electrons LC values	earth radii
L	FL	McIlwain parameter L	earth radii
ψ	FLAT	Latitude array	deg
L_{max}	FLMAX	Maximum L value of flux map	earth radii
L_{min}	FLMIN	Minimum L value of flux map	earth radii
λ	FLON	Longitude array	deg
	FXELG	Flux rate electrons $L > 1.5$	electrons/cm^2-sec
	FXELL	Flux rate electrons $L \leq 1.5$	electrons/cm^2-sec
	FXETG	Flux rate threat electrons $B \geq BC1$	electrons/cm^2-sec
	FXETL	Flux rate threat electrons $B < BC1$	electrons/cm^2-sec

TABLE 1. Nomenclature (continued)

MATH SYMBOL	PROGRAM SYMBOL	DEFINITION	UNITS
	FXPHE	Flux rate high energy protons	protons/cm^2-sec
	FXPLE	Flux rate low energy protons	protons/cm^2-sec
	NPOINT	Total number of trajectory points	--
Ω_E	OMEGA	Earth rotational rate	deg/sec
i	ORBINC	Orbit inclination	deg
	PHEB	High energy proton map B values	gauss
	PHEFX	High energy proton map flux values	protons/cm^2-sec
	PHEL	High energy proton map L values	earth radii
π	PI	Value of π	--
	PLEB	Low energy proton map B values	gauss
	PLEFX	Low energy proton map flux values	protons/cm^2-sec
	PLEL	Low energy proton map L values	earth radii
r	R	Distance from radius of earth to vehicle	km
	RADIAN	Conversion factor radian to degrees	deg/rad
r_e	RE	Radius of earth	km
	SUMELG	Integrated flux electrons $L > 1.5$	electrons/cm^2
	SUMELL	Integrated flux electrons $L \leq 1.5$	electrons/cm^2
	SUMETG	Integrated flux threat electrons $B \geq BC1$	electrons/cm^2
	SUMETL	Integrated flux threat electrons $B < BC1$	electrons/cm^2
	SUMPHE	Integrated flux high energy protons	protons/cm^2

TABLE 1. Nomenclature (continued)

MATH SYMBOL	PROGRAM SYMBOL	DEFINITION	UNITS
	SUMPLE	Integrated flux low energy protons	protons/cm^2
t	T	Time array	sec
t	TIME	Time	sec
	TLAST	Previous time	sec
	TMAX	Maximum trajectory time	sec
	TMIN	Minimum trajectory time	sec
2π	TPI	2 times π	--
ψ	XL	Latitude	deg
ψ_1	XLATO	Starting latitude	deg
λ	XLONG	Longitude	deg
λ_1	XLONGO	Starting longitude	deg



$\Delta\nu$ = geocentric angle between displacement vectors of initial point P_1 and point P_2

$\Delta\theta$ = polar angle

β_1 = heading at point P_1

ψ_1 = latitude at initial point P_1

ψ_2 = latitude at point P_2

λ_1 = longitude at initial point P_1

λ_2 = longitude at point P_2

ω = rotational velocity of the earth

Δt = time required for vehicle to cover the true-anomaly arc

i = orbital inclination measured counter clockwise from equator

t = time

r_e = radius of earth

h = altitude of orbit

FIGURE 1. The method and derivation used to generate trajectory points for the circular orbits.

The change in true-anomaly $\Delta\nu$:

$$\Delta\nu = \sqrt{\frac{9.81 \times 10^{-3}}{r_e + h}} \left(\frac{r_e t}{r_e + h}\right)$$

From the spherical law of cosines

$$\cos a = \cos b \cos c + \sin b \sin c \cos A$$

The following relationships can be rewritten:

Consider the spherical triangle P_1NP_2

a. Solution of latitude Ψ_2

$$\cos \overline{P_2N} = \cos \overline{P_1P_2} \cos \overline{P_1N} + \sin \overline{P_1P_2} \sin \overline{P_1N} \cos \beta_1$$

Substituting $\Delta\nu = \overline{P_1P_2}$

$$\sin \Psi_1 = \cos \overline{P_1N}$$

$$\sin \Psi_2 = \cos \overline{P_2N}$$

$$\cos \Psi_2 = \cos\Delta\nu \sin \Psi_1 + \sin\Delta\nu \cos \Psi_1 \cos \beta_1$$

Thus the latitude Ψ_2 at point P_2 can be solved, given the initial heading β_1, initial latitude Ψ_1, and the change in true-anomaly $\Delta\nu$

b. Solution of longitude λ_2

$$\lambda_2 = \lambda_1 + \Delta\theta - \omega\Delta t$$

First, solve for the polar angle $\Delta\theta$

$$\cos \overline{P_1P_2} = \cos \overline{P_1N} \cos \overline{P_2N} + \sin \overline{P_1N} \sin \overline{P_2N} \cos\Delta\theta$$

which give $\cos\Delta\theta = \dfrac{\cos\Delta\nu - \sin \Psi_1 \sin \Psi_2}{\cos \Psi_1 \cos \Psi_2}$

Thus the longitude λ_2 at point P_2 can be found, given the initial latitude Ψ_1, final latitude Ψ_2, and the change in true-anomaly $\Delta\nu$.

The following angle adjustments are made:

c. Adjustment for heading β_1 which is dependent on the value of the orbital inclination i.

i in first quadrant, set $\beta_1 = \beta_1$

i in second quadrant, set $\beta_1 = 2\pi - |\beta_1|$

i in third quadrant, set $\beta_1 = \pi + |\beta_1|$

i in fourth quadrant, set $\beta_1 = \pi - |\beta_1|$

d. Adjustment for polar angle $\Delta\theta$, which is dependent on the change in true anomaly $\Delta\nu$

(1) For easterly flights, i.e., $0 < \beta_1 \leq \pi$

if $0 < \Delta\nu \leq \pi$, set $\Delta\theta = \Delta\theta$

if $\pi < \Delta\nu \leq 2\pi$, set $\Delta\theta = 2\pi - |\Delta\theta|$

(2) For westerly flights, i.e., $\pi < \beta_1 \leq 2\pi$

if $0 < \Delta\nu \leq \pi$, set $\Delta\theta = 2\pi - |\Delta\theta|$

if $\pi < \Delta\nu \leq 2\pi$, set $\Delta\theta = \Delta\theta$

2. Generation of Hohman transfer orbits and elliptical orbits. These calculations are performed in the subroutine SATRAC, a description of which is given in Ref. 4.

3. Table look-up of flux values. Flux values are given as functions of B (magnetic flux density) and of L (McIlwain parameter). These values are read in as inputs to the computer. The flux values are first converted to their logarithms (base e). A table look-up is then performed using linear interpolation of the logarithmic fluxes. The anti-log of this Interpolated value is the final flux.

For $B_i \leq B \leq B_{i+1}$

$L_j \leq L \leq L_{j+1}$

$$\text{Log flux} = \frac{L_{j+1} - L}{L_{j+1} - L_j}\left\{\frac{B_{i+1} - B}{B_{i+1} - B_i} F(B_i, L_j) + \frac{B - B_i}{B_{i+1} - B_i} F(B_{i+1}, L_j)\right\}$$

$$+ \frac{L - L_j}{L_{j+1} - L_j}\left\{\frac{B_{i+1} - B}{B_{i+1} - B_i} F(B_i, L_{j+1}) \frac{B - B_i}{B_{i+1} - B_i} F(B_{i+1}, L_{j+1})\right\}$$

where $F(B_i, L_j)$ is the logarithm of the flux at $B = B_i$ and $L = L_j$. This method of interpolation is used because it has been shown to give better results.

RESULTS

The interpolated flux values are compared to the values given in the flux maps, and have been shown to be accurate. The total time-integrated flux cannot be compared to any empirical data because no such data exist. The counters on the vehicles only measure, at any point, the value of flux and not the total-integrated flux.

INPUT PREPARATION AND OUTPUT DESCRIPTION

INPUT DATA PREPARATION

Input of Trajectory Points

The first card specifies the input type, by the BCD word POINTS in Columns 1-6. The second card specifies the number of trajectory points in format I10. Thereafter, each successive card specifies a trajectory point, in format F10.0.

Columns	Program Symbol	Description	Unit
1-10	TIME	Time	sec
11-20	XL	Latitude	deg
21-30	XLONG	Longitude	deg
31-40	ALT	Altitude	km

Input for Generation of Circular Orbits

The first card specifies the input type, by the BCD word CIRCULAR in Columns 1-8. The second card specifies the circular orbit, in format F10.0 as follows:

Columns	Program Symbol	Description	Unit
1-10	XLATO	Initial Latitude	deg
11-20	XLONGO	Initial Longitude	deg
21-30	ORBINC	Orbit Inclination	deg
31-40	ALT	Altitude	km
41-50	DT	Time Interval	sec
51-60	TMAX	Final Time	sec
61-70	TMIN	Initial Time	sec

Input for Generation of Hohman Transfer Orbit and/or Elliptic Orbits

The first card specifies the input type, by the BCD word TRANSFER in Columns 1-8. The second card specifies the Hohman transfer orbit and/or elliptic orbit, in format F10.0 as follows:

Columns	Program Symbol	Description	Unit
1-10	XLATO	Initial latitude	deg
11-20	XLONGO	Initial longitude	deg.
21-30	ORBINC	Orbit inclination	deg
31-40	ALT1	Perigee or initial altitude	km
41-50	ALT2	Apogee or final altitude	km
51-60	DV	Interval between points in the ephemeris	deg
61-70	NPOINT	Total number of trajectory points	--

Note that a Hohman transfer orbit is completed in 180 degrees, and is just half an elliptic orbit. For example, if DV is taken as 3 degrees, NPOINT is set equal to 180/3 = 60 to obtain a transfer orbit only. For one elliptic orbit, NPOINT would be 120; for two orbits, 240, and so forth.

OUTPUT DATA

The output listings are self-explanatory. For each point in space, as specified by time, latitude, longitude, and altitude, the proton and electron fluxes and their time-integrated total fluxes are printed.

For purposes of study, the AFSWC threat electron fluxes are separated into two groups, depending on a set of B cutoff values. The group of fluxes for B greater than B cutoff refers to those electrons in transient conditions at lower altitudes. These are the electrons that would be lost before they drifted around the earth.

The electron fluxes are also separated into two groups, depending on the value of L. The group of fluxes for L less than 1.5 refers to the fluxes based on the fission spectrum. The group of fluxes for L greater than 1.5 refers to the fluxes based on the exponential energy spectrum.

PROGRAMMING INFORMATION

This is a standard FORTRAN job. There are no additional tape requirements other than those required for a standard FORMON job, and no special operating instructions. Logical tape 5 is used as the input tape, using the FORTRAN statement READ INPUT TAPE. Logical tape 6 is used as the output tape, using the FORTRAN statement WRITE OUTPUT TAPE. Logical tape 14 is used as the punch output tape.

This is a small program so that no segmentation is required

The storage requirements are:

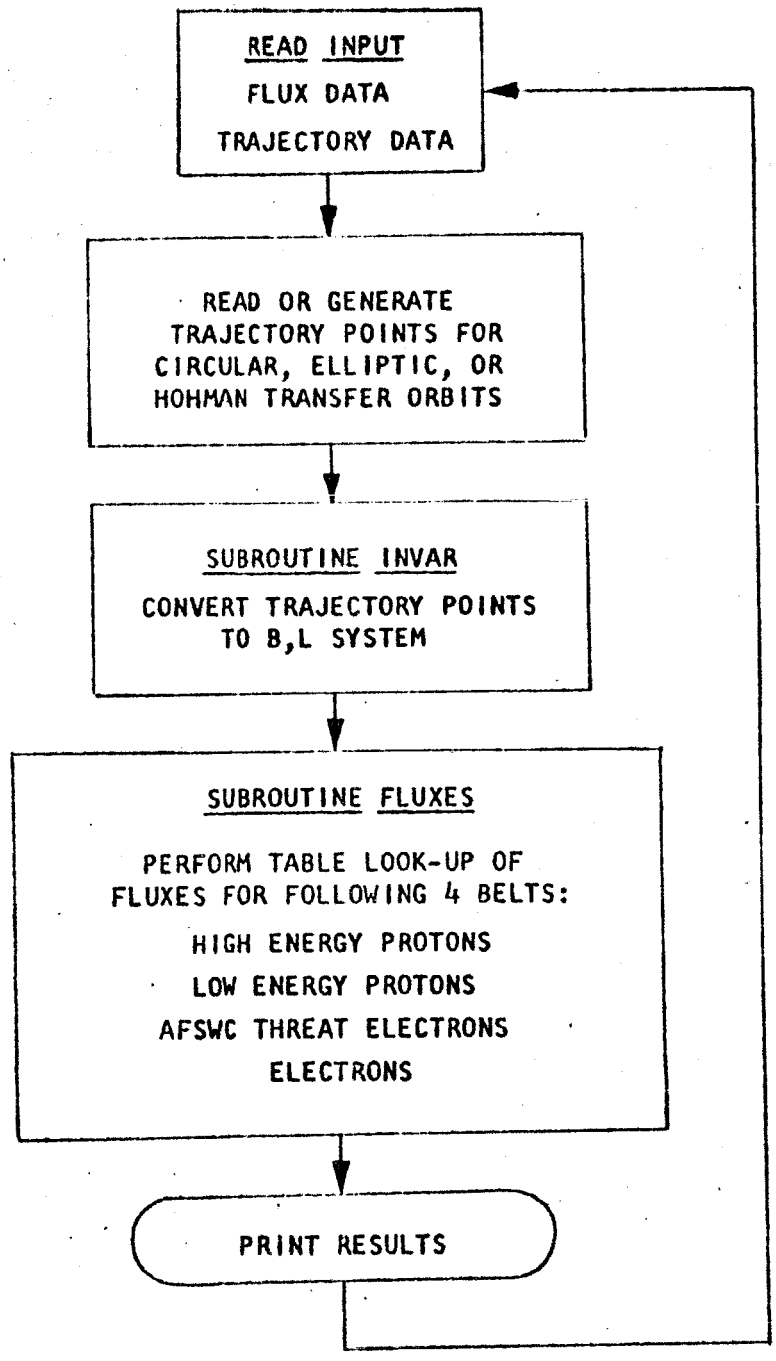
Main Program	1,300
Subroutines	9,400
Common	17,800

The following library routines are required:

SQRT	DEXP(2)
COS	(DFSB)
LOG	(DFAD)
EXP	DSQRT
EXP(3)	ACOS
SIN	EXIT
(DFMP)	DSIN
(DFDP)	DLOG
(TSHM)	SINH
(RTN)	(FPT)
DCOS	(TSB)

(STHM)	(RLR)
(FIL)	(STB)
ASIN	(WLR)

FLOW CHARTS


READ INPUT
FLUX DATA
TRAJECTORY DATA
READ OR GENERATE TRAJECTORY POINTS FOR CIRCULAR, ELLIPTIC, OR HOHMAN TRANSFER ORBITS
SUBROUTINE INVAR
CONVERT TRAJECTORY POINTS TO B,L SYSTEM
SUBROUTINE FLUXES
PERFORM TABLE LOOK-UP OF FLUXES FOR FOLLOWING 4 BELTS:
HIGH ENERGY PROTONS
LOW ENERGY PROTONS
AFSWC THREAT ELECTRONS
ELECTRONS
PRINT RESULTS

PROGRAM LISTING

```
      DIMENSION DUMMY(100), T(1500), FLAT(1500), FLON(1500), ALTI(1500)
      COMMON DUMMY,T,FLAT,FLON,ALTI
      COMMON XLATO,XLONGO,ORBINC,ALT1,ALT2,DV,NPOINT
      COMMON HEADG
      EQUIVALENCE (XLATO,AT1),(XLONGO,ONG1),(ORBINC,HDG1),(NPOINT,N)
      DIMENSION PHEB(70),PLEB(70),ETHB(70),ELEB(70),
     1PHEL(50),PLEL(50),ETHL(50),ELEL(50),
     2PHEFX(2000),PLEFX(2000),ETHFX(2000),ELEFX(2000),
     3ETHBC(50),ETHB1(50),ETHLC(50),ETHL1(50)
      COMMON PHEB,PHEL,PHEFX,
     1PLEB,PLEL,PLEFX,
     2ETHB,ETHL,ETHFX,ETHBC,ETHLC,ETHB1,ETHL1,
     3ELEB,ELEL,ELEFX
C
C     VAN ALLEN RADIATION PROGRAM
C     COMPUTES PROTON AND ELECTRON FLUXES
C
C     INPUT REQUIRED---
C
C     REFER TO LISTING OF SUBROUTINE FLUXES FOR
C     PROPER SETUP OF FLUX DATA
C
C     THE FOLLOWING INSTRUCTIONS REFER TO THE SETUP
C     OF TRAJECTORY DATA
C
C     FIRST CARD SPECIFIES TYPE OF INPUT BY A BCD WORD
C       POINTS   - TRAJECTORY POINTS ARE GIVEN AS INPUT
C       CIRCULAR - CIRCULAR ORBITS ARE TO BE GENERATED
C       TRANSFER - TRANSFER (AND ELLIPTICAL) ORBITS TO BE GENERATED
C
C     1 IF FIRST CARD IS -POINTS- ,THE SECOND CARD GIVES
C       THE NUMBER OF POINTS SPECIFIED,FORMAT (I10)
C       ALL SUCCEEDING CARDS SUPPLY FOLLOWING DATA IN FORMAT (4F10.0)
C         1      T = TIME (SEC)
C         2   FLAT = LATITUDE (DEG)
C         3   FLON = LONGITUDE (DEG)
C         4   ALTI = ALTITUDE (KM)
C
C     2 IF FIRST CARD IS -CIRCULAR- ,ONLY ONE ADDITIONAL
C       CARD IS REQUIRED GIVING THE FOLLOWING DATA IN FORMAT (7F10.0)
C         1   XLATO = INITIAL LATITUDE (DEG)
C         2  XLONGO = INITIAL LONGITUDE (DEG)
C         3  ORBINC = ORBIT INCLINATION (DEG)
C         4      ALT = ALTITUDE (KM)
C         5       DT = TIME INTERVAL DESIRED (SEC)
C         6     TMAX = ENDING TIME (SEC)
C         7     TMIN = STARTING TIME (SEC)
C
C     3 IF FIRST CARD IS -TRANSFER- ,ONLY ONE ADDITIONAL
C       CARD IS REQUIRED GIVING THE FOLLOWING DATA IN FORMAT (6F10.0,I10)
C         1   XLATO = INITIAL LATITUDE (DEG)
C         2  XLONGO = INITIAL LONGITUDE (DEG)
C         3  ORBINC = ORBIT INCLINATION (DEG)
```

```
C        4    ALT1 = PERIGEE OR INITIAL ALTITUDE (SEC)
C        5    ALT2 = APOGEE OR FINAL ALTITUDE (SEC)
C        6      DV = INTERVAL BET. POINTS IN THE EPHEMERIS (DEG)
C        7  NPOINT = TOTAL NUMBER OF TRAJECTORY POINTS
C
C     CONSTANTS
C
    1 RE=6371.2
      RADIAN=57.29578
      PI=3.1415927
      TPI=6.2831954
      OMEGA=4.167E-3
C
C     READ IN FLUX DATA
C
      CALL FLUXES (BB,FL,DTIME,
     1FXPHE,DFXPHE,SUMPHE,FXPLE,DFXPLE,SUMPLE,
     2FXETL,DFXETL,SUMETL,FXETG,DFXETG,SUMETG,
     3FXELL,DFXELL,SUMELL,FXELG,DFXELG,SUMELG)
    5 WRITE OUTPUT TAPE 6,6
    6 FORMAT (1H1,5X,27HVAN ALLEN RADIATION PROGRAM//)
   10 READ INPUT TAPE 5,11,ORBIT
   11 FORMAT (A6)
   20 IF (ORBIT-6HPOINTS) 21,100,21
   21 IF (ORBIT-6HCIRCUL) 22,200,22
   22 IF (ORBIT-6HTRANSF) 23,300,23
   23 WRITE OUTPUT TAPE 6,24
   24 FORMAT (1H1,27HTYPE OF INPUT NOT SPECIFIED)
   25 CALL EXIT
C
C     TRAJECTORY POINTS GIVEN AS INPUT
C
  100 READ INPUT TAPE 5,101,NPOINT
  101 FORMAT (I10)
  102 READ INPUT TAPE 5,103,(T(I),FLAT(I),FLON(I),ALTI(I),I=1,NPOINT)
  103 FORMAT (4F10.0)
  110 WRITE OUTPUT TAPE 6,111,NPOINT
  111 FORMAT (10X,I4,1X,36HTRAJECTORY POINTS ARE GIVEN AS INPUT)
      GO TO 501
C
C     CIRCULAR ORBITS ARE TO BE GENERATED
C
C     REFERENCE-BOEINT COORDINATION SHEET NO. 2-5342-1-71
C     --AS 1065 CALCULATION OF ORBIT EQUATION--
C     BY ZACHARY S. PRICE OF SPACE MECHANICS GROUP
C
  200 READ INPUT TAPE 5,201,XLATO,XLONGO,ORBINC,ALT,DT,TMAX,TMIN
  201 FORMAT (7F10.0)
      ALTNM=ALT/1.853
  210 WRITE OUTPUT TAPE 6,211
  211 FORMAT (10X,15HCIRCULAR ORBITS///)
  212 WRITE OUTPUT TAPE 6,213,XLATO,XLONGO,ORBINC,ALT,DT,TMAX,TMIN
  213 FORMAT (15X,16HINITIAL LATITUDE,4X,F10.2,2X,5H(DEG)/
     115X,17HINITIAL LONGITUDE,3X,F10.2,2X,5H(DEG)//
```

```
     215X,17HORBIT INCLINATION,3X,F10.2,2X,5H(DEG)/
     315X,8HALTITUDE,12X,F10.2,2X,4H(KM)//
     415X,13HTIME INTERVAL,7X,F10.2,2X,5H(SEC)/
     515X,11HENDING TIME,9X,F10.2,2X,5H(SEC)/
     615X,13HSTARTING TIME,7X,F10.2,2X,5H(SEC))
  220 J=1
      T(1)=TMIN
C
C     SINCE ASINF GIVES PRINCIPAL VALUES ONLY
C     THE INITIAL HEADING B1 HAS TO BE ADJUSTED
C     DEPENDENT ON THE ORBIT INCLINATION
C     B1 GREATER THAN PI MEANS WESTERLY FLIGHT
C
  230 SINB1=COSF(ORBINC/RADIAN)/COSF(XLATO/RADIAN)
      SINB1 = MIN1F(1.0,MAX1F(-1.0,SINB1))
      B1=ASINF(SINB1)
  231 IF (ORBINC) 232,233,233
  232 ORBINC=ORBINC+360.
  233 IF (ORBINC-90.) 235,239,234
  234 IF (ORBINC-180.) 235,235,236
  235 B1=TPI-ABSF(B1)
      GO TO 239
  236 IF (ORBINC-270.) 237,237,238
  237 B1=PI+ABSF(B1)
      GO TO 239
  238 B1=PI-ABSF(B1)
  239 CONTINUE
  240 XPHI=SQRTF(9.81E-3/(RE+ALT))*RE*T(J)/(RE+ALT)
      XPHI=MODF(XPHI,TPI)
  241 SINXL2=COSF(XPHI)*SINF(XLATO/RADIAN)+SINF(XPHI)*COSF(XLATO/RADIAN)
     1*COSF(B1)
      SINXL2=MIN1F(1.0,MAX1F(-1.0,SINXL2))
      XL2=ASINF(SINXL2)
C
C     SINCE ACOSF GIVES PRINCIPAL VALUES ONLY
C     THE POLAR ANGLE DTHETA HAS TO BE ADJUSTED
C     DEPENDENT ON THE CHANGE IN TRUE ANOMALY XPHI
C     AND DEPENDENT ON B1 (I.E.,EASTERLY OR WESTERLY FLIGHT)
C
  242 COSDT=(COSF(XPHI)-SINF(XLATO/RADIAN)*SINXL2)/(COSF(XLATO/RADIAN)
     1*COSF(XL2))
      COSDT=MIN1F(1.0,MAX1F(-1.0,COSDT))
      DTHETA=ACOSF(COSDT)
  250 IF (B1-PI) 251,251,252
  251 IF (XPHI-PI) 254,254,253
  252 IF (XPHI-PI) 253,253,254
  253 DTHETA=TPI-ABSF(DTHETA)
  254 CONTINUE
C
C     NEW LATITUDE AND LONGITUDE ARE DETERMINED
C
  260 FLAT(J)=XL2*RADIAN
      FLON(J)=XLONGO+DTHETA*RADIAN-OMEGA*T(J)
      FLON(J)=MODF(FLON(J),360.)
```

```
      IF (T(J)-TMAX) 270,500,500
  270 J=J+1
      T(J)=T(J-1)+DT
      GO TO 240
C
C     HOHMAN TRANSFER TRAJECTORY AND ELLIPTIC ORBITS
C
C
C     REFERENCE-BOEING COORDINATION SHEET NO. 2-5342-1-107
C     --SUBROUTINE TO CALCULATE LOCATION IN
C     TRANSFER TRAJECTORY--BY STURE FOLKESON OF
C     SPACE MECHANICS GROUP, DATED 4-18-63
C
  300 CALL SATRAC
  310 IF (FLOATF(NPOINT)-180.0/DV-10.0) 311,313,313
  311 WRITE OUTPUT TAPE 6,312
  312 FORMAT (10X,27HHOHMAN TRANSFER TRAJECTORY///)
      GO TO 315
  313 WRITE OUTPUT TAPE 6,314
  314 FORMAT(11X,46HHOHMAN TRANSFER TRAJECTORY AND ELLIPTIC ORBITS///)
  315 WRITE OUTPUT TAPE 6,316,XLATO,XLONGO,HEADG,ALT1    ,ALT2,DV,NPOINT
  316 FORMAT (15X,16HINITIAL LATITUDE,4X,F10.2,2X,5H(DEG)/
     115X,17HINITIAL LONGITUDE,3X,F10.2,2X,5H(DEG)//
     215X,17HORBIT INCLINATION,3X,F10.2,2X,5H(DEG)/
     315X,7HPERIGEE,13X,F10.2,2X,4H(KM)/
     415X,6HAPOGEE,14X,F10.2,2X,4H(KM)//
     515X,16HANGULAR INTERVAL,4X,F10.2,2X,5H(DEG)/
     615X,17HTRAJECTORY POINTS,3X,I10,2X)
      NPOINT=NPOINT+1
      GO TO 501
C
C     INITIALIZATION
C
  500 NPOINT=J
  501 I=1
      II=1
      NCOUNT=0
      SUMPHE=0.
      SUMPLE=0.
      SUMETL=0.
      SUMETG=0.
      SUMELE=0.
      SUMEGR=0.
  510 DO 800 L=1,NPOINT
C
C     TRAJECTORY POINT IS SELECTED
C
  520 TIME=T(L)
      XLONG=MODF(360.+FLON(L),360.)
      XL=FLAT(L)
  521 IF (ORBIT-6HCIRCUL) 522,523,522
  522 ALT=ALTI(L)
      ALTNM=ALT/1.853
  523 CONTINUE
```

```
  530 R=RE+ALT
  531 GO TO (532,533),I
  532 TLAST = TIME
      I = 2
  533 CONTINUE
      IF (R-7.0*RE) 600,700,700
C
C     B AND EL ARE COMPUTED FROM INVAR PACKAGE
C
  600 CALL INVAR (XL,XLONG,ALT,.01,BB,FL)
      DTIME=TIME-TLAST
C
C     COMPUTATION OF PROTON AND ELECTRON FLUXES
C
C
      CALL FLUXES (BB,FL,DTIME,
     1FXPHE,DFXPHE,SUMPHE,FXPLE,DFXPLE,SUMPLE,
     2FXETL,DFXETL,SUMETL,FXETG,DFXETG,SUMETG,
     3FXELL,DFXELL,SUMELL,FXELG,DFXELG,SUMELG)
C
C     THE HEADING PLUS 8 TRAJECTORY POINTS ARE PRINTED
C     PER PAGE
C
  700 CONTINUE
  703 GO TO (710,720),II
  710 WRITE OUTPUT TAPE 6,711
  711 FORMAT (1H1,14X,9H--XLONG--3X,6H--XL--4X,7H--ALT--5X,
     19H--BVALU--3X,8H-ELVALU-14X,7H-FVALU-3X,
     28H--FLUX--5X,8H-FLXSUM-/
     36X,4HTIME5X,9HLONGITUDE2X,8HLATITUDE3X,
     48HALTITUDE4X,9HMAG. FLUX3X,9HMC ILWAIN2X,
     535X,6HSUM OF/
     64X,9H(SECONDS)2X,19H(DEGREES) (DEGREES)1X,12H(KILOMETERS)3X,
     77HDENSITY4X,9HPARAMETER2X,12X,4HFLUX5X,
     89HFLUX*TIME4X,9HFLUX*TIME/
     935X,13H(NAUT. MILES))
  712 WRITE OUTPUT TAPE 6,713
  713 FORMAT (86X,19HHIGH ENERGY PROTONS/
     186X,18HLOW ENERGY PROTONS/
     286X,31HTHREAT ELECTRONS B LESS THAN BC/
     386X,34HTHREAT ELECTRONS B GREATER THAN BC/
     486X,25HL LESS THAN 1.5 ELECTRONS/
     586X,26HL GREATER THAN 1.5 ELECTRONS)
  716 II=2
  720 NCOUNT=NCOUNT+1
      IF (NCOUNT-6) 730,721,721
  721 II=1
      NCOUNT=0
  730 WRITE OUTPUT TAPE 6,731,TIME,XLONG,XL,ALT,
     1BB,FL,FXPHE,DFXPHE,SUMPHE,ALTNM,
     2FXPLE,DFXPLE,SUMPLE,
     3FXETL,DFXETL,SUMETL,FXETG,DFXETG,SUMETG,
     4FXELL,DFXELL,SUMELL,
     5FXELG,DFXELG,SUMELG
```

```
731 FORMAT (/2X,F12.3,2X,F8.2,2X,F8.2,2X,F10.2,2X,
   1F10.5,2X,F10.5,12X,1PE10.5,2X,1PE12.5,2X,1PE12.5/36X,0PF10.2,
   236X,1PE10.5,2X,1PE12.5,2X,1PE12.5,
   34(/82X,1PE10.5,2X,1PE12.5,2X,1PE12.5))
732 TLAST=TIME
800 CONTINUE
    GO TO 5
    END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE FLUXES (BB,FL,DTIME,
     1FXPHE,DFXPHE,SUMPHE,FXPLE,DFXPLE,SUMPLE,
     2FXETL,DFXETL,SUMETL,FXETG,DFXETG,SUMETG,
     3FXELL,DFXELL,SUMELL,FXELG,DFXELG,SUMELG)
      DIMENSION DUMMY(100), T(1500), FLAT(1500), FLON(1500), ALTI(1500)
      COMMON DUMMY,T,FLAT,FLON,ALTI
      COMMON XLATO,XLONGC,CRBINC,ALT1,ALT2,DV,NPOINT
      COMMON HEADG
      EQUIVALENCE (XLATO,AT1),(XLONGC,ONG1),(CRBINC,HDG1),(NPOINT,N)
      DIMENSION PHEB(70),PLEB(70),ETHB(70),ELEB(70),
     1PHEL(50),PLEL(50),ETHL(50),ELEL(50),
     2PHEFX(2000),PLEFX(2000),ETHFX(2000),ELEFX(2000),
     3ETHBC(50),ETHB1(50),ETHLC(50),ETHL1(50)
      COMMON PHEB,PHEL,PHEFX,
     1PLEB,PLEL,PLEFX,
     2ETHB,ETHL,ETHFX,ETHBC,ETHLC,ETHB1,ETHL1,
     3ELEB,ELEL,ELEFX
      DIMENSION FXMAP(4)
      DIMENSION BBMAX(4),BBMIN(4),FLMAX(4),FLMIN(4)
C
C     SUBROUTINE FLUXES COMPUTES PROTON AND ELECTRON FLUXES
C     AND THEIR RESPECTIVE TIME-INTEGRATED FLUXES
C     FROM THE FOLLOWING BELTS--
C         HIGH ENERGY PROTONS
C         LOW ENERGY PROTONS
C         AFSWC THREAT ELECTRONS
C         ELECTRONS
C
C     INITIALLY THIS SUBROUTINE WILL READ IN THE FLUX TABLE
C     DATA FOR THE ABOVE BELTS IN THAT ORDER
C        1  B VALUES
C        2  L VALUES
C        3  FLUX VALUES
C
C     FOR THE AFSWC THREAT ELECTRON BELT THE FOLLOWING
C     ADDITIONAL DATA ARE REQUIRED
C        4  B CUTOFF VALUES
C        5  L CUTOFF VALUES
C        6  B1 CUTOFF VALUES
C        7  L1 CUTOFF VALUES
C
C
C     INPUT CARD FORMATS
C
C     FIRST CARD SPECIFIES TYPE OF INPUT,I.E.,BINARY OR BCD
C     TO BE PUNCHED ON COLUMNS 1,11,21,31 FOR THE
C     4 BELTS RESPECTIVELY
C     E.G.,IF ALL BINARY,THEN THIS CARD HAS THE WORD
C     -BINARY- PUNCHED 4 TIMES,OTHERWISE LEAVE BLANK
C
C     FOLLOWING THIS CARD ARE THE FLUX TABLE DATA
C
C     IF CARDS ARE BCD THEN FORMAT IS (7E10.3)
```

```
C     BINARY CARDS HAVE 24 WORDS EACH.
C
C     BINARY CARDS ARE PUNCHED IF INPUT IS BCD
C
C     SUBROUTINE FXREAD READS FLUX DATA IN BINARY FORM
C     SUBROUTINE FXRITE WRITES FLUX DATA IN BINARY FORM
C
      IF (CELL-6HFINISH) 1000,2000,1000
C
C     STATEMENT 1000 GIVES THE NUMBER OF VALUES OF
C       BB,FL,AND FLUX FOR EACH OF THE FLUX MAPS
C     IF FLUX MAPS ARE CHANGED,CHANGE THESE VALUES ACCORDINGLY
C
 1000 NPHEB = 66
      NPHEL = 27
      NPHEFX = 1782
      NPLEB = 29
      NPLEL = 39
      NPLEFX=1131
      NETHB = 29
      NETHL = 41
      NETHFX = 1189
      NELEB = 24
      NELEL = 40
      NELEFX = 960
      NETHBC = 48
      NETHLC = 48
      NETHB1 = 41
      NETHL1 = 41
      CELL=6HFINISH
C
C     STATEMENTS 1010-1040 GIVE THE BOUNDARY VALUES OF BB AND FL
C     IF FLUX MAPS ARE CHANGED,CHANGE THE LIMITS ACCORDINGLY
C
C     LIMITS FOR HIGH ENERGY PROTON FLUX MAP
C
 1010 BBMAX(1)=.272
      BBMIN(1)=.012
      FLMAX(1)=2.70
      FLMIN(1)=1.12
C
C     LIMITS FOR LOW ENERGY PROTON FLUX MAP
C
 1020 BBMAX(2)=.280
      BBMIN(2)=.000
      FLMAX(2)=5.25
      FLMIN(2)=1.12
C
C     LIMITS FOR AFSWC THREAT ELECTRON FLUX MAP
C
 1030 BBMAX(3)=.560
      BBMIN(3)=.000
      FLMAX(3)=5.02
      FLMIN(3)=1.02
```

```
C
C     LIMITS FOR ELECTRON FLUX MAP
C
 1040 BBMAX(4)=.460
      BBMIN(4)=.000
      FLMAX(4)=5.75
      FLMIN(4)=1.15
 1090 READ INPUT TAPE 5,1091,(FXMAP(I),I=1,4)
 1091 FORMAT (4(A6,4X))
C
C     WRITE DUMMY BCD RECORD IF BINARY CARDS ARE
C     TO BE PUNCHED
C
 1092 DO 1097 I=1,4
 1093 IF (FXMAP(I)-6HBINARY) 1094,1097,1094
 1094 WRITE OUTPUT TAPE 14,1095
 1095 FORMAT (16HDUMMY BCD RECORD)
 1096 GO TO 1100
 1097 CONTINUE
C
C     READ HIGH ENERGY PROTON FLUX DATA
C
 1100 IF (FXMAP(1)-6HBINARY) 1150,1101,1150
 1101 CALL FXREAD (PHEB,NPHEB)
      CALL FXREAD (PHEL,NPHEL)
      CALL FXREAD (PHEFX,NPHEFX)
 1102 GO TO 1200
 1150 READ INPUT TAPE 5,1151,(PHEB(I),I=1,NPHEB)
      READ INPUT TAPE 5,1151,(PHEL(I),I=1,NPHEL)
      READ INPUT TAPE 5,1151,(PHEFX(I),I=1,NPHEFX)
 1151 FORMAT((7E10.3))
 1160 CALL FXRITE (PHEB,NPHEB)
      CALL FXRITE (PHEL,NPHEL)
      CALL FXRITE (PHEFX,NPHEFX)
C
C     READ LOW ENERGY PROTON FLUX DATA
C
 1200 IF (FXMAP(2)-6HBINARY) 1250,1201,1250
 1201 CALL FXREAD (PLEB,NPLEB)
      CALL FXREAD (PLEL,NPLEL)
      CALL FXREAD (PLEFX,NPLEFX)
 1202 GO TO 1300
 1250 READ INPUT TAPE 5,1251,(PLEB(I),I=1,NPLEB)
      READ INPUT TAPE 5,1251,(PLEL(I),I=1,NPLEL)
      READ INPUT TAPE 5,1251,(PLEFX(I),I=1,NPLEFX)
 1251 FORMAT((7E10.3))
 1260 CALL FXRITE (PLEB,NPLEB)
      CALL FXRITE (PLEL,NPLEL)
      CALL FXRITE (PLEFX,NPLEFX)
C
C     READ AFSWC THREAT ELECTRON FLUX DATA
C
 1300 IF (FXMAP(3)-6HBINARY) 1350,1301,1350
 1301 CALL FXREAD (ETHB,NETHB)
```

```
      CALL FXREAD (ETHL,NETHL)
      CALL FXREAD (ETHFX,NETHFX)
      CALL FXREAD (ETHBC,NETHBC)
      CALL FXREAD (ETHLC,NETHLC)
      CALL FXREAD (ETHB1,NETHB1)
      CALL FXREAD (ETHL1,NETHL1)
 1302 GO TO 1400
 1350 READ INPUT TAPE 5,1351,(ETHB(I),I=1,NETHB)
      READ INPUT TAPE 5,1351,(ETHL(I),I=1,NETHL)
      READ INPUT TAPE 5,1351,(ETHFX(I),I=1,NETHFX)
      READ INPUT TAPE 5,1351,(ETHBC(I),I=1,NETHBC)
      READ INPUT TAPE 5,1351,(ETHLC(I),I=1,NETHLC)
      READ INPUT TAPE 5,1351,(ETHB1(I),I=1,NETHB1)
      READ INPUT TAPE 5,1351,(ETHL1(I),I=1,NETHL1)
 1351 FORMAT((7E10.3))
 1360 CALL FXRITE (ETHB,NETHB)
      CALL FXRITE (ETHL,NETHL)
      CALL FXRITE (ETHFX,NETHFX)
      CALL FXRITE (ETHBC,NETHBC)
      CALL FXRITE (ETHLC,NETHLC)
      CALL FXRITE (ETHB1,NETHB1)
      CALL FXRITE (ETHL1,NETHL1)
C
C     READ ELECTRON FLUX DATA
C
 1400 IF (FXMAP(4)-6HBINARY) 1450,1401,1450
 1401 CALL FXREAD (ELEB,NELEB)
      CALL FXREAD (ELEL,NELEL)
      CALL FXREAD (ELEFX,NELEFX)
 1402 GO TO 1500
 1450 READ INPUT TAPE 5,1451,(ELEB(I),I=1,NELEB)
      READ INPUT TAPE 5,1451,(ELEL(I),I=1,NELEL)
      READ INPUT TAPE 5,1451,(ELEFX(I),I=1,NELEFX)
 1451 FORMAT((7E10.3))
 1460 CALL FXRITE (ELEB,NELEB)
      CALL FXRITE (ELEL,NELEL)
      CALL FXRITE (ELEFX,NELEFX)
 1500 CONTINUE
C
C     CONVERT FLUX TO LOGF(FLUX) FOR LOGARITHMIC INTERPOLATION
C
 1910 DO 1912 I=1,NPHEFX
      IF (PHEFX(I)) 1912,1912,1911
 1911 PHEFX(I)=LOGF(PHEFX(I))
 1912 CONTINUE
 1920 DO 1922 I=1,NPLEFX
      IF(PLEFX(I)) 1922,1922,1921
 1921 PLEFX(I)=LOGF(PLEFX(I))
 1922 CONTINUE
 1930 DO 1932 I=1,NETHFX
      IF (ETHFX(I)) 1932,1932,1931
 1931 ETHFX(I)=LOGF(ETHFX(I))
 1932 CONTINUE
 1940 DO 1942 I=1,NELEFX
```

```
      IF (ELEFX(I)) 1942,1942,1941
 1941 ELEFX(I)=LOGF(ELEFX(I))
 1942 CONTINUE
      GO TO 2500
C
C     CALCULATION OF HIGH ENERGY PROTON FLUXES
C
 2000 CONTINUE
 2100 IF (FL-FLMAX(1)) 2101,2101,2110
 2101 IF (FL-FLMIN(1)) 2110,2102,2102
 2102 IF (BB-BBMAX(1)) 2103,2103,2110
 2103 IF (BB-BBMIN(1)) 2110,2140,2140
 2110 FXPHE=0.
      DFXPHE=0.
      GO TO 2200
 2140 CALL TABLE (1,BB,FL,PHEB,PHEL,PHEFX,NPHEB,FXPHE)
 2141 IF (FXPHE) 2150,2150,2142
 2142 FXPHE=EXPF(FXPHE)
 2150 DFXPHE=FXPHE*DTIME
      SUMPHE=SUMPHE+DFXPHE
C
C     CALCULATION OF LOW ENERGY PROTON FLUXES
C
 2200 IF (FL-FLMAX(2)) 2201,2201,2210
 2201 IF (FL-FLMIN(2)) 2210,2202,2202
 2202 IF (BB-BBMAX(2)) 2203,2203,2210
 2203 IF (BB-BBMIN(2)) 2210,2240,2240
 2210 FXPLE=0.
      DFXPLE=0.
      GO TO 2300
 2240 CALL TABLE (2,BB,FL,PLEB,PLEL,PLEFX,NPLEB,FXPLE)
 2241 IF (FXPLE) 2250,2250,2242
 2242 FXPLE=EXPF(FXPLE)
 2250 DFXPLE=FXPLE*DTIME
      SUMPLE=SUMPLE+DFXPLE
C
C     CALCULATION OF AFSWC THREAT ELECTRON FLUXES
C
 2300 IF (FL-FLMAX(3)) 2301,2301,2310
 2301 IF (FL-FLMIN(3)) 2310,2302,2302
 2302 IF (BB-BBMAX(3)) 2303,2303,2310
 2303 IF (BB-BBMIN(3)) 2310,2320,2320
 2310 FXETL=0.
      DFXETL=0.
      FXETG=0.
      DFXETG=0.
      GO TO 2400
C
C     INTERPOLATE FOR B CUTOFF TO SET FLUX=0.
C
 2320 IF (BB-ETHBC(1)) 2340,2321,2321
 2321 IL=1
 2322 IF (FL-ETHLC(IL)) 2325,2324,2323
 2323 IL=IL+1
```

```
      GO TO 2322
 2324 BC=ETHBC(IL)
      GO TO 2326
 2325 BC=ETHBC(IL-1)+(ETHBC(IL)-ETHBC(IL-1))*(FL-ETHLC(IL-1))/
     1(ETHLC(IL)-ETHLC(IL-1))
 2326 IF (BB-BC) 2340,2340,2310
 2340 CALL TABLE (3,BB,FL,ETHB,ETHL,ETHFX,NETHB,FXETH)
 2351 IF (FXETH) 2341,2341,2352
 2352 FXETH=EXPF(FXETH)
C
C     SEPARATE FLUXES INTO 2 REGIONS DEPENDING ON VALUE OF B
C
 2341 IL1=1
 2342 IF (FL-ETHL1(IL1)) 2345,2343,2343
 2343 IL1=IL1+1
 2344 GO TO 2342
 2345 BC1=ETHB1(IL1-1)+(ETHB1(IL1)-ETHB1(IL1-1))*
     1(FL-ETHL1(IL1-1))/(ETHL1(IL1)-ETHL1(IL1-1))
 2346 IF (BB-BC1) 2347,2348,2348
C
C     B LESS THAN B1 CUTOFF
C
 2347 FXETL=FXETH
      DFXETL=FXETL*DTIME
      SUMETL=SUMETL+DFXETL
      FXETG=0.
      DFXETG=0.
      GO TO 2349
C
C     B GREATER THAN B1 CUTOFF
C
 2348 FXETG=FXETH
      DFXETG=FXETG*DTIME
      SUMETG=SUMETG+DFXETG
      FXETL=0.
      DFXETL=0.
 2349 CONTINUE
C
C     CALCULATION OF ELECTRON FLUXES
C
 2400 IF (FL-FLMAX(4)) 2401,2401,2410
 2401 IF (FL-FLMIN(4)) 2410,2402,2402
 2402 IF (BB-BBMAX(4)) 2403,2403,2410
 2403 IF (BB-BBMIN(4)) 2410,2440,2440
 2410 FXELL=0.
      DFXELL=0.
      FXELG=0.
      DFXELG=0.
      GO TO 2500
 2440 CALL TABLE (4,BB,FL,ELEB,ELEL,ELEFX,NELEB,FXELE)
 2441 IF (FXELE) 2450,2450,2442
 2442 FXELE=EXPF(FXELE)
C
C     SEPARATE FLUXES INTO 2 GROUPS DEPENDING ON WHETHER
```

```
C     THE VALUE OF L IS LESS THAN OR GREATER THAN 1.5
C
 2450 IF (FL-1.5) 2451,2451,2455
C
C     L IS LESS THAN 1.5
C
 2451 FXELL=FXELE
      DFXELL=FXELL*CTIME
      SUMELL=SUMELL+DFXELL
      FXELG=0.
      DFXELG=C.
      GO TO 2500
C
C     L IS GREATER THAN 1.5
C
 2455 FXELG=FXELE
      DFXELG=FXELG*CTIME
      SUMELG=SUMELG+DFXELG
      FXELL=0.
      DFXELL=C.
 2500 CONTINUE
      RETURN
      END(1,0,0,1,0,0,0,C,0,1,0,0,0,0,0)
```

```
      SUBROUTINE TABLE (IF,BB,FL,BBX,FLX,FX,NB,FLUX)
      DIMENSION IBB(4),ILL(4),BBX(2),FLX(2),FX(2)
C
C     DEFINITIONS OF SYMBOLS IN CALLING SEQUENCE
C
C       1     IF=NUMBER WHICH IDENTIFIES FLUX MAP
C                  1=HIGH ENERGY PROTONS
C                  2=LOW ENERGY PROTONS
C                  3=AFSWC THREAT ELECTRONS
C                  4=ELECTRONS
C       2     BB=B VALUE
C       3     FL=L VALUE
C       4    BBX=B ARRAY
C       5    FLX=L ARRAY
C       6     FX=FLUX ARRAY
C       7     NB=NUMBER OF VALUES OF B
C       8   FLUX=CALCULATED INTERPOLATED VALUE OF FLUX
C
C     B AND L ARRAYS MUST BE IN MONOTONELY INCREASING ORDER
C
C     INITIALIZE INDEX VALUES IBB AND ILL
C
  100 IF (CELL-6HFINISH) 101,200,101
  101 DO 103 I=1,4
  102 IBB(I)=0
      ILL(I)=0
  103 CONTINUE
  104 CELL=6HFINISH
C
C     SET INDEXES TO PREVIOUS VALUES
C
  200 IB=IBB(IF)
      IL=ILL(IF)
C
C     SEARCH B ARRAY TO SELECT INDEX IB
C
  300 IF (BB-BBX(IB+1)) 310,301,301
  301 IB=IB+1
      GO TO 300
  310 IF (BB-BBX(IB)) 311,400,400
  311 IB=IB-1
      GO TO 310
C
C     SEARCH L ARRAY TO SELECT INDEX IL
C
  400 IF (FL-FLX(IL+1)) 410,401,401
  401 IL=IL+1
      GO TO 400
  410 IF (FL-FLX(IL)) 411,500,500
  411 IL=IL-1
      GO TO 410
C
C     SAVE NEW INDEXES IB AND IL FOR NEXT TIME
```

```
C
  500 IBB(IF)=IB
      ILL(IF)=IL
C
C     COMPUTE INDEX VALUES FOR FLUXES
C
  600 K1=(IL-1)*NB+IB
      K2=K1+1
      K3=K1+NB
      K4=K3+1
C
C     CALCULATE INTERPOLATED VALUE OF FLUX
C
  700 B1=(BBX(IB+1)-BB)/(BBX(IB+1)-BBX(IB))
      B2=(BB-BBX(IB))/(BBX(IB+1)-BBX(IB))
      FL1=(FLX(IL+1)-FL)/(FLX(IL+1)-FLX(IL))
      FL2=(FL-FLX(IL))/(FLX(IL+1)-FLX(IL))
      FLUX=FL1*(B1*FX(K1)+B2*FX(K2))+FL2*(B1*FX(K3)+B2*FX(K4))
      RETURN
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE FXREAD (A,N)
      DIMENSION A(24)
C
C     SUBROUTINE FXREAD READS IN FLUX DATA
C     ON 24-WORD BINARY RECORDS
C
  100 K1=1
  101 DO 107 I=1,N,24
  102 K2=K1+23
  103 IF (K2-N) 105,105,104
  104 K2=N
  105 READ TAPE 5,(A(I),I=K1,K2)
  106 K1=K1+24
  107 CONTINUE
  108 RETURN
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0)
```

```
      SUBROUTINE FXRITE (A,N)

      SUBROUTINE FXRITE (A,N)
      DIMENSION A(24)
C
C     SUBROUTINE FXRITE PUNCHES FLUX DATA
C     ON 24-WORD BINARY RECORDS
C
  100 K1=1
  101 DO 107 I=1,N,24
  102 K2=K1+23
  105 WRITE TAPE 14,(A(I),I=K1,K2)
  106 K1=K1+24
  107 CONTINUE
  108 RETURN
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE INVAR(FLAT,FLONG,ALT,ERR,BB,FL)
      SUBROUTINE INVAR(FLAT,FLONG,ALT,ERR,BB,FL)
C         REVISED FEB.1963
C      NOTE,ERROR IN L IS TYPICALLY LESS THAN 10.*ERR*L (PERCENT)
C        FLAT=LATITUDE IN DEGREES  , FLONG=LONGITUDE IN DEGREES
C       ALT=ALTITUDE=DISTANCE FROM SURFACE OF EARTH IN KILOMETERS
      DIMENSION V(3,3),B(200),ARC(200),VN(3),VP(3),BEG(200),BEND(200),
     1BLOG(200),ECO(200),R1(3),R2(3),R3(3)
      V(1,2)=ALT/6371.2
      V(2,2)=(90.-FLAT)/57.2957795
      V(3,2)=FLONG/57.2957795
      ARC(1)=0.
      ARC(2)=(1.0+V(1,2))*SQRTF(ERR)*0.3
      DCLT=1.5708-0.2007*COSF(V(3,2)+1.239)
      IF(V(2,2)-DCLT)10,10,11
   11 ARC(2)=-ARC(2)
   10 CALL START (R1,R2,R3,B,ARC,ERR,V)
      DO 12 I=1,3
      VP(I)=V(I,2)
   12 VN(I)=V(I,3)
      CALL LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      IF(J-200)16,17,17
   17 FL=-1.0
      GO TO 18
   16 JUP=J
      DO 40 J=1,JUP
      ARC(J)=ABSF(ARC(J))
   40 BLOG(J)=LOGF(B(J))
      JEP=JUP-1
      DO 21 J=2,JEP
      ASUM=ARC(J)+ARC(J+1)
      DX=BLOG(J-1)-BLOG(J)
      DN=ASUM*ARC(J)*ARC(J+1)
      BCO    =((BLOG(J-1)-BLOG(J+1))*ARC(J)**2-DX*ASUM**2)/DN
      CCC    =(DX*ARC(J+1)-(BLOG(J)-BLOG(J+1))*ARC(J))/DN
      SA=.75*ARC(J)
      SC=SA+.25*ASUM
      DCO=BLOG(J-1)-CCO    *SA*SC
      ECO(J)=BCO    +CCC    *(SA+SC)
      BEG(J)=EXPF(DCO+ECO(J)*.5*ARC(J))
   21 BEND(J)=EXPF(DCO+ECO(J)*.5*(ASUM+ARC(J)))
      BEG(JUP)=BEND(JEP)
      BEND(JUP)=B(JUP)
      ECO(JUP)=(2.0/ARC(JUP))*LOGF(BEND(JUP)/BEG(JUP))
      CALL INTEG (ARC,BEG,BEND,B,JEP,ECO,FLINT)
   27 CALL CARMEL (B(2),FLINT,FL)
   18 BB=B(2)
      RETURN
      END(1,0,0,0,0,0,0,C,C,0,1,0,0,0,0,0)
```

INVAR000
INVARA00
INVARA01
INVARB01
INVARC01
INVAR001
INVAR002
INVAR003
INVAR004
INVAR005
INVAR006
INVAR007
INVAR008
INVAR009
INVAR010
INVAR011
INVAR012
INVAR013
INVAR014
INVAR015
INVARA15
INVARB15
INVARC15
INVAR016
INVAR017
INVAR018
INVAR019
INVAR020
INVAR021
INVAR022
INVAR023
INVAR024
INVAR025
INVAR026
INVAR027
INVAR028
INVAR029
INVAR030
INVAR031
INVAR032
INVAR033
INVAR034
INVAR035
INVAR036
INVAR037
INVAR038
INVAR039

```
      SUBROUTINE LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      SUBROUTINE LINES (R1,R2,R3,B,ARC,ERR,J,VP,VN)
      DIMENSION B(200),ARC(200),R1(3),R2(3),R3(3),VN(3),VP(3),RA(3)
      CRE=0.25
      IF(ERR-0.15625)74,75,75
   74 CRE=    (ERR**0.333333333)
   75 A3=ARC(3)
      AAB=ABSF(A3)
      SNA=A3/AAB
      A1=ARC(1)
      A2=ARC(2)
      AC6=A3*A3/6.0
      J=3
      ILP=1
      IS=1
      GO TO 87
   66 IS=1
      J=J+1
      AC6=A3*A3/6.0
      ARCJ=A1+A2+A3
      AD=(ASUM+A1)/AA
      BD=ASUM/BB
      CD=A1/CC
   36 DO 5 I=1,3
      DD=R1(I)/AA-R2(I)/BB+R3(I)/CC
      GO TO(6,8),IS
    6 RT=R1(I)-(AD*R1(I)-BD*R2(I)+CD*R3(I)-DD*ARCJ)*ARCJ
      RA(I)=R1(I)
      R1(I)=R2(I)
      R2(I)=R3(I)
      R3(I)=RT
      VP(I)=VN(I)
    8 RBAR=(R2(I)+R3(I))/2.-DD*AC6
    5 VN(I)=VP(I)+A3*RBAR
   87 IF(VN(2))76,77,77
   76 VN(2)=-VN(2)
   77 IF(VN(2)-3.141592653)78,78,79
   79 VN(2)=6.283185307-VN(2)
      GO TO 77
   78 IF(VN(3))80,81,81
   80 VN(3)=VN(3)+6.283185307
      GO TO 78
   81  IF(VN(3)-6.283185307)82,82,83
   83 VN(3)=VN(3)-6.283185307
      GO TO 81
   82 GO TO (9,10),IS
    9 SIT=ABSF(SINF(VN(2)))
      PRE1=VN(1)
      PRE2=PRE1*VN(2)
      PRE3=PRE1*SIT*VN(3)
      SSQ=SIT*SIT
      CER=(6356.912+SSQ*(21.3677+.108*SSQ))/6371.2
      AER=VN(1)-CER
      CALL MAGNET(AER,SIT,VN(3),BR,BT,BP,B(J),VN(2))
      R3(1)=BR/B(J)
```

```
      DN=B(J)*VN(1)                                                     LINES053
      R3(2)=BT/DN                                                       LINES054
      R3(3)=BP/(DN*SIT)                                                 LINES055
      ASUM=A3+A2                                                        LINES056
      AA=ASUM*A2                                                        LINES057
      BB=A3*A2                                                          LINES058
      CC=ASUM*A3                                                        LINES059
      IS=2                                                              LINES060
      GO TO 36                                                          LINES061
   10 SIT=ABSF(SINF(VN(2)))                                             LINES062
      B(J)=B(J)*((PRE1/VN(1))**3)                                       LINES063
   59 QRT=.5*ABSF(R3(1))/(.1+ABSF(R3(2)*VN(1)))                         LINES064
      X=(ABSF(VN(1)-PRE1)+QRT*ABSF(VN(1)*VN(2)-PRE2)+ABSF(VN(1)*SIT*VN(3LINES065
     1)-PRE3))/(AAB*ERR*SQRTF(1.+QRT*QRT))                              LINES066
      GO TO (90,93,90),ILP                                              LINES067
   93 IF(X-3.3)90,89,89                                                 LINES068
   89 A3=A3*0.2*(8.0+X)/(0.8+X)                                         LINES069
      J=J-1                                                             LINES070
      ILP=3                                                             LINES071
      ASUM=A2+A1                                                        LINES072
      AA=ASUM*A1                                                        LINES073
      BB=A2*A1                                                          LINES074
      CC=ASUM*A2                                                        LINES075
      DO 91 I=1,3                                                       LINES076
      VN(I)=VP(I)                                                       LINES077
      R3(I)=R2(I)                                                       LINES078
      R2(I)=R1(I)                                                       LINES079
   91 R1(I)=RA(I)                                                       LINES080
      GO TO 73                                                          LINES081
   90 IF(J-2CC)67,60,60                                                 LINES082
   67 A1=A2                                                             LINES083
      IF(B(J)-B(2))49,49,60                                             LINES084
   49 ILP=2                                                             LINES085
      A2=A3                                                             LINES086
      A3=A3*.2*(8.+X)/(.8+X)                                            LINES087
      AM=(2.-R3(2)*VN(1))*VN(1)*CRE                                     LINES088
      IF(ABSF(A3)-AM)84,84,72                                           LINES089
   72 A3=SNA*AM                                                         LINES090
   84 IF(SNA*R3(1)+.5)85,85,73                                          LINES091
   85 AM=-.5*SNA*VN(1)/R3(1)                                            LINES092
      IF(ABSF(A3)-AM)73,73,86                                           LINES093
   86 A3=SNA*AM                                                         LINES094
   73 ARC(J+1)=A3                                                       LINES095
      AAB=ABSF(A3)                                                      LINES096
      GO TO 66                                                          LINES097
   60 RETURN                                                            LINES098
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE START (R1,R2,R3,B,ARC,ERR,V)                           START000
      DIMENSION B(200),ARC(200),V(3,3),R1(3),R2(3),R3(3)                START001
      SIT=ABSF(SINF(V(2,2)))                                            START002
      AER=V(1,2)                                                        START003
      SSQ=SIT*SIT                                                       START004
      OER=(6356.912+SSQ*(21.3677+.1C8*SSQ))/6371.2                      START005
      V(1,2)=AER+OER                                                    START006
   10 IF(V(3,2))11,12,12                                                STARTA06
   11 V(3,2)=V(3,2)+6.283185307                                         STARTB06
      GO TO 1C                                                          STARTC06
   12 CALL MAGNET(AER,SIT,V(3,2),BR,BT,BP,B(2),V(2,2))                  START007
      R2(1)=BR/B(2)                                                     START008
      CN=B(2)*V(1,2)                                                    START009
      R2(2)=BT/CN                                                       START010
      R2(3)=BP/(CN*SIT)                                                 START011
      IS=0                                                              START012
    1  CO 2 I=1,3                                                       START013
    2 V(I,1)=V(I,2)-ARC(2)*R2(I)                                        START014
      SIT=ABSF(SINF(V(2,1)))                                            START015
    3  SSQ=SIT*SIT                                                      START016
      OER=(6356.912+SSQ*(21.3677+.1C8*SSQ))/6371.2                      START017
      AER=V(1,1)-OER                                                    START018
      CALL MAGNET(AER,SIT,V(3,1),BR,BT,BP,B(1),V(2,1))                  START019
      IF(B(1)-B(2))4,5,5                                                START020
    4  ARC(2)=-ARC(2)                                                   START021
      GO TO 1                                                           START022
    5 R1(1)=BR/B(1)                                                     START023
      ARC(3)=ARC(2)                                                     START024
      CN=B(1)*V(1,1)                                                    START025
      R1(2)=BT/CN                                                       START026
      R1(3)=BP/(CN*SIT)                                                 START027
      DO 6 I=1,3                                                        START028
    6 V(I,1)=V(I,2)-ARC(2)*(R1(I)+R2(I))/2.                             START029
      SIT=ABSF(SINF(V(2,1)))                                            START030
      IS=IS+1                                                           START031
      GO TO (3,7),IS                                                    START032
    7  CO 8 I=1,3                                                       START033
    8 V(I,3)=V(I,2)+ARC(3)*((1.5)*R2(I)-.5*R1(I))                       START034
      RETURN                                                            START035
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE MAGNET (R,S,PHI,BR,BTHET,BPHI,BB,THET)                 MAGNT000
      DIMENSION DP(49),P(49),G(49),H(49),CONST(49),AOR(7),SP(7),CP(7)   MAGNT001
      IF(KIP+1234)150,151,150                                           MAGNT002
  150 KIP=-1234                                                         MAGNT003
      DO 152 N=1,49                                                     MAGNT004
      G(N)=0.0                                                          MAGNT005
  152 H(N)=0.0                                                          MAGNT006
C     JENSEN AND CAIN COEFFICIENTS FOR 1960 (JUNE 1962)                 MAGNT007
C     G(I) = G(N,M) AND H(I) = H(N,M) WHERE I = N+7*(M-1)               MAGNT008
      G( 2)= 3.04112050E-01                                             MAGNT009
      G( 9)= 2.14736858E-02                                             MAGNT010
      G( 3)= 2.40353671E-02                                             MAGNT011
      G(10)=-5.12533379E-02                                             MAGNT012
      G(17)=-1.33811969E-02                                             MAGNT013
      G( 4)=-3.15178651E-02                                             MAGNT014
      G(11)= 6.21300906E-02                                             MAGNT015
      G(18)=-2.48981333E-02                                             MAGNT016
      G(25)=-6.49565905E-03                                             MAGNT017
      G( 5)=-4.17943639E-02                                             MAGNT018
      G(12)=-4.52983660E-02                                             MAGNT019
      G(19)=-2.17947447E-02                                             MAGNT020
      G(26)= 7.00825405E-03                                             MAGNT021
      G(33)=-2.04395562E-03                                             MAGNT022
      G( 6)= 1.62556271E-02                                             MAGNT023
      G(13)=-3.44067606E-02                                             MAGNT024
      G(20)=-1.94470026E-02                                             MAGNT025
      G(27)=-6.08211374E-04                                             MAGNT026
      G(34)= 2.77533549E-03                                             MAGNT027
      G(41)= 6.96802467E-04                                             MAGNT028
      G( 7)=-1.95231736E-02                                             MAGNT029
      G(14)=-4.85326147E-03                                             MAGNT030
      G(21)= 3.21172428E-03                                             MAGNT031
      G(28)= 2.14128828E-02                                             MAGNT032
      G(35)= 1.05051275E-03                                             MAGNT033
      G(42)= 2.26829448E-04                                             MAGNT034
      G(49)= 1.11471358E-03                                             MAGNT035
      H( 9)=-5.79890501E-02                                             MAGNT036
      H(10)= 3.31240714E-02                                             MAGNT037
      H(17)=-1.57893822E-03                                             MAGNT038
      H(11)= 1.48696943E-02                                             MAGNT039
      H(18)=-4.07490158E-03                                             MAGNT040
      H(25)= 2.10318235E-04                                             MAGNT041
      H(12)=-1.18245456E-02                                             MAGNT042
      H(19)= 1.00057732E-02                                             MAGNT043
      H(26)= 4.30380863E-04                                             MAGNT044
      H(33)= 1.38503490E-03                                             MAGNT045
      H(13)=-7.95897466E-04                                             MAGNT046
      H(20)=-2.00044021E-03                                             MAGNT047
      H(27)= 4.59718859E-03                                             MAGNT048
      H(34)= 2.42063078E-03                                             MAGNT049
      H(41)=-1.21806522E-03                                             MAGNT050
      H(14)=-5.75830293E-03                                             MAGNT051
      H(21)=-8.73461401E-03                                             MAGNT052
      H(28)=-3.40604073E-03                                             MAGNT053
```

```
      H(35)=-1.18162456E-04                                          MAGNT054
      H(42)=-1.11623013E-03                                          MAGNT055
      H(49)=-3.24831891E-04                                          MAGNT056
      P(1)=1.0                                                       MAGNT057
      DP(1)=0.0                                                      MAGNT058
      SP(1)=0.0                                                      MAGNT059
      CP(1)=1.0                                                      MAGNT060
      CONST(9)=0.0                                                   MAGNT061
      CONST(16)=0.0                                                  MAGNT062
      DO 80 N=3,7                                                    MAGNT063
      FN=N                                                           MAGNT064
      DO 80 M=1,N                                                    MAGNT065
      FM=M                                                           MAGNT066
      I=N+7*(M-1)                                                    MAGNT067
   80 CONST(I)=((FN-2.0)**2-(FM-1.0)**2)/((FN+FN-3.0)*(FN+FN-5.0))   MAGNT068
  151 C=SQRTF(ABSF(1.0-S*S))                                         MAGNT069
      IF(THET-1.570796327) 154,154,156                               MAGNT070
  156 C=-C                                                           MAGNT071
  154 AR=1./(1.+R)                                                   MAGNT072
  155 SP(2)=SINF(PHI)                                                MAGNT073
      CP(2)=COSF(PHI)                                                MAGNT074
      AOR(1)=AR*AR                                                   MAGNT075
      AOR(2)=AR*AOR(1)                                               MAGNT076
      DO 90 M=3,7                                                    MAGNT077
      N=M-1                                                          MAGNT078
      SP(M)=SP(2)*CP(N)+CP(2)*SP(N)                                  MAGNT079
      CP(M)=CP(2)*CP(N)-SP(2)*SP(N)                                  MAGNT080
   90 AOR(M)=AR*AOR(N)                                               MAGNT081
      BR=0.0                                                         MAGNT082
      BTHET=0.0                                                      MAGNT083
      BPHI=0.0                                                       MAGNT084
      DO 32 N=2,7                                                    MAGNT085
      FN=N                                                           MAGNT086
      SUMR=0.0                                                       MAGNT087
      SUMT=0.0                                                       MAGNT088
      SUMP=0.0                                                       MAGNT089
      DO 33 M=1,N                                                    MAGNT090
      IF(N-M)87,88,87                                                MAGNT091
   88 I=8*N-7                                                        MAGNT092
      L=I-8                                                          MAGNT093
      P(I)=S*P(L)                                                    MAGNT094
      DP(I)=S*DP(L)+C*P(L)                                           MAGNT095
      GO TO 89                                                       MAGNT096
   87 I=N+7*(M-1)                                                    MAGNT097
      J=I-1                                                          MAGNT098
      K=I-2                                                          MAGNT099
      P(I)=C*P(J)-CONST(I)*P(K)                                      MAGNT100
      DP(I)=C*DP(J)-S*P(J)-CONST(I)*DP(K)                            MAGNT101
   89 FM=M-1                                                         MAGNT102
      TS=G(I)*CP(M)+H(I)*SP(M)                                       MAGNT103
      SUMR=SUMR+P(I)*TS                                              MAGNT104
      SUMT=SUMT+DP(I)*TS                                             MAGNT105
   33 SUMP=SUMP+FM*P(I)*(-G(I)*SP(M)+H(I)*CP(M))                     MAGNT106
      BR=BR+AOR(N)*FN*SUMR                                           MAGNT107
```

```
      BTHET=BTHET-AOR(N)*SUMT                                   MAGNT108
   32 BPHI=BPHI-AOR(N)*SUMP                                     MAGNT109
      BPHI=BPHI/S                                               MAGNT110
      BB=SQRTF(BR**2+BTHET**2+BPHI**2)                          MAGNT111
      RETURN                                                    MAGNT112
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE INTEG (ARC,BEG,BEND,B,JEP,ECO,FI)                      INTEG001
      DIMENSION ARC(200),BEG(200),BEND(200),B(200),ECO(200)             INTEG002
    4 KK=JEP                                                            INTEG003
    6 IF(KK-4)14,11,20                                                  INTEG004
   11 KK=KK-1                                                           INTEG005
   14 A=B(KK-1)/B(2)                                                    INTEG006
      X2=B(KK)/B(2)                                                     INTEG007
      X3=B(KK+1)/B(2)                                                   INTEG008
      ASUM=ARC(KK)+ARC(KK+1)                                            INTEG009
      DN=ARC(KK)*ARC(KK+1)*ASUM                                         INTEG010
      BB=(-A*ARC(KK+1)*(ARC(KK)+ASUM)+X2*ASUM**2-X3*ARC(KK)**2)/DN      INTEG011
      C=(A*ARC(KK+1)-X2*ASUM+X3*ARC(KK))/DN                             INTEG012
      FI=1.570796326*(1.-A+BB*BB/(4.*C))/SQRTF(ABSF(C))                 INTEG013
      RETURN                                                            INTEG014
   20 T=SQRTF(1.-BEND(2)/B(2))                                          INTEG015
      FI=(2.*T-LOGF((1.+T)/(1.-T)))/ECO(2)                              INTEG016
      IF(B(2)-BEND(KK))21,21,25                                         INTEG017
   25 KK=KK+1                                                           INTEG018
   21 T=SQRTF(ABSF(1.0-BEG(KK)/B(2)))                                   INTEG019
      FI=FI-(2.*T-LOGF((1.+T)/(1.-T)))/ECO(KK)                          INTEG020
      KK=KK-1                                                           INTEG021
   22 DO 5 I=3,KK                                                       INTEG022
      ARG1=1.-BEND(I)/B(2)                                              INTEG023
      IF(ARG1)26,26,27                                                  INTEG024
   26 TE=1.E-5                                                          INTEG025
      GO TO 28                                                          INTEG026
   27 TE=SQRTF(ARG1)                                                    INTEG027
   28 ARG1=1.-BEG(I)/B(2)                                               INTEG028
      IF(ARG1)29,29,31                                                  INTEG029
   31 TB=SQRTF(ARG1)                                                    INTEG030
      GO TO 32                                                          INTEG031
   29 TB=1.E-5                                                          INTEG032
   32 IF(ABSF(ECO(I))-2.E-5) 23,23,24                                   INTEG033
   23 FI=FI+((TE+TB)*(ARC(I)+ARC(I+1)))/4.                              INTEG034
      GO TO 5                                                           INTEG035
   24 FI=FI+(2.*(TE-TB)-LOGF((1.+TE)*(1.-TB)/((1.-TE)*(1.+TB))))/ECO(I) INTEG036
    5 CONTINUE                                                          INTEG037
   30 RETURN                                                            INTEG038
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE CARMEL (B,XI,VL)                                       CARMLOOO
C     COMPUTE L                                                         CARMLOO1
      IF(XI-1.0E-36)14,14,15                                            CARMLOO2
   14 VL=(0.311653/B)**(1./3.)                                          CARMLOO2
      RETURN                                                            CARMLOO2
   15 XX=3.0*LOGF(XI)                                                   CARMLOO2
      XX=XX+LOGF(B/0.311653)                                            CARMLOE2
      IF(XX+22.)1,1,8                                                   CARMLOO3
    8 IF(XX+3.)2,2,9                                                    CARMLOO4
    9 IF(XX-3.)3,3,10                                                   CARMLOO5
   10 IF(XX-11.7)4,4,11                                                 CARMLOO6
   11 IF(XX-23.)5,5,6                                                   CARMLOO7
    1 GG=.333338*XX+.30062102                                           CARMLOO8
      GO TO 7                                                           CARMLOO9
    2 GG=((((((((-8.1537735E-14*XX+8.3232531E-13)*XX+1.0066362E-9)*XX+  CARMLO10
     18.1048663E-8)*XX+3.2916354E-6)*XX+8.2711096E-5)*XX+1.3714667E-3)* CARMLO11
     2XX+.01501724 5)*XX+.43432642)*XX+.62337691                        CARMLO12
      GO TO 7                                                           CARMLO13
    3 GG=((((((((2.6047023E-10*XX+2.3028767E-9)*XX-2.1997983E-8)*XX-    CARMLO14
     15.3977642E-7)*XX-3.3408822E-6)*XX+3.8379917E-5)*XX+1.1784234E-3)* CARMLO15
     2XX+1.4492441E-2)*XX+.43352788)*XX+.6228644                        CARMLO16
      GO TO 7                                                           CARMLO17
    4 GG=((((((((6.3271665E-10*XX-3.958306E-8)*XX+9.9766148E-07)*XX-    CARMLO18
     11.2531932E-5)*XX+7.9451313E-5)*XX-3.2077032E-4)*XX+2.1680398E-3)* CARMLO19
     2XX+1.2817956E-2)*XX+.43510529)*XX+.6222355                        CARMLO20
      GO TO 7                                                           CARMLO21
    5 GG=(((((2.8212095E-8*XX-3.8049276E-6)*XX+2.170224E-4)*XX-6.7310339CARMLO22
     1E-3)*XX+.12038224)*XX-.18461796)*XX+2.0007187                     CARMLO23
      GO TO 7                                                           CARMLO24
    6 GG=XX-3.0460681                                                   CARMLO25
    7 VL=(((1.0+EXP F(GG))*0.311653)/B)**(1./3.)                        CARMLO26
C     END COMPUTE L                                                     CARMLO27
      RETURN                                                            CARMLO28
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE SATRAC
C
C     SUBROUTINE SATRAC COMPUTES TRANSFER ORBIT
C     AND ADDITIONAL ELLIPTICAL ORBITS
C
C     REFERENCE-BOEING COORDINATION SHEET NO. 2-5342-1-107
C     --SUBROUTINE TO CALCULATE LOCATION IN TRANSFER
C     TRAJECTORY--BY STURE FOLKESON OF SPACE MECHANICS
C     GROUP, DATED 18 APRIL 1963
C
C     CHANGES WERE MADE IN STATEMENT NUMBERS 90,100,130,140,590,600
C     THE CONSTANTS EARTHR, GRAVCO, AND OMEGA ARE ASSEMBLED
C     THE INITIAL VELOCITY V1 IS COMPUTED
C     ONLY 1 INPUT CARD IS REQUIRED (7 INPUT DATA)
C     FORMAT IS (6E10.0,I10)
C       1   AT1 = INITIAL LATITUDE (DEG)
C       2  ONG1 = INITIAL LONGITUDE (DEG)
C       3  HDG1 = ORBIT INCLINATION (DEG)
C       4  ALT1 = INITIAL ALTITUDE (KM)
C       5  ALT2 = FINAL ALTITUDE (KM)
C       6    DV = INTERVAL BETWEEN POINTS IN THE EPHEMERIS (DEG)
C       7     N = TOTAL NUMBER OF EPHEMERIS POINTS
C
C
C     STAEMENTS 710,730,820 CHANGED TO INCLUDE INITIAL POINT
C
      DIMENSION DUMMY(100), T(1500), FLAT(1500), FLON(1500), ALTI(1500)
      COMMON DUMMY, T, FLAT, FLON, ALTI
      COMMON AT1,ONG1,HDG1,ALT1,ALT2,DV,N
      COMMON HEADG
      EQUIVALENCE (XLATO,AT1),(XLONGO,ONG1),(ORBINC,HDG1),(NPOINT,N)
    1 DIMENSION RV(1000),KTYPE(100),SNW(2),CSW(2),TRA(2,4),JD(2)
    2 TRA(1,1) = 6H CIRCL
    3 TRA(2,1) = 1HE
    4 TRA(1,2) = 6HN ELLI
    5 TRA(2,2) = 3HPSE
    6 TRA(1,3) = 6H PARAB
    7 TRA(2,3) = 3HOLA
    8 TRA(1,4) = 6H HYPER
    9 TRA(2,4) = 4HBOLA
   10 BETA = 2HHS
   11 ETOL = .00015
   20 HOUR = 3600.
   30 ONE = 1.0
D  40 PI = 3.141592653589793
   50 SEAMIL = 3280.83989
   60 TWO = 2.0
D  70 TPI = TWO*PI
D  80 DEG = 180./PI
C  81 READ IN RED CARD  ADJUSTABLE CONSTANTS
   90 CONTINUE
      EARTHR=2.0925738E7
      GRAVCO=1.407654E16
      OMEGA=0.729211508E-4
```

```
  100 CONTINUE
C 121 INPUT INITIAL CONDITIONS - YELLOW CARD
  130 READ INPUT TAPE 5,140,AT1,ONG1,HDG1,ALT1,ALT2,DV,N
  140 FORMAT (6F10.0,I10)
      HEADG = HDG1
      L=2
      ALT1FT=ALT1*SEAMIL
      ALT2FT=ALT2*SEAMIL
      V1=SQRTF((2.0*GRAVCO*(EARTHR+ALT2FT))/((EARTHR+ALT1FT)
     1*(2.0*EARTHR+ALT1FT+ALT2FT)))
  142 XI = HDG1/DEG
  143 PSI = AT1/DEG
D 144 SNB = COSF(XI)/COSF(PSI)
  145 IND = TWO*XI/PI+ONE
  146 IF (ABSF(SNB)-1.0) 156, 156, 150
  150 WRITE OUTPUT TAPE 6,151
  151 FORMAT(67HORBIT INCLINATION LESS THAN INITIAL LATITUDE. ORBIT DOES
 1511 NOT EXIST.)
  156 GO TO (157,159,162,162), IND
  157 HDG1 = ASINF(SNB)*DEG
  158 GO TO 190
  159 HDG1 = (TPI+ASINF(SNB))*DEG
  161 GO TO 190
  162 HDG1 = (PI-ASINF(SNB))*DEG
  190 R1=EARTHR+ALT1*SEAMIL
D 200 ALFA = TWO/R1-V1**2/GRAVCO
D 210 GMMA1 = FTP1/DEG
D 220 CSG1 = COSF(GMMA1)
D 230 RALFA = R1*ALFA
D 240 ECC2 = ONE-(TWO*RALFA-RALFA**2)*CSG1**2
D 250 ECC = SQRTF(ECC2)
  260 IF (ECC-ETOL) 270, 270, 360
C 261 CIRCULAR ORBIT
  270 J = 1
  280 TRANOM = 6HINDETM
  290 A = ONE/ALFA
  300 TP = A**3/GRAVCO
  310 PERSEC = TPI*SQRTF(TP)
  320 PERIOD = PERSEC/HOUR
  330 GO TO (331,450), J
  331 CONTINUE
  350 GO TO 580
D 360 PARA = (TWO*R1-R1*RALFA)*CSG1**2
D 370 CSW1 = (PARA/R1-ONE)/ECC
  380 W1 = ACOSF(CSW1)
  390 IF (GMMA1) 400, 410, 410
  400 W1 = TPI-W1
  410 TRANOM = W1*DEG
D 411 IF (ABSF(ALFA)-1.E-12) 470, 420, 420
  420 IF (ALFA) 530, 470, 430
C 421 ELLIPTICAL ORBIT
  430 J = 2
  440 GO TO 290
  450 CONTINUE
```

```
  460 GC TO 580
C 461 PARABOLIC TRAJECTORY
  470 J = 3
  480 A = 6HINFNTY
  481 C = PARA/TWO
  482 TP = TWO*C**3/GRAVCO
  483 PERSEC = TPI*SQRTF(TP)
  490 PERIOC = A
  520 GC TC 580
C 521 HYPERBOLIC TRAJECTCRY
  530 J = 4
  540 A = ONE/ALFA
  541 TP = -A**3/GRAVCC
  542 PERSEC = TPI*SQRTF(TP)
  550 PERIOC = 6HINFNTY
  580 WRITE OUTPUT TAPE 6, 581, TRA(1,J), TRA(2,J)
  581 FCRMAT(1H ,10X,19HTHE TRAJECTCRY IS A,2A6//)
C 582 NUMBER OF OUTPUT PCINTS - GREEN CARD
  590 CCNTINUE
  600 CCNTINUE
  610 GC TC (620,670,760), L
C 611 EACH RANGE ANGLE SPECIFIED - WHITE CARDS.L=1
  620 DC 65C I=1, N
  630 READ INPUT TAPE 5, 640, RV(I), KNEXT
  640 FCRMAT(F10.4,69X,I1)
  650 CONTINUE
  660 GC TO 790
C 661 RUN N NUMBER OF POINTS DV DEGREES APART.L=2
  670 IF (DV) 680, 680, 710
  680 WRITE OUTPUT TAPE 6, 690
  690 FORMAT(1H ,63HTHIS IS RIDICULOUS - THE TRUE ANOMALY IS CONSTANT OR
  6901 DECREASING)
  700 CALL EXIT
  710 CONTINUE
      T(1) = C.
      FLAT(1) = AT1
      FLCN(1) = CNG1
      ALTI(1) = ALT1
      RV(2) = DV
  730 DC 740 I=2,N
  740 RV(I+1) = RV(I)+DV
  75C GC TO 790
C 751 FIND LOCATION OF SPECIFIED ALTITUDE OR FLIGHT PATH ANGLE.L=3
  760 DC 78C I=1, N
  770 READ INPUT TAPE 5, 64C, RV(I), KTYPE(I)
  78C CONTINUE
  790 PSI1 = AT1/DEG
  800 AMCA1 = ONG1/DEG
  810 BETA1 = HDG1/DEG
C 811 START PROCESSING LCCP
  82C M=N+1
      DC 1320 I=2,M
  830 GC TC (84C,84C,124C), L
  840 DELTAV = RV(I)/DEG
```

```
  850 GC TC (860,960,960,960), J
C 851 CIRCULAR CASE.J=1
  860 VEL = V1
  870 ALT = ALT1
  880 FTP = 0.
  890 DELTAT = PERSEC*DELTAV/TPI
  891 NREV = DELTAV/TPI
  892 REV = NREV
  893 DELTAV = DELTAV-REV*TPI
  900 CALL LOC(DELTAV,DELTAT,PSI1,PSI2,AMDA1,AMDA2,BETA1,BETA2,OMEGA)
C 901 COMPUTE SATELLITE LOCATION IN LATITUDE, LONGITUDE AND HEADING
  910 AT = PSI2*DEG
  920 CNG = AMDA2*DEG
  930 HDG = BETA2*DEG
  940 ETIME = DELTAT/HOUR
  950 GC TO 1240
  960 W2 = W1+DELTAV
D 970 NREV = W2/TPI
  980 REV = NREV
D 990 W2 = W2-REV*TPI
D1000 R2 = PARA/(ONE+ECC*COSF(W2))
 1010 ALT = (R2-EARTHR)/SEAMIL
D1020 VEL2 = (TWO/R2-ALFA)*GRAVCO
D1030 VEL = SQRTF(VEL2)
D1040 CSG22 = PARA/(TWO*R2-ALFA*R2**2)
D1050 CSG2 = SQRTF(CSG22)
 1060 GMMA2 = ACOSF(CSG2)
 1070 IF (W2-PI) 1090, 1090, 1080
 1080 GMMA2 = -GMMA2
D1090 NREV = DELTAV/TPI
 1100 REV = NREV
D1110 DELTAV = DELTAV-REV*TPI
D1120 SNW1 = SINF(W1)
D1130 SNW2 = SINF(W2)
D1140 CSW2 = COSF(W2)
 1150 SNW(1) = SNW1
 1160 SNW(2) = SNW2
 1161 CSW(1) = CSW1
 1162 CSW(2) = CSW2
 1163 JC(1) = 0
 1170 JC(2) = 0
 1171 IF (J-3) 1172, 1180, 1180
 1172 IF (R1-A) 1173, 1176, 1175
 1173 JC(1) = 1
 1174 GC TC 1176
 1175 JC(1) = 2
 1176 IF (R2-A) 1177, 1180, 1179
 1177 JC(2) = 1
 1178 GC TC 1180
 1179 JC(2) = 2
 1180 CALL TEMPUS(J,JD,SNW,CSW,ECC,DM)
 1190 DELTAT = (DM/TPI+REV)*PERSEC
 1230 CALL LOC(DELTAV,DELTAT,PSI1,PSI2,AMDA1,AMDA2,BETA1,BETA2,OMEGA)
 1240 T(1)=DELTAT
```

```
      SUBROUTINE SATRAC

1250 FLAT(I) = PSI2*DEG
1260 FLON(I) = AMDA2*DEG
1270 ALTI(I) = ALT
1320 CONTINUE
1330 IF (KNEXT-1) 1340, 590, 130
1340 RETURN
     END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE SATRAC

 1250 FLAT(I) = PSI2*DEG
 1260 FLON(I) = AMDA2*DEG
 1270 ALTI(I) = ALT
 1320 CONTINUE
 1330 IF (KNEXT-1) 1340, 590, 130
 1340 RETURN
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE TEMPUS(J,JD,SNW,CSW,ECC,DM)
    1 DIMENSICN E(2), F(2), Z(2) ,SF(2), JD(2), SNW(2), CSW(2), SNE(2)
   10 PI = 3.14159265
   20 HPI = PI/2.0
   30 TPI = 2.0*PI
   40 CNE = 1.0
   50 IF (J-3) 60, 240, 360
C  51 ELLIPTIC CASE J=2
D  60 TE2 = CNE-ECC**2
D  70 TE = SCRTF(TE2)
   80 CO 210 I=1,2
   81 N = JC(I)
   90 SINE = TE*SNW(I)/(CNE+ECC*CSW(I))
  100 TEST = ABSF(SINE)+1.0E-08
  110 IF (TEST-ONE) 120,111,111
  111 IF (SNW(I)) 193, 191, 191
  120 EA = ASINF(SINE)
      I = I
      WRITE OUTPUT TAPE 6, 1000, I, N, SNDW, SINE, EA
 1000 FCRMAT(1H0,10X,2I3,3F15.9)
  130 IF (SNW(I)) 140, 150, 190
  140 GC TC(150,170), N
  150 EA = TPI+EA
  160 GC TO 2C0
  170 EA = PI-EA
  180 GC TC 2C0
  190 GO TO(200,170), N
  191 EA = HPI
  192 GC TO 200
  193 EA = PI+HPI
  194 GO TO 2C0
  200 E(I) = EA
  210 SNE(I) = SINE
  211 IF (E(2)-E(1)) 212, 214, 214
  212 PPC = 1.0
  213 GC TO 220
  214 PPC = 0.0
  220 DM = E(2)-E(1)-ECC*(SNE(2)-SNE(1))+PPC*TPI
      WRITE OUTPUT TAPE 6, 1001, PPC
 1001 FCRMAT(1H ,70X,F7.1)
  230 RETURN
C 231 PARABCLIC CASE J=3
  240 CC 33C I=1, 2
  250 TW2 = (CNE-CSW(I))/(ONE+CSW(I))
D 260 TW = SQRTF(TW2)
  270 Z(I) = TW
  280 IF (SNW(I)) 290, 310,330
  290 Z(I) = -Z(I)
  300 GC TO 330
  310 Z(I) = C.C
  320 GC TO 330
  330 CCNTINUE
  340 DM = Z(2)-Z(1)+(Z(2)**3-Z(1)**3)/3.0
  350 RETURN
```

```
C 351 HYPERBOLIC CASE J=4
D 360 ER2 = (ECC-ONE)/(ECC+ONE)
D 370 ER = SQRTF(ER2)
  380 DO 481 I=1, 2
  390 TW2 = (ONE-CSW(I))/(ONE+CSW(I))
D 400 TW = SQRTF(TW2)
D 410 X = ER*TW
D 420 Y = (ONE+X)/(ONE-X)
D 430 FA = LOGF(Y)
  440 IF (SNW(I)) 450, 470, 480
  450 FA = -FA
  460 GO TO 480
  470 FA = 0.0
  480 SF(I) = SINHF(FA)
  481 F(I) = FA
  490 DM = ECC*(SF(2)-SF(1))-(F(2)-F(1))
  500 RETURN
      END(1,0,0,0,0,0,0,0,0,1,0,0,0,0,0)
```

```
      SUBROUTINE LCC (DELTAV,DELTAT,PSI1,PSI2,AMDA1,AMDA2,BETA1,BETA2,OM
     1EGA)
C     CALCULATION OF LATITUDE, LONGITUDE, AND HEADING
D   1 PI = 3.141592653585793
D   2 TPI = 2.0*PI
    3 DV = DELTAV
    4 P1 = PSI1
    5 B1 = BETA1
D  10 CSV = COSF(DV)
D  20 SNP1 = SINF(P1)
D  30 SNV = SINF(DV)
D  40 CSP1 = COSF(P1)
   50 IF (CSP1) 90, 60, 90
   60 WRITE OUTPUT TAPE 6, 70
   70 FORMAT(1H0,40HINITIAL CONDITIONS ARE AT THE NORTH POLE)
   80 CALL EXIT
D  90 CSB1 = COSF(B1)
D 100 TC = CSV*SNP1+SNV*CSP1*CSB1
  110 PSI2 = ASINF(TC)
D 120 CSP22= 1.0-TC**2
D 121 CSP2 = SQRTF(CSP22)
  130 IF (CSP2) 170, 140, 170
  140 WRITE OUTPUT TAPE 6, 150
  150 FORMAT(1H ,10X,54HNEXT POINT AT A POLE - DISREGARD LONGITUDE AND H
  151EADING)
  160 RETURN
  170 IF (DELTAV-TPI) 180, 190, 190
  180 IF (DELTAV) 210, 190, 200
  190 BETA2 = BETA1
  191 GO TO 270
D 200 TE = (TC*CSV-SNP1)/(CSP2*SNV)
  201 TE = MIN1F(1.0,MAX1F(-1.0,TE))
  202 GO TO 220
  210 WRITE OUTPUT TAPE 6, 211
  211 FORMAT(1H ,10X,67HCHECK THE RANGE ANGLE FOR THIS POINT - NEGATIVE
  211 OR TOO CLOSE TO 360)
  212 RETURN
  220 IF (DELTAV-PI) 260, 230, 260
  230 IF (BETA1-PI) 240, 240, 250
  240 BETA2 = PI-BETA1
  241 GO TO 270
  250 BETA2 = TPI+PI-BETA1
  251 GO TO 270
  260 IF (BETA1-PI) 261, 261, 263
  261 BETA2 = ACOSF(TE)
  262 GO TO 270
  263 BETA2 = TPI-ACOSF(TE)
D 270 TF = (CSV-SNP1*TC)/(CSP1*CSP2)
  271 TF = MIN1F(1.0,MAX1F(-1.0,TF))
  280 DTETA = ACOSF(TF)
  290 IF (DELTAV-PI) 310, 310, 300
  300 DTETA = TPI-DTETA
  310 IF (BETA1-PI) 330, 330, 320
  320 DTETA = -DTETA
```

```
330 IF (AMDA1) 340, 35C, 350
340 AMDA1 = TPI+AMDA1
350 AMDA2 = AMDA1+DTETA-DELTAT*OMEGA
360 IL = AMCA2/TPI
370 IF (IL) 380, 400, 380
380 XL = IL
390 AMDA2 = AMDA2-XL*TPI
400 IF (ABSF(AMDA2)-PI)  450, 450, 410
410 IF (AMDA2) 420, 44C, 440
420 AMDA2 = TPI+AMDA2
430 RETURN
440 AMDA2 = AMDA2 - TPI
450 RETURN
    END(1,0,0,0,0,0,0,C,0,1,0,0,0,0,0)
```

SAMPLE INPUT DATA

POINTS

19

0.	13.8027	-161.8474	185.3
62.6303	16.2881	-156.4527	203.0812
125.9436	18.6195	-150.9205	256.8106
190.6508	20.7688	-145.2394	347.6615
257.5175	22.7077	-139.4044	477.6500
327.9340	24.4080	-133.4175	649.7323
401.2553	25.8430	-127.2896	867.9463
480.2561	26.9881	-121.0416	1137.6215
565.7888	27.8224	-114.7042	1465.6637
659.5704	28.3297	-108.3182	1860.9505
763.7665	28.5000	-101.9143	2334.8721
881.1525	28.3297	-95.6015	2902.0813
1015.3731	27.8224	-89.3840	3581.5392
1171.2675	26.9881	-83.3398	4397.9978
1355.4376	25.8430	-77.5299	5384.1248
1577.0314	24.4080	-72.0176	6583.6141
1849.0611	22.7077	-66.8730	8055.8221
2190.4722	20.7688	-62.1819	9882.8586
2629.6528	18.6195	-58.0612	12180.6557

22

SAMPLE PROBLEM

VAN ALLEN RADIATION PROGRAM

19 TRAJECTORY POINTS ARE GIVEN AS INPUT

TIME (SECONDS)	--XLONG-- LONGITUDE (DEGREES)	--XL-- LATITUDE (DEGREES)	--ALT-- ALTITUDE (KILOMETERS) (NAUT. MILES)	--BVALU-- MAG. FLUX DENSITY	-ELVALU- MC ILWAIN PARAMETER	-FVALU- FLUX	--FLUX-- FLUX*TIME	-FLXSUM- SUM OF FLUX*TIME
						HIGH ENERGY PROTONS		
						LOW ENERGY PROTONS		
						THREAT ELECTRONS B LESS THAN BC		
						THREAT ELECTRONS B GREATER THAN BC		
						L LESS THAN 1.5 ELECTRONS		
						L GREATER THAN 1.5 ELECTRONS		
0.	198.15	13.80	185.30	0.31362	1.06972	0.	0.	0.
			100.00			0.	0.	0.
						0.	0.	0.
						1.819E 05	0.	0.
						0.	0.	0.
						0.	0.	0.
62.630	203.55	16.29	203.08	0.31911	1.10819	0.	0.	0.
			109.60			0.	0.	0.
						0.	0.	0.
						1.044E 05	6.53855E 06	6.53855E 06
						0.	0.	0.
						0.	0.	0.
125.944	209.08	18.62	256.81	0.32221	1.15869	0.	0.	0.
			138.59			0.	0.	0.
						0.	0.	0.
						8.721E 04	5.52172E 06	1.20603E 07
						0.	0.	0.
						0.	0.	0.
190.651	214.76	20.77	347.66	0.32226	1.22229	0.	0.	0.
			187.62			0.	0.	0.
						0.	0.	0.
						9.716E 04	6.28690E 06	1.83472E 07
						0.	0.	0.
						0.	0.	0.
257.517	220.60	22.71	477.65	0.31831	1.29986	0.	0.	0.
			257.77			0.	0.	0.
						0.	0.	0.
						1.481E 05	9.90217E 06	2.82493E 07
						0.	0.	0.
						0.	0.	0.
327.934	226.58	24.41	649.73	0.30929	1.39503	0.	0.	0.
			350.64			0.	0.	0.
						0.	0.	0.
						3.455E 05	2.43274E 07	5.25767E 07
						0.	0.	0.
						0.	0.	0.

TIME (SECONDS)	--XLONG-- LONGITUDE (DEGREES)	--XL-- LATITUDE (DEGREES)	--ALT-- ALTITUDE (KILOMETERS) (NAUT. MILES)	--BVALU-- MAG. FLUX DENSITY	-ELVALU- MC ILWAIN PARAMETER	-FVALU- FLUX	--FLUX-- FLUX*TIME	-FLXSUM- SUM OF FLUX*TIME
						HIGH ENERGY PROTONS		
						LOW ENERGY PROTONS		
						THREAT ELECTRONS B LESS THAN BC		
						THREAT ELECTRONS B GREATER THAN BC		
						L LESS THAN 1.5 ELECTRONS		
						L GREATER THAN 1.5 ELECTRONS		
401.255	232.71	25.84	867.95	0.29443	1.51048	0.	0.	0.
			468.40			0.	0.	0.
						0.	0.	0.
						1.239E 06	9.08169E 07	1.43394E 08
						0.	0.	0.
						1.532E 00	1.12351E 02	1.12351E 02
480.256	238.96	26.99	1137.62	0.27364	1.64886	0.	0.	0.
			613.93			5.376E 00	4.24690E 02	4.24690E 02
						0.	0.	0.
						5.779E 06	4.56525E 08	5.99918E 08
						0.	0.	0.
						2.296E 04	1.81392E 06	1.81403E 06
565.789	245.30	27.82	1465.66	0.24761	1.81150	7.391E 00	0.	6.32182E 02
			790.97			7.423E 02	6.34937E 04	6.39183E 04
						1.463E 07	1.25161E 09	1.25161E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						1.141E 05	9.75807E 06	1.15721E 07
659.570	251.68	28.33	1860.95	0.21770	1.99703	2.237E 01	0.	2.72981E 03
			1004.29			1.023E 04	9.59047E 05	1.02296E 06
						1.629E 07	1.52765E 09	2.77926E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						7.889E 04	7.39806E 06	1.89702E 07
763.767	258.09	28.50	2334.87	0.18575	2.20028	1.597E 01	0.	4.39337E 03
			1260.05			2.014E 04	2.09855E 06	3.12152E 06
						1.405E 07	1.46401E 09	4.24327E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						3.363E 04	3.50424E 06	2.24744E 07
881.153	264.40	28.33	2902.08	0.15366	2.41132	0.	0.	4.39337E 03
			1566.15			2.008E 04	2.35712E 06	5.47864E 06
						1.083E 07	1.27093E 09	5.51419E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						1.717E 04	2.01597E 06	2.44904E 07

TIME (SECONDS)	--XLONG-- LONGITUDE (DEGREES)	--XL-- LATITUDE (DEGREES)	--ALT-- ALTITUDE (KILOMETERS) (NAUT. MILES)	--BVALU-- MAG. FLUX DENSITY	-ELVALU- MC ILWAIN PARAMETER	-FVALU- FLUX	--FLUX-- FLUX*TIME	-FLXSUM- SUM OF FLUX*TIME
						HIGH ENERGY PROTONS		
						LOW ENERGY PROTONS		
						THREAT ELECTRONS B LESS THAN BC		
						THREAT ELECTRONS B GREATER THAN BC		
						L LESS THAN 1.5 ELECTRONS		
						L GREATER THAN 1.5 ELECTRONS		
1015.373	270.62	27.82	3581.54	0.12318	2.62112	0.	0.	4.39337E 03
			1932.83			1.630E 04	2.18747E 06	7.66610E 06
						8.313E 06	1.11575E 09	6.62994E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						8.041E 03	1.07921E 06	2.55696E 07
1171.267	276.66	26.99	4398.00	0.09564	2.82218	0.	0.	4.39337E 03
			2373.45			1.036E 04	1.61531E 06	9.28142E 06
						6.199E 06	9.66387E 08	7.59633E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						1.362E 03	2.12389E 05	2.57820E 07
1355.438	282.47	25.84	5384.12	0.07188	3.01415	0.	0.	4.39337E 03
			2905.63			8.049E 03	1.48243E 06	1.07638E 07
						3.018E 06	5.55827E 08	8.15216E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						8.406E 03	1.54820E 06	2.73302E 07
1577.031	287.98	24.41	6583.61	0.05224	3.20562	0.	0.	4.39337E 03
			3552.95			5.913E 03	1.31039E 06	1.20742E 07
						2.053E 06	4.54957E 08	8.60711E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						5.962E 04	1.32103E 07	4.05405E 07
1849.061	293.13	22.71	8055.82	0.03666	3.41171	0.	0.	4.39337E 03
			4347.45			4.059E 03	1.10419E 06	1.31784E 07
						1.416E 06	3.85079E 08	8.99219E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						3.511E 05	9.55018E 07	1.36042E 08
2190.472	297.82	20.77	9882.86	0.02479	3.65337	0.	0.	4.39337E 03
			5333.44			2.634E 03	8.99443E 05	1.40779E 07
						9.208E 05	3.14382E 08	9.30657E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						1.226E 06	4.18462E 08	5.54504E 08

TIME (SECONDS)	--XLCNG-- LONGITUDE (DEGREES)	--XL-- LATITUDE (DEGREES)	--ALT-- ALTITUDE (KILOMETERS) (NAUT. MILES)	--BVALU-- MAG. FLUX DENSITY	-ELVALU- MC ILWAIN PARAMETER	-FVALU- FLUX	--FLUX-- FLUX*TIME	-FLXSUM- SUM OF FLUX*TIME
						HIGH ENERGY PROTONS		
						LOW ENERGY PROTONS		
						THREAT ELECTRONS B LESS THAN BC		
						THREAT ELECTRONS B GREATER THAN BC		
						L LESS THAN 1.5 ELECTRONS		
						L GREATER THAN 1.5 ELECTRONS		
2629.653	301.94	18.62	12180.66 6573.48	0.01610	3.95723	0.	0.	4.39337E 03
						1.263E 03	5.54699E 05	1.46326E 07
						4.455E 05	1.95637E 08	9.50221E 09
						0.	0.	5.99918E 08
						0.	0.	0.
						3.952E 06	1.73560E 09	2.29010E 09

APPENDIX A

FLUX DATA SET-UP

SPECIFICATION OF DATA MODE

One card precedes all flux data input. This card specifies whether data is in binary mode or BCD mode. If all four flux maps are in binary mode, then the word BINARY is punched four times, beginning in Columns 1, 11, 21, 31. If all maps are in BCD mode, then the word BCD is punched four times. It is not necessary that all maps be in the same mode; intermixing is allowed. For example, if the AFSWC threat electron map is in the BCD mode while the other three are all binary, this card would contain the word BINARY punched three times in columns 1, 11, and 31, while column 21 has the word BCD punched. The following reference is used:

Columns 1-10 refer to high energy protons

11-20 refer to low energy protons

21-30 refer to AFSWC threat electrons

31-40 refer to electrons

PUNCHING OF BINARY DATA

In order to compress the size of the input deck, if flux data are read in the BCD mode, then the subroutine FLUXES will automatically repunch these data in binary mode, to be used in subsequent runs. In the Boeing FORMON system, however, these binary cards must be gang-punched with a 7 and 9 on Column 1 before they can

be read in as column binary cards.

FLUX DATA SET-UP

The flux table is stored columnwise in matrix form.

$$
\begin{array}{c|cccc|}
 & L_1 & L_2 & \cdots & L_j \\
\hline
B_1 & F(B_1, L_1) & F(B_1, L_2) & & \\
B_2 & F(B_2, L_1) & F(B_2, L_2) & & \\
\vdots & & & & \\
B_i & F(B_i, L_1) & F(B_i, L_2) & & \\
\hline
\end{array}
\qquad \text{where } L_j > L_{j-1}, \quad B_i > B_{i-1}
$$

The data cards are processed in the following order:

a. B values

b. L values

c. Flux values

In the case of the AFSWC threat electron belt, the following additional data are required:

d. BC values

e. LC values

f. Bl values

g. Ll values

BC and LC are sets of values defining the boundary of this theoretical belt. B1 and L1 are sets of values referring to transient electrons as explained on page 19.

APPENDIX B

NEW FLUX TABLES

If it is desired to change the flux tables, then some minor changes have to be made in the subroutine FLUXES as well as in the flux data input. Enumerated below are the required changes:

Statement Number	Description
1000	Number of values of B, L, and flux for all flux maps
1010	B and L limits for high energy protons
1020	B and L limits for low energy protons
1030	B and L limits for AFSWC threat electrons
1040	B and L limits for electrons

Refer to program listing of subroutine FLUXES for instructions for making these changes.