AD-A060 743 NAVAL RESEARCH LAB WASHINGTON D C F/G 22/2

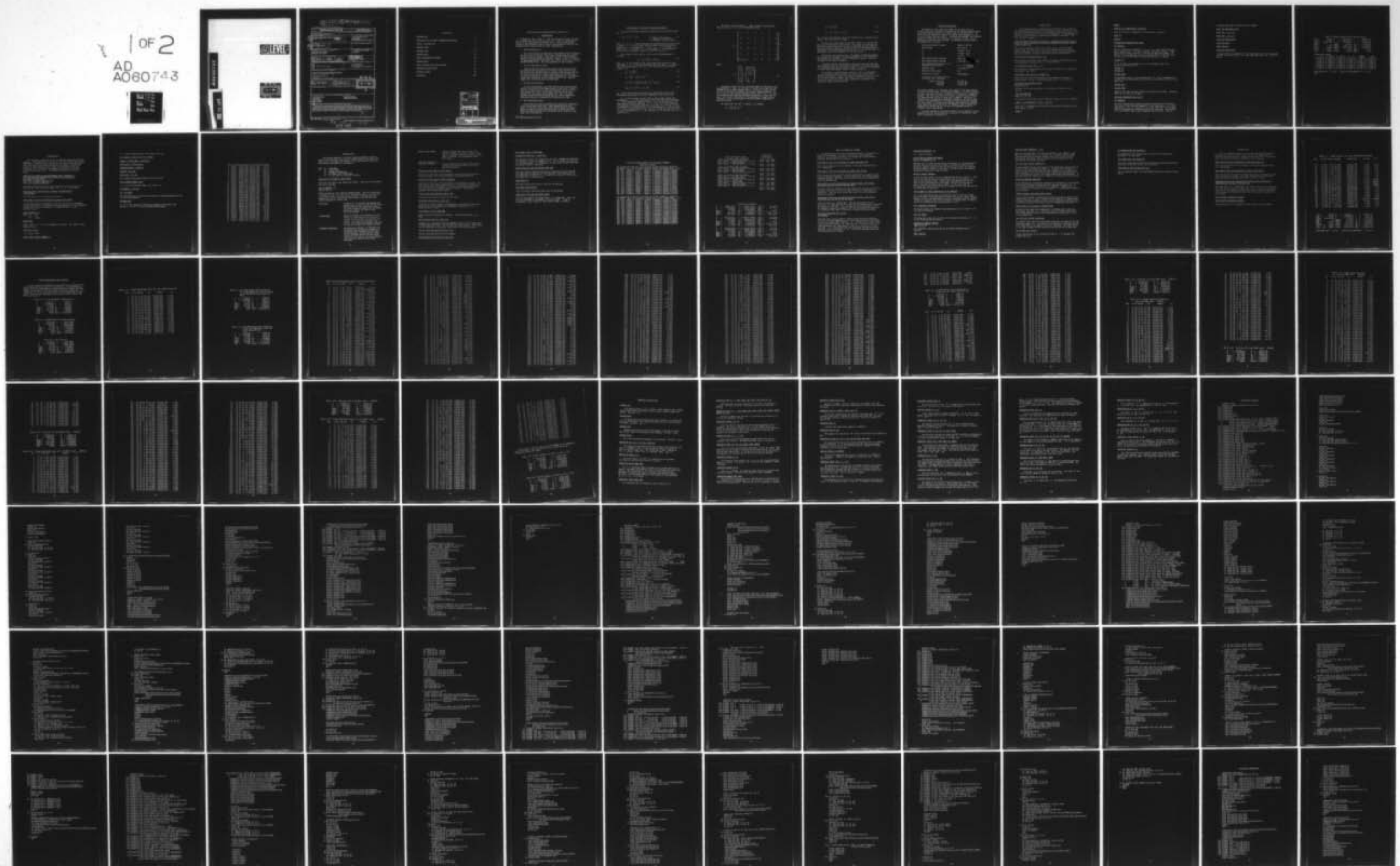
THE ORBIT DETERMINATION OF SOLRAD 11.(U)

AUG 78 R R DASENBROCK

UNCLASSIFIED NRL-MR-3809 SBIE-AD-E000 222 NL

1 OF 2

AD A060743



(18) SBIE (19) AD-E000 222

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 3809	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) THE ORBIT DETERMINATION OF SOLRAD 11.		5. TYPE OF REPORT & PERIOD COVERED A continuing report on an NRL problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Robert R. Dasenbrock		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 54-A-1-20A
11. CONTROLLING OFFICE NAME AND ADDRESS (14) NRL-MR-3809		12. REPORT DATE (11) August 1978
		13. NUMBER OF PAGES 125
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (9) Memorandum rept.,		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. (12) 125 p.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) (16) X0804SL (17) X0804SL		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Orbit determination Least squares Kalman filtering Orbit prediction Trajectory prediction Differential corrections		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The orbit determination scheme used for the Navy's SOLRAD HI satellite is discussed in detail. A standard batch and sequential orbit determination method is described. Each program used in the data reduction scheme is discussed and a guide to its operation is presented. Results from the early portion of the mission are included. These include the initial raw ranges and resulting orbits.

CONTENTS

THE ORBIT DETERMINATION OF SOLRAD 11

Introduction

On March 14, 1976 SOLRAD 11 was co-launched with LES-8 and LES-9 into a synchronous orbit with the Titan IIIC. From this parking orbit Solrad 11 was subsequently maneuvered into its final configuration. Basically four different types or orbits were encountered from the time between orbital injection and the final configuration.

1. The Synchronous Arc:

   This was basically an eight hour coast between orbital injection and the firing of the PKM. Very little ranging data was received and thus little more than a confirmation of the Standard Orbital Parameter Message (SOPM) was obtained. Tables (1) and (3) indicate the pre-launch nominal synchronous coast orbit and the orbit actually obtained respectively. The close agreement is quite amazing.

2. The Post-PKM Transfer Orbit:

   After the PKM was fired and the twenty minute hydrazine burn was completed the spacecraft was allowed to coast for five days. During that time the spacecraft was visible from Blossom Point, MD, for four separate passes. Ample ranging data was received over the period to permit an accurate orbit determination. Considerable trouble was encountered in the ranging system at this time and roughly 50% of the data had to be discarded.

3. Pre-AKM Transfer Orbit:

   At the second perigee after PKM a twenty minute hyrazine burn occurred which boasted the apogee height an additional 10,000 km. This was a highly eccentric coasting trajectory and was visible from Blossom Point for a period after the burn and was again visible for a very short period before AKM. The latter pass provided enough additional ranging data to yield a solution of sufficient accuracy to determine the firing time for AKM.

4. Post-AKM Phasing Orbit:

   The firing of the AKM circularized the trajectory resulting in a near-zero eccentricity orbit. Short hydrazine burns were given to both the A and B spacecraft over the period of several months in order to separate the spacecraft to their final 180 degree configuration and also raising both their respective semi-major axes to 125,000 km.

Note: Manuscript submitted June 7, 1978.

Description of the Orbit Determination Process

A linear, multistage, stochastic process may be described by [see Ref. 1]

$$x_{i+1} = \Phi_i x_i + \Gamma_i w_i \qquad \begin{array}{l} i = 0, \ldots, n-1 \\ p = \text{number of measurements} \\ n = \text{dimension of state vector} \end{array} \tag{1}$$

where x_i is a $n \times 1$ matrix denoting the satellite position state vector at stage i, Φ_i is the state transition matrix which describes the evolution of x from the stage i to $i + 1$. Γ_i is a known $n \times r$ matrix and w_i is a known $r \times 1$ matrix where r is the number of statistically independent components of the driving noise.

The estimate of the state $\hat{x}_i$ due to a set of measurements is given by

$$\hat{x}_i = \overline{x}_i + K_i(z_i - H_i \overline{x}_i) \tag{2}$$

where $\overline{x}_i$ is the estimate of the state before the inclusion of the measurements, K_i is the proportionality matrix or gain, z_i denotes the actual measurements, and $H_i \overline{x}_i$ are the predicted measurements based on $\overline{x}_i$.

$$K_i = P_i H_i^t R^{-1} \tag{3}$$

$$P_i = (M_i^{-1} + H_i^t R^{-1} H_i)^{-1} \tag{4}$$

$$P_i = M_i - M_i H_i^t (H_i M_i H_i^t + R_i)^{-1} H_i M_i \tag{5}$$

$$M_{i+1} = \Phi_i P_i \Phi_i^t + \Gamma_i q_i \Gamma_i^t \tag{6}$$

H_{ij} is the sensitivity matrix and is the sensitivity of the i^{th} measurement with respect to the j^{th} component of the state vector.

M_i is the covariance of the error of the estimate before the measurements. P_i is the covariance of the error after the measurement. R_i is the measurement weighting matrix. Eq. (5) represents the well known matrix inversion identity. If $p < n$ Eq. (5) is advantageous as the dimension of the matrix quantity $(H_i M_i H_i^t + R_i)$ is $p \times p$. If the measurements are processed one at a time the quantity is a scalar since $p = 1$. On the other hand if $p \geq n$ Eq. (4) is used as $(M_i^{-1} + H_i^t R^{-1} H_i)$ is always of dimension $n \times n$. The components of the process noise are denoted by q_i.

The state transition matrix Φ takes on the following simple form in the $(n, \xi, \eta, i, \Omega, U)$ coordinate system.

$$\Phi_i = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ (t_i - t_{i-1}) & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \quad (7)$$

where

$$x = \begin{pmatrix} n \\ \xi \\ \eta \\ i \\ \Omega \\ U \end{pmatrix} = \begin{pmatrix} (\mu/a^3)^{1/2} \\ e \cos \omega \\ e \sin \omega \\ i \\ \Omega \\ M+\omega \end{pmatrix} \quad (8)$$

In general either one or an arbitrary number of observations may be processed at a time. In the case of Solrad 11 the range data was assembled from a single ground station. To process a small span of data it was necessary to include the a-priori weighting matrix M^{-1} in Eq. (4) as the tracking covariance matrix $(H^t R^{-1} H)^{-1}$ was often ill conditioned and thus non-invertable using only a single pass of data. If at least three passes (3/4 revolution) of range data are available, a fair solution may be obtained without weighting the previous solution.

In other words set $M^{-1} = 0$ and Eq. (4) becomes

$$P_i = (H_i^t R_i^{-1} H_i)^{-1}$$

$$K_i = P_i H_i^t R_i^{-1} \tag{9}$$

$$\hat{x}_i = \bar{x}_i + K_i[z_i - H_i x_i] \tag{10}$$

Eqs. (9) and (10) represent the normal equations for a weighted least squares orbit determination scheme.

The user has several choices available to him. If a new pass of data is to be processed he may either: do a batch fit using the new pass and at least three prior passes of range data and not weight the a-priori covariance matrix or: weight the a-priori covariance matrix and process the new observations recursively. If the previous data are unavailable, the latter procedure is the only choice.

The user has full control over: the number of range data points to be processed, the number of orbital parameters to be determined in the solution, and whether or not to include the a-priori in the new estimate of the orbital parameters.

At present the NRL Blossom Point tracking facility provides only range magnitude data for the SOLRAD HI satellite. The orbit determination program accepts only this type of data. It may easily be modified to accept angle measurements should a future need arise.

The program is highly modular in design. This is advantageous as new algorithms become available they can easily replace their obsolete counterparts. Many of the Fortran subroutines that were used in the orbit prediction module were taken from TRIP (reference 3). The routines that make up the entire program are constantly being updated to improve their efficiency.

Orbital Perturbations

The numerical orbit prediction module was designed to be as accurate and fast as possible yet minimizing the required program length since it was to operate within the limited space of the NRL PDP-10. The force field was modeled to include the leading zonal harmonics J_2, J_3, J_4 and the lunar and solar effects. An analytical ephemeris was provided to give the lunar and solar position as a function of time. Solar radiation pressure was also included. The physical constants used in the program are as follows:

Earth gravitational constant	$398601.5\ km^3/sec^2$
Earth radius	6378.135 km
J_2	1.08265×10^{-3}
J_3	-2.5450×10^{-6}
J_4	-1.6715×10^{-6}
Lunar gravitational constant	4902.7 km [illegible]
Solar gravitational constant	1.3271545 × [illegible] /sec^2
Solar radiation pressure at 1 A.U. assuming no reflectivity	$4.7 \times 10^{-6}\ \frac{newtons}{meter^2}$
Satellite area	$2.25\ meters^2$
Satellite reflectivity	0.6
Ellipticity of earth	0.0818188108
Coordinates of tracking antennae at Blossom Point, Maryland:	
Geodetic latitude	38.4314 deg.
longitude	282.9135 deg.
altitude	-0.0247 km.

The orbital elements are referenced with respect to the mean equator-of-date coordinate system. An analytical ephemeris is used to provide the lunar and solar position as required by the numerical integration program. The solar positions are accurate to seconds of arc whereas the lunar positions can be in error up to a degree. While this is acceptable for the numerical integration program it can lead to poor results in the computation of the times of satellite passage through the lunar shadow. An accurate lunar model would require an ephemeris tape. While the use of the entire JPL ephemeris tape is beyond the capability of the NRL PDP-10, the use of an abridged version is presently being investigated.

Although atmospheric retardation effects are not important in the SOLRAD-HI mission, the program does incorporate an atmospheric model. This is described in reference (2).

PROGRAM LOAD

This program provides a convenient way to enter a state vector on the keyboard and form an ASCII file which may be used as input to either the orbit determination program or the ephemeris generation program. The operation of this routine is nearly self explanatory. The program may be executed on the NRL PDP-10 by entering the command " RUN LOAD". The first message to appear is:

ENTER THE NAME OF THE ASCII OUTPUT FILE. THIS CAN BE THE INPUT FILE TO EITHER THE ORBIT DETERMINATION PROGRAM OR EPHEMERIS GENERATION PROGRAM: FORMAT (XXXXXX.EXT)

The user must name the output file at this point. This will later become either the input file to the orbit determination or ephemeris generation program. The file contains the initial weighting covariance matrix which may be required by the orbit determination program.

ENTER SATELLITE IDENTIFICATION - (A5)

Here the user will enter a five character satellite identification code on the keyboard in format (A5), i.e., enter "SRIIA".

ENTER MODIFIED JULIAN DATE - (I5)

The five digit integer modified Julian Date of epoch will be entered at this point, i.e., 42787.

ENTER HOURS, MIN, AND SEC AS HHMMSS.SSS

Enter the time of day corresponding to the modified Julian Date entered above. Be sure to include leading zeros, i.e., 081305.321.

RECTANGULAR OR KEPLERIAN INPUT?
Type "R" or "K"

The state vector may be entered as a Cartesian or Keplerian state vector.

FEET OR KILOMETERS
Type "F" or "K"

The Cartesian state vector may be entered in terms of feet or kilometers.

ENTER X IN KILOMETERS - (FEET) - F20.10)

Enter the X component of the satellite's position with respect to the mean equator of epoch.

ENTER Y

ENTER Z

ENTER XD IN KILOMETERS/SEC (FEET/SEC)

Enter the inertial X component of the satellite's velocity.

ENTER YD

ENTER ZD

PERTURBATION PARAMETERS ARE 100011

ANY CHANGES?

The six perturbation parameters are J_2, J_3, J_4, drag, solar, and lunar forces respectively. "1" means included - "0" means excluded. 100011 means that the effects due to J_2, the sun and moon are included. If more or less perturbations are to be included enter "Y" for this query in which case the following six additional queries will appear.

INCLUDE J2?

If the leading zonal harmonic J2 is to be included in the orbit prediction process type "Y".

INCLUDE J3?

INCLUDE J4?

INCLUDE DRAG?

If atmospheric drag is to be included type "Y". This is automatically deleted from the perturbation model for all orbits above 7000 km in semi-major axis.

INCLUDE SUN?

INCLUDE MOON?

Normally the lunar and solar effects will always be included. Therefore enter "Y" for both of the above.

NON-FIXED PARAMETERS ARE 1111110

ANY CHANGES?

The orbit determination program includes the provision for fitting any one or all of the following components of the state vector, i.e., the mean motion, e cos (ω), e sin (ω), I, Ω, M+ω, and BIAS. 1111110 means that all components are determined except the range bias: By entering "Y" to this query fewer or more components may be fitted. Then the following additional seven queries will appear:

SOLVE FOR SEMI-MAJOR AXIS?

SOLVE FOR e cos (ω)?

SOLVE FOR e sin (ω)?

SOLVE FOR INCLINATION?

SOLVE FOR NODE?

SOLVE FOR M+ω?

SOLVE FOR RANGE BIAS?

A range bias may be determined if the range data exists for a time span for more than several days. For short data spans type "N" for this query.

Table (1) Sample Output of the Program LOAD

```
SAT =        SR11A   A    =     125300.00  X    =    115563.90
MJD =        43508   E    =    0.00020000  Y    =    -38891.83
TSEC=    50723.210   I    =       27.5000  Z    =    -28915.45
NOBS=            0   NODE=         8.5000  XD   =    0.6791356
RMS =        0.000   PERI=       205.0000  YD   =    1.4867899
BIAS=        0.000   MEAN=       125.0000  ZD   =    0.7132165
FIRST    1   40587   TSEC=          0.000  RNG =          0.00
LAST     1   40587   TSEC=          0.000  RNG =          0.00

COVARIANCE MATRIX
       N       E COS(W)    E SIN(W)        I          NODE         M+W
 1.000E-18  0.000E-01  0.000E-01  0.000E-01  0.000E-01  0.000E-01
 0.000E-01  1.000E-09  0.000E-01  0.000E-01  0.000E-01  0.000E-01
 0.000E-01  0.000E-01  1.000E-09  0.000E-01  0.000E-01  0.000E-01
 0.000E-01  0.000E-01  0.000E-01  1.000E-07  0.000E-01  0.000E-01
 0.000E-01  0.000E-01  0.000E-01  0.000E-01  1.000E-07  0.000E-01
 0.000E-01  0.000E-01  0.000E-01  0.000E-01  0.000E-01  1.000E-07
 0.000E-01  0.000E-01  0.000E-01  0.000E-01  0.000E-01  0.000E-01

PERTURBATIONS = 100011    NON-FIXED PARAMETERS = 1111110
END
```

PROGRAM ENTER

This program is primarily a means of entering range data into the computer. The program may be run (on the PDP-10) by the single command "RUN ENTER". Time and range information are input whereas predicted range and range rate for each observation are output. Azimuth and elevation of the satellite with respect to Blossom Point are also presented. These quantities are computed with respect to the most recent orbital state vector. The program procedure is described below.

ENTER ALL FILE NAMES IN THE ALPHANUMERIC FORM: XXXXXX.EXT
WILL THE NEW RANGE OBSERVATIONS BE ADDED TO AN EXISTING FILE
OR ENTERED TO A NEW FILE?
TYPE "A" TO ADD TO EXISTING FILE
TYPE "B" TO BEGIN A NEW FILE

The range observations may be either appended to an existing file or entered into a new file by answering "A" or "B" to the query.

ENTER THE NAME OF THE NEW FILE IN WHICH THE RANGE DATA IS
TO BE ENTERED

The file name is to be entered at this point.

ENTER NAME OF THE FILE CONTAINING THE INPUT STATE VECTOR

The new observations are compared to the most recent state propagated forward numerically to the time of each new observation. Since numerical integration is used to propagate it is important that a recent solution is used.

FOR ALL QUERIES "?"
Type "Y" - Yes
"N" - No
"S" - Stop

ENTER YEAR

Enter "1977" or "77" to correspond to the year. This query is only presented once.

ENTER DAY OF YEAR

i.e., enter "320".

ENTER TIME IN FORM: HHMMSS.SSS

i.e., enter "070926.214" for 7 hr 9 min 26.214 sec.

The program responds with (for example)

YR/DAY = 77/320 H/M/S = 7/9/26.214

RANGE RATE = 0.20322 km/sec

PREDICTED RANGE = 125124.19

AZIMUTH = 99.0 DEG

EVELATION = 25.2 DEG

The azimuth is measured clockwise from true north.

ENTER OBSERVED RANGE IN KM

i.e., enter the observed range, e.g. "125123.9"

DIFFERENCE = -0.29 KM

O.K. TO STORE?

The observation may be stored at this point or deleted depending on the size of the difference.

ANY MORE OBS?

A "Y" to this query will return the program for more data input whereas a "N" or "S" will terminate the procedure.

Table (2) Sample Output of the Program ENTER

1	43411	17933.348	121917.797	-18.71
2	43411	17958.859	121911.052	-18.72
3	43411	25664.599	120030.453	-22.03
4	43411	25690.192	120025.057	-21.90
5	43411	32802.015	118753.038	-24.93
6	43411	32827.597	118749.590	-24.89
7	43411	41997.287	118089.597	-28.65
8	43411	51111.260	118678.839	-30.34
9	43411	51136.844	118681.837	-30.70
10	43411	58564.527	120000.624	-30.58
11	43411	65956.113	121837.153	-29.02
12	43411	65981.622	121843.598	-29.49
13	43412	43890.749	122446.181	-2.54
14	43412	43916.335	122440.785	-1.65
15	43412	58943.265	120059.234	1.71
16	43412	58968.846	120057.884	1.41
17	43412	63099.340	120047.392	-0.55
18	43412	63124.852	120048.141	-0.75
19	43412	70759.595	120879.466	0.48
20	43412	70785.186	120883.363	-0.15
21	43412	78907.740	122796.039	1.41
22	43412	78933.322	122802.784	0.88
23	43413	74528.037	123596.934	23.37
24	43413	74553.644	123593.187	22.77
25	43413	82333.903	123025.680	19.89
26	43413	82359.439	123025.235	19.96
27	43414	4707.618	123375.388	15.87
28	43415	2487.473	122484.704	-9.79
29	43415	2513.054	122479.158	-8.09
30	43415	9899.778	120758.799	34.46
31	43415	18560.154	119761.240	-10.98
32	43415	18585.733	119760.490	-10.98

PROGRAM GENER

This program generates a satellite position ephemeris using as input either the output file from the program LOAD or the output file from the orbit determination program. This routine may be initiated by entering ".RUN GENER" on the keyboard.

TYPE "A" - ALERT MODE
"B" - ECLIPSE MODE
"C" - EPHEMERIS GENERATION
"D" - TO UPDATE STATE VECTOR ONLY
"E" - TO LIST STATE AT SPECIFIED INTERVALS

ENTER ASCII FILE NAME OF INPUT VECTOR

Enter the file name of the input state vector. This will be any name of the user's choosing.

FOR ALL QUERIES "?"
Type Y or N

GENER incorporates five different program modes. The user selects one by typing "A", "B", "C", "D", or "E" to the first query. The specific use of each option is described briefly below. Each provides the capability to save a copy of the state vector at the end point for further updating at a later time.

ALERT MODE: If option "A" is selected the program will compute the rise and set times for the satellite described by the input state vector. The times listed are accurate to the nearest minute and indicate the time of zero elevation passage.

ECLIPSE MODE: Option "B" permits the calculation of all eclipses of the satellite by the earth or or moon (if any) within the selected time period. Four times are given for each eclipse, the beginning of penumbra and umbra and the end of umbra and penumbra passage.

EPHEMERIS GENERATION: The selection of option "C" initiates the calculation of the satellite ephemeris at two minute time intervals over the time period selected. An abbreviated listing is also available which lists the state vector at 1, 15, 29, 31, 45, and 59 minutes past each hour instead of two minute intervals.

UPDATE STATE VECTOR: Option "D" allows the user to update the orbital elements of a satellite over a time span to a selected stop time. Only the final state is printed. No intermediate states are listed.

LIST STATE VECTOR AT SPECIFIED INTERVALS: By using option "E" the state vector may be listed on the teletype and/or disk file at specified intervals.

<u>ENTER ASCII FILE NAME OF INPUT VECTOR</u>

Enter the 10 character file designation containing the starting state vector and epoch. This must be entered in the form XXXXXX.EXT.

<u>ENTER ASCII FILE NAME OF OUTPUT EPHEMERIS</u>

Enter the 10 character file designation of the ephemeris listing. This will normally be a rather lengthy file as it contains the satellite's state vector every two minutes. The disk storage space required for a one day listing will be approximately 95 blocks. 35 blocks are required for an abbreviated listing.

<u>INITIAL STATE AND EPOCH FOR "SR11A" ARE</u>

The state vector will be listed at this point.

<u>ENTER MOD JULIAN DATE OF START TIME</u>

Enter the five digit integer corresponding to the modified Julian date of the start time (beginning of the ephemeris generation); e.g., 43295 for June 1, 1977.

<u>ENTER HOURS & MIN IN FORM HHMM</u>

Enter the hours and minutes of the day. Include leading zeros, e.g., 0905.

<u>MOVE STARTING POINT BY X.XXXX DAYS</u>

Normally the starting point of the ephemeris may be several hours ahead or behind the epoch time. This message only confirms this. The program may be terminated at this point if any input error has been made.

<u>INITIAL STATE AND EPOCH FOR "SR11A" ARE</u>

The new starting state vector will appear.

<u>ENTER MODIFIED JULIAN DATE OF STOP TIME</u>

ENTER HOURS & MIN IN FORM HHMM

INTEGRATION TIME SPAN = X.XXXX DAYS

This message confirms the integration time span. Remember 95 blocks/day are required to store the ephemeris file. If the integration time span is some large number, e.g. 10000 days, an input error has been made and the program should be terminated.

ROUTINE DOES NOT WORK FOR REVERSE TIME SPAN

If by any chance a negative time span is requested the above message will appear and the program will be terminated. This restriction is only applicable to option "C". The numerical integration scheme works in reverse for all other options.

FINAL STATE IS

The final state vector will be listed on the terminal.

SAVE FINAL STATE VECTOR?

If "Y" is given to the above query then the following:

ENTER ASCII FILE NAME OF OUTPUT VECTOR

List the file name of the output state, i.e. "ABIRD.OUT". This file will be catalogued as a permanent file on the disk. It may be subsequently used as input to either ENTER, GENER, or DIFFCR.

Table (3) Sample Output of the Program GENER.
Ephemeris option.

```
SAT SR11B MJD 43399 LAT 18.42 LON 351.82 RNG 121253.46 121049.56
LSP BLMPT  87.7  25.0 121461.38 ITALY 219.0  58.3 INDIA 283.9  12.4
GMT 1101 -375586.6  67713.9  12645.7 -0.986410 0.150739 0.065362
MIN  661 -101491.7  60033.1  39268.0 -1.027696-1.346648-0.598549
MIN  663 -101614.9  59871.4  39196.2 -1.025165-1.348142-0.599526
MIN  665 -101737.7  59709.5  39124.2 -1.022632-1.349631-0.600502
MIN  667 -101860.3  59547.5  39052.0 -1.020096-1.351117-0.601476
MIN  669 -101982.6  59385.2  38979.8 -1.017556-1.352599-0.602448
MIN  671 -102104.5  59222.8  38907.5 -1.015013-1.354076-0.603418
MIN  673 -102226.2  59060.3  38835.0 -1.012468-1.355550-0.604387
MIN  675 -102347.5  58897.5  38762.4 -1.009919-1.357020-0.605353
MIN  677 -102468.6  58734.6  38689.7 -1.007367-1.358485-0.606318
MIN  679 -102589.3  58571.5  38616.9 -1.004812-1.359947-0.607282
MIN  681 -102709.7  58408.2  38544.0 -1.002354-1.361413-0.608261
MIN  683 -102829.8  58244.7  38470.9 -0.999793-1.362867-0.609221
MIN  685 -102949.7  58081.1  38397.7 -0.997229-1.364316-0.610179
MIN  687 -103069.2  57917.3  38324.5 -0.994662-1.365761-0.611134
MIN  689 -103188.4  57753.3  38251.1 -0.992092-1.367202-0.612089

SAT SR11B MJD 43399 LAT 17.89 LON 345.77 RNG 120822.20 120628.84
LSP BLMPT  92.0  29.5 121020.79 ITALY 227.7  54.7 INDIA 285.4   6.8
GMT 1131 -375813.2  65974.4  12058.5 -0.986469 0.150419 0.065223
MIN  691 -103307.3  57589.2  38177.6 -0.989520-1.368639-0.613041
MIN  693 -103425.9  57424.9  38103.9 -0.986944-1.370072-0.613991
MIN  695 -103544.1  57260.4  38030.2 -0.984366-1.371501-0.614940
MIN  697 -103662.1  57095.7  37956.4 -0.981784-1.372926-0.615886
MIN  699 -103779.8  56930.9  37882.4 -0.979200-1.374346-0.616831
MIN  701 -103897.1  56765.9  37808.3 -0.976613-1.375763-0.617774
MIN  703 -104014.2  56600.7  37734.1 -0.974023-1.377175-0.618715
MIN  705 -104130.9  56435.3  37659.8 -0.971430-1.378583-0.619654
MIN  707 -104247.3  56269.8  37585.4 -0.968834-1.379987-0.620592
MIN  709 -104363.4  56104.1  37510.9 -0.966235-1.381387-0.621527
MIN  711 -104479.2  55938.3  37436.2 -0.963633-1.382783-0.622461
MIN  713 -104594.7  55772.3  37361.5 -0.961028-1.384175-0.623393
MIN  715 -104709.8  55606.1  37286.6 -0.958420-1.385562-0.624323
MIN  717 -104824.7  55439.7  37211.7 -0.955810-1.386946-0.625251
MIN  719 -104939.2  55273.2  37136.6 -0.953196-1.388325-0.626177
```

Table (4) Eclipse option.

```
SATELLITE ENTERED EARTH PENUMBRA
MJD = 43428      HR/MIN/SEC= 5/11/23.818
SATELLITE LEFT EARTH PENUMBRA
MJD = 43428      HR/MIN/SEC= 6/17/50.350

SATELLITE ENTERED MOON PENUMBRA
MJD = 43428      HR/MIN/SEC=11/25/53.602
SATELLITE LEFT MOON PENUMBRA
MJD = 43428      HR/MIN/SEC=12/36/46.835

SATELLITE ENTERED EARTH PENUMBRA
MJD = 43433      HR/MIN/SEC= 9/ 3/28.662
SATELLITE ENTERED EARTH UMBRA
MJD = 43433      HR/MIN/SEC= 9/22/10.134
SATELLITE LEFT EARTH UMBRA
MJD = 43433      HR/MIN/SEC=10/16/52.792
SATELLITE LEFT EARTH PENUMBRA
MJD = 43433      HR/MIN/SEC=10/35/34.182
```

Table (5)
Alert option.

```
SR11B  RISE TIME
43400  16HR  8MIN

SR11B   SET TIME
43401   3HR 13MIN

SR11B  RISE TIME
43401  22HR 31MIN

SR11B   SET TIME
43402  10HR 46MIN

SR11B  RISE TIME
43403   1HR 24MIN

SR11B   SET TIME
43403  18HR 39MIN

SR11B  RISE TIME
43404   6HR 55MIN
```

Table (6) List option.

A	=	125091.2610	X =	-101491.71000	XLAT =		18.4184
E	=	0.00645635	Y =	60033.06980	XLON =		351.8231
I	=	27.3292	Z =	39268.04980	FCAL =		0
NODE	=	9.5158	XD =	-1.02769560	MJD =		43399
PERI	=	138.2987	YD =	-1.34664790	H/M/S =		11/01/ 0.00
MEAN	=	358.2364	ZD =	-0.59854850	STEP =		2767.0304
A	=	125119.1870	X =	-108333.19400	XLAT =		16.2384
E	=	0.00668185	Y =	50031.69080	XLON =		327.5566
I	=	27.3306	Z =	34754.87770	FCAL =		8
NODE	=	9.5130	XD =	-0.87101722	MJD =		43399
PERI	=	137.2624	YD =	-1.42901762	H/M/S =		13/01/ 0.00
MEAN	=	5.1466	ZD =	-0.65398462	STEP =		3600.0000
A	=	125143.0360	X =	-114011.99300	XLAT =		13.9035
E	=	0.00689160	Y =	39492.17850	XLON =		303.1581
I	=	27.3324	Z =	29867.58970	FCAL =		8
NODE	=	9.5105	XD =	-0.70502405	MJD =		43399
PERI	=	136.2493	YD =	-1.49599279	H/M/S =		15/01/ 0.00
MEAN	=	12.0290	ZD =	-0.70237432	STEP =		3600.0000

ORBIT DETERMINATION PROGRAM

The operation of the orbit determination program is designed to be nearly self explantory. The user has control of the range data reduction via an interactive mode. The program may be initiated by entering " RUN DIFFCR" on the terminal.

ENTER NAME OF THE FILE CONTAINING THE RANGE MAGNITUDE DATA

Enter the 10 character file designation of the range magnitude data, i.e., "ABIRD.RNG".

ENTER NAME OF THE FILE CONTAINING THE INPUT STATE VECTOR

This file will be the a-priori orbital state vector as required by the orbit determination program. It may correspond to the Standard Orbital Parameter Message. The ASCII file referred to here will be generated using the program LOAD.

ENTER NAME OF THE FILE CONTAINING THE UPDATED OUTPUT STATE VECTOR REFERENCED NEAR THE START OF THE DATA SPAN

The state vector as referenced to the beginning of the data span will be recorded in an ASCII file by this name of the user's choosing, i.e., "ABIRD.STA". This may later become the input file to the ephemeris generation program.

ENTER NAME OF THE FILE CONTAINING THE UPDATED OUTPUT STATE VECTOR REFERENCED NEAR THE END OF THE DATA SPAN

Enter any name, i.e., "ABIRD.END". The new state vector will be equivalent to "ABIRD.STA" but its epoch will be referenced to a time near the end of the data span. This also may be used as the input file to the ephemeris generation program or the orbit determination program in its sequential mode should a new batch of range data be processed.

NON-FIXED PARAMETERS ARE 1111010
ANY CHANGES?

The input state vector contains a seven integer key which indicates which of the seven components of the state vector are to be held fixed in the orbit determination process. "1" indicates that it may vary in the orbit fitting scheme. "0" indicates it is to be held fixed. The seven parameters are: mean motion, e cos (ω), e sin (ω), I, Ω, M+ω, and BIAS. e.g. 1111010 means that the nodal angle and the bias parameter are to be held fixed. If "Y" is given to the above query the following message will appear.

ENTER NEW PARAMETERS - I7

i.e., enter "1110010".

ENTER STARTING OBSERVATION NUMBER
Enter 1 if beginning - I4

Normally one will begin processing the observations at the beginning of the data span in which case the integer "1" should be entered on the keyboard. However should you wish to begin at one of the later points enter the number assigned to that point. The data points are numbered sequentially from one in steps of unity.

WEIGHT A-PRIORI ESTIMATE?

If the new observations are to be weighted with the old, type "Y". This will be the case if only a few observations are to be processed, i.e., when in the sequential mode. However the range observations may be used to solely determine the state update if sufficient data exists over a time span of several days or more. In this case type "N". Naturally the a-priori estimate may be weighted with the new observations in either mode of operation.

ENTER NUMBER OF RANGE OBSERVATIONS TO BE PROCESSED

All of the observations may be processed at once in which case enter the total number of observations in integer form (or any larger number). However if only a few observations are to be processed, and the a-priori estimate is weighted, then enter the number in integer form. As few as one or all observations may be processed at a time.

LIST INDIVIDUAL RESIDUALS?

The observed minus the computed range magnitudes will be listed if "Y" is given to this query.

EDIT O-C ARRAY?

The range data points may be edited at this point by entering "Y". If so the following message appears:

AUTOMATIC OR MANUAL EDITING?
Type "A" or "M"

The individual range data points may be deleted automatically or manually.

RMS = XXX.XXX

ENTER DELETION THRESHOLD - F10.3

Here the user must enter the deletion threshold. For example, if the RMS for a batch of data is 2.5 km and 5.0 is entered, then all range observations having a residual greater than 5.0 km will be deleted. The same bad data points must be deleted at each iteration of the differential corrections procedure.

LIST THE STATE VECTOR "ABIRD.STA"?

The new estimate of the state vector at the end of each iteration may be listed on the line printer by typing Y. In either case it will be written on the ASCII file under the name "ABIRD.STA" or any name of the user's choosing. This state vector has an epoch time one minute past the first hour prior to the first range observation being processed. In the batch mode this quantity will normally be listed on the line printer.

RETURN FOR ANOTHER ITERATION?

If the differential corrections procedure has converged to a satisfactory solution the iteration procedure may be terminated by typing "N". If "Y" is entered another iteration will be computed. Normally four to five iterations are necessary for convergence to a sufficiently accurate state estimate.

COMPUTE TRACKING COVARIANCE MATRIX AGAIN?

If the initial vector is sufficiently accurate this matrix need only be computed once and "N" will be entered on the keyboard. If however each successive iteration produces a state vector sufficiently different from the previous iteration this quantity may be recomputed by entering "Y".

UPDATE EPOCH TO TIME NEAR LAST OBSERVATION?

If desired, the epoch may be updated to a time near that of the last observation processed; i.e., one minute past the nearest previous hour. This should always be done if the observations are processed a few at a time sequentially.

LIST THE STATE VECTOR "ABIRD.END"

The updated state vector and covariance matrix may be listed on the line printer by typing "Y". In either case the results are inserted into the file "ABIRD.END" or any name of the user's choosing. This state vector is the same as ABIRD.STA but referenced near the end of the data span.

ACCEPT MORE DATA POINTS?

If more data points are to be processed type "Y". To terminate the program enter "N".

LIST OBSERVATIONS AND RESIDUALS?

All observations and residuals may be listed on the teletype by entering "Y" to this query.

EDIT RANGE DATA FILE XXXXXX.EXT

The entire range data file may be edited at this point and a new file absent of the bad data points will be created.

ENTER NAME OF EDITED RANGE DATA FILE

The ten character name in the form XXXXXX.EXT will be entered at this point.

PROGRAM BATCH

This is a simplified version of the orbit determination program. After the user answers several initial queries the program runs to completion with no additional interaction with the user. This program may be initiated by entering "RUN BATCH" on the terminal.

ENTER NAME OF THE FILE CONTINING THE RANGE MAGNITUDE DATA

Enter the 10 character file designation of the range magnitude data, i.e., "ABIRD.RNG".

ENTER NAME OF THE FILE CONTAINING THE INPUT STATE VECTOR

This file will be the a-priori orbital state vector as required by the orbit determination program. The ASCII file referred to here may be generated using the program LOAD, GENER, DIFFCR, or BATCH.

ENTER NAME OF THE FILE CONTAINING THE UPDATED OUTPUT STATE VECTOR REFERENCED NEAR THE END OF THE DATA SPAN

Enter any name, i.e., "ABIRD.END". The state vector will be referenced to a time near the end of the data span. This also may be used as the input file to the ephemeris generation program or the orbit determination program in its sequential mode should a new batch of range data be processed.

ENTER STARTING OBSERVATION NUMBER

ENTER STOPPING OBSERVATION NUMBER

The program will run to completion after entering the above query.

Table (7). Sample output of the orbit determination program.

TAG	YR	DAY	HR	MN	SECOND	RANGE(KM)	O-C(KM)	R(K)
70	77	277	4	28	53.230	120173.305	0.235	4.16
71	77	277	6	59	46.279	120744.259	-0.520	4.04
72	77	277	7	0	12.200	120748.306	-0.125	4.04
73	77	277	9	53	38.276	122895.120	-0.599	3.91
74	77	277	9	54	4.236	122901.866	-0.455	3.91
75	77	278	3	23	6.804	121790.235	-0.318	3.19
76	77	278	3	23	32.415	121783.340	-0.077	3.19
77	77	278	6	8	27.149	119389.347	0.691	3.09
78	77	278	6	8	52.684	119384.401	0.676	3.09
79	77	278	10	39	13.433	118129.020	11.656	******
80	77	278	10	39	39.040	118129.469	10.942	******
81	77	278	13	21	40.733	119239.451	10.094	******
82	77	278	13	22	6.269	119243.798	9.931	******
83	77	278	16	27	52.809	121745.416	-0.269	2.75
84	77	278	16	29	52.432	121776.894	0.165	2.74
85	77	278	16	30	17.968	121783.490	0.129	2.74
86	77	278	16	37	0.429	121887.668	-0.497	2.74
87	77	278	16	37	25.939	121894.863	0.039	2.74
88	77	279	9	55	53.161	122641.496	-2.154	2.24
89	77	279	12	16	5.763	120760.598	-0.570	2.18
90	77	279	12	18	48.871	122731.218	1999.896	******
91	77	279	12	33	4.749	120579.673	-0.506	2.18
92	77	279	12	33	30.328	120575.326	-0.480	2.18
93	77	279	17	1	17.101	119957.304	1.347	2.07
94	77	279	17	2	32.534	119964.649	0.461	2.07

SAT =		SR11A	A =	124875.01	X =	124193.62
MJD =		43425	E =	0.00445603	Y =	2691.80
TSEC=		25260.000	I =	27.3254	Z =	-8705.50
NOBS=		37	NODE=	9.0366	XD =	0.0161565
RMS =		0.775	PERI=	42.7876	YD =	1.5962983
BIAS=		0.000	MEAN=	308.8517	ZD =	0.8132601
FIRST	70	43420	TSEC=	16133.230	RNG =	120173.30
LAST	112	43425	TSEC=	28343.913	RNG =	121999.49

PERTURBATIONS = 100011 NON-FIXED PARAMETERS = 1111110

Orbital Ranging Data and Solutions

In this section the ranging data and respective solutions for the early portion of the mission will be presented. Listed immediately below are the pre-launch nominal, Standard Orbital Parameter Message (SOPM), and the best solution obtained by using the ranging data. For comparison, all three state vectors have been referenced at the same epoch. Note the close agreement between the pre-launch nominal and the final solution.

Table (8) Pre-launch Nominal Synchronous Arc

A	=	42747.86	X	=	-27868.34
E	=	0.01344603	Y	=	-29851.15
I	=	25.1831	Z	=	-10539.28
NODE	=	13.6786	XD	=	2.2991557
PERI	=	210.1283	YD	=	-1.7756603
MEAN	=	5.6798	ZD	=	-1.0668875
MJD	=	42852	H/M/S	=	8/31/40.24

Table (9) Standard Orbital Parameter Message

A	=	42726.3497	X	=	-27823.50290
E	=	0.01346675	Y	=	-29861.12180
I	=	25.1916	Z	=	-10539.68950
NODE	=	13.7258	XD	=	2.30177241
PERI	=	210.5023	YD	=	-1.77352623
MEAN	=	5.3258	ZD	=	-1.06732072
MJD	=	42852	H/M/S	=	8/31/40.24

Table (10) Final Solution

A	=	42721.8002	X	=	-27825.81610
E	=	0.01334121	Y	=	-29860.66150
I	=	25.1917	Z	=	-10539.26070
NODE	=	13.7258	ZD	=	2.30139433
PERI	=	210.2434	YD	=	-1.77360111
MEAN	=	5.5764	ZD	=	-1.06731634
MJD	=	42852	H/M/S	=	8/31/40.24

Table (11) Range magnitude data for the synchronous arc.

TAG	YR	DY	HR	MN	SEC	RANGE	O-C
1	76	75	12	55	48.117	40066.960	-1.790
2	76	75	13	34	11.687	40031.140	0.070
3	76	75	13	34	37.612	40030.540	0.190
4	76	75	14	34	59.397	39841.220	-2.240
5	76	75	14	35	25.375	39839.420	-0.070
6	76	75	14	36	25.352	39835.520	0.370
7	76	75	14	39	15.688	39822.480	-0.420
8	76	75	14	39	41.657	39820.780	-0.050
9	76	75	14	40	49.066	39816.090	0.280
10	76	75	14	41	14.604	39813.790	-0.210
11	76	75	15	38	37.352	39505.900	-0.850
12	76	75	15	39	3.353	39503.050	-0.250
13	76	75	16	45	29.423	39058.310	-1.990
14	76	75	16	45	55.410	39054.710	-0.650
15	76	75	16	47	5.179	39046.770	-0.040
16	76	75	16	47	31.175	39043.770	0.040
17	76	75	16	48	46.659	39035.080	0.230
18	76	75	16	49	11.201	39031.630	-0.410
19	76	75	18	45	47.179	38216.340	0.300
20	76	75	18	47	42.312	38203.600	-2.810
21	76	75	18	48	8.315	38200.750	-0.190
22	76	75	18	49	21.532	38192.810	-0.150

Table (12) Post-PKM Transfer Orbit referenced to a time immediately after the firing of the PKM and the first hydrazine burn.

A	=	78183.36	X	=	-27186.43
E	=	0.45029499	Y	=	30106.06
I	=	25.2381	Z	=	16840.14
NODE	=	13.8108	XD	=	-3.0524713
PERI	=	94.5052	YD	=	-1.8641317
MEAN	=	7.3612	ZD	=	-0.5098235
MJD	=	42853	H/M/S	=	02/01/00.000

Table (13) Post-PKM Transfer Orbit referenced to the time immediately before the second hydrazine burn.

A	=	78174.27	X	=	-12246.34
E	=	0.45073484	Y	=	36876.96
I	=	25.2535	Z	=	18270.89
NODE	=	13.8022	XD	=	-3.4968687
PERI	=	94.2400	YD	=	-1.1073740
MEAN	=	359.9640	ZD	=	-0.1137500
MJD	=	42858	H/M/S	=	01/35/00.000

Table (14) Range Magnitude Data for the Transfer Orbit

TAG	YR	DY	HR	MN	SEC	RANGE	O-C
1	76	76	2	48	4.173	39485.058	0.184
2	76	76	2	48	4.173	38135.999	-1348.875
3	76	76	2	48	30.022	38153.237	-1349.598
4	76	76	2	48	30.022	39502.296	-0.538
5	76	76	2	49	46.114	38206.301	-1349.698
6	76	76	2	49	46.114	39555.360	-0.639
7	76	76	2	50	11.427	38224.737	-1349.044
8	76	76	2	50	11.427	39573.797	0.016
9	76	76	2	51	13.909	38270.156	-1347.725
10	76	76	2	51	16.601	39619.215	-0.572
11	76	76	2	51	41.920	38288.594	-1349.151
12	76	76	2	51	41.920	39637.653	-0.092
13	76	76	2	52	53.644	38339.708	-1349.168
14	76	76	2	52	53.644	39688.767	-0.109
15	76	76	2	53	18.182	39705.406	-1.051
16	76	76	2	53	18.182	38356.348	-1350.108
17	76	76	2	53	18.182	39705.406	-1.051
18	76	76	3	45	37.440	40941.760	-1347.098
19	76	76	3	45	37.440	42290.815	1.957
20	76	76	3	46	3.296	40962.890	-1349.705
21	76	76	3	46	3.296	42311.951	-0.644
22	76	76	3	47	34.574	41049.080	-1347.598
23	75	76	3	47	34.574	42398.141	1.463
24	76	76	3	48	0.327	41072.470	-1348.011
25	76	76	3	48	0.327	42421.525	1.044
26	76	76	3	49	6.321	41134.370	-1347.269
27	76	76	3	49	6.321	42483.432	1.793
28	76	76	3	49	31.874	42506.666	1.285
29	76	75	3	49	31.874	41157.610	-1347.771
30	76	76	3	49	31.874	42506.666	1.285
31	76	76	6	36	29.596	52268.520	-1346.989
32	76	76	6	36	54.149	52297.300	-1348.220
33	76	76	6	37	54.338	52371.640	-1347.479
34	76	76	6	37	54.338	53719.957	0.838
35	76	76	6	38	20.328	52402.820	-1348.094
36	76	76	6	38	20.328	53751.130	0.216
37	76	76	6	41	20.521	52623.170	-1348.406
38	76	76	6	41	48.306	52654.200	-1351.435
39	76	76	6	43	27.398	52778.760	-1348.420
40	76	76	6	45	36.340	52937.650	-1347.957
41	76	76	6	50	53.353	53328.283	-1347.275
42	76	76	9	26	54.838	66616.883	151.844
43	76	76	9	27	20.020	66649.411	152.424
44	76	76	9	28	28.598	66735.901	151.916
45	76	76	9	28	30.328	66585.992	-0.188

46	76	76	9	28	30.328	66595.992	-0.188
47	76	76	9	28	53.102	66767.978	152.908
48	76	76	9	28	56.307	66619.070	-1.065
49	76	76	9	42	45.821	67818.901	147.980
50	76	76	9	42	45.821	67668.243	-2.679
51	76	76	9	43	10.341	67850.679	148.684
52	76	76	9	43	10.341	67700.021	-1.974
53	76	76	10	7	1.130	69663.524	150.622
54	76	76	10	7	4.308	69513.615	-3.304
55	76	76	10	7	4.308	69547.000	30.080
56	76	76	10	7	27.105	69697.551	151.820
57	76	76	10	28	41.272	72652.000	1498.631
58	76	76	10	31	1.626	75831.290	4501.186
59	76	76	10	31	7.401	74344.441	3007.068
60	76	76	10	31	30.340	72867.000	1500.749
61	76	76	10	31	30.342	72867.560	1501.306
62	76	76	11	13	46.907	74697.040	152.202
63	76	76	11	14	12.905	74729.417	152.171
64	76	76	11	15	12.585	74804.215	152.586
65	76	76	11	15	37.052	74835.540	153.423
66	76	76	11	15	40.070	74685.633	-0.245
67	76	76	11	30	33.096	75949.000	152.742
68	76	76	11	30	35.664	75799.800	0.356
69	76	76	11	30	59.056	75921.000	152.537
70	76	76	11	32	12.292	76070.840	151.545
71	76	76	11	32	14.841	75920.928	-1.528
72	76	76	11	32	39.404	75953.755	0.843
73	76	76	11	32	39.930	76103.670	150.229
74	76	76	12	2	41.168	79679.441	1503.602
75	76	76	12	2	43.800	78180.000	0.930
76	76	76	12	3	7.151	79710.620	1502.895
77	76	76	12	3	9.950	78211.000	-0.160
78	76	76	12	4	7.036	79783.920	1502.724
79	76	76	12	4	9.590	78284.000	-0.329
80	76	76	12	4	32.560	79815.546	1503.044
81	76	76	12	4	34.160	79316.000	1.536
82	76	76	12	47	11.283	84423.800	3000.102
83	76	76	12	49	5.049	84560.800	3000.113
84	76	76	12	49	31.048	85470.000	3878.024
85	76	76	13	3	52.669	85474.113	2948.888
86	76	76	13	3	52.669	84125.000	1499.775
87	76	76	13	4	18.519	85503.792	2847.680
88	76	76	13	4	18.519	84154.000	1497.888
89	76	77	9	56	26.134	111439.733	-2.633
90	76	77	9	56	52.902	111430.289	-2.336
91	76	77	10	39	23.409	110490.140	-1.489
92	76	77	10	39	49.824	110480.547	-1.597
93	76	77	11	14	30.085	111048.504	1347.967
94	76	77	11	14	56.838	111039.360	1348.921
95	76	77	11	33	50.623	109260.542	-1.169
96	76	77	11	34	16.376	109251.398	-0.560

97	76	77	11	52	21.557	108840.083	-0.560
98	76	77	11	52	21.557	110189.149	1348.507
99	76	77	11	52	47.310	108830.640	-0.236
100	76	77	11	52	47.310	110179.705	1348.829
101	76	77	12	23	52.655	108124.479	1.007
102	76	77	12	23	52.655	109473.544	1350.073
103	76	77	12	24	18.408	108112.787	-0.924
104	76	77	12	25	34.343	109433.672	1348.736
105	76	77	12	25	59.656	108075.013	-0.332
106	76	77	12	25	59.656	109424.079	1348.735
107	76	77	12	39	29.265	107769.225	0.447
108	76	77	12	39	29.265	107769.225	0.447
109	76	77	12	39	54.584	107759.331	0.133
110	76	77	12	39	54.584	109108.397	1349.199
111	76	77	12	39	54.584	107759.331	0.133
112	76	77	12	42	35.412	110545.020	2846.661
113	76	77	12	42	35.412	107747.000	48.641
114	76	77	13	27	21.927	106686.075	0.016
115	76	77	13	27	47.808	106677.231	0.881
116	76	77	13	27	47.808	106677.231	0.881
117	76	77	13	42	6.439	109201.932	2847.176
118	76	77	13	58	57.118	105973.768	-3.781
119	76	77	13	58	57.118	108821.796	2844.248
120	76	77	13	59	23.111	105968.671	0.804
121	76	77	13	59	23.111	108816.699	2848.833
122	76	77	14	2	12.129	108753.143	2848.208
123	76	77	14	2	38.883	108742.651	2847.675
124	76	77	14	12	6.568	108531.747	2847.825
125	76	77	14	12	32.321	108522.000	2847.642
126	76	77	14	41	14.329	105037.216	0.339
127	76	77	14	41	40.290	105028.676	1.381
128	76	77	14	42	52.257	105001.691	0.955
129	76	77	14	43	17.795	104991.948	0.635
130	76	77	14	46	15.386	104925.993	0.187
131	76	77	14	46	41.139	104916.700	0.391
132	76	77	14	55	48.574	104714.000	-0.610
133	76	77	14	56	14.343	104706.000	0.876
134	76	77	15	8	26.615	104436.582	0.788
135	76	77	15	8	51.928	104426.689	0.198
136	76	77	15	9	49.533	104405.254	-0.071
137	76	77	15	10	14.846	104396.000	-0.025
138	76	77	15	58	11.829	103341.591	0.770
139	76	77	15	58	37.583	103332.447	1.070
140	76	77	15	59	32.845	103311.461	0.348
141	76	77	15	59	58.596	103301.868	0.198
142	76	77	16	32	1.896	103945.073	1349.981
143	76	77	16	32	1.896	102596.007	0.914
144	76	77	16	32	27.889	103935.329	1349.812
145	76	77	16	32	27.889	102586.284	0.767
146	76	77	16	36	1.136	103856.104	1349.181
147	76	77	16	36	1.136	102507.118	0.195
148	76	77	16	36	26.641	103846.751	1349.233

149	76	77	16	36	26.641	102497.675	0.157
150	76	77	16	49	52.006	102200.431	0.443
151	76	77	16	50	17.555	102190.088	-0.441
152	76	77	16	56	15.826	102057.879	0.130
153	76	77	16	56	41.806	102048.586	0.477
154	76	77	17	3	3.528	101906.334	0.040
155	76	77	17	3	28.937	101896.891	0.049
156	76	77	17	11	21.121	101720.913	0.041
157	76	77	17	11	47.327	101711.020	-0.068
158	76	77	17	24	14.338	101430.565	-0.740
159	76	77	17	24	40.091	102770.786	1349.159
160	76	77	17	24	40.091	101421.720	0.092
161	76	77	17	27	41.413	101353.517	0.092
162	76	77	17	28	7.262	101343.624	-0.068
163	76	77	17	38	22.944	102460.201	1349.050
164	76	77	17	38	22.944	101111.135	-0.016
165	76	77	17	38	48.825	102450.458	1349.114
166	76	77	17	38	48.825	101101.392	0.048
167	76	77	17	49	59.397	102194.435	1348.178
168	76	77	17	49	59.397	100845.369	-0.887
169	76	77	17	50	25.775	100835.626	-0.556
170	76	77	17	57	56.898	100662.196	-1.183
171	76	77	17	58	21.308	100653.202	-0.799
172	76	77	18	4	15.008	100517.398	-0.378
173	76	77	18	4	40.329	100507.203	-0.796
174	76	77	18	12	30.856	100324.180	-1.516
175	76	77	18	12	56.824	100314.736	-0.863
176	76	77	18	13	56.118	100291.503	-1.028
177	76	77	18	14	21.431	100281.609	-1.068
178	76	77	18	47	36.424	99492.406	-1.021
179	76	77	18	48	2.281	99480.464	-2.556
180	76	79	6	34	10.533	96979.564	0.581
181	76	79	6	34	35.846	96991.200	0.167
182	76	79	6	35	32.105	97018.537	0.739
183	76	79	6	35	57.418	97030.800	0.967
184	76	79	6	36	58.361	97059.159	0.368
185	76	79	6	37	23.120	97071.300	0.751
186	76	79	6	38	14.584	97095.733	0.758
187	76	79	6	38	39.905	97106.826	-0.160
188	76	79	6	39	58.584	97144.150	-0.131
189	76	79	6	40	23.897	97156.741	0.469
190	76	79	6	43	2.338	97231.600	0.375
191	76	79	6	43	27.859	97243.600	0.317
192	76	79	6	45	10.938	97291.900	-0.041
193	76	79	7	16	22.132	98164.344	0.074
194	76	79	7	16	47.541	98176.185	0.201
195	76	79	7	39	59.571	98813.094	-0.082
196	76	79	7	40	25.356	98824.726	-0.175
197	76	79	7	42	33.557	98882.946	-0.207
198	76	79	7	42	59.342	98895.238	0.376
199	76	79	7	44	10.338	98927.315	0.228

200	76	79	7	45	7.320	98951.599	-1.340
201	76	79	7	45	30.281	98963.440	0.089
202	76	79	7	49	39.322	99076.762	0.594
203	76	79	7	50	5.156	99087.405	-0.452
204	76	79	8	19	40.856	99885.902	-0.334
205	76	79	8	20	6.817	99897.594	-0.249
206	76	79	8	29	35.151	100151.000	-0.527
207	76	79	8	30	1.904	100162.760	-0.691
208	76	79	8	30	55.005	100187.193	0.080
209	76	79	8	31	20.539	100198.136	-0.354
210	76	79	8	42	18.365	100490.733	-0.398
211	76	79	8	42	44.119	100502.575	0.003
212	76	79	8	43	47.130	100531.000	0.441
213	76	79	8	44	13.884	100542.148	-0.292
214	76	79	8	51	39.054	100739.411	-0.555
215	76	79	8	52	4.382	100750.503	-0.692
216	76	79	9	15	49.909	101381.267	-0.695
217	76	79	9	16	15.056	101392.959	-0.112
218	76	79	9	17	20.685	101421.739	-0.322
219	76	79	9	17	46.663	101433.580	0.044
220	76	79	9	26	40.912	101668.917	-0.500
221	76	79	9	27	6.058	101679.860	-0.655
222	76	79	9	28	12.055	101709.839	0.197
223	76	79	9	28	38.106	101721.681	0.542
224	76	79	9	48	19.941	102241.971	-0.494
225	76	79	9	48	45.127	102253.814	0.243
226	76	79	9	59	43.050	102543.412	-0.311
227	76	79	10	0	8.587	102555.404	0.418
228	76	79	10	1	20.440	102586.282	-0.394
229	76	79	10	1	46.401	102597.974	-0.153
230	76	79	10	19	35.097	103069.398	-0.240
231	76	79	10	20	1.284	103080.940	-0.257
232	76	79	10	27	34.355	103280.902	-0.329
233	76	79	10	27	59.894	103291.994	-0.516
234	76	79	10	45	23.280	103753.822	0.201
235	76	79	10	45	48.817	103764.765	-0.151
236	76	79	10	56	34.939	104050.169	-0.662
237	76	79	10	57	0.916	104061.711	-0.622
238	76	79	11	6	33.768	104315.785	-0.308
239	76	79	11	6	58.305	104327.000	0.032
240	76	79	11	29	38.633	104930.360	-0.238
241	76	79	11	30	3.192	104941.752	0.243
242	76	79	11	39	30.327	105193.577	-0.018
243	76	79	11	39	55.864	105205.419	0.467
244	76	79	11	49	31.053	105461.292	0.431
245	76	79	11	49	56.606	105472.534	0.299
246	76	79	11	59	26.368	105726.159	0.209
247	76	79	11	59	51.906	105737.851	0.524
248	76	79	12	21	46.868	106322.596	-0.937
249	76	79	12	22	11.406	106334.440	-0.038
250	76	79	12	37	31.086	106745.003	0.205

251	76	79	12	37	56.832	106756.096	-0.191
252	76	79	12	51	14.579	107112.249	0.014
253	76	79	12	51	39.133	107123.342	0.153
254	76	79	13	8	41.398	107579.626	0.560
255	76	79	13	9	7.375	107590.718	0.074
256	76	79	13	18	53.064	107851.837	0.283
257	76	79	13	19	19.025	107863.229	0.116
258	76	79	13	41	46.928	108463.714	1.480
259	76	79	13	42	14.088	108474.956	0.675
260	76	79	13	43	45.389	108515.278	0.506
261	76	79	13	44	11.366	108526.070	-0.220
262	76	79	13	59	36.360	108936.187	0.527
263	76	79	14	0	2.339	108947.579	0.446
264	76	79	14	1	33.456	108987.751	0.388
265	76	79	14	1	59.425	108999.743	0.918
266	76	79	14	32	25.160	109800.039	-0.379
267	76	79	14	32	51.264	109811.731	-0.077
268	76	79	14	47	16.154	110187.970	0.107
269	76	79	14	47	42.288	110199.512	0.329
270	76	79	15	0	21.400	110526.136	-0.646
271	76	79	15	0	47.603	110538.427	0.380
272	76	79	15	3	52.684	110616.973	-0.549
273	76	79	15	4	18.789	110627.616	-1.103
274	76	80	13	20	38.396	80985.056	0.382
275	76	80	13	21	4.150	80955.504	0.844
276	76	80	14	8	33.104	77612.221	0.727
277	76	80	15	4	31.016	74974.197	1350.226
278	76	80	15	4	56.561	74943.619	1350.133
279	76	80	15	4	56.561	73594.500	1.014
280	76	80	15	29	20.890	73193.130	1350.182
281	76	80	15	29	46.852	73162.552	1350.688
282	76	80	15	29	46.852	71813.400	1.537
283	76	80	16	17	54.078	69696.651	1349.641
284	76	80	16	18	20.097	66817.594	-1498.137
285	76	80	16	18	20.097	68316.500	0.769
286	76	80	16	50	32.041	67341.182	1349.321
287	76	80	16	50	58.018	67309.853	1349.249
288	76	80	16	50	58.018	65960.700	0.096
289	76	80	16	52	7.089	67225.762	1348.268
290	76	80	16	52	33.072	64347.454	-1498.775
291	76	80	16	52	33.072	65846.400	0.172
292	76	80	17	26	1.316	64779.605	1349.209
293	76	80	17	27	4.404	64748.576	1394.028
294	76	80	17	27	4.404	63399.500	44.952
295	76	80	17	27	39.355	64661.337	1348.807
296	76	80	17	28	5.333	64630.158	1348.858
297	76	80	17	28	5.333	63281.000	-0.300
298	76	80	17	54	17.177	62773.394	1379.884
299	76	80	17	54	51.104	62742.215	1389.402
300	76	80	17	54	51.104	61339.100	-13.712
301	76	80	18	24	47.896	60551.182	1349.391
302	76	80	18	25	13.832	60520.603	1349.786

303	76	80	18	25	13.832	59171.500	0.683
304	76	80	18	59	58.135	58039.371	1348.271
305	76	80	19	0	24.128	55161.064	-1499.252
306	76	80	19	0	24.128	56660.000	-0.316
307	76	80	20	37	29.327	48416.033	-1496.812
308	76	80	20	37	54.858	48387.253	-1496.961
309	76	80	20	37	54.858	49886.200	1.986

Table (15) Pre-AKM Transfer Orbit referenced to a time after the second hydrazine burn.

A	=	82918.51	X	=	-17080.74
E	=	0.48212085	Y	=	34982.73
I	=	25.6832	Z	=	18334.89
NODE	=	14.3575	XD	=	-3.4280831
PERI	=	94.0499	YD	=	-1.3826542
MEAN	=	1.9900	ZD	=	-0.2353574
MJD	=	42858	H/M/S	=	01/57/42.000

TAG	YR	DY	HR	MN	SEC	RANGE	O-C
1	76	81	3	19	40.299	38938.844	0.119
2	76	81	4	46	14.055	43341.296	1.292
3	76	81	4	46	40.920	43367.828	0.244
4	76	81	5	46	36.349	47382.349	1.404
5	76	81	5	47	2.356	47413.077	1.096
6	76	81	6	16	28.396	49570.084	-2.402
7	76	81	6	16	54.373	48253.246	-1351.736
8	76	81	6	16	54.373	49602.312	-2.670
9	76	81	6	18	57.624	49755.955	-3.468
10	76	81	6	18	57.624	51254.918	1495.495
11	76	81	6	19	22.161	49788.333	-1.887
12	76	81	6	19	22.161	51287.295	1497.075
13	76	81	7	1	31.274	53044.529	0.730
14	76	81	7	1	56.796	53077.506	0.172
15	76	81	7	2	47.561	53143.910	-0.164
16	76	81	7	2	47.561	54492.976	1348.902
17	76	81	7	3	13.545	53177.187	-1.067
18	76	81	7	4	16.017	53261.129	0.648
19	76	81	7	4	16.017	54610.195	1349.714
20	76	81	7	4	41.554	53293.956	-0.158
21	76	81	7	5	44.134	53378.048	1.464
22	76	81	7	6	10.127	53411.924	1.065
23	76	81	7	19	40.358	54486.231	1.126
24	76	81	7	20	6.335	54520.707	0.982
25	76	81	7	45	46.062	56589.425	0.333
26	76	81	7	46	12.024	56623.751	-0.499
27	76	81	8	7	0.163	58324.773	1.581

28	76	81	8	7	26.144	58359.800	1.077
29	76	81	8	21	49.800	59543.280	0.149
30	76	81	8	22	14.327	59578.000	1.146
31	76	81	8	30	9.619	60231.903	0.689
32	76	81	8	30	35.565	60267.200	0.220
33	76	81	8	40	14.416	61065.326	-0.687
34	76	81	8	40	40.377	61101.901	0.006
35	76	81	8	49	17.868	61817.655	-0.249
36	76	81	8	49	43.069	61853.331	0.524
37	76	81	9	10	4.918	63547.158	-1.029
38	76	81	9	10	29.450	63581.600	-0.681
39	76	81	9	20	16.159	64398.419	0.287
40	76	81	9	20	42.392	64434.544	-0.086
41	76	81	9	30	0.613	65211.006	-0.570
42	76	81	9	30	25.150	65246.900	1.161
43	76	81	9	40	2.534	66049.526	-0.294
44	76	81	9	40	27.056	66084.600	0.623
45	76	81	9	47	19.405	66659.004	0.632
46	76	81	9	47	45.555	66694.500	-0.300
47	76	81	10	10	25.823	68587.119	-2.195
48	76	81	10	10	50.344	68624.100	0.650
49	76	81	10	19	59.905	69387.565	-0.665
50	76	81	10	20	24.449	69422.641	0.267
51	76	81	10	29	38.005	70193.107	1.047
52	76	81	10	30	4.921	70228.600	-0.867
53	76	81	10	39	34.048	71018.436	-1.481
54	76	81	10	40	0.025	71056.000	0.027
55	76	81	10	51	19.169	71998.753	0.973
56	76	81	10	51	45.152	72032.900	-0.877
57	76	81	11	11	29.192	73671.899	0.769
58	76	81	11	11	55.154	73707.154	0.196
59	76	81	11	19	59.178	74373.413	-0.867
60	76	81	11	20	25.140	74410.588	0.550
61	76	81	11	27	31.406	74995.783	-0.818
62	76	81	11	27	56.943	75030.700	-1.008
63	76	81	11	46	1.784	76519.028	-0.299
64	76	81	11	46	26.322	76553.800	0.914
65	76	81	11	49	59.156	76844.453	0.664
66	76	81	11	50	25.134	76878.779	-0.495
67	76	81	12	18	21.829	79159.300	0.542
68	76	81	12	18	47.790	79193.626	-0.247
69	76	81	12	20	20.604	79319.090	-0.280
70	76	81	12	20	46.629	79355.365	0.820
71	76	82	9	34	22.112	121315.816	-1.848
72	76	82	9	34	48.097	121319.669	5.387
73	76	82	9	49	0.339	121200.846	-2.526
74	76	82	9	49	25.055	121198.898	-1.260
75	76	82	10	0	8.306	121116.455	-0.211
76	76	82	10	0	34.347	121112.257	-1.036
77	76	82	10	13	8.609	121016.624	0.706
78	76	82	10	13	34.593	121013.326	0.749

Table (16) Solution for post-AKM coast. (SR11A)

A	=	121483.82	X	=	35715.93
E	=	0.0133391	Y	=	-105579.61
I	=	25.1481	Z	=	-52211.00
NODE	=	14.9160	XD	=	1.7007914
PERI	=	103.2368	YD	=	0.5490189
MEAN	=	169.9170	ZD	=	0.0435333
MJD	=	42859	H/M/S	=	10/40/0

Table (17) Range magnitude data for the post-AKM coast. (SR11A)

TAG	YR	DY	HR	MN	SEC	RANGE	O-C
1	76	82	10	51	48.344	120737.817	-0.931
2	76	82	10	52	13.888	120735.268	-0.848
3	76	82	12	56	18.777	120342.240	0.320
4	76	82	12	56	43.323	120341.941	0.045
5	76	82	13	12	34.602	120347.637	0.056
6	76	82	13	13	0.564	120348.236	0.320
7	76	82	14	49	54.170	120661.070	-0.265
8	76	82	14	50	20.276	120662.869	-0.920
9	76	82	15	4	21.841	120747.410	-0.271
10	76	82	15	4	46.337	120749.658	-0.604
11	76	82	15	43	2.582	121026.667	0.660
12	76	82	15	43	27.135	121030.565	1.257
13	76	82	15	58	51.436	121159.475	0.658
14	76	82	15	59	17.430	121163.372	0.768
15	76	82	16	45	5.024	121605.993	0.603
16	76	82	16	45	30.571	121609.741	-0.143
17	76	82	17	4	53.122	121821.994	0.810
18	76	82	17	5	19.265	121826.648	0.563
19	76	83	13	27	50.112	120049.920	-0.876
20	76	83	13	28	16.339	120041.377	-1.015
21	76	83	13	29	17.637	120023.261	0.490
22	76	83	13	50	57.352	119714.003	100.595
23	76	83	13	51	23.566	119704.560	99.269
24	76	83	16	39	0.116	119930.476	2848.222
25	76	83	16	39	26.093	119926.420	2848.794
26	76	83	19	8	34.640	116199.987	1.171
27	76	83	19	9	0.643	116199.837	1.429
28	76	83	19	41	45.356	116204.952	1.664
29	76	83	19	42	10.943	116205.685	1.872
30	76	83	19	43	31.162	116207.201	1.666
31	76	83	19	43	57.107	116207.931	1.813
32	76	83	20	40	40.320	116387.376	2.991

33	76	83	20	41	6.288	116389.175	2.675
34	76	83	22	29	56.846	117240.416	1.973
35	76	83	22	30	22.776	117239.988	-2.968
36	76	83	22	31	24.412	117254.375	0.658
37	76	83	22	31	50.569	117259.453	1.157
38	76	83	22	58	35.085	117555.198	1.695
39	76	83	22	59	1.095	117559.845	1.336
40	76	83	22	59	55.147	117567.789	-1.145
41	76	83	23	0	20.052	117573.785	0.038
42	76	83	23	37	41.900	118031.868	1.718
43	76	83	23	38	6.448	118034.114	-1.266
44	76	84	0	7	2.105	118571.195	155.302
45	76	84	0	7	28.126	118577.790	156.045
46	76	84	18	1	58.809	117573.336	2.812
47	76	84	18	2	23.403	117567.490	3.554
48	76	84	19	17	31.177	116395.151	0.023
49	76	84	19	17	57.146	116388.556	-0.129
50	76	84	20	42	15.366	115288.809	61.735
51	76	84	20	42	41.128	115233.413	11.702
52	76	84	21	41	3.800	114559.373	2.725
53	76	84	21	41	28.324	114555.026	2.557
54	76	84	23	4	22.104	113863.255	1.162
55	76	84	23	4	48.088	113862.057	2.699
56	76	85	0	30	1.088	113516.245	1.690
57	76	85	0	30	27.047	113516.245	2.427
58	76	85	5	0	27.143	115076.985	-0.084
59	76	85	5	0	53.644	115081.033	-1.633
60	76	85	6	11	14.311	116074.095	2.102
61	76	85	6	11	40.337	116080.841	2.198
62	76	85	7	24	40.080	117274.164	0.970
63	76	85	7	25	6.169	117282.259	1.569
64	76	85	8	23	12.146	118310.247	1.246
65	76	85	8	23	37.878	118317.592	0.950
66	76	86	1	7	33.829	119733.084	-44.489
67	76	86	1	7	33.829	119760.343	-17.230
68	76	86	1	7	59.825	119768.588	-2.915
69	76	86	1	7	59.825	119755.846	-15.657
70	76	86	2	16	20.153	118866.062	-2.339
71	76	86	2	16	20.153	118854.071	-14.330

Table (18) Solution for post-AKM coast. (SR11B)

A =	121553.30	X =	35731.60
E =	0.0127867	Y =	-105541.39
I =	25.1794	Z =	-52259.67
NODE =	14.7547	XD =	1.7017354
PERI =	105.6393	YD =	0.5480135
MEAN =	167.6265	ZD =	0.0453877
MJD =	42859	H/M/S =	10/40/00

Table (19) Range magnitude data for the post-AKM coast. (SR11B)

TAG	YR	DAY	HR	MN	SEC	RANGE	O-C
1	76	82	14	18	35.144	120511.323	-1.413
2	76	82	14	19	0.650	120513.572	-0.931
3	76	82	14	22	18.939	120529.311	0.767
4	76	82	14	22	44.901	120530.810	0.388
5	76	82	16	33	6.589	121493.444	1.437
6	76	82	16	33	32.552	121496.891	0.486
7	76	83	13	52	28.904	119686.422	-1.497
8	76	83	13	52	54.884	119673.778	-1.151
9	76	84	0	29	48.022	118885.000	-0.479
10	76	84	0	30	13.544	118890.773	-0.651
11	76	84	1	9	41.375	119453.334	0.288
12	76	84	1	10	6.920	119458.731	-0.456
13	76	84	1	11	23.514	119477.767	0.156
14	76	84	1	11	48.070	119483.763	0.243
15	76	84	1	20	0.142	119604.429	2.300
16	76	84	1	56	1.055	120124.569	-0.464
17	76	84	1	56	26.592	120132.514	1.306
18	76	84	17	26	48.127	118278.769	0.703
19	76	84	17	27	14.382	118271.874	0.958
20	76	84	17	29	17.408	118239.046	1.623
21	76	84	17	29	39.828	118230.952	-0.369
22	76	84	18	31	27.901	117236.690	0.012
23	76	84	18	31	53.016	117230.245	0.159
24	76	84	20	11	27.913	115760.212	-0.299
25	76	84	20	11	52.402	115754.816	-0.185
26	76	84	21	16	13.573	114955.270	-0.656
27	76	84	21	16	39.329	114951.223	0.121
28	76	84	22	21	12.821	114313.864	-0.013
29	76	84	22	21	37.387	114308.318	-2.110
30	76	84	23	22	9.080	113891.606	0.714
31	76	84	23	22	35.049	113887.109	-1.452
32	76	85	0	27	47.402	113653.721	1.114
33	76	85	0	28	12.073	113653.273	1.412
34	76	85	1	54	19.855	113706.035	-0.716
35	76	85	1	54	44.386	113707.683	-0.335
36	76	85	3	20	28.085	114181.206	1.441
37	76	85	3	20	53.655	114183.004	-0.106
38	76	85	4	24	37.857	114786.487	-1.020
39	76	85	4	25	2.436	114793.382	1.347
40	76	85	5	37	10.892	115704.751	0.847
41	76	85	5	37	36.903	115710.747	0.725
42	76	85	6	34	55.803	117264.871	689.282
43	76	85	6	35	20.386	116580.445	-1.699
44	76	85	8	26	18.892	118488.174	-1.300
45	76	85	8	26	44.885	118495.669	-1.598
46	76	86	1	21	18.370	119566.977	-0.122
47	76	86	1	21	18.370	119554.985	-12.114

48	76	86	1	21	44.353	119562.481	1.352
49	76	86	1	21	44.353	119550.489	-10.641
50	76	86	2	22	52.340	118756.788	-10.775
51	76	86	2	22	52.372	118768.780	1.223
52	76	86	2	23	18.362	118763.234	0.892
53	76	86	2	23	18.362	118751.242	-11.100
54	76	86	3	14	46.057	118182.985	-11.168
55	76	86	3	15	12.024	118178.939	-10.905
56	76	86	3	17	25.905	118156.004	-11.762
57	76	86	3	17	51.041	118151.957	-11.689
58	76	86	6	14	45.027	117260.074	-13.002
59	76	86	6	15	11.006	117259.924	-13.298
60	76	86	7	11	19.089	117381.490	-13.536
61	76	86	7	11	45.042	117382.689	-14.068

Table (20) Solution for post-HB3 coast. (SR11A)

A	=	122729.54	X	=	41814.17
E	=	0.0035917	Y	=	-103476.49
I	=	25.1741	Z	=	-52037.14
NODE	=	14.7574	XD	=	1.6789547
PERI	=	117.5648	YD	=	0.6322660
MEAN	=	158.8512	ZD	=	0.0863569
MJD	=	42864	H/M/S	=	8/41/00

Table (21) Range magnitude data for post-HB3 coast. (SR11A)

TAG	YR	DY	HR	MN	SEC	RANGE	O-C
1	76	87	9	52	14.824	121470.809	-3.541
2	76	87	9	52	40.791	121467.362	-2.501
3	76	87	10	55	15.345	120898.505	-1.300
4	76	87	10	55	40.135	120896.707	0.115
5	76	87	11	57	14.264	120504.408	-2.040
6	76	87	11	57	39.809	120502.481	-1.906
7	76	87	13	3	46.670	120297.721	0.804
8	76	87	13	4	12.648	120297.122	0.811
9	76	87	14	26	25.609	120361.277	-1.350
10	76	87	14	26	51.566	120361.877	-2.053
11	76	87	16	38	59.297	121203.095	-2.478
12	76	87	16	39	24.826	121210.889	1.270
13	76	87	17	37	32.635	121830.110	-1.631
14	76	87	17	37	58.889	121837.755	0.842
15	76	88	12	56	47.164	121759.695	2.746
16	76	88	12	57	13.314	121751.265	2.563
17	76	88	12	58	8.178	121734.477	3.068
18	76	88	12	58	34.330	121606.915	-116.256
19	76	88	15	8	53.395	119799.916	311.858
20	76	88	15	9	19.393	119485.731	4.074
21	76	88	22	53	27.300	119025.852	2.413

22	76	88	22	53	53.392	119029.149	0.414
23	76	89	2	8	14.040	121866.235	1.932
24	76	89	2	8	39.619	121872.831	1.865
25	76	89	17	59	9.542	120617.300	2.128
26	76	89	17	59	35.617	120608.456	0.231
27	76	89	20	39	58.872	118229.453	3.092
28	76	89	20	40	21.627	118224.207	2.773
29	76	89	22	12	31.370	117180.479	1.295
30	76	89	22	12	56.921	117117.331	-57.838
31	76	89	23	1	7.945	116777.558	2.053
32	76	89	23	1	33.056	116773.511	0.986
33	76	90	0	45	32.793	116320.674	1.424
34	76	90	1	12	16.404	116301.616	1.672
35	76	90	1	12	42.398	116302.516	2.546
36	76	90	4	48	3.311	117616.078	2.101
37	76	90	4	48	29.312	117620.874	1.837
38	76	90	4	53	32.352	117516.247	-162.380
39	76	91	0	56	7.316	121077.010	-1.222
40	76	91	0	56	33.296	121072.064	0.519
41	76	91	1	11	11.415	120850.067	2.373
42	76	91	1	11	37.393	120840.923	-0.217
43	76	91	1	15	24.364	120787.710	3.663
44	76	91	1	15	50.566	120779.765	2.288
45	76	91	1	17	48.870	120750.386	2.520
46	76	91	1	18	13.416	120745.289	3.555
47	76	91	2	49	40.686	119498.453	1.172
48	76	91	2	50	6.841	119491.857	-0.225
49	76	91	3	53	48.437	118827.239	-0.384
50	76	91	3	54	14.415	118823.042	-0.761
51	76	91	3	55	47.088	118809.701	-0.556
52	76	91	3	56	13.157	118804.155	-2.314
53	76	91	5	10	2.854	118319.691	0.628
54	76	91	5	10	27.438	118318.791	1.524
55	76	91	6	35	22.085	118174.741	-1.057
56	76	91	6	35	48.113	118176.090	-0.185
57	76	91	8	14	38.275	118602.395	2.534
58	76	91	8	15	3.849	118604.344	1.335
59	76	91	9	37	46.816	119409.886	0.683
60	76	91	9	38	12.802	119414.683	0.302
61	76	91	10	32	12.909	120129.388	1.600
62	76	91	10	32	38.902	120135.534	1.518
63	76	92	9	17	16.512	121790.238	1.635
64	76	92	9	17	41.114	121785.291	1.325
65	76	92	10	34	50.653	121021.270	0.063
66	76	92	10	35	16.647	121017.973	0.376
67	76	92	11	27	42.289	120644.581	1.470
68	76	92	11	28	8.072	120643.832	3.260
69	76	92	12	33	7.880	120364.425	1.738
70	76	92	12	33	32.626	120362.626	1.010
71	76	92	13	38	28.089	120308.514	2.552
72	76	92	15	10	44.111	120618.199	2.875

73	76	92	15	11	10.089	120620.298	2.469
74	76	92	17	30	5.386	121876.728	2.226
75	76	92	17	30	31.617	121883.474	3.749
76	76	93	12	49	8.044	121607.065	-0.796
77	76	93	12	49	34.025	121598.522	-1.084
78	76	93	14	30	17.642	119807.411	-0.769
79	76	93	14	30	42.181	119801.715	0.128
80	76	93	17	2	50.256	117919.318	-0.416
81	76	93	17	3	18.938	117912.872	-3.025
82	76	93	19	14	47.384	117424.960	-2.334
83	76	93	19	15	12.098	117425.410	-2.152
84	76	93	20	7	37.049	117548.025	-3.208
85	76	93	20	8	2.589	117550.123	-2.828
86	76	93	20	9	9.006	117554.171	-3.299
87	76	93	20	9	34.552	117555.670	-3.559
88	76	94	18	41	48.089	119868.418	-7.075
89	76	94	18	42	13.618	119863.022	-5.982
90	76	94	20	0	8.120	118729.207	-5.350
91	76	94	20	0	34.080	118723.661	-4.968
92	76	94	22	26	40.543	117082.747	-6.374
93	76	94	22	27	6.537	117080.498	-5.058
94	76	95	1	0	17.880	116430.698	-5.780
95	76	95	2	52	22.149	116796.445	-5.934
96	76	95	2	52	48.129	116797.494	-7.691
97	76	95	5	16	34.401	118263.030	-7.337
98	76	95	5	16	59.811	118269.475	-6.618
99	76	95	7	13	13.641	120064.333	-8.454
100	76	96	4	5	56.682	118826.040	3.250
101	76	96	4	6	22.624	118822.443	2.998
102	76	96	6	4	42.890	118330.933	3.454
103	76	96	6	5	8.835	118332.282	4.984
104	76	96	8	5	7.455	118743.447	3.894
105	76	96	8	5	33.433	118748.244	5.583
106	76	96	10	6	5.368	120011.120	7.070
107	76	96	10	6	30.908	120016.066	6.313
108	76	97	9	14	23.514	121537.213	10.996
109	76	97	9	14	48.828	121531.817	10.350
110	76	97	11	18	1.021	120447.168	11.897
111	76	97	11	18	26.540	120445.946	13.258
112	76	97	13	28	2.684	120093.263	13.031
113	76	97	13	28	28.665	120093.564	13.001
114	76	97	14	40	2.891	120289.455	13.676
115	76	97	14	40	28.853	120291.276	13.478
116	76	98	14	21	3.787	119788.224	-7.574
117	76	98	14	21	28.311	119782.378	-6.902
118	76	98	16	12	20.922	118304.251	-6.900
119	76	98	16	12	46.916	118298.555	-8.102
120	76	98	22	5	7.174	117193.970	-1355.541
121	76	99	0	38	34.413	120588.070	-6.873
122	76	99	0	39	0.546	120594.666	-6.840
123	76	99	1	52	53.659	121725.033	-7.680
124	76	99	2	8	34.039	121966.516	-6.342
125	76	99	2	8	59.635	121972.662	-6.712

Table (22) Solution for post-HB3 coast. (SR11B)

A	= 122647.62	X	= 42219.19
E	= 0.0042631	Y	= -103300.54
I	= 25.1859	Z	= -52047.49
NODE	= 14.8762	XD	= 1.6762578
PERI	= 117.5897	YD	= 0.6377430
MEAN	= 158.8965	ZD	= 0.0874778
MJD	= 42864	H/M/S	= 8/41/00

Table (23) Range magnitude data for post-HB3 coast. (SR11B)

TAG	YR	DAY	HR	MN	SEC	RANGE	O-C
1	76	87	9	48	52.597	121490.895	-0.718
2	76	87	9	49	17.328	121486.249	-1.086
3	76	87	11	1	26.645	120841.095	-1.837
4	76	87	11	1	51.166	120838.097	-1.818
5	76	87	11	52	26.372	120527.064	0.175
6	76	87	11	52	55.415	120524.514	0.024
7	76	87	13	7	3.110	120319.156	21.055
8	76	87	13	7	29.132	120319.756	22.135
9	76	87	14	21	47.007	120362.177	-0.959
10	76	87	14	22	12.605	120363.226	-1.135
11	76	87	16	33	48.576	121183.758	1.329
12	76	87	16	34	13.343	121187.505	1.216
13	76	87	16	35	51.361	121201.296	-0.343
14	76	87	16	36	17.345	121207.591	1.863
15	76	87	17	32	56.866	121807.776	1.097
16	76	87	17	33	21.405	121813.172	1.708
17	76	87	17	35	1.347	121830.710	-0.305
18	76	87	17	35	27.311	121837.905	1.796
19	76	88	13	59	8.104	120563.487	2.650
20	76	88	23	0	59.786	119056.880	2.845
21	76	88	23	1	24.329	119062.426	3.304
22	76	89	21	37	6.301	117368.299	-0.745
23	76	90	1	8	31.080	116134.032	-0.800
24	76	90	1	8	57.093	116134.482	-0.295
25	76	91	1	1	30.832	120880.645	0.424
26	76	91	1	1	56.348	120874.500	0.772
27	76	91	1	3	8.860	120855.913	0.618
28	76	91	1	3	34.838	120849.317	0.619
29	76	91	2	52	26.360	119357.250	0.049
30	76	91	2	52	52.331	119352.454	0.334
31	76	91	4	0	13.625	118669.399	0.225
32	76	91	4	0	39.595	118666.101	0.567
33	76	91	5	5	28.797	118241.445	0.385
34	76	91	5	5	54.922	118239.647	0.584
35	76	91	6	42	20.175	118093.048	0.996
36	76	91	6	42	46.272	118093.347	0.584

37	76	91	8	23	12.128	118579.311	0.993
38	76	91	8	23	38.144	118582.609	0.842
39	76	91	9	49	43.838	119475.241	1.481
40	76	91	9	50	8.416	119480.337	1.418
41	76	91	10	35	3.591	120089.815	-0.165
42	76	91	10	35	29.578	120097.460	1.176
43	76	91	10	37	26.874	120126.240	1.414
44	76	91	10	37	52.870	120132.386	1.213
45	76	92	9	23	9.910	121724.584	-1.188
46	76	92	9	23	35.889	121719.937	-1.027
47	76	92	10	29	34.340	121067.738	-1.586
48	76	92	10	30	0.094	121064.441	-1.210
49	76	92	11	36	52.641	120593.017	-1.236
50	76	92	11	37	17.304	120590.319	-1.685
51	76	92	12	29	21.080	120374.918	-0.972
52	76	92	12	29	46.396	120373.569	-1.137
53	76	92	13	45	34.360	120314.510	-1.085
54	76	92	13	46	0.118	120315.109	-1.029
55	76	92	15	2	7.147	120565.736	-2.332
56	76	92	15	2	32.684	120567.834	-2.500
57	76	92	17	34	17.648	121919.898	-2.216
58	76	92	17	34	43.645	121927.093	-0.252
59	76	93	12	55	3.281	121451.473	0.956
60	76	93	12	55	29.151	121441.579	-0.732
61	76	93	14	20	41.296	119912.638	-0.715
62	76	93	14	21	6.848	119906.429	0.138
63	76	93	17	5	58.370	117830.279	2.152
64	76	93	17	6	24.331	117827.132	2.435
65	76	93	19	10	1.514	117350.012	4.116
66	76	94	18	38	18.832	119785.076	2.391
67	76	94	18	38	44.825	119778.181	2.133
68	76	94	19	54	52.846	118659.955	1.659
69	76	94	19	55	18.806	118652.160	-0.132
70	76	94	22	23	47.672	116957.733	1.083
71	76	94	22	24	13.769	116954.736	1.751
72	76	95	1	3	14.181	116269.710	0.662
73	76	95	1	3	40.158	116269.710	0.607
74	76	95	2	50	12.119	116612.223	-0.764
75	76	95	2	50	38.109	116614.321	-1.390
76	76	95	5	11	55.355	118023.196	-1.466
77	76	95	5	12	20.862	118030.241	-0.072
78	76	95	7	18	13.284	119970.948	-1.247
79	76	95	7	18	39.296	119979.641	0.039
80	76	95	7	20	51.070	120017.265	0.089
81	76	95	7	21	17.285	120025.360	0.700
82	76	96	4	10	6.093	118721.562	-2.255
83	76	96	4	10	31.407	118718.565	-2.080
84	76	96	5	52	41.307	118263.330	-3.639
85	76	96	5	53	7.614	118263.481	-2.948
86	76	96	8	10	4.790	118706.123	6.878
87	76	96	8	10	38.825	118702.975	-0.485

88	76	96	10	1	8.146	119957.626	0.138
89	76	96	10	1	33.686	119862.872	-0.219
90	76	97	9	18	8.168	121514.279	1.704
91	76	97	9	18	34.020	121510.382	2.642
92	76	97	11	13	9.889	120481.044	0.159
93	76	97	11	13	34.417	120478.946	0.693
94	76	97	13	50	6.857	120106.154	1.950
95	76	97	13	50	32.885	120107.503	2.503
96	76	97	14	45	39.923	120293.075	3.912
97	76	97	14	46	5.887	120295.623	4.369
98	76	98	12	45	13.048	121446.526	2.475
99	76	98	12	45	39.019	121439.931	4.066
100	76	98	14	16	35.925	119817.154	-4.753
101	76	98	14	17	1.912	119811.608	-3.254
102	76	98	16	15	30.637	118210.566	-2.891
103	76	98	16	15	56.616	118205.919	-3.047
104	76	98	21	48	27.088	118242.344	-1.880
105	76	99	0	44	12.449	120520.917	-0.392
106	76	99	0	44	38.300	120527.512	-0.266
107	76	99	1	58	25.552	121648.286	-3.197
108	76	99	2	0	31.302	121681.413	-2.029
109	76	99	2	1	47.357	121702.099	-0.663
110	76	99	2	2	12.196	121707.196	-1.874

As of July 1, 1977 the orbits for both SOLRAD 11A and SOLRAD 11B are listed below. The two satellites are in their final configurations nearly 180° apart.

Table (24) Orbital Elements for SR11A on July 1, 1977

A	= 125154.3660	X	= -25186.54390
E	= 0.00758780	Y	= 107802.61600
I	= 27.1049	Z	= 56588.50950
NODE	= 10.2113	XD	= -1.75178682
PERI	= 122.6350	YD	= -0.39585865
MEAN	= 330.4147	ZD	= -0.04045244
MJD	= 43325	H/M/S	= 00/00/00.000

Table (25) Orbital Elements for SR11B on July 1, 1977

A	= 125231.0700	X	= 26849.90660
E	= 0.00386447	Y	= -108788.07300
I	= 26.9783	Z	= -56941.97580
NODE	= 10.5011	XD	= 1.73073720
PERI	= 111.1721	YD	= 0.40311423
MEAN	= 161.6871	ZD	= 0.04119660
MJD	= 43325	H/M/S	= 00/00/00.000

SUBROUTINE DESCRIPTIONS

PROGRAM LOAD

This program allows a user to input a state vector to disk storage in the proper format to be later read as input to ENTER, GENER, and DIFFCR. See Sect. (V).

PROGRAM ENTER

The range data may be entered into disk storage in its proper formant by this program. The range observations are later used as input to DIFFCR. See Sect. (VI).

PROGRAM GENER

Ephemeris generation, state vector updates, rise and set times, and eclipse times are determined by this program. See Sect. (VII).

PROGRAM DIFFCR

The orbit determination program is located here. See Sect. (VIII).

SUBROUTINE AZEL (R, T, AZ, ELV, IND, KEY)

The azimuth AZ and elevation ELV of a satellite with position R at time T is computed. Key is 1 for ELV greater than zero and KEY is zero for ELV less than zero. The station key is IND. Blossom Pt: IND = 1, ARCETRI, ITALY: IND = 2, AMEDABAD, INDIA: IND = 3.

SUBROUTINE PAPER (X, T)

The state vector X at time T is listed on the line printer. Both Keplerian and Cartesian version is listed.

SUBROUTINE BIGLET (XMB, XML)

The 7 x 7 covariance XMB is reduced in size to XML whose size is equal to the number of parameters free in the differential corrections process. XML is generally a 6 x 6 square array since the bias parameter is usually held fixed in the orbit determination process. The state error covariance matrix is with respect to the coordinate set (n, ξ, η, i, Ω, U, BIAS).

SUBROUTINE LITBIG (XML, XMB)

The covariance XML is increased in size to XMB (7 x 7).

SUBROUTINE INOD (X, T, IDEN, NOBS, RMS, BIAS, LUIN, WM, LF, LR)

INOD represents an input subroutine to the orbit determination program. The state vector $\vec{X}$ is read from the disk file into core storage.

SUBROUTINE OUTOD (X, T, IDEN, NOBS, RMS, BIAS, LUOUT, WM, KSTART, KSTOP, NPRINT, LF, LR)

Similar to INOD except the data is read from core storage out to the disk for permanent storage.

SUBROUTINE RTTIME (TA, TB)

From the time TA, the time one minute from the nearest hour is computed. This time is denoted as TB. For the SOLRAD Project all epochs are referenced one minute past the nearest hour to the beginning or end of the data span. This is for convience only and thus not required for any phase of the project.

SUBROUTINE FANDG (X, Z, T, TN)

The state vector $\vec{X}$ is propagated through time (T-TN) via the standard f and g series. The resulting state vector is stored as Z.

SUBROUTINE FORCE (K2, K3, K4, KDRAG, KSUN, KMOON)

This is a data initialization subroutine. K2, K3, K4, KDRAG, KSUN, KMOON are keys to include or exclude the perturbations J2, J3, J4, drag, Sun and Moon respectively. If the keys are set to one the respective perturbations are included and if set to zero they are excluded.

SUBROUTINE DEGRAD (A, B)

The Keplerian state vector A (a, e, i, Ω, ω, M) is converted from degrees to radians. The result is B. The first two components remain unchanged.

SUBROUTINE RADDEG (A, B)

Opposite to DEGRAD. The Keplerian state vector is converted from radians to degrees. The first two components remain unchanged.

SUBROUTINE ELRELN (ELR, ELN)

The Keplerian state vector ELR as defined above is converted to a non-singular set of elements ELN, i.e., n, e cos ω, e sin ω, i, Ω, M+ω, where n is the mean motion. Both ELR and ELN are espressed in radians.

SUBROUTINE ELNELR (ELN, ELR)

Opposite to ELRELN. The non-singular set of elements (for near-zero eccentricity) are converted to the Keplerian set which are expressed in radians.

SUBROUTINE FOUR (X, TSTART, TSTOP, INT, DT)

The driving subroutine for the numerical integrator RUK. $\vec{X}$ is the Cartesian state vector, TSTART the starting time, TSTOP the stop time, INT the print interval, and DT the step size. The print interval is thus INT·DT.

SUBROUTINE RUK (X)

A fourth order Runge Kutta numerical integrator.

SUBROUTINE DFQ (X, DX)

DFQ computes the derivatives $\vec{DX}$ which are required by the subroutine RUK.

SUBROUTINE DIFFEQ (D, U, V, W, UD, VD, WD, UDD, VDD, WDD)

The accelerations due to the various perturbing forces are computed by this subroutine. D is the time; U,V,W are the inertial position components; UD, VD, WD are the inertial velocity components; and UDD, VDD, WDD are the accelerations.

FUNCTION STPSZE (X, STEPCN)

The step size required by the numerical integrator is computed by this subroutine. STEPCN is the number of radians per step - usually set to 0.05.

SUBROUTINE ORBIT (ELD, X, T, TN)

The Keplerian state vector ELD is propagated analytically through the time period T-TN. Perturbations due to J_2 only are included. The perturbations include only those terms not involving eccentricity. It is therefore valid only for low altitude, near circular orbits and for short time spans. ELD is expressed in degrees.

SUBROUTINE STATEL (X, ELD)

The Keplerian state vector ELD is computed from the Cartesian state vector $\vec{X}$. The units for both $\vec{X}$ and ELD are degrees and kilometers.

SUBROUTINE ELSTAT (ELD, X)

The Cartesian state vector $\vec{X}$ is computed from the Keplerian state vector ELD. The units are in kilometers and degrees.

FUNCTION ARKTNS (N, X, Y)

A four quadrant inverse tangent subroutine. If N = 180 a value between the range -180^o to $+180^o$ is returned. If N = 360 the range is 0^o to 360^o.

SUBROUTINE POSION (T, X1, Y1, Z1)

The geodetic station location (X1, Y1, Z1) of Blossom Point, Maryland is computed at time T. The position is respect to inertial space. (True Eq. of Date.)

SUBROUTINE PLACE (T, X1, Y1, Z1, XLAT, XLON)

The geocentric latitude and longitude XLAT and XLON are computed for a satellite at position S1, Y1, Z1 at time T. T is the total number of seconds measured from January 1, 0 hours 1970.

SUBROUTINE GHA70 (TSEC, TDAY,GHAN, DA, OMEGA)

The Greenwich hour angle GHAN is computed from the time TDAY and TSEC. TDAY represents the number of whole days from Jan. 1, 0 hours 1970. TSEC represents the number of seconds of the fractional day. For example for 0200 June 1, 1977. The modified Julian date (MJD) is 43295, TDAY = 2708, TSEC - 7200.

SUBROUTINE SUN (T, XS)

The position of the sun XS is given at time T. This subroutine calls the additional subroutine SUNPS (XJD, XS) for the solar position. For computational efficiency SUN merely interpolates the values obtained from SUNPS. Although SUN may be called thousands of time over an integration span, SUNPS may be called no more than several times.

SUBROUTINE MOON (T, XM)

The lunar positions XM is computed at time T. Moon is also an interpolation routine which calls an additional routine MOONPS.

FUNCTION FASTRG (ELN, T, TN)

The range of the satellite from Blossom Point is computed at time T. The satellite state vector is ELN at epoch time TN. The state vector is propagated analytically including no perturbations. This routine is used to compute the tracking covariance matrix $(H^T R^{-1} H)^{-1}$

where H is the sensitivity matrix and R is the observational variance matrix. ELN represents the state vector by the non-singular set of elements (n, ξ, η, i, Ω, U). For computational speed the routine is written in single precision. The units for ELN are kilometers and degrees.

SUBROUTINE FLSTAT (ELR, X)

Similar to ELSTAT but the computations are performed in single precision. ELR represents the Keplerian state vector in radians.

SUBROUTINE DERIV (ELN, TIME, TN, H, NR, NC)

The sensitivity matrix H is computed for each of the times given in the array TIME. H is a NR × NC array, where NR is the number of data points and NC is the number of state variables to be determined in the solution. NC can be from one to seven (usually six). H_{ij} is the sensitivity of the i^{th} range value with respect to a change in the j^{th} component of the state vector. ELN is the state vector with epoch TN.

SUBROUTINE REDUCE (KA, KB, KC, KD, KE, KF, KG, NC, NUMBER)

The number of free parameters, NUMBER, represented by the number of non-zero numbers in the calling sequence KA, KB, ... KG is determined.

SUBROUTINE ERASE (DX, DE, NC)

The full 7 × 1 array DE is determined from DX the correction in the state vector due to the observations. DE represents the full correction. The components of DE which were held fixed in the differential corrections process are set to zero.

SUBROUTINE ORDER (Y, TIME, KMAX, TMAX)

The set of observations Y are ordered in ascending order with respect to TIME. The length of TIME and Y is given by KMAX. The time of the latest observation is given by TMAX.

SUBROUTINE OUT (A, NR, NC)

The array A is listed on the line printer. The number of rows is NR and the number of columns is given by NC.

SUBROUTINE TRAFER (A, B, NR, NC)

The array A is copied into B. The dimensions of both are NR x NC.

SUBROUTINE TRNPSE (A, B, NR, NC)

The transpose of A is computed and stored in B. The dimensions of A are $NR \times NC$. The dimensions of B are $NC \times NR$.

SUBROUTINE ADD (A, B, C, NR, NC)

The sum of A and B is stored into C; i.e., $C = A + B$. The dimensions of A, B, and C are $NR \times NC$.

SUBROUTINE SUB (A, B, C, NR, NC)

The difference of A and B is stored into C; i.e., $C = A - B$.

SUBROUTINE MULT (A, B, C, NR, NS, NC)

The matrix product of A and B is computed and stored into C. The dimensions of A are $NR \times NS$. The dimensions of B are $NS \times NC$ and the dimensions of the result are $NR \times NC$. In other words $C = AB$.

SUBROUTINE IVERSE (BSAVE, B, NX)

This is a matrix inversion subroutine. The inverse of BSAVE is computed and stored into B. The dimensions of each are $NX \times NX$. To reduce the possibility of underflow and/or overflow on the PDP-10 the matrix is scaled before the inverse is carried out.

SUBROUTINE SHADOW (T, X)

The times for passage into and out of the lunar and earth penumbra and umbra are computed. This subroutine makes use of the functions DARSIN, ADOT, DOT, and FNORM. The subroutines ITRATE, FG, and SHAD are also used.

PROGRAM LISTING

```
      PROGRAM LOAD
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  500 FORMAT(3A5)
  300 FORMAT(A1)
  100 FORMAT(A5)
  101 FORMAT(I5)
  102 FORMAT(D20.10)
  104 FORMAT(2I2,F6.3)
  103 FORMAT(A10)
  200 FORMAT(/,41H ENTER THE NAME OF THE ASCII  OUTPUT FILE,
     *       /,41H THIS CAN BE THE INPUT FILE TO EITHER THE,
     *       /,41H ORBIT DETERMINATION PROGRAM OR EPHEMERIS,
     *       /,41H GENERATION PROGRAM:  FORMAT (XXXXXX.EXT))
  201 FORMAT(/,38H ENTER SATELLITE IDENTIFICATION - (A5))
  202 FORMAT(/,38H ENTER MODIFIED JULIAN DATE      - (I5))
  203 FORMAT(/,40H ENTER HOURS, MIN, AND SEC AS HHMMSS.SSS)
  601 FORMAT(/,48H RECTANGULAR OR KEPLERIAN INPUT ?  (TYPE R OR K))
  204 FORMAT(/,8H ENTER X)
  205 FORMAT(/,8H ENTER Y)
  206 FORMAT(/,8H ENTER Z)
  207 FORMAT(/,9H ENTER XD)
  208 FORMAT(/,9H ENTER YD)
  209 FORMAT(/,9H ENTER ZD)
  214 FORMAT(/,28H ENTER SEMI-MAJOR AXIS IN KM)
  215 FORMAT(/,19H ENTER ECCENTRICITY)
  216 FORMAT(/,18H ENTER INCLINATION)
  217 FORMAT(/,11H ENTER NODE)
  218 FORMAT(/,20H ENTER PERIGEE ANGLE)
  219 FORMAT(/,19H ENTER MEAN ANOMALY)
  210 FORMAT(/,11H ENTER BIAS)
  301 FORMAT(/,13H INCLUDE J2 ?)
  302 FORMAT(/,13H INCLUDE J3 ?)
  303 FORMAT(/,13H INCLUDE J4 ?)
  304 FORMAT(/,15H INCLUDE DRAG ?)
  305 FORMAT(/,14H INCLUDE SUN ?)
  306 FORMAT(/,15H INCLUDE MOON ?)
  307 FORMAT(/,36H FEET OR KILOMETERS ?  - TYPE F OR K)
  401 FORMAT(/,14H SOLVE FOR A ?)
  402 FORMAT(/,21H SOLVE FOR E COS(*) ?)
  403 FORMAT(/,21H SOLVE FOR E SIN(*) ?)
  404 FORMAT(/,24H SOLVE FOR INCLINATION ?)
  405 FORMAT(/,24H SOLVE FOR NODAL ANGLE ?)
  406 FORMAT(/,16H SOLVE FOR M+* ?)
  407 FORMAT(/,17H SOLVE FOR BIAS ?)
  501 FORMAT(/,28H PERTURBATION PARAMETERS ARE,2X,6I1)
  502 FORMAT(/,25H NON-FIXED PARAMETERS ARE,2X,7I1)
  503 FORMAT(14H ANY CHANGES ?)
      DIMENSION X(7),XM(49),LP(5),LR(7),ELD(7)
      COMMON/KOUNT/KOUNT(7)
      LOGICAL KILO
```

```
      DATA KOUNT/1,1,1,1,1,1,1/
      DATA KAA/"202517170312/
      DATA KBB/"202624067744/
      DATA KCC/"202344020100/
      DATA LB/1,1,1,1,1,1,0/
      DATA LF/1,0,0,0,1,1/

      TYPE 200
      ACCEPT 103,XNAME
      OPEN (UNIT=21,ACCESS='SEQOUT',FILE=XNAME)

      TYPE 201
      ACCEPT 100,IDEN
      TYPE 202
      ACCEPT 101,MJD
      TYPE 203
      ACCEPT 104,MH,MM,FSEC

      TYPE 601
      ACCEPT 300,LNY
      IF (LNY.EQ.1HK) GO TO 8
      IF (LNY.EQ.1HR) GO TO 4

    4 TYPE 307
      ACCEPT 300,LNY
      IF (LNY.EQ.1HK) KILO=.TRUE.
      IF (LNY.EQ.1HF) KILO=.FALSE.
      IF (LNY.NE.1HF.AND.LNY.NE.1HK) GO TO 4

      TYPE 204
      ACCEPT 102,X(1)
      TYPE 205
      ACCEPT 102,X(2)
      TYPE 206
      ACCEPT 102,X(3)
      TYPE 207
      ACCEPT 102,X(4)
      TYPE 208
      ACCEPT 102,X(5)
      TYPE 209
      ACCEPT 102,X(6)
      IF (KILO) GO TO 6
      DO 5 K=1,6
    5 X(K)=3.048D-4*X(K)
      GO TO 6

    8 TYPE 214
      ACCEPT 102,ELD(1)
      TYPE 215
      ACCEPT 102,ELD(2)
      TYPE 216
      ACCEPT 102,ELD(3)
      TYPE 217
```

```
      ACCEPT 102,ELD(4)
      TYPE 218
      ACCEPT 102,ELD(5)
      TYPE 219
      ACCEPT 102,ELD(6)
      CALL ELSTAT(ELD,X)

    6 BIAS=0.0D0

      TYPE 501,(LF(K),K=1,6)
      TYPE 503
   10 TYPE 500,KAA,KBB,KCC
      ACCEPT 300,LNY
      IF (LNY.EQ.1HN) GO TO 30
      IF (LNY.NE.1HY) GO TO 10

      DO 20 K=1,6
   20 LF(K)=0
      TYPE 301
      TYPE 500,KAA,KBB,KCC
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(1)=1
      TYPE 302
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(2)=1
      TYPE 303
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(3)=1
      TYPE 304
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(4)=1
      TYPE 305
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(5)=1
      TYPE 306
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LF(6)=1

   30 TYPE 502,(LR(K),K=1,7)
      TYPE 503
   40 TYPE 500,KAA,KBB,KCC
      ACCEPT 300,LNY
      IF (LNY.EQ.1HN) GO TO 60
      IF (LNY.NE.1HY) GO TO 40

      DO 50 K=1,7
   50 LR(K)=0
      TYPE 401
      TYPE 500,KAA,KBB,KCC
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(1)=1
      TYPE 402
      ACCEPT 300,LNY
```

```
      IF (LNY.EQ.1HY) LR(2)=1
      TYPE 403
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(3)=1
      TYPE 404
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(4)=1
      TYPE 405
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(5)=1
      TYPE 406
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(6)=1
      TYPE 407
      ACCEPT 300,LNY
      IF (LNY.EQ.1HY) LR(7)=1

   60 T=(MJD-40587)*86400.0D0+MH*3600+MM*60+TSEC
      NOBS=0
      RMS=0.0D0
      LUOUT=21
      DO 70 K=1,7
      DO 70 J=1,7
      K1=K+7*(J-1)
   70 XM(K1)=0.0D0
      XM(1)=1.0D-18
      XM(9)=1.0D-09
      XM(17)=1.0D-09
      XM(25)=1.0D-07
      XM(33)=1.0D-07
      XM(41)=1.0D-07
      XM(49)=1.0D-02
      KSTART=1
      KSTOP =1
      NPRINT=1
      CALL OUTOD (X,T,IDEN,NOBS,RMS,BIAS,LUOUT,
     *            XM,KSTART,KSTOP,NPRINT,LF,LR)
      CLOSE (UNIT=21)
      RETURN
      END

      SUBROUTINE STATEL (X,ELD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(7), ELD(7), B(3)
      DATA  PI/3.14159265358979D0/
      DATA  TPI/6.28318530717959D0/
      DATA  RTD/57.2957795130820D0/
      DATA XMU/398601.5D0/
      B(1)=X(2)*X(6)-X(3)*X(5)
      B(2)=X(3)*X(4)-X(1)*X(6)
      B(3)=X(1)*X(5)-X(2)*X(4)
      R2=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)
      V2=X(4)*X(4)+X(5)*X(5)+X(6)*X(6)
```

```
      B2=B(1)*B(1)+B(2)*B(2)+B(3)*B(3)
      AA=X(1)*X(4)+X(2)*X(5)+X(3)*X(6)
      R1=DSQRT(R2)
      B1=DSQRT(B2)
      AA=AA/XMU
      P1=B2/XMU
      C3=V2-2.0D0*XMU/R1
      SMA=-XMU/C3
      ECC=DSQRT(DABS(1.0D0+C3*P1/XMU))
      XINC=ARKTNS(180,B(3),DSQRT(B(1)**2+B(2)**2))
      XNODE=ARKTNS(360,-B(2),B(1))
      THETA=ARKTNS(360,(P1-R1),B1*AA)
      ARGLAT=ARKTNS(360,X(2)*B(1)-X(1)*B(2),X(3)*B1)
      PERI=ARGLAT-THETA
      IF (PERI.LT.0.0D0) PERI=PERI+TPI
      F1=AA*XMU/DSQRT(XMU*SMA)
      F2=1.0D0-R1/SMA
      IF (DABS(ECC).GE.1.0D-8) GO TO 10
      COSE=1.0D0
      SINE=0.0D0
      GO TO 20
   10 SINE=F1/ECC
      COSE=F2/ECC
   20 E=ARKTNS(360,COSE,SINE)
      XMEAN=(E-ECC*SINE)
      IF (ECC.EQ.0.0D0) XMEAN=THETA
      ELD(1)=SMA
      ELD(2)=ECC
      ELD(3)=XINC*RTD
      ELD(4)=XNODE*RTD
      ELD(5)=PERI*RTD
      ELD(6)=XMEAN*RTD
      RETURN
      END

      FUNCTION ARKTNS (N,X,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA  PI/3.14159265358979D0/
      DATA  TPI/6.28318530717958D0/
      IF (X.NE.0.0D0) GO TO 10
      IF (Y.GT.0.0D0) T=0.5D0*PI
      IF (Y.LT.0.0D0) T=1.5D0*PI
      IF (Y.EQ.0.0D0) T=0.0D0
      GO TO 20
   10 T=DATAN(Y/X)
      IF (X.LT.0.0D0) T=T+PI
      IF (T.LT.0.0D0) T=T+TPI
   20 IF (N.EQ.360) GO TO 30
      IF (T.GT.PI) T=T-TPI
   30 ARKTNS=T
      RETURN
      END
```

```
      SUBROUTINE OUTOD(X,T,IDEN,NOBS,RMS,BIAS,
     *    LUOUT,WM,KSTART,KSTOP,NPRINT,LF,LR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (2H  )
  101 FORMAT (6H SAT =,7X,A5,2X,5HA     =,F12.2,2X,5HX     =,F12.2)
  102 FORMAT (6H MJD =,  I12,2X,5HE     =,F12.8,2X,5HY     =,F12.2)
  103 FORMAT (6H TSEC=,F12.3,2X,5HI     =,F12.4,2X,5HZ     =,F12.2)
  104 FORMAT (6H NOBS=,  I12,2X,5HNODE=,F12.4,2X,5HXD    =,F12.7)
  105 FORMAT (6H RMS =,F12.3,2X,5HPERI=,F12.4,2X,5HYD    =,F12.7)
  106 FORMAT (6H BIAS=,F12.3,2X,5HMEAN=,F12.4,2X,5HZD    =,F12.7)
  107 FORMAT (18H COVARIANCE MATRIX)
  108 FORMAT (8X,1HN,2X,2X,8HE COS(W),2X,8HE SIN(W),
     *        7X,1HI,6X,4HNODE,7X3HM+W,5X,4HBIAS)
  109 FORMAT (1X,7(1PE10.3))
  110 FORMAT(6H FIRST,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
  111 FORMAT(6H LAST ,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
  112 FORMAT (17H PERTURBATIONS = ,6I1,2X,
     *        24H NON-FIXED PARAMETERS = ,7I1)
  113 FORMAT (4H END)
      COMMON/TM/TIME(1)/Y/Y(1)/TSTEP/TSTEP
      DIMENSION X(7),EL(7),WM(49),LF(6),LR(7)
      REWIND LUOUT
      CALL STATEL(X,EL)
      MJD=IDINT(T/86400.0D0)+40587
      TSEC=DMOD(T,86400.0D0)
      JF=40587+TIME(KSTART)/86400.0D0
      JL=40587+TIME(KSTOP )/86400.0D0
      TF=DMOD (TIME(KSTART),86400.0D0)
      TL=DMOD (TIME(KSTOP ),86400.0D0)
      RF=Y(KSTART)
      RL=Y(KSTOP )
      WRITE (LUOUT,100)
      WRITE (LUOUT,101),IDEN,EL(1),X(1)
      WRITE (LUOUT,102), MJD,EL(2),X(2)
      WRITE (LUOUT,103),TSEC,EL(3),X(3)
      WRITE (LUOUT,104),NOBS,EL(4),X(4)
      WRITE (LUOUT,105), RMS,EL(5),X(5)
      WRITE (LUOUT,106),BIAS,EL(6),X(6)
      WRITE (LUOUT,110),KSTART,JF,TF,RF
      WRITE (LUOUT,111),KSTOP, JL,TL,RL
      WRITE (LUOUT,100)
      WRITE (LUOUT,107)
      WRITE (LUOUT,108)
      DO 10 J=1,7
   10 WRITE (LUOUT,109),(WM(7*K+J-7),K=1,7)
      WRITE (LUOUT,100)
      WRITE (LUOUT,112),(LF(K),K=1,6),(LR(K),K=1,7)
      WRITE (LUOUT,113)
      END FILE LUOUT
      IF (NPRINT.EQ.0) RETURN
      TYPE 100
      TYPE 101,IDEN,EL(1),X(1)
      TYPE 102, MJD,EL(2),X(2)
```

```
      TYPE 103,TSEC,EL(3),X(3)
      TYPE 104,NOBS,EL(4),X(4)
      TYPE 105, RMS,EL(5),X(5)
      TYPE 106,BIAS,EL(6),X(6)
      TYPE 110,KSTART,JF,TF,RF
      TYPE 111,KSTOP ,JL,TL,RL
      TYPE 100
      TYPE 112,(LF(K),K=1,6),(LR(K),K=1,7)
      RETURN
      END

      SUBROUTINE ELSTAT (ELD,X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(7), ELD(7), A(3,2)
      DATA XMU/398601.5D0/
      DATA  DTR/0.017453292519943D0/
      SNI=DSIN (ELD(3)*DTR)
      CNI=DCOS (ELD(3)*DTR)
      SOM=DSIN (ELD(4)*DTR)
      COM=DCOS (ELD(4)*DTR)
      XM=DMOD(ELD(6),360.0D0)*DTR
      ECC=ELD(2)
      E=XKEP(ECC,XM,1.0D-10)
      SINE=DSIN(E)
      COSE=DCOS(E)
      STA=DSQRT(1.0D0-ECC**2)*SINE/(1.0D0-ECC*COSE)
      CTA=(COSE-ECC)/(1.0D0-ECC*COSE)
      TAA=ARKTNS(180,CTA,STA)
      TBB=TAA+DTR*ELD(5)
      CBA=DCOS (TBB)
      SBA=DSIN (TBB)
      A(1,1)=+COM*CBA-SOM*CNI*SBA
      A(2,1)=+SOM*CBA+COM*CNI*SBA
      A(3,1)=+SNI*SBA
      A(1,2)=-COM*SBA-SOM*CNI*CBA
      A(2,2)=-SOM*SBA+COM*CNI*CBA
      A(3,2)=+SNI*CBA
      P=ELD(1)*(1.0D0-ECC**2)
      R=P/(1.0D0+ECC*CTA)
      VR=ECC*STA*DSQRT(XMU/P)
      VT=DSQRT(XMU*(2.0D0/R-1.0D0/ELD(1))-VR*VR)
      DO 10 K=1,3
      X(K)=R*A(K,1)
   10 X(K+3)=VR*A(K,1)+VT*A(K,2)
      RETURN
      END
      DOUBLE PRECISION FUNCTION XKEP (ECC,XM,TOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (10X,40H ** KEPLERS EQUATION DID NOT CONVERGE **)
      EOLD=XM
      DO 10 K=1,100
      SEC=DSIN(EOLD)*ECC
      CEC=DCOS(EOLD)*ECC
```

```
      ENEW=(XM+SEC-EOLD*CEC)/(1.0D0-CEC)
      DE=DABS(ENEW-EOLD)
      IF (DE.LE.TOL) GO TO 20
   10 EOLD=ENEW
      TYPE 100
      STOP
   20 XKEP=ENEW
      RETURN
      END
```

```
      PROGRAM ENTER
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  820 FORMAT(/)
  870 FORMAT(I4)
  850 FORMAT(A1)
  851 FORMAT(A10)
  855 FORMAT(A2)
  802 FORMAT(2I2,F6.3)
  804 FORMAT(F10.3)
  805 FORMAT(I4,I7,2F11.3,F9.2)
  890 FORMAT(/,20H FOR ALL QUERIES "?",
     *       /,15H TYPE "Y" - YES,
     *       /,15H      "N" - NO ,
     *       /,16H      "S" - STOP)
  800 FORMAT(/,18H ENTER DAY OF YEAR)
  833 FORMAT(/,33H ENTER YEAR,  E.G. "1977"  - (I4))
  801 FORMAT( 31H ENTER TIME IN FORM: HHMMSS.SSS)
  815 FORMAT(/,44H WILL THE NEW RANGE OBSERVATIONS BE ADDED TO,
     *       /,44H AN EXISTING FILE OR ENTERED TO A NEW FILE ?)
  816 FORMAT( 33H TYPE "A" TO ADD TO EXISTING FILE,
     *       /,29H TYPE "B" TO BEGIN A NEW FILE)
  860 FORMAT(/,40H THE RANGE DATA WAS STORED IN FILE      ,A10)
  875 FORMAT(/,40H THE UPDATED STATE VECTOR WAS STORED IN ,A10)
  803 FORMAT(27H ENTER OBSERVED RANGE IN KM)
  810 FORMAT(8H YR/DAY=,I2,1H/,I3,2X,
     *       7H H/M/S=,I2,1H/,I2,1H/,F6.3)
  710 FORMAT(18H PREDICTED RANGE =,F10.2,3H KM,
     *       3X,12H ELEVATION =,F6.1,4H DEG)
  713 FORMAT(18H RANGE RATE      =,F10.5,7H KM/SEC,
     *       4X,12H AZIMUTH   =,F6.1,4H DEG)
  711 FORMAT(18H OBSERVED  RANGE =,F10.2)
  712 FORMAT(18H DIFFERENCE      =,F10.2,3H KM)
  813 FORMAT(16H O.K. TO STORE ?)
  814 FORMAT(16H ANY MORE OBS  ?)
  986 FORMAT(/,30H NUMERICAL VERSION OF "RANGIN")
  988 FORMAT(/,34H ENTER ALL ASCII FILE NAMES IN THE,
     *       /,31H ALPHANUMERIC FORM:  XXXXXX.EXT)
  992 FORMAT(/,40H ENTER THE NAME OF THE NEW ASCII FILE IN,
     *       /,38H WHICH THE RANGE DATA IS TO BE ENTERED)
  993 FORMAT(/,36H ENTER NAME OF THE EXISTING FILE  IN,
     *       /,36H WHICH THE RANGE DATA IS TO BE ADDED)
  994 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *       /,23H THE INPUT STATE VECTOR)
      DIMENSION H(7),HT(7),XK(7),WA(49),LF(6),LR(7),TIME(1)
      DIMENSION XM(49),P(49),Q(49),WB(49),WC(1),YMF(1),ZA(6)
      DIMENSION ELD(7),ELR(7),ELN(7),X(7),DE(7),DX(7),ELX(7)
      COMMON/INIT/INIT/XYZ/XE,YE,ZE/MAXDIM/MAXDIM
      COMMON/LOC/XLAT,XLON,ALT,LSTA1
      COMMON/BIAS/BIAS/NRDATA/NRDATA
      COMMON/XMU/XMU,RE,XJ2
```

```
      DIMENSION MYST(20)
      LOGICAL PAST
      DATA MYST/40931,41316,41682,42047,42412,
     *          42777,43143,43508,43873,44238,
     *          44604,44969,45334,45699,46065,
     *          0,0,0,0,0/
      PAST=.FALSE.
      KTAG=1
      TYPE 986
      TYPE 988
      TYPE 815
      TYPE 816
      ACCEPT 850,LNA
      IF (LNA.EQ.1HA) BCCESS='APPEND'
      IF (LNA.EQ.1HB) BCCESS='SEQOUT'
      IF (LNA.EQ.1HA) CCCESS='SEQINOUT'
      IF (LNA.EQ.1HB) CCCESS='SEQIN'
      IF (LNA.EQ.1HA) TYPE 993
      IF (LNA.EQ.1HB) TYPE 992
      ACCEPT 851,YNAME
      IF (LNA.NE.1HA) GO TO 21
      OPEN(UNIT=21,ACCESS='SEQIN',FILE=YNAME)
      DO 10 K=1,2000
      READ (21,805,END=11) KKA,KKB,XXA,XXB,XXC
   10 KTAG=KKA+1
   11 CLOSE (UNIT=21)
   21 TYPE 994
      ACCEPT 851,XNAME
      IF (XNAME.EQ.YNAME) GO TO 21
      OPEN (UNIT=22,ACCESS=CCCESS ,FILE=XNAME)

C     ENTER STATION COORDINATES
      LSTA1=5HBLMPT
      XLAT=+38.431414D0
      XLON=282.913583D0
      ALT=-0.0247D0

      LRDATA=21
      LUDCIN=22

C     READ IN FROM DISK FILE INITIAL GUESS FOR ELEMENTS
      CALL INOD(X,EPOCH,ISAT,NOBS,RMS,BIAS,LUDCIN,XM,LF,LR)
      CALL FORCE(LF(1),LF(2),LF(3),LF(4),LF(5),LF(6))
      CALL STATEL(X,ELD)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)
      DDTT=STPSZE(X,0.05D0)
      BIAS=0.0D0
      ELN(7)=BIAS
      TYPE 890

C     ACCEPT TIME AND RANGE
   30 TYPE 833
```

```
      ACCEPT 870,MYSZ
      MYSZ=MOD(MYSZ,100)
      MYSN=MYSZ-70
      IF (MYSN.LE.0.OR.MYSN.GE.16) GO TO 30
   40 TYPE 800
      ACCEPT 870,MDAY
      TYPE 801
      ACCEPT 802,JR,JM,SEC
      TYPE 810,MYSZ,MDAY,JR,JM,SEC
      MJD=MYST(MYSN)+MDAY
      TSEC=JR*3600.0D0+JM*60.0D0+SEC
      TIME(1)=(MJD-40587)*86400.0D0+TSEC
      RA=RANGE(ELN,TIME(1)+1.0D0,EPOCH)
      RB=RANGE(ELN,TIME(1)-1.0D0,EPOCH)
      RRATE=(RA-RB)/2.0D0

      DINCRM=TIME(1)-EPOCH
      IF (DABS(DINCRM).GE.5.0D+6) GO TO 40
  878 FORMAT(20H UPDATE STATE VECTOR,F8.2,6H HOURS)
      DHOUR=DINCRM/3600.0D0
      IF (DABS(DINCRM).GT.14400.0D0) TYPE 878,DHOUR
      CALL FOUR(X,EPOCH,TIME(1),-1,DDIT)
      EPOCH=TIME(1)
      CALL STATEL(X,ELD)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)
      YPRED=RANGE(ELN,EPOCH,EPOCH)

      CALL AZEL(X,TIME(1),AZIM,ELEV,1,KEY)
      TYPE 713,RRATE,AZIM
      TYPE 710,YPRED,ELEV
      TYPE 803
      ACCEPT 804,YOBS
      YMF(1)=YOBS-YPRED
      TYPE 712,YMF(1)

   44 TYPE 813
      ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 46
      IF (LNY.EQ.1HE) GO TO 50
      IF (LNY.EQ.1HS) GO TO 50
      IF (LNY.NE.1HY) GO TO 44
      IF (PAST) BCCESS='APPEND'
      OPEN (UNIT=21,ACCESS=BCCESS   ,FILE=YNAME)
      WRITE (LRDATA,805) KTAG,MJD,TSEC,YOBS,YMF(1)
      CLOSE (UNIT=21)
      PAST=.TRUE.
      KTAG=KTAG+1

   46 TYPE 814
      ACCEPT 850,LNY
      IF (LNY.EQ.1HY) GO TO 40
      IF (LNY.EQ.1HE) GO TO 50
```

```
      IF (LNY.EQ.1HS) GO TO 50
      IF (LNY.EQ.1HN) GO TO 50
      GO TO 46

50    CLOSE (UNIT=22)
      TYPE 860,YNAME
      STOP
      END

      SUBROUTINE AZEL(R,T,AZ,ELV,IND,KEY)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION R(3),RZ(3),S(3),PHI(3),THETA(3),XH(3)
      COMMON/LOC/XLAT,XLON,ALT,LSTA1
      COMMON/MOC/YLAT,YLON,BLT,LSTA2
      COMMON/NOC/ZLAT,ZLON,CLT,LSTA3
      COMMON/XMU/XMU,RE,XJ2
      DATA DTR/.01745329252D0/
      DATA EP/0.08181333402D0/
      DATA PI2/1.570796327D0/
      THETA(1)=XLAT*DTR
      THETA(2)=YLAT*DTR
      THETA(3)=ZLAT*DTR
      PHI(1)=XLON*DTR
      PHI(2)=YLON*DTR
      PHI(3)=ZLON*DTR
      XH(1)=ALT
      XH(2)=BLT
      XH(3)=CLT
      TD=IDINT(T/86400.0D0)
      TF= DMOD(T,86400.0D0)
      CALL GHA70(TF,TD,G,0.0D0,OM)
      G=G*DTR
      SL=DSIN(THETA(IND))
      CL=DCOS(THETA(IND))
      SP=DSIN(PHI(IND))
      CP=DCOS(PHI(IND))
      SG=DSIN(G)
      CG=DCOS(G)
      RZ(1)=+R(1)*CG+R(2)*SG
      RZ(2)=-R(1)*SG+R(2)*CG
      RZ(3)=+R(3)
      XN=RE/DSQRT(1.D0-EP*EP*DSIN(THETA(IND))**2)
      RZ(1)=RZ(1)-(XN+XH(IND))*CL*CP
      RZ(2)=RZ(2)-(XN+XH(IND))*CL*SP
      RZ(3)=RZ(3)-(XN*(1.D0-EP*EP)+XH(IND))*SL
      SP=DSIN(PHI(IND)+PI2)
      CP=DCOS(PHI(IND)+PI2)
      SL=DSIN(PI2-THETA(IND))
      CL=DCOS(PI2-THETA(IND))
      S(1)=+CP*RZ(1)+SP*RZ(2)
      S(2)=-SP*RZ(1)+CP*RZ(2)
      S(3)=+RZ(3)
      RZ(1)=+S(1)
```

```
RZ(2)=+CL*S(2)+SL*S(3)
RZ(3)=-SL*S(2)+CL*S(3)
AZ=ARKTNS(360,RZ(2),RZ(1))/DTR
RM=DSQRT(RZ(1)*RZ(1)+RZ(2)*RZ(2)+RZ(3)*RZ(3))
SEL=RZ(3)/RM
CEL=DSQRT(1.D0-SEL*SEL)
ELV=ARKTNS(180,CEL,SEL)/DTR
KEY=0
IF (ELV.GT.0.D00) KEY=1
RETURN
END


DOUBLE PRECISION FUNCTION RANGE(ELN,T,TN)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/BIAS/BIAS
DIMENSION ELN(7),ELR(7),ELD(7),RX(7)
CALL ELNELR(ELN,ELR)
CALL RADDEG(ELR,ELD)
CALL POSION(T,XSTAT,YSTAT,ZSTAT)
CALL ORBIT(ELD,RX,T,TN)
R2=(XSTAT-RX(1))**2+(YSTAT-RX(2))**2+(ZSTAT-RX(3))**2
RANGE=DSQRT(R2)+BIAS
RETURN
END
```

```
      PROGRAM GENER
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  108 FORMAT(2I2,F6.3)
  101 FORMAT(D15.10)
  131 FORMAT(///)
  103 FORMAT(A2)
  104 FORMAT(3A5)
  107 FORMAT(A10)
  111 FORMAT(I5)
  125 FORMAT(A1)
  130 FORMAT(4H END)
  126 FORMAT(16H TYPE "Y" OR "N")
  106 FORMAT(//,15H FINAL STATE IS)
  300 FORMAT(/,17H ABREV PRINTOUT ?)
  400 FORMAT(/,20H FOR ALL QUERIES "?")
  371 FORMAT(/,24H WANT LISTING  ON DISK ?)
  370 FORMAT(/,24H WANT TELETYPE LISTING ?)
  372 FORMAT(/,40H ENTER ASCII FILE NAME OF OUTPUT LISTING)
  409 FORMAT(/,39H ENTER PRINT INTERVAL IN HOURS - HH.HHH)
  105 FORMAT(//,29H INITIAL STATE AND EPOCH FOR ,A5,4H ARE)
  100 FORMAT(/,35H ENTER INTEGRATION INTERVAL IN DAYS)
  102 FORMAT(/,33H FOURTH OR NINTH ORDER INTEGRATOR)
  113 FORMAT(/,23H MOVE STARTING POINT BY,F10.4,5H DAYS)
  120 FORMAT(/,44H ROUTINE DOES NOT WORK FOR REVERSE TIME SPAN)
  110 FORMAT(/,40H ENTER HOURS,MIN,SEC IN FORM  HHMMSS.SSS)
  109 FORMAT(/,36H ENTER MOD JULIAN DATE OF START TIME)
  181 FORMAT(/,34H ENTER HOUR OF DAY IN FORM HH - I2)
  112 FORMAT(/,36H ENTER MOD JULIAN DATE OF STOP  TIME)
  200 FORMAT(/,39H ENTER ASCII FILE NAME OF INPUT  VECTOR)
  210 FORMAT(/,42H ENTER ASCII FILE NAME OF OUTPUT EPHEMERIS)
  230 FORMAT(/,39H ENTER ASCII FILE NAME OF OUTPUT VECTOR)
  114 FORMAT(/,24H INTEGRATION TIME SPAN =,F10.4,5H DAYS)
  310 FORMAT(/,23H TYPE "A" -  ALERT MODE,
     *       /,25H      "B" -  ECLIPSE MODE,
     *       /,33H      "C" -  EPHEMERIS GENERATION,
     *       /,40H      "D" -  TO UPDATE STATE VECTOR ONLY,
     *       /,49H      "E" -  TO LIST STATE AT SPECIFIED INTERVALS)
  127 FORMAT(/,26H SAVE FINAL STATE VECTOR ?)
      DIMENSION X(7),ELD(7),WM(49),LF(6),LR(7)
      COMMON/LOC/XLAT,XLON,ALT,LSTA1
      COMMON/MOC/YLAT,YLON,BLT,LSTA2
      COMMON/NOC/ZLAT,ZLON,CLT,LSTA3
      COMMON/SHADO/KKEEYY/DXDYDZ/DX,DY,DZ
      COMMON/TSTEP/TSTEP/BIAS/BIAS
      COMMON/ABREV/ABREV/KEY/KEYS,KEYM
      LOGICAL ABREV,ALERT,ECLPS,EPHEM,UPDAT,KLIST,MLIST,NLIST
      DATA KAA/"202517170312/
      DATA KBB/"202346420336/
      DATA KCC/"711011634500/
```

```
      LSTA1=5HBLMPT
      XLAT=+38.431414D0
      XLON=282.913583D0
      ALT =-0.0247D0
      LSTA2=5HITALY
      YLAT=+43.754D0
      YLON=+11.255D0
      BLT =+0.1804D0
      LSTA3=5HINDIA
      ZLAT=+23.033D0
      ZLON=+72.583D0
      CLT =+0.0D0
      BIAS=0.0D0
      LUIN=21
      LUOUT=22
      LUSAV=23
      KKEEYY=0
      KEYS=0
      KEYM=0
    4 TYPE 310
      ACCEPT 125,LNA
      ALERT=.FALSE.
      ECLPS=.FALSE.
      EPHEM=.FALSE.
      UPDAT=.FALSE.
      KLIST=.FALSE.
      IF (LNA.EQ.1HA) ALERT=.TRUE.
      IF (LNA.EQ.1HB) ECLPS=.TRUE.
      IF (LNA.EQ.1HC) EPHEM=.TRUE.
      IF (LNA.EQ.1HD) UPDAT=.TRUE.
      IF (LNA.EQ.1HE) KLIST=.TRUE.

      TYPE 200
      ACCEPT 107,XNAME
      OPEN (UNIT=21,ACCESS='SEQIN ',FILE=XNAME)
      IF (.NOT.EPHEM) GO TO 7

    5 TYPE 210
      ACCEPT 107,YNAME
      IF (XNAME.EQ.YNAME) GO TO 5
      OPEN (UNIT=22,ACCESS='SEQOUT',FILE=YNAME)

    7 REWIND LUIN
      TYPE 400
      TYPE 126
      IF (EPHEM) REWIND LUOUT
      CALL INGEN(X,TSTART,ISAT,NOBS,RMS,BIAS,LUIN,
     *           WM,KSTART,KSTOP,JF,JL,TF,TL,RF,RL,
     *           LF,LR)
      CALL FORCE(LF(1),LF(2),LF(3),LF(4),LF(5),LF(6))
      IF (ALERT) DDTT=STPSZE(X,0.06D0)
      IF (ECLPS) DDTT=STPSZE(X,0.020D0)
      IF (EPHEM) DDTT=STPSZE(X,0.032D0)
```

```
      IF (UPDAT) DDTT=STPSZE(X,0.04D0)
      IF (KLIST) DDTT=STPSZE(X,0.04D0)
      TSTEP=DDTT
      TYPE 105,ISAT
      CALL PAPER(X,TSTART)

      IF (ALERT) GO TO 75
      IF (ECLPS) GO TO 35
      IF (EPHEM) GO TO 12
      IF (UPDAT) GO TO 35
      IF (KLIST) GO TO 90
      GO TO 4

C     EPHEMERIS GENERATION CODE CONTINUES HERE
   12 TYPE 109
      ACCEPT 111,MJD
      IF (IABS(MJD-43500).GT.10000) GO TO 12
      TYPE 181
      ACCEPT 108,MH
      TBEGIN=(MJD-40587)*86400.0D0+MH*3600.0D0+60.0D0
      TEL=(TBEGIN-TSTART)/86400.0D0
      TYPE 113,TEL
      IF (TEL.EQ.0.0D0) GO TO 6
      CALL FOUR(X,TSTART,TBEGIN,-1,DDTT)
      TYPE 105,ISAT
      CALL PAPER(X,TBEGIN)
    6 TYPE 300
      ACCEPT 125,LNY
      IF (LNY.EQ.1HY) ABREV=.TRUE.
      IF (LNY.EQ.1HN) ABREV=.FALSE.
      IF (LNY.NE.1HN.AND.LNY.NE.1HY) GO TO 6
    8 TYPE 112
      ACCEPT 111,MJD
      IF (IABS(MJD-43500).GT.10000) GO TO 8
      TYPE 110
      ACCEPT 108,MH,MN,SEC
      TSTOP =(MJD-40587)*86400.0D0+MH*3600.0D0+MN*60.0D0+SEC
      TEL=(TSTOP-TBEGIN)/86400.0D0
      TYPE 114,TEL
      IF (TEL.LE.0.0D0) TYPE 120
      IF (TEL.LE.0.0D0) STOP
      CALL FIND4(X,TBEGIN,TSTOP,DDTT,ISAT,LUOUT)
      WRITE(LUOUT,130)
      CLOSE(UNIT=22)
      GO TO 40

C     ECLIPSE AND UPDATE CODE CONTINUES HERE
   35 IF (ECLPS) KKEEYY=1
      IF (UPDAT) KKEEYY=0
      TYPE 112
      ACCEPT 111,MJD
      IF (IABS(MJD-43500).GT.10000) GO TO 35
      TYPE 110
```

```
      ACCEPT 108,MH,MN,SEC
      TSTOP =(MJD-40587)*86400.0D0+MH*3600.0D0+MN*60.0D0+SEC
      TEL=(TSTOP-TSTART)/86400.0D0
      TYPE 114,TEL
      CALL FOUR(X,TSTART,TSTOP,-1,DDTT)
      GO TO 40

C     LISTING OPTION STARTS HERE
   90 KKEEYY=0
      TYPE 112
      ACCEPT 111,MJD
      IF (IABS(MJD-43500).GT.10000) GO TO 90
      TYPE 110
      ACCEPT 108,MH,MN,SEC
      TSTOP =(MJD-40587)*86400.0D0+MH*3600.0D0+MN*60.0D0+SEC
      TEL=(TSTOP-TSTART)/86400.0D0
      TYPE 114,TEL
   91 TYPE 409
      ACCEPT 101,XSTEP
      IF (DABS(XSTEP).GT.1.0D+4) GO TO 91
      ZSTEP=XSTEP*3600
      IF (TEL.LE.0.0D0.OR.XSTEP.LE.0.0D0) TYPE 120
      IF (TEL.LE.0.0D0.OR.XSTEP.LE.0.0D0) GO TO 90
      MLIST=.FALSE.
      NLIST=.FALSE.
      TYPE 370
      ACCEPT 125,LNY
      IF (LNY.EQ.1HY) MLIST=.TRUE.
      TYPE 371
      ACCEPT 125,LNY
      IF (LNY.EQ.1HY) NLIST=.TRUE.
      IF (.NOT.NLIST) GO TO 99
   98 TYPE 372
      ACCEPT 107,YNAME
      IF (XNAME.EQ.YNAME) GO TO 98
      OPEN(UNIT=22,ACCESS='SEQOUT',FILE=YNAME)
   99 T1=TSTART
      TYPE 131
      IF (MLIST) CALL PAPER(X,TSTART)
      IF (NLIST) CALL DAPER(X,TSTART,LUOUT)
   92 T2=T1+ZSTEP
      CALL FOUR(X,T1,T2,-1,DDTT)
      IF (DABS(T2-TSTOP).LE.1.0D0) GO TO 94
      IF (MLIST) CALL PAPER(X,T2)
      IF (NLIST) CALL DAPER(X,T2,LUOUT)
      IF (DABS(T2-TSTOP).LT.DABS(ZSTEP-1.0D0)) GO TO 94
      IF (T2.GT.TSTOP) GO TO 94
      T1=T2
      GO TO 92
   94 CALL FOUR(X,T2,TSTOP,-1,DDTT)
      IF (MLIST) CALL PAPER(X,TSTOP)
      IF (NLIST) CALL DAPER(X,TSTOP,LUOUT)
      TYPE 131
```

```
      IF (NLIST) CLOSE(UNIT=22)
      GO TO 40
C     ALERT PROGRAM STARTS HERE
   75 TYPE 112
      ACCEPT 111,MJD
      TYPE 110
      ACCEPT 108,MH,MN,SEC
      TSTOP =(MJD-40587)*86400.0D0+MH*3600.0D0+MN*60.0D0+SEC
      TEL=(TSTOP-TSTART)/86400.0D0
      TYPE 114,TEL
      CALL SIX(X,TSTART,TSTOP,-1,DDTT,ISAT)
C     ALL MODES INCLUDE THE FOLLOWING CODE
   40 CLOSE (UNIT=21)
   45 TYPE 106
      CALL PAPER(X,TSTOP)
      TYPE 127
      ACCEPT 125,LNY
      IF (LNY.EQ.1HN) RETURN
   50 TYPE 230
      ACCEPT 107,ZNAME
      IF (ZNAME.EQ.YNAME) GO TO 50
      OPEN (UNIT=23,ACCESS='SEQOUT',FILE=ZNAME)
      REWIND LUSAV
      CALL OUTGEN(X,TSTOP,ISAT,NOBS,RMS,BIAS,LUSAV,
     *            WM,KSTART,KSTOP,JF,JL,TF,TL,RF,RL,
     *            LF,LR)
      CLOSE (UNIT=23)
      STOP
      END

      SUBROUTINE FIND4(X,TBEGIN,TSTOP,DT,ISAT,LUOUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/TSTEP/TSTEP/NCAL/NCAL
      COMMON/KEY/KEYS,KEYM
      DIMENSION X(7),Z(7)
      NCAL=0
      KEYS=0
      KEYM=0
      CALL RTTIME(TBEGIN,TSTART)
      IF (DABS(TBEGIN-TSTART).LT.0.01D0) GO TO 10
      CALL FOUR(X,TBEGIN,TSTART,-1,DT)
   10 XMSTEP=(TSTOP-TSTART)/DT
      YMSTEP=DABS(XMSTEP)+0.0001D0
      NMSTEP=IDINT(YMSTEP)
      IF (NMSTEP.LT.1) NMSTEP=1
      TMSTEP=(TSTOP-TSTART)/NMSTEP
      TSTEP=TMSTEP
      CALL OUTPD(X,TSTART,ISAT,LUOUT)
      X(7)=TSTART
      TNEXT=TSTART+120.0D0
   20 DIFF=DABS(TNEXT-X(7))
```

```
      IF (TNEXT.LT.X(7)) GO TO 40
      IF (DIFF.GT.TSTEP/2) GO TO 50
   40 CALL FANDG(X,Z,TNEXT,X(7))
      CALL OUTPD(Z,TNEXT,ISAT,LUOUT)
      TNEXT=TNEXT+120.0D0
      GO TO 20
   50 IF (DABS(TSTOP-X(7)).LE.1.0D0) GO TO 70
      IF (TSTOP.GT.TSTART.AND.X(7).GE.TSTOP) GO TO 70
      IF (TSTOP.LT.TSTART.AND.X(7).LE.TSTOP) GO TO 70
      CALL RUK(X)
      GO TO 20

   70 RETURN
      END

      SUBROUTINE SIX(X,TSTART,TSTOP,INT,DT,ISAT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/TSTEP/TSTEP/NCAL/NCAL
      COMMON/KEY/KEYS,KEYM
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION X(7)
      LOGICAL LPRINT,SEE
      NCAL=0
      KEYS=0
      KEYM=0
      LCON=1
      DDT=DT
      ELLAST=0.0D0
      SEE=.TRUE.
      LPRINT=.TRUE.
      IF (INT.EQ.-1) LPRINT=.FALSE.
      IF (DABS(DDT).LT.1.0D-4) DDT=STPSZE(X,0.06D0)
      XMSTEP=(TSTOP-TSTART)/DDT
      YMSTEP=DABS(XMSTEP)+0.0001D0
      NMSTEP=IDINT(YMSTEP)
      IF (NMSTEP.LT.1) NMSTEP=1
      TMSTEP=(TSTOP-TSTART)/NMSTEP
      TSTEP=TMSTEP
      X(7)=TSTART
      IF (LPRINT) CALL PAPER(X,X(7))
   10 CALL RUK(X)
      CALL AZEL(X,X(7),AZ,EL,1,MEY)
      IF (ELLAST.EQ.0.0D0) GO TO 15
      DELDT=(EL-ELLAST)/TSTEP
      IF (EL.GT.0.0D0.AND.SEE) GO TO 15
      IF (EL.LE.0.0D0.AND..NOT.SEE) GO TO 15
      TME=X(7)-EL/DELDT
      IF (EL.GT.0.0D0) KEY=+1
      IF (EL.LT.0.0D0) KEY=-1
      CALL INSPCT(TME,KEY,ISAT)
   15 IF (EL.GT.0.0D0) SEE=.TRUE.
      IF (EL.LE.0.0D0) SEE=.FALSE.
      ELLAST=EL
```

```
      IF (DABS(TSTOP-X(7)).LE.1.0D0) GO TO 30
      IF (TSTOP.GT.TSTART.AND.X(7).GE.TSTOP) GO TO 30
      IF (TSTOP.LT.TSTART.AND.X(7).LE.TSTOP) GO TO 30
      IF (INT.EQ.0) GO TO 20
      IF (LCON/INT*INT.NE.LCON) GO TO 20
      IF (LPRINT) CALL PAPER(X,X(7))
   20 LCON=LCON+1
      GO TO 10
   30 IF (LPRINT) CALL PAPER(X,X(7))
      RETURN
      END

      SUBROUTINE INSPCT(TME,KEY,ISAT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT(1X,I5,2X,I2,2HHR,1X,I2,3HMIN,1X,F6.3)
  101 FORMAT(/,1X,A5,1X,10H RISE TIME)
  102 FORMAT(/,1X,A5,1X,10H  SET TIME)
      IF (KEY.GT.0) TYPE 101,ISAT
      IF (KEY.LT.0) TYPE 102,ISAT
      MJD=40587+TME/86400.0D0
      TSEC=DMOD(TME,86400.0D0)+30.0D0
      MH=TSEC/3600.0D0
      MM=DMOD(TSEC,3600.0D0)/60.0D0
      TYPE 100,MJD,MH,MM
      RETURN
      END

      SUBROUTINE OUTPD(X,T,ISAT,LUOUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT(2H  )
  101 FORMAT(4H SAT,1X,A5,4H MJD,1X,I5,4H LAT,F6.2,
     *       4H LON,F7.2,4H RNG,2F10.2)
  102 FORMAT(4H LSP,1X,A5,2F6.1,F10.2,2(1X,A5,2F6.1))
  103 FORMAT(4H GMT,1X,4I1,1X,3F9.1,1X,3F9.6)
  104 FORMAT(4H MIN,1X, I4,1X,3F9.1,1X,3F9.6)
      COMMON/LOC/XLAT,XLON,ALY,LSTA1
      COMMON/MOC/YLAT,YLON,BLT,LSTA2
      COMMON/NOC/ZLAT,ZLON,CLT,LSTA3
      DIMENSION X(7),PSUN(6),PMOON(6),XLOAD(15,6)
      COMMON/ABREV/ABREV
      LOGICAL ABREV

      MJD=IDINT(T/86400.0D0)+40587
      MIN=DMOD(T,86400.0D0)/60.0D0+0.001D0
      JEY=MOD(MIN,30)

      J=(JEY+1)/2
      DO 5 K=1,6
    5 XLOAD(J,K)=X(K)

      IF (JEY.NE.1.AND.JEY.NE.15.AND.JEY.NE.29) GO TO 6
      CALL POSION(T,XSTA,YSTA,ZSTA)
      R2=(X(1)-XSTA)**2+(X(2)-YSTA)**2+(X(3)-ZSTA)**2
```

```
      R1=DSQRT(R2)
      IF (JEY.EQ.01) R01=R1
      IF (JEY.EQ.15) R15=R1
      IF (JEY.EQ.29) R29=R1

    6 IF (JEY.NE.1) GO TO 7
      CALL SUN (T,PSUN)
      CALL MOON(T,PMOON)
      RM=DSQRT(PSUN(1)**2+PSUN(2)**2+PSUN(3)**2)
      SA=PSUN(1)/RM
      SB=PSUN(2)/RM
      SC=PSUN(3)/RM
      CALL AZEL(X,T,AZ1,EL1,1,KEY1)
      CALL AZEL(X,T,AZ2,EL2,2,KEY2)
      CALL AZEL(X,T,AZ3,EL3,3,KEY3)
      CALL PLACE(T,X(1),X(2),X(3),XLT,XLN)

      NIN=MIN
      KHR=MIN/60
      LH10=KHR/10
      LH1 =MOD(KHR,10)
      LM10=MOD(MIN,60)/10
      LM1 =MOD(MIN,10)

    7 IF (JEY.NE.29) RETURN
      WRITE (LUOUT,100)
      WRITE (LUOUT,101),ISAT,MJD,XLT,XLN,R15,R29
      WRITE (LUOUT,102),LSTA1,AZ1,EL1,R01,LSTA2,AZ2,EL2,
     *                  LSTA3,AZ3,EL3
      WRITE (LUOUT,103),LH10,LH1,LM10,LM1,(PMOON(K),K=1,3),
     *                  SA,SB,SC
      DO 10 J=1,15
      JIN=NIN+2*(J-1)
      IF (J.NE.1.AND.J.NE.8.AND.J.NE.15.AND.ABREV) GO TO 10
      WRITE (LUOUT,104),JIN,(XLOAD(J,K),K=1,6)
   10 CONTINUE

      RETURN
      END

      SUBROUTINE AZEL(R,T,AZ,ELV,IND,KEY)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION R(3),RZ(3),S(3),PHI(3),THETA(3),XH(3)
      COMMON/LOC/XLAT,XLON,ALT,LSTA1
      COMMON/MOC/YLAT,YLON,BLT,LSTA2
      COMMON/NOC/ZLAT,ZLON,CLT,LSTA3
      COMMON/XMU/XMU,RE,XJ2
      DATA DTR/.01745329252D0/
      DATA EP/0.08181333402D0/
      DATA PI2/1.570796327D0/
      THETA(1)=XLAT*DTR
      THETA(2)=YLAT*DTR
      THETA(3)=ZLAT*DTR
```

```
      PHI(1)=XLON*DTR
      PHI(2)=YLON*DTR
      PHI(3)=ZLON*DTR
      XH(1)=ALT
      XH(2)=BLT
      XH(3)=CLT
      TD=IDINT(T/86400.0D0)
      TF= DMOD(T,86400.0D0)
      CALL GHA70(TF,TD,G,0.0D0,JM)
      G=G*DTR
      SL=DSIN(THETA(IND))
      CL=DCOS(THETA(IND))
      SP=DSIN(PHI(IND))
      CP=DCOS(PHI(IND))
      SG=DSIN(G)
      CG=DCOS(G)
      RZ(1)=+R(1)*CG+R(2)*SG
      RZ(2)=-R(1)*SG+R(2)*CG
      RZ(3)=+R(3)
      XN=RE/DSQRT(1.D0-EP*EP*DSIN(THETA(IND))**2)
      RZ(1)=RZ(1)-(XN+XH(IND))*CL*CP
      RZ(2)=RZ(2)-(XN+XH(IND))*CL*SP
      RZ(3)=RZ(3)-(XN*(1.D0-EP*EP)+XH(IND))*SL
      SP=DSIN(PHI(IND)+PI2)
      CP=DCOS(PHI(IND)+PI2)
      SL=DSIN(PI2-THETA(IND))
      CL=DCOS(PI2-THETA(IND))
      S(1)=+CP*RZ(1)+SP*RZ(2)
      S(2)=-SP*RZ(1)+CP*RZ(2)
      S(3)=+RZ(3)
      RZ(1)=+S(1)
      RZ(2)=+CL*S(2)+SL*S(3)
      RZ(3)=-SL*S(2)+CL*S(3)
      AZ=ARKTNS(360,RZ(2),RZ(1))/DTR
      RM=DSQRT(RZ(1)*RZ(1)+RZ(2)*RZ(2)+RZ(3)*RZ(3))
      SEL=RZ(3)/RM
      CEL=DSQRT(1.D0-SEL*SEL)
      ELV=ARKTNS(180,CEL,SEL)/DTR
      KEY=0
      IF (ELV.GT.0.0D0) KEY=1
      RETURN
      END

      SUBROUTINE INGEN(X,T,IDEN,NOBS,RMS,BIAS,
     *    LUOUT,WM,KSTART,KSTOP,JF,JL,TF,TL,
     *    RF,RL,LF,LR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (2H )
  101 FORMAT (6H SAT =,7X,A5,2X,5HA    =,F12.2,2X,5HX    =,F12.2)
  102 FORMAT (6H MJD =,  I12,2X,5HE    =,F12.8,2X,5HY    =,F12.2)
  103 FORMAT (6H TSEC=,F12.3,2X,5HI    =,F12.4,2X,5HZ    =,F12.2)
  104 FORMAT (6H NOBS=,  I12,2X,5HNODE=,F12.4,2X,5HXD   =,F12.7)
  105 FORMAT (6H RMS =,F12.3,2X,5HPERI=,F12.4,2X,5HYD   =,F12.7)
```

```
106 FORMAT (6H BIAS=,F12.3,2X,5HMEAN=,F12.4,2X,5HZD  =,F12.7)
107 FORMAT (18H COVARIANCE MATRIX)
108 FORMAT (8X,1HN,2X,2X,8HE COS(W),2X,8HE SIN(W),
   *          7X,1HI,6X,4HNODE,7X3HM+W,6X,4HBIAS)
109 FORMAT (1X,7(1PE10.3))
110 FORMAT(6H FIRST,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
111 FORMAT(6H LAST ,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
112 FORMAT (17H PERTURBATIONS = ,6I1,2X,
   *          24H NON-FIXED PARAMETERS = ,7I1)
    COMMON/TSTEP/TSTEP/KOUNT/KOUNT(7)
    DIMENSION X(7),EL(7),WM(49),LF(6),LR(7)
    REWIND LUOUT
    READ (LUOUT,100)
    READ (LUOUT,101),IDEN,EL(1),X(1)
    READ (LUOUT,102), MJD,EL(2),X(2)
    READ (LUOUT,103),TSEC,EL(3),X(3)
    READ (LUOUT,104),NOBS,EL(4),X(4)
    READ (LUOUT,105), RMS,EL(5),X(5)
    READ (LUOUT,106),BIAS,EL(6),X(6)
    READ (LUOUT,110),KSTART,JF,TF,RF
    READ (LUOUT,111),KSTOP, JL,TL,RL
    T=(MJD-40587)*86400.0D0+TSEC
    READ (LUOUT,100)
    READ (LUOUT,107)
    READ (LUOUT,108)
    DO 10 J=1,7
 10 READ (LUOUT,109),(WM(7*K+J-7),K=1,7)
    READ (LUOUT,100)
    READ (LUOUT,112),(LF(K),K=1,6),(LR(K),K=1,7)
    DO 30 K=1,7
 30 KOUNT(K)=LR(K)
    RETURN
    END


    SUBROUTINE OUTGEN(X,T,IDEN,NOBS,RMS,BIAS,
   *     LUOUT,WM,KSTART,KSTOP,JF,JL,TF,TL,
   *     RF,RL,LF,LR)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
100 FORMAT (2H  )
101 FORMAT (6H SAT =,7X,A5,2X,5HA    =,F12.2,2X,5HX    =,F12.2)
102 FORMAT (6H MJD =,  I12,2X,5HE    =,F12.8,2X,5HY    =,F12.2)
103 FORMAT (6H TSEC=,F12.3,2X,5HI    =,F12.4,2X,5HZ    =,F12.2)
104 FORMAT (6H NOBS=,  I12,2X,5HNODE=,F12.4,2X,5HXD   =,F12.7)
105 FORMAT (6H RMS =,F12.3,2X,5HPERI=,F12.4,2X,5HYD   =,F12.7)
106 FORMAT (6H BIAS=,F12.3,2X,5HMEAN=,F12.4,2X,5HZD   =,F12.7)
107 FORMAT (18H COVARIANCE MATRIX)
108 FORMAT (8X,1HN,2X,2X,8HE COS(W),2X,8HE SIN(W),
   *          7X,1HI,6X,4HNODE,7X,3HM+W,6X,4HBIAS)
109 FORMAT (1X,7(1PE10.3))
110 FORMAT(6H FIRST,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
111 FORMAT(6H LAST ,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
112 FORMAT (17H PERTURBATIONS = ,6I1,2X,
```

```
     *          24H NON-FIXED PARAMETERS = ,7I1)
 113 FORMAT (4H END)
     COMMON/TSTEP/TSTEP
     DIMENSION X(7),EL(7),WM(49),LF(6),LR(7)
     REWIND LUOUT
     CALL STATEL(X,EL)
     MJD=IDINT(T/86400.0D0)+40587
     TSEC=DMOD(T,86400.0D0)
     WRITE (LUOUT,100)
     WRITE (LUOUT,101),IDEN,EL(1),X(1)
     WRITE (LUOUT,102), MJD,EL(2),X(2)
     WRITE (LUOUT,103),TSEC,EL(3),X(3)
     WRITE (LUOUT,104),NOBS,EL(4),X(4)
     WRITE (LUOUT,105), RMS,EL(5),X(5)
     WRITE (LUOUT,106),BIAS,EL(6),X(6)
     WRITE (LUOUT,110),KSTART,JF,TF,RF
     WRITE (LUOUT,111),KSTOP, JL,TL,RL
     WRITE (LUOUT,100)
     WRITE (LUOUT,107)
     WRITE (LUOUT,108)
     DO 10 J=1,7
  10 WRITE (LUOUT,109), (WM(7*K+J-7),K=1,7)
     WRITE (LUOUT,100)
     WRITE (LUOUT,112), (LF(K),K=1,6), (LR(K),K=1,7)
     WRITE (LUOUT,113)
     END FILE LUOUT
     RETURN
     END


     SUBROUTINE DAPER(X,T,LUOUT)
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 100 FORMAT (1H )
 101 FORMAT (7H A     =,F14.4,3X,4H X =,F14.5,3X,6HKLAT =,F14.4)
 102 FORMAT (7H E     =,F14.8,3X,4H Y =,F14.5,3X,6HKLON =,F14.4)
 103 FORMAT (7H I     =,F14.4,3X,4H Z =,F14.5,3X,6HFCAL =,I14)
 104 FORMAT (7H NODE =,F14.4,3X,4HXD =,F14.8,3X,6HMJD  =,I14)
 105 FORMAT (7H PERI =,F14.4,3X,4HYD =,F14.8,3X,
     *          7HH/M/S =,2X,2I1,1H/,2I1,1H/,F5.2)
 106 FORMAT (7H MEAN =,F14.4,3X,4HZD =,F14.8,3X,6HSTEP =,F14.4)
     COMMON/TSTEP/TSTEP/NCAL/NCAL
     DIMENSION X(7),ELD(7),R(7)
     CALL STATEL(X,ELD)
     MJD=IDINT(T/86400.0D0)+40587
     TSEC=DMOD(T,86400.0D0)
     SC=DMOD(TSEC,60.0D0)
     MH=TSEC/3600
     MM=DMOD(TSEC/60,60.0D0)
     MHT=MH/10
     MHO=MOD(MH,10)
     MMT=MM/10
     MMO=MOD(MM,10)
     CALL PLACE(T,X(1),X(2),X(3),XLT,XLN)
```

```
WRITE (LUOUT,100)
WRITE (LUOUT,101) ELD(1),X(1),XLT
WRITE (LUOUT,102) ELD(2),X(2),XLN
WRITE (LUOUT,103) ELD(3),X(3),NCAL
WRITE (LUOUT,104) ELD(4),X(4),MJD
WRITE (LUOUT,105) ELD(5),X(5),MHT,MHO,MMT,MMO,SC
WRITE (LUOUT,106) ELD(6),X(6),TSTEP
RETURN
END
```

```
      PROGRAM BATCH
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  820 FORMAT(/)
  870 FORMAT(I4)
  850 FORMAT(A1)
  851 FORMAT(A10)
  855 FORMAT(A2)
  891 FORMAT(3A5)
  892 FORMAT(4A5)
  830 FORMAT(7I1)
  805 FORMAT(I4,I7,2F11.3,F9.2)
  808 FORMAT(/,16H UPDATE EPOCH BY,F10.4,5H DAYS)
  810 FORMAT(/,35H INITIAL STATE VECTOR AND EPOCH ARE)
  878 FORMAT(/,39H TOTAL NUMBER OF OBSERVATIONS ALLOWED =,I4)
  879 FORMAT(/,20H LAST OBSERVATION IS,I4)
  877 FORMAT(/,24H NO OBSERVATIONS IN FILE)
  880 FORMAT(/,24H INPUT ERROR DURING READ)
  946 FORMAT(/,25H FADING MEMORY SOLUTION ?)
  888 FORMAT(/,22H LIST THE STATE VECTOR,2H ",A10,3H"  )
  905 FORMAT(/,27H NON-FIXED PARAMETERS ARE =,2X,7I1)
  909 FORMAT(  2X,I5,4X,I2,I4,I3,I3,F7.3,F14.3,F10.3,3X,F6.2)
  911 FORMAT(/,4X,3HTAG,4X,2HYR,1X,3HDAY,3H HR,3H MN,
     *        7H SECOND,5X,9HRANGE(KM),4X,7HO-C(KM),4X,4HR(K))
  990 FORMAT(/,34H ENTER STARTING OBSERVATION NUMBER)
  991 FORMAT(/,34H ENTER STOPPING OBSERVATION NUMBER)
  992 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *        /,25H THE RANGE MAGNITUDE DATA)
  994 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *        /,23H THE INPUT STATE VECTOR)
  998 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *        /,36H THE UPDATED OUTPUT STATE REFERENCED,
     *        /,30H NEAR THE END OF THE DATA SPAN)
      DIMENSION XM(49),XMI(49),P(49),PI(49),WM(49),LR(7)
      DIMENSION WB(49),WC(49),PHI(49),PHT(49)
      DIMENSION ELD(7),ELR(7),ELN(7),X(7),DE(7),DX(7)
      DIMENSION H(1400),HT(1400),WA(49),LF(6)
      COMMON/TM/TIME(200)/Y/Y(200)/YMF/YMF(200)/XK/XK(1400)
      COMMON/INIT/INIT/XYZ/XE,YE,ZE/MAXDIM/MAXDIM
      COMMON/LOC/XLAT,XLON,ALT,LSTA1/SHADO/KKEEYY
      COMMON/BIAS/BIAS/NRDATA/NRDATA/R/R(200)/RS/RS(200)

      TYPE 992
      ACCEPT 851,XNAME
      OPEN (UNIT=21,ACCESS='SEQIN' ,FILE=XNAME)
    1 TYPE 994
      ACCEPT 851,YNAME
      IF (YNAME.EQ.XNAME) GO TO 1
      OPEN (UNIT=22,ACCESS='SEQIN' ,FILE=YNAME)
    2 TYPE 998
      ACCEPT 851,ZNAME
```

```
      IF (ZNAME.EQ.XNAME) GO TO 2
      IF (ZNAME.EQ.YNAME) GO TO 2
      OPEN (UNIT=23,ACCESS='SEQOUT',FILE=ZNAME)

C     ENTER STATION COORDINATES
      LSTA1=5HBLMPT
      XLAT=+38.431414D0
      XLON=282.913583D0
      ALT=-0.0247D0

      MAXDIM=200
      MAXOBS=200
      KKEEYY=0
      LRDATA=21
      LUDCIN=22
      LUOLD=23
      NPRINT=1
      RMS=0.0D0
      KITER=0
      NC=7

C     ACCEPT LSTART AND LSTOP
    4 TYPE 990
      ACCEPT 870,LSTART
      TYPE 991
      ACCEPT 870,LSTOP
      IF (LSTOP.LT.LSTART) GO TO 4

C     READ IN RANGE MAGNITUDE DATA
      REWIND LRDATA
      REWIND LUDCIN
      REWIND LUOLD
      ND=0
      MCOUNT=0
      TMIN=+1.0D+20
      TMAX=-1.0D+20
   10 READ (LRDATA,305,END=20,ERR=500),KNUM,MJD,TSEC,DIST
      IF (MJD.EQ.0) GO TO 10
      MCOUNT=MCOUNT+1
      IF (MCOUNT.LT.LSTART) GO TO 10
      IF (MCOUNT.GT.LSTOP)  GO TO 20
      ND=ND+1
      Y(ND)=DIST
      R(ND)=1.0D0
      TIME(ND)=(MJD-40587)*86400.0D0+TSEC
      IF (TIME(ND).GT.TMAX) TMAX=TIME(ND)
      IF (TIME(ND).LT.TMIN) TMIN=TIME(ND)
      IF (ND.GE.MAXOBS) GO TO 12
      GO TO 10
   12 TYPE 878,MAXOBS
   20 NRDATA=ND
      IF (ND.LE.0) TYPE 877
      IF (ND.LE.0) GO TO 4
```

```
      LSTOPP=LSTART+ND-1
      IF (LSTOPP.NE.LSTOP) TYPE 879,LSTOPP
      LSTOP=LSTOPP
      KSTART=1
      KSTOP=LSTOP-LSTART+1
      CALL ORDER(Y,TIME,ND,WMAX)
C     ENTER "FADING MEMORY SOLUTION"
    7 TYPE 946
      ACCEPT 850,LFM
      IF (LFM.NE.1HN.AND.LFM.NE.1HY) GO TO 7
C     READ IN FROM DISK FILE INITIAL GUESS FOR ELEMENTS
    8 CALL INOD(X,EPOCH,ISAT,NOBS,RMS,BIAS,LUDCIN,WM,LF,LR)
      CALL FORCE(LF(1),LF(2),LF(3),LF(4),LF(5),LF(6))
      CALL REDUCE(LR(1),LR(2),LR(3),LR(4),
     *            LR(5),LR(6),LR(7),NC,NU)
      DDTT=STPSZE(X,0.05D0)
C     COMPUTE INITIAL WPHI,WQ,XMI,P,PI
   22 DO 30 K=1,7
      DO 30 J=1,7
      K1=J+7*(K-1)
      P(K1) =0.0D0
      PI(K1)=0.0D0
      XMI(K1)=0.0D0
   30 XM(K1)=0.0D0
      DO 24 K=1,MAXOBS
   24 YMF(K)=0.0D0
C     PRINT INITIAL RESULTS
      DTIM=TIME(1)-EPOCH
      IF (DTIM.GE.0.0D0.AND.DTIM.LE.14400.0D0) GO TO 26
      CALL RTTIME(TIME(1),TB)
      DTIM=(TB-EPOCH)/86400.0D0
      TYPE 808,DTIM
      CALL FOUR (X,EPOCH,TB,-1,DDTT)
      EPOCH=TB
   26 TYPE 810
      CALL OUTOD(X,EPOCH,ISAT,ND,RMS,BIAS,LUOLD,
     *           WM,LSTART,LSTOP,NPRINT,LF,LR)
      CALL STATEL(X,ELD)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)
      BIAS=0.0D0
      ELN(7)=BIAS
C     ENTER SPECIAL VARIANCES FOR INCL AND NODE HERE
      DO 33 K=1,NU
      DO 33 J=1,NU
      K1=J+NU*(K-1)
      XM(K1)=0.0D0
      IF (K.EQ.J) XM(K1)=1.0D+8
```

```
      IF (K.EQ.J.AND.K.EQ.4) XM(K1)=2.0D-9
   33 IF (K.EQ.J.AND.K.EQ.5) XM(K1)=2.0D-9

C     COMPUTE OBSERVED MINUS COMPUTED RANGES
   40 KITER=KITER+1
      TLAST=EPOCH
      CALL ELNELR(ELN,ELR)
      CALL RADDEG(ELR,ELD)
      CALL ELSTAT(ELD,X)
      DO 80 K=1,KSTOP
      TSTART=TLAST
      TSTOP=TIME(K)
      CALL FOUR(X,TSTART,TSTOP,-1,DDTT)
      CALL POSION(TSTOP,XSA,YSA,ZSA)
      RG2=(X(1)-XSA)**2+(X(2)-YSA)**2+(X(3)-ZSA)**2
      YMF(K)=Y(K)-DSQRT(RG2)-BIAS
   80 TLAST=TSTOP

C     COMPUTE RESIDUALS, EDIT DATA POINTS, AND FADING MEMORY
    9 RMS=0.0D0
      ZND=0.0D0
      DO 84 K=1,KSTOP
      IF (LFM.EQ.1HN) GO TO 81
      REVNUM=TIME(ND)-TIME(K)
      REVNUM=REVNUM/(5.0D0*86400.0D0)
      IF (DABS(REVNUM).GT.0.0D0) PFAC=(4.0)**SNGL(REVNUM)
      IF (DABS(REVNUM).LE.0.0D0) PFAC=1.0D0
   81 IF (LFM.EQ.1HN) PFAC=1.0
      RS(K)=R(K)*PFAC
      ZND=ZND+1.0D0/RS(K)
   84 RMS=RMS+YMF(K)**2/RS(K)
      RMS=DSQRT(RMS/ZND)
      NX=ND/5+1
      IF (ND/5*5.EQ.ND) NX=ND/5
      CALL OUT(YMF,NX,5)
      CALL EDIT(RMS,XNAME,NU,LSTART,KSTOP,ND,1,LLE,KITER)
      RMS=0.0D0
      ZND=0.0D0
      JCOUNT=0
      DO 87 K=1,KSTOP
      IF (RS(K).LT.1.0D8) JCOUNT=JCOUNT+1
      ZND=ZND+1.0D0/RS(K)
   87 RMS=RMS+YMF(K)**2/RS(K)
      RMS=DSQRT(RMS/ZND)

C     COMPUTE SENSITIVITY MATRIX H AND ITS TRANSPOSE HT
      CALL DERIV(ELN,TIME,EPOCH,H,ND,NC)
      CALL TRNPSE(H,HT,ND,NU)
      CALL IVERSE(XM,XMI,NU)
      CALL DIVR(HT,RS,HT,NU,ND)
      CALL MULT(HT,H,WA,NU,ND,NU)
      CALL ADD(XMI,WA,PI,NU,NU)
      CALL IVERSE(PI,P,NU)
```

```
      CALL MULT(P,HT,XK,NU,NU,ND)
      CALL MULT(XK,YMF,DX,NU,ND,1)
      CALL ERASE(DX,DE,NC)
      CALL ADD(ELN,DE,ELN,NC,1)
      CALL ELNELR(ELN,ELR)
      CALL RADDEG(ELR,ELD)
      CALL ELSTAT(ELD,X)
      BIAS=ELN(7)

C     OUTPUT STATE VECTOR WITH OLD EPOCH
      TYPE 888,ZNAME
      NPRINT=1
      CALL LITBIG(P,WM)
      CALL OUTOD(X,EPOCH,ISAT,JCOUNT,RMS,BIAS,LUOLD,
     *           WM,LSTART,LSTOP,NPRINT,LF,LR)
      IF (RMS.LE.1.3D0.AND.KITER.GE.2) GO TO 42
      IF (KITER.LT.5) GO TO 40

C     UPDATE X AND XM TO NEW EPOCH ONE MINUTE PAST HOUR
   42 CALL RTTIME(TIME(KSTOP),TNEXT)
      CALL FOUR (X,EPOCH,TNEXT,-1,DDTT)
      EPOCH=TNEXT

C     PRINT RESULTS
      TYPE 888,ZNAME
      NPRINT=1
      CALL TRAFER(P,XM,NU,NU)
      CALL LITBIG(XM,WM)
      CALL OUTOD(X,EPOCH,ISAT,JCOUNT,RMS,BIAS,LUOLD,
     *           WM,LSTART,LSTOP,NPRINT,LF,LR)

      TYPE 820
      TYPE 911
      TYPE 820
      DO 202 K=1,KSTOP
      CALL YDHMS(TIME(K),KY,MDAY,MH,MM,SEC)
      LS=LSTART-1+K
  202 TYPE 909,LS,KY,MDAY,MH,MM,SEC,Y(K),YMF(K),RS(K)
      TYPE 820
      TYPE 820

      CLOSE (UNIT=21)
      CLOSE (UNIT=22)
      CLOSE (UNIT=23)
      STOP
  500 TYPE 880
      STOP
      END

      SUBROUTINE EDIT(RMS,XNAME,NU,LSTART,KSTOP,MAXDAT,KEY,LLE,KITER)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   80 FORMAT (A10)
   81 FORMAT (A1)
```

```
82 FORMAT (3A5)
83 FORMAT (F10.2)
84 FORMAT (1H )
90 FORMAT (I4,I7,2F11.3,F9.3)
91 FORMAT (2X,I5,4X,I2,I4,I3,I3,F7.3,F16.3,F12.3,F10.1)
98 FORMAT (/,6H RMS =,F10.3)
99 FORMAT (/,34H THE FOLLOWING OBS WILL BE DELETED,/)
   COMMON/YMF/YMF(1)/Y/Y(1)/TM/TIME(1)/R/R(1)/RS/RS(1)
   LOGICAL NPRINT,UNDER,EARLY

   NPRINT=.TRUE.
   EARLY=.TRUE.
   KF=KSTOP

40 IF (KITER.EQ.1) THRESH=50.0D0
   IF (KITER.EQ.2) THRESH=25.0D0
   IF (KITER.EQ.3) THRESH=14.0D0
   IF (KITER.EQ.4) THRESH= 9.0D0
   IF (KITER.EQ.5) THRESH= 6.0D0
   IF (KITER.GE.6) THRESH= 5.0D0
   TYPE 99
   K=0
50 IF (K.GE.KF) GO TO 60
   UNDER=.FALSE.
   K=K+1
   XHRESH=THRESH*SQRT(SNGL(RS(K))/SNGL(RS(MAXDAT)))
   IF (DABS(YMF(K)).LE.XHRESH) UNDER=.TRUE.
   MJD=40587+TIME(K)/86400.0D0
   IF (MJD.EQ.40587) GO TO 50
   CALL YDHMS(TIME(K),KR,MDAY,MH,MM,SEC)
   LW=LSTART+K-1
   IF (.NOT.UNDER) TYPE 91,LW,KR,MDAY,MH,MM,SEC,Y(K),YMF(K),XHRESH
   IF (UNDER) GO TO 50
   RS(K)=2.0D8
   GO TO 50

60 RETURN
   END
```

```
      PROGRAM DIFFCR
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  820 FORMAT(/)
  870 FORMAT(I4)
  850 FORMAT(A1)
  851 FORMAT(A10)
  855 FORMAT(A2)
  891 FORMAT(3A5)
  892 FORMAT(4A5)
  830 FORMAT(7I1)
  916 FORMAT(D20.8)
  805 FORMAT(I4,I7,2F11.3,F9.2)
  808 FORMAT(/,16H UPDATE EPOCH BY,F10.4,5H DAYS)
  810 FORMAT(/,35H INITIAL STATE VECTOR AND EPOCH ARE)
  840 FORMAT(/,26H ACCEPT MORE DATA POINTS ?)
  860 FORMAT(/,42H ENTER NUM OF OBSERVATIONS TO BE PROCESSED)
  877 FORMAT(/,24H NO OBSERVATIONS IN FILE)
  878 FORMAT(/,39H TOTAL NUMBER OF OBSERVATIONS ALLOWED =,I4)
  912 FORMAT(/,38H CONSTRAIN INCLINATION AND NODE ONLY ?)
  913 FORMAT(/,24H ENTER VARIANCE FOR NODE)
  914 FORMAT(  50H SUGGEST:1.0D-6 LOOSE,1.0D-8 AVERAGE,1.0D-10 TIGHT)
  915 FORMAT(/,31H ENTER VARIANCE FOR INCLINATION)
  879 FORMAT(/,36H LAST OBSERVATION IN FILE IS NUMBER ,I4)
  880 FORMAT(/,24H INPUT ERROR DURING READ)
  885 FORMAT(/,28H LIST INDIVIDUAL RESIDUALS ?)
  888 FORMAT(/,22H LIST THE STATE VECTOR,2H ",A10,3H" ?)
  900 FORMAT(/,31H RETURN FOR ANOTHER ITERATION ?)
  905 FORMAT(/,27H NON-FIXED PARAMETERS ARE =,2X,7I1)
  946 FORMAT(/,25H FADING MEMORY SOLUTION ?)
  906 FORMAT(  14H ANY CHANGES ?)
  907 FORMAT(/,27H ENTER NEW PARAMETERS - 7I1)
  908 FORMAT(/,34H LIST OBSERVATIONS AND RESIDUALS ?,/)
  909 FORMAT(  2X,I5,4X,I2,I4,I3,I3,F7.3,F16.3,F12.3,3X,F6.2)
  910 FORMAT(/,45H UPDATE EPOCH TO TIME NEAR LAST OBSERVATION ?)
  911 FORMAT(/,4X,3HTAG,4X,2HYR,1X,3HDAY,3H HR,3H MN,
     *        7H SECOND,7X,9HRANGE(KM),6X,7HO-C(KM),4X,4HR(K))
  920 FORMAT(/,33H COMPUTE TRACK COV MATRIX AGAIN ?)
  930 FORMAT(/,37H ANALYTICAL OR NUMERICAL O-C RANGES ?)
  940 FORMAT(/,33H ANALYTICAL OR NUMERICAL UPDATE ?)
  970 FORMAT(/,27H WEIGHT A-PRIORI ESTIMATE ?)
  980 FORMAT(/,30H RETURN TO STARTING POSITION ?)
  990 FORMAT(/,34H ENTER STARTING OBSERVATION NUMBER)
  991 FORMAT(/,34H ENTER STOPPING OBSERVATION NUMBER)
  992 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *       /,25H THE RANGE MAGNITUDE DATA)
  994 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *       /,23H THE INPUT STATE VECTOR)
  996 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *       /,36H THE UPDATED OUTPUT STATE REFERENCED,
     *       /,32H NEAR THE START OF THE DATA SPAN)
```

```
998 FORMAT(/,36H ENTER NAME OF ASCII FILE CONTAINING,
     *       /,36H THE UPDATED OUTPUT STATE REFERENCED,
     *       /,30H NEAR THE END OF THE DATA SPAN)
      DIMENSION XM(49),XMI(49),P(49),PI(49),WM(49),LR(7)
      DIMENSION WB(49),WC(49),WPHI(49),PHI(49),PHT(49)
      DIMENSION ELD(7),ELR(7),ELN(7),X(7),DE(7),DX(7)
      DIMENSION H(1400),HT(1400),WA(49),LF(6)
      COMMON/TM/TIME(200)/Y/Y(200)/YMF/YMF(200)/XK/XK(1400)
      COMMON/INIT/INIT/XYZ/XE,YE,ZE/MAXDIM/MAXDIM
      COMMON/LOC/XLAT,XLON,ALT,LSTA1/SHADO/KKEEYY
      COMMON/BIAS/BIAS/NRDATA/NRDATA/R/R(200)/RS/RS(200)
      LOGICAL BEDIT,YEDIT
      DATA KAA/"202517170312/
      DATA KBB/"202624067744/
      DATA KCC/"202344020100/
      DATA KDD/"202024067744/
      DATA KEE/"202346420336/
      DATA KFF/"711011634500/

      TYPE 992
      ACCEPT 851,XNAME
      OPEN (UNIT=21,ACCESS='SEQIN' ,FILE=XNAME)
    1 TYPE 994
      ACCEPT 851,YNAME
      IF (YNAME.EQ.XNAME) GO TO 1
      OPEN (UNIT=22,ACCESS='SEQIN' ,FILE=YNAME)
    2 TYPE 996
      ACCEPT 851,ZNAME
      IF (ZNAME.EQ.XNAME) GO TO 2
      IF (ZNAME.EQ.YNAME) GO TO 2
      OPEN (UNIT=23,ACCESS='SEQOUT',FILE=ZNAME)
    3 TYPE 998
      ACCEPT 851,WNAME
      IF (WNAME.EQ.XNAME) GO TO 3
      IF (WNAME.EQ.YNAME) GO TO 3
      IF (WNAME.EQ.ZNAME) GO TO 3
      OPEN (UNIT=24,ACCESS='SEQOUT',FILE=WNAME)

C     ENTER STATION COORDINATES
      LSTA1=5HBLMPT
      XLAT=+38.431414D0
      XLON=282.913583D0
      ALT=-0.0247D0

      LRDATA=21
      LUDCIN=22
      LUOLD=23
      LUNEW=24
      BEDIT=.FALSE.
      YEDIT=.FALSE.
      REWIND LRDATA
      REWIND LUDCIN
      REWIND LUOLD
```

```
      REWIND LUNEW
      MAXDIM=200
      MAXOBS=200
      MCOUNT=0
      KKEEYY=0
      NPRINT=1
      RMS=0.0D0
      F0=1.5D0
      NC=7
C     READ IN FROM DISK FILE INITIAL GUESS FOR ELEMENTS
      CALL INOD(X,EPOCH,ISAT,NOBS,RMS,BIAS,LUDCIN,WM,LF,LR)
      CALL FORCE(LF(1),LF(2),LF(3),LF(4),LF(5),LF(6))
   11 TYPE 905,(LR(K),K=1,7)
      TYPE 906
      GO TO 9
   23 TYPE 891,KAA,KBB,KCC
    9 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 21
      IF (LNY.NE.1HY) GO TO 23
      TYPE 907
      ACCEPT 830,(LR(K),K=1,7)
      TYPE 905,  (LR(K),K=1,7)
   21 CALL REDUCE(LR(1),LR(2),LR(3),LR(4),
     *             LR(5),LR(6),LR(7),NC,NU)
      DDTT=STPSZE(X,0.05D0)
C     COMPUTE INITIAL WPHI,WQ,XMI,P,PI
   22 DO 30 K=1,7
      DO 30 J=1,7
      K1=J+7*(K-1)
      P(K1) =0.0D0
      PI(K1)=0.0D0
      XMI(K1)=0.0D0
      WPHI(K1)=0.0D0
      IF (K.NE.J) GO TO 30
      WPHI(K1)=1.0D0
   30 CONTINUE
      DO 24 K=1,MAXDIM
   24 YMF(K)=0.0D0
      CALL BIGLIT(WM,XM)
C     ZERO OUT COVARIANCE ?
      TYPE 970
      GO TO 28
   27 TYPE 891,KAA,KBB,KCC
   28 ACCEPT 850,KNY
      IF (KNY.EQ.1HY) GO TO 32
      IF (KNY.EQ.1HN) GO TO 29
      GO TO 27
   29 DO 31 K=1,NU
      DO 31 J=1,NU
      K1=J+NU*(K-1)
```

```
      XM(K1)=0.0D0
   31 IF (K.EQ.J) XM(K1)=1.0D+8
      GO TO 100
C     ENTER SPECIAL VARIANCES FOR INCL AND NODE HERE
   32 TYPE 912
      ACCEPT 850,KNY
      IF (KNY.EQ.1HN) GO TO 100
      IF (KNY.NE.1HY) GO TO 32
      TYPE 913
      TYPE 914
      ACCEPT 916,XMINCL
      TYPE 915
      ACCEPT 916,XMNODE
      DO 33 K=1,NU
      DO 33 J=1,NU
      K1=J+NU*(K-1)
      XM(K1)=0.0D0
      IF (K.EQ.J) XM(K1)=1.0D+8
      IF (K.EQ.J.AND.K.EQ.4) XM(K1)=XMINCL
   33 IF (K.EQ.J.AND.K.EQ.5) XM(K1)=XMNODE
      GO TO 100
C     ACCEPT LSTART, LSTOP AND READ RANGE DATA
  100 IF (YEDIT) GO TO 200
      TYPE 990
      ACCEPT 870,LSTART
      TYPE 991
      ACCEPT 870,LSTOP
      IF (LSTOP.LT.LSTART) GO TO 100
  110 ND=0
    7 TYPE 946
      ACCEPT 850,LFM
      IF (LFM.NE.1HN.AND.LFM.NE.1HY) GO TO 7
      IF (LSTART.EQ.1) BEDIT=.TRUE.
      IF (LSTART.NE.1) BEDIT=.FALSE.
   10 READ (LRDATA,305,END=20,ERR=500),KNUM,MJD,TSEC,DIST
      IF (MJD.EQ.0) GO TO 10
      MCOUNT=MCOUNT+1
      IF (MCOUNT.LT.LSTART) GO TO 10
      ND=ND+1
      Y(ND)=DIST
      R(ND)=1.0D0
      TIME(ND)=(MJD-40587)*86400.0D0+TSEC
      IF (ND.GE.MAXOBS) GO TO 12
      IF (MCOUNT.GE.LSTOP)  GO TO 41
      GO TO 10
   12 TYPE 878,MAXOBS
      GO TO 41
   20 YEDIT=.TRUE.
   41 NRDATA=ND
      IF (ND.LE.0) TYPE 877
      IF (ND.LE.0) GO TO 200
```

```
      LSTOPP=LSTART+ND-1
      IF (LSTOPP.NE.LSTOP) TYPE 879,LSTOPP
      LSTOP=LSTOPP
      KSTART=1
      KSTOP=LSTOP-LSTART+1
      CALL ORDER(Y,TIME,ND,WMAX)
C     UPDATE STATE TO FIRST OBSERVATION
      DTIM=TIME(KSTART)-EPOCH
      IF (DTIM.GE.0.0D0.AND.DTIM.LE.14400.0D0) GO TO 26
      CALL RTTIME(TIME(KSTART),TB)
      DTIM=(TB-EPOCH)/86400.0D0
      TYPE 808,DTIM
      CALL FOUR (X,EPOCH,TB,-1,DDTT)
      WPHI(6)=TB-EPOCH
      EPOCH=TB
      CALL BIGLIT(WPHI,PHI)
      CALL TRNPSE(PHI,PHT,NU,NU)
      CALL MULT(XM,PHT,WA,NU,NU,NU)
      CALL MULT(PHI,WA,WB,NU,NU,NU)
      NU2=NU**2
      F1=F0*DABS(WPHI(6)/86400.0D0)+1.0D0
      DO 25 K=1,NU2
   25 XM(K)=F1*WB(K)
   26 TYPE 810
      CALL LITBIG(XM,WM)
      CALL OUTOD(X,EPOCH,ISAT,ND,RMS,BIAS,LUOLD,
     *           WM,LSTART,LSTOP,NPRINT,LF,LR)
      CALL STATEL(X,ELD)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)
      BIAS=0.0D0
      ELN(7)=BIAS
      GO TO 40


C     COMPUTE OBSERVED MINUS COMPUTED RANGES
   40 TLAST=EPOCH
      CALL ELNELR(ELN,ELR)
      CALL RADDEG(ELR,ELD)
      CALL ELSTAT(ELD,X)
      DO 80 K=KSTART,KSTOP
      TSTART=TLAST
      TSTOP=TIME(K)
      CALL FOUR(X,TSTART,TSTOP,-1,DDTT)
      CALL POSION(TSTOP,XSA,YSA,ZSA)
      RG2=(X(1)-XSA)**2+(X(2)-YSA)**2+(X(3)-ZSA)**2
      YMF(K)=Y(K)-DSQRT(RG2)-BIAS
   80 TLAST=TSTOP

C     COMPUTE RESIDUALS AND EDIT DATA POINTS
   83 RMS=0.0D0
```

```
      ZND=0.0D0
      DO 84 K=KSTART,KSTOP
      RS(K)=R(K)
      IF (LPM.EQ.1HN) GO TO 81
      REVNUM=TIME(KSTOP)-TIME(K)
      REVNUM=REVNUM/(5.0D0*86400.0D0)
      IF (DABS(REVNUM).GT.0.0D0) PFAC=(4.0)**SNGL(REVNUM)
      IF (DABS(REVNUM).LE.0.0D0) PFAC=1.0D0
      RS(K)=R(K)*PFAC
   81 ZND=ZND+1.0D0/RS(K)
   84 RMS=RMS+YMF(K)**2/RS(K)
      RMS=DSQRT(RMS/ZND)
      IF (ND.LE.25) GO TO 97
      TYPE 885
      GO TO 64
   63 TYPE 891,KAA,KBB,KCC
   64 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 78
      IF (LNY.NE.1HY) GO TO 63
   97 NX=ND/5+1
      IF (ND/5*5.EQ.ND) NX=ND/5
      CALL OUT(YMF,NX,5)
   78 CALL EDIT(RMS,XNAME,NU,LSTART,KSTOP,1,LLE)
      RMS=0.0D0
      ZND=0.0D0
      JCOUNT=0
      DO 87 K=KSTART,KSTOP
      IF (RS(K).LT.1.0D8) JCOUNT=JCOUNT+1
      ZND=ZND+1.0D0/RS(K)
   87 RMS=RMS+YMF(K)**2/RS(K)
      RMS=DSQRT(RMS/ZND)

C     COMPUTE STATE UPDATE FOR ELN
      CALL DERIV(ELN,TIME,EPOCH,H,ND,NU)
      CALL TRNPSE(H,HT,ND,NU)
      IF (ND.LE.NU) GO TO 50
      CALL IVERSE(XM,XMI,NU)
      CALL DIVR(HT,RS,HT,NU,ND)
      CALL MULT(HT,H,WA,NU,ND,NU)
      CALL ADD(XMI,WA,PI,NU,NU)
      CALL IVERSE(PI,P,NU)
      GO TO 70
   50 CALL MULT(H,XM,WA,ND,NU,NU)
      CALL MULT(XM,HT,WB,NU,NU,ND)
      CALL MULT(H,WB,WC,ND,NU,ND)
      DO 60 K=1,ND
      DO 60 J=1,ND
      K1=J+ND*(K-1)
   60 IF (K.EQ.J) WC(K1)=WC(K1)+RS(K)
      CALL IVERSE(WC,WC,ND)
      CALL MULT(WC,WA,WB,ND,ND,NU)
      CALL MULT(HT,WB,WA,NU,ND,NU)
      CALL MULT(XM,WA,WB,NU,NU,NU)
```

```
      CALL SUB(XM,WB,P,NU,NU)
   70 CALL TRNPSE(H,HT,ND,NU)
      CALL MULT(P,HT,XK,NU,NU,ND)
      CALL DIVR(XK,RS,XK,NU,ND)
      CALL MULT(XK,YMP,DX,NU,ND,1)
      CALL ERASE(DX,DE,NC)
      CALL ADD(ELN,DE,ELN,NC,1)
      CALL ELNELR(ELN,ELR)
      CALL RADDEG(ELR,ELD)
      CALL ELSTAT(ELD,X)
      BIAS=ELN(7)
C     OUTPUT STATE VECTOR WITH OLD EPOCH
      TYPE 888,ZNAME
      GO TO 66
   65 TYPE 891,KAA,KBB,KCC
   66 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) NPRINT=0
      IF (LNY.EQ.1HY) NPRINT=1
      IF (LNY.NE.1HY.AND.LNY.NE.1HN) GO TO 65
      CALL LITBIG(P,WM)
      CALL OUTOD(X,EPOCH,ISAT,JCOUNT,RMS,BIAS,LUOLD,
     *              WM,LSTART,LSTOP,NPRINT,LF,LR)
C     RETURN FOR ANOTHER ITERATION
      TYPE 900
      GO TO 67
   88 TYPE 891,KAA,KBB,KCC
   67 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 86
      IF (LNY.EQ.1HY) GO TO 40
      GO TO 88
C     UPDATE X AND XM TO NEW EPOCH ONE MINUTE PAST HOUR
   86 TYPE 910
      GO TO 69
   91 TYPE 891,KAA,KBB,KCC
   69 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) CALL TRAFER(P,XM,NU,NU)
      IF (LNY.EQ.1HN) GO TO 180
      IF (LNY.NE.1HY) GO TO 91
      CALL RTTIME(TIME(KSTOP),TNEXT)
      CALL FOUR (X,EPOCH,TNEXT,-1,DDTT)
      WPHI(6)=TNEXT-EPOCH
      EPOCH=TNEXT
      CALL BIGLIT(WPHI,PHI)
      CALL TRNPSE(PHI,PHT,NU,NU)
      CALL MULT(P,PHT,WA,NU,NU,NU)
      CALL MULT(PHI,WA,WB,NU,NU,NU)
      NU2=NU**2
      F1=F0*DABS(WPHI(6)/86400.0D0)+1.0D0
      DO 92 K=1,NU2
   92 XM(K)=F1*WB(K)
```

```
      TYPE 888,WNAME
      GO TO 52
   51 TYPE 891,KAA,KBB,KCC
   52 ACCEPT 850,LNY
      IF (LNY.EQ.1HN) NPRINT=0
      IF (LNY.EQ.1HY) NPRINT=1
      IF (LNY.NE.1HY.AND.LNY.NE.1HN) GO TO 51
      CALL LITBIG(XM,WM)
      CALL OUTOD(X,EPOCH,ISAT,JCOUNT,RMS,BIAS,LUNEW,
     *           WM,LSTART,LSTOP,NPRINT,LF,LR)
      CALL STATEL(X,ELD)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)

  180 IF (YEDIT) GO TO 200
      TYPE 840
      ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 200
      IF (LNY.NE.1HY) GO TO 180
      TYPE 860
      ACCEPT 870,ND
      IF (ND.LE.0) GO TO 200
      LSTART=LSTOP+1
      LSTOP=LSTART+ND-1
      MCOUNT=LSTART-1
      GO TO 110

C     PRINT LISTING OF FINAL RESULTS
  200 TYPE 908
      ACCEPT 850,LNY
      IF (LNY.EQ.1HN) GO TO 203
      IF (LNY.NE.1HY) GO TO 200
      TYPE 820
      TYPE 911
      TYPE 820
      DO 202 K=KSTART,KSTOP
      CALL YDHMS(TIME(K),KY,MDAY,MH,MM,SEC)
      LS=LSTART-1+K
  202 TYPE 909,LS,KY,MDAY,MH,MM,SEC,Y(K),YMF(K),RS(K)
      TYPE 820
      TYPE 820

  203 IF (BEDIT.AND.YEDIT) CALL EDIT(RMS,XNAME,NU,
     *                     LSTART,LSTOP,2,LLE)
      CLOSE (UNIT=21)
      CLOSE (UNIT=22)
      CLOSE (UNIT=23)
      CLOSE (UNIT=24)
      STOP
  500 TYPE 880
      STOP
      END
```

```
      SUBROUTINE EDIT(RMS,XNAME,NU,LSTART,KSTOP,KEY,LLE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   80 FORMAT (A10)
   81 FORMAT (A1)
   82 FORMAT (3A5)
   83 FORMAT (F10.2)
   84 FORMAT (1H )
   90 FORMAT (I4,I7,2F11.3,F9.3)
   91 FORMAT (2X,I5,4X,I2,I4,I3,I3,F7.3,F16.3,F12.3,3X,F6.2)
   98 FORMAT (/,6H RMS =,F10.3)
   97 FORMAT (/,26H ALL OBS WITH A RESIDUAL >,F7.3,6H TAGED,/)
   99 FORMAT (/,34H THE FOLLOWING OBS WILL BE DELETED,/)
  101 FORMAT (/,37H ENTER NAME OF EDITED RANGE DATA FILE)
  102 FORMAT (/,33H ENTER DELETION THRESHOLD - F10.3)
  103 FORMAT (/,30H AUTOMATIC OR MANUAL EDITING ?)
  106 FORMAT (/,21H EDIT RANGE DATA FILE,2X,1H",A10,1H")
  107 FORMAT (/,17H EDIT O-C ARRAY ?)
  104 FORMAT (   12H TYPE A OR M)
  105 FORMAT (    9H DELETE ?)
      COMMON/YMF/YMF(1)/Y/Y(1)/TM/TIME(1)/RS/RS(1)
      LOGICAL NPRINT,UNDER,EARLY,LATE
      DATA KAA/"202517170312/
      DATA KBB/"202624067744/
      DATA KCC/"202344020100/

      NPRINT=.TRUE.
      EARLY=.FALSE.
      LATE =.FALSE.
      KSTART=1
      KS=1
      KF=KSTOP
      IF (KEY.EQ.1) EARLY=.TRUE.
      IF (KEY.EQ.2) LATE =.TRUE.
      IF (EARLY) GO TO 29
      IF (LATE)  GO TO 20
      RETURN

   20 TYPE 106,XNAME
      GO TO 22
   21 TYPE 82,KAA,KBB,KCC
   22 ACCEPT 81,LLE
      IF (LLE.EQ.1HN) RETURN
      IF (LLE.NE.1HY) GO TO 21
   10 TYPE 101
      ACCEPT 80,UNAME
      IF (XNAME.EQ.UNAME) GO TO 10
      OPEN (UNIT=11,DEVICE='DSK',ACCESS='SEQOUT',FILE=UNAME)
      REWIND 11
      GO TO 28

   29 TYPE 107
      GO TO 26
   25 TYPE 82,KAA,KBB,KCC
```

```
26 ACCEPT 81,LLE
   IF (LLE.EQ.1HN) RETURN
   IF (LLE.NE.1HY) GO TO 25

28 TYPE 103
30 TYPE 104
   ACCEPT 81,LNX
   IF (LNX.EQ.1HA) GO TO 40
   IF (LNX.EQ.1HM) GO TO 55
   GO TO 30

40 TYPE 98,RMS
   TYPE 102
   ACCEPT 83,THRESH
   TYPE 99
   K=0
   KN=0
50 IF (K.GE.KF) GO TO 60
   UNDER=.FALSE.
   K=K+1
   IF (DABS(YMF(K)).LE.THRESH) UNDER=.TRUE.
   MJD=40587+TIME(K)/86400.0D0
   IF (MJD.EQ.40587) GO TO 50
   CALL YDHMS(TIME(K),KY,MDAY,MH,MM,SEC)
   IF (LATE.AND.UNDER) KN=KN+1
   IF (LATE.AND.UNDER) WRITE (11,90) KN,MJD,TSEC,Y(K),YMF(K)
   LT=LSTART+K-1
   IF (.NOT.UNDER) TYPE 91,LT,KY,MDAY,MH,MM,SEC,Y(K),YMF(K),RS(K)
   IF (LATE.OR.UNDER) GO TO 50
   RS(K)=2.0D8
   GO TO 50

55 TYPE 98,RMS
   TYPE 102
   ACCEPT 83,THRESH
   TYPE 97,THRESH
   K=0
   KN=0
57 IF (K.GE.KF) GO TO 60
   UNDER=.FALSE.
   K=K+1
   IF (DABS(YMF(K)).LE.THRESH) UNDER=.TRUE.
   MJD=40587+TIME(K)/86400.0D0
   IF (MJD.EQ.40587) GO TO 57
   CALL YDHMS(TIME(K),KY,MDAY,MH,MM,SEC)
   IF (UNDER) GO TO 56
   LT=LSTART+K-1
   TYPE 91,LT,KY,MDAY,MH,MM,SEC,Y(K),YMF(K),RS(K)
   IF (NPRINT) TYPE 105
   GO TO 52
51 TYPE 82,KAA,KBB,KCC
52 ACCEPT 81,LNX
   NPRINT=.FALSE.
```

```
   IF (LNX.EQ.1HN) UNDER=.TRUE.
   IF (LNX.NE.1HY.AND.LNX.NE.1HN) GO TO 51
56 IF (LATE.AND.UNDER) KN=KN+1
   IF (LATE.AND.UNDER) WRITE (11,90) KN,MJD,TSEC,Y(K),YMF(K)
   IF (LATE.OR.UNDER) GO TO 57
   RS(K)=2.0D8
   GO TO 57

60 IF (LATE) CLOSE (UNIT=11,DEVICE='DSK')
   RETURN
70 STOP
   END
```

ADDITIONAL SUBROUTINES

```
      SUBROUTINE PAPER(X,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (1H )
  101 FORMAT (7H A    =,F14.4,3X,4H X =,F14.5,3X,6HXLAT =,F14.4)
  102 FORMAT (7H E    =,F14.8,3X,4H Y =,F14.5,3X,6HXLON =,F14.4)
  103 FORMAT (7H I    =,F14.4,3X,4H Z =,F14.5,3X,6HPCAL =,I14)
  104 FORMAT (7H NODE =,F14.4,3X,4HXD =,F14.8,3X,6HMJD  =,I14)
  105 FORMAT (7H PERI =,F14.4,3X,4HYD =,F14.8,3X,
     *        7HH/M/S =,2X,2I1,1H/,2I1,1H/,F5.2)
  106 FORMAT (7H MEAN =,F14.4,3X,4HZD =,F14.8,3X,6HSTEP =,F14.4)
      COMMON/TSTEP/TSTEP/NCAL/NCAL
      DIMENSION X(7),ELD(7),R(7)
      CALL STATEL(X,ELD)
      MJD=IDINT(T/86400.0D0)+40587
      TSEC=DMOD(T,86400.0D0)
      SC=DMOD(TSEC,60.0D0)
      MH=TSEC/3600
      MM=DMOD(TSEC/60,60.0D0)
      MHT=MH/10
      MHO=MOD(MH,10)
      MMT=MM/10
      MMO=MOD(MM,10)
      CALL PLACE(T,X(1),X(2),X(3),XLT,XLN)
      TYPE 100
      TYPE 101,ELD(1),X(1),XLT
      TYPE 102,ELD(2),X(2),XLN
      TYPE 103,ELD(3),X(3),NCAL
      TYPE 104,ELD(4),X(4),MJD
      TYPE 105,ELD(5),X(5),MHT,MHO,MMT,MMO,SC
      TYPE 106,ELD(6),X(6),TSTEP
      RETURN
      END

      SUBROUTINE INOD(X,T,IDEN,NOBS,RMS,BIAS,LUIN,WM,LF,LR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(7),ELD(7),WM(49)
      DIMENSION LF(6),LR(7)
      COMMON/KOUNT/KOUNT(7)
      REAL XS(6)
  100 FORMAT (1H )
  101 FORMAT (6X,7X,A5,26X,F12.2)
  102 FORMAT (6X,   I12,26X,F12.2)
  103 FORMAT (6X,F12.3,26X,F12.2)
  104 FORMAT (6X,   I12,26X,F12.7)
  105 FORMAT (6X,F12.3,26X,F12.7)
  106 FORMAT (6X,F12.3,26X,F12.7)
  109 FORMAT (1X,7(1PE10.3))
  110 FORMAT (17X,6I1,26X,7I1)
      REWIND LUIN
      READ (LUIN,100)
```

```
      READ (LUIN,101),IDEN,XS(1)
      READ (LUIN,102), MJD,XS(2)
      READ (LUIN,103),TSEC,XS(3)
      READ (LUIN,104),NOBS,XS(4)
      READ (LUIN,105), RMS,XS(5)
      READ (LUIN,106),BIAS,XS(6)
      T=(MJD-40587)*86400.0D0+TSEC
      DO 5 J=1,6
    5 X(J)=XS(J)
      DO 10 J=1,5
   10 READ (LUIN,100)
      DO 20 J=1,7
   20 READ (LUIN,109),(WM(7*K+J-7),K=1,7)
      READ (LUIN,100)
      READ (LUIN,110),(LF(K),K=1,6),(LR(K),K=1,7)
      DO 30 K=1,7
   30 KOUNT(K)=LR(K)
      RETURN
      END

      SUBROUTINE RTTIME(TA,TB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      TI=TA+0.01D0
      TDAY=IDINT(TI/86400.0D0)
      TSEC=DMOD (TI,86400.0D0)
      HOUR=IDINT(TSEC/3600.0D0)
      TB=86400.0D0*TDAY+3600.0D0*HOUR+60.0D0
      RETURN
      END

      SUBROUTINE FANDG(X,Z,T,TN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION X(7),Z(7)
      DT1=T-TN
      IF (DABS(DT1).LT.1.0D-5) DT1=0.0D0
      DT2=DT1*DT1
      DT3=DT1*DT2
      RN1=DSQRT(X(1)**2+X(2)**2+X(3)**2)
      VR1=(X(1)*X(4)+X(2)*X(5)+X(3)*X(6))/RN1
      RN2=RN1*RN1
      RN3=RN1*RN2
      RN4=RN1*RN3
      XKA=XMU/RN3
      XKB=XKA*VR1/RN1
      F1=1.0D0-XKA*DT2/2+XKB*DT3/2
      G1=DT1-XKA*DT3/6
      F2=-XKA*DT1+3*XKB*DT2/2
      G2=1.0D0-XKA*DT2/2
      Z(1)=X(1)*F1+X(4)*G1
      Z(2)=X(2)*F1+X(5)*G1
      Z(3)=X(3)*F1+X(6)*G1
      Z(4)=X(1)*F2+X(4)*G2
```

AD-A060 743 NAVAL RESEARCH LAB WASHINGTON D C F/G 22/2

THE ORBIT DETERMINATION OF SOLRAD 11.(U)

AUG 78 R R DASENBROCK

UNCLASSIFIED NRL-MR-3809 SBIE-AD-E000 222 NL

2 OF 2

AD A060743

END

DATE FILMED

01 -79

DDC

```
      Z(5)=X(2)*F2+X(5)*G2
      Z(6)=X(3)*F2+X(6)*G2
      RETURN
      END

      SUBROUTINE OUTOD(X,T,IDEN,NOBS,RMS,BIAS,
     *   LUOUT,WM,LSTART,LSTOP,NPRINT,LF,LR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (2H  )
  101 FORMAT (6H SAT =,7X,A5,2X,5HA    =,F12.2,2X,5HX    =,F12.2)
  102 FORMAT (6H MJD =,  I12,2X,5HE    =,F12.8,2X,5HY    =,F12.2)
  103 FORMAT (6H TSEC=,F12.3,2X,5HI    =,F12.4,2X,5HZ    =,F12.2)
  104 FORMAT (6H NOBS=,  I12,2X,5HNODE=,F12.4,2X,5HXD   =,F12.7)
  105 FORMAT (6H RMS =,F12.3,2X,5HPERI=,F12.4,2X,5HYD   =,F12.7)
  106 FORMAT (6H BIAS=,F12.3,2X,5HMEAN=,F12.4,2X,5HZD   =,F12.7)
  107 FORMAT (18H COVARIANCE MATRIX)
  108 FORMAT (8X,1HN,2X,2X,8HE COS(W),2X,8HE SIN(W),
     *        7X,1HI,6X,4HNODE,7X3HM+W,6X,4HBIAS)
  109 FORMAT (1X,7(1PE10.3))
  110 FORMAT(6H FIRST,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
  111 FORMAT(6H LAST ,I4,I8,2X,5HTSEC=,F12.3,2X,5HRNG =,F12.2)
  112 FORMAT (17H PERTURBATIONS = ,6I1,2X,
     *        24H NON-FIXED PARAMETERS = ,7I1)
  113 FORMAT (4H END)
      COMMON/TM/TIME(1)/Y/Y(1)/TSTEP/TSTEP
      DIMENSION X(7),EL(7),WM(49),LF(6),LR(7)
      REWIND LUOUT
      CALL STATEL(X,EL)
      MJD=IDINT(T/86400.0D0)+40587
      TSEC=DMOD(T,86400.0D0)
      KSTART=1
      KSTOP=LSTOP-LSTART+1
      JF=40587+TIME(KSTART)/86400.0D0
      JL=40587+TIME(KSTOP )/86400.0D0
      TF=DMOD (TIME(KSTART),86400.0D0)
      TL=DMOD (TIME(KSTOP ),86400.0D0)
      RF=Y(KSTART)
      RL=Y(KSTOP )
      WRITE (LUOUT,100)
      WRITE (LUOUT,101),IDEN,EL(1),X(1)
      WRITE (LUOUT,102), MJD,EL(2),X(2)
      WRITE (LUOUT,103),TSEC,EL(3),X(3)
      WRITE (LUOUT,104),NOBS,EL(4),X(4)
      WRITE (LUOUT,105), RMS,EL(5),X(5)
      WRITE (LUOUT,106),BIAS,EL(6),X(6)
      WRITE (LUOUT,110),LSTART,JF,TF,RF
      WRITE (LUOUT,111),LSTOP, JL,TL,RL
      WRITE (LUOUT,100)
      WRITE (LUOUT,107)
      WRITE (LUOUT,108)
      DO 10 J=1,7
   10 WRITE (LUOUT,109),(WM(7*K+J-7),K=1,7)
      WRITE (LUOUT,100)
```

```
WRITE (LUOUT,112), (LF(K),K=1,6), (LR(K),K=1,7)
WRITE (LUOUT,113)
END FILE LUOUT
IF (NPRINT.EQ.0) RETURN
TYPE 100
TYPE 101,IDEN,EL(1),X(1)
TYPE 102, MJD,EL(2),X(2)
TYPE 103,TSEC,EL(3),X(3)
TYPE 104,NOBS,EL(4),X(4)
TYPE 105, RMS,EL(5),X(5)
TYPE 106,BIAS,EL(6),X(6)
TYPE 110,LSTART,JF,TF,RF
TYPE 111,LSTOP ,JL,TL,RL
TYPE 100
TYPE 112,(LF(K),K=1,6),(LR(K),K=1,7)
RETURN
END
SUBROUTINE FORCE(K2,K3,K4,KDRAG,KSUN,KMOON)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/LOG/LJ2,LJ3,LJ4,LDRAG,LSUN,LMOON
COMMON/FAC/FAC2,FAC3,FAC4,DXMU,XMUS,XMUM
COMMON/XMU/XMU,RE,XJ2/KEY/KEYS,KEYM
LOGICAL LJ2,LJ3,LJ4,LDRAG,LSUN,LMOON
DATA  RE/6378.135D0/
DATA  DXMU/398601.5D0/
DATA  XMU/398601.5D0/
DATA  XMUM/4902.778D0/
DATA  XMUS/1.3271545D11/
DATA  XJ2/1.08265D-3/
DATA  XJ3/-2.5450D-6/
DATA  XJ4/-1.6715D-6/
FAC2=XMU*XJ2*RE**2
FAC3=XMU*XJ3*RE**3
FAC4=XMU*XJ4*RE**4
KEYS=0
KEYM=0
LJ2=.FALSE.
LJ3=.FALSE.
LJ4=.FALSE.
LDRAG=.FALSE.
LSUN=.FALSE.
LMOON=.FALSE.
IF (K2.EQ.0) LJ2=.TRUE.
IF (K3.EQ.0) LJ3=.TRUE.
IF (K4.EQ.0) LJ4=.TRUE.
IF (KDRAG.EQ.0) LDRAG=.TRUE.
IF (KSUN.EQ.0) LSUN=.TRUE.
IF (KMOON.EQ.0) LMOON=.TRUE.
RETURN
END

SUBROUTINE DEGRAD(A,B)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      DIMENSION A(1),B(1)
      DATA  DTR/0.01745329251994 3D0/
      B(1)=A(1)
      B(2)=A(2)
      DO 10 K=3,6
   10 B(K)=A(K)*DTR
      RETURN
      END

      SUBROUTINE RADDEG(A,B)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1)
      DATA  RTD/57.2957795130 82D0/
      B(1)=A(1)
      B(2)=A(2)
      DO 10 K=3,6
   10 B(K)=A(K)*RTD
      RETURN
      END

      SUBROUTINE ELRELN(ELR,ELN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION ELR(1),ELN(1)
      ELN(1)=DSQRT(XMU/ELR(1)**3)
      ELN(2)=ELR(2)*DCOS(ELR(5))
      ELN(3)=ELR(2)*DSIN(ELR(5))
      ELN(4)=ELR(3)
      ELN(5)=ELR(4)
      ELN(6)=ELR(5)+ELR(6)
      RETURN
      END

      SUBROUTINE ELNELR(ELN,ELR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION ELN(1),ELR(1)
      DATA  TPI/6.283185307179 5D0/
      ELR(1)=(XMU/ELN(1)**2)**(1.0D0/3.0D0)
      ELR(2)=DSQRT(ELN(2)**2+ELN(3)**2)
      ELR(3)=ELN(4)
      ELR(4)=ELN(5)
      ELR(5)=ARKTNS(360,ELN(2),ELN(3))
      ELR(6)=ELN(6)-ELR(5)
      IF (ELR(6).LT.0.0D0) ELR(6)=ELR(6)+TPI
      RETURN
      END

      SUBROUTINE SOLVE(X,TSTART,TSTOP,INT,DT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/TSTEP/TSTEP/NCAL/NCAL
      COMMON/OBSC/OBSC/PDNINE/PDNINE,ISAT,LUOUT
      DIMENSION X(7)
```

```
      LOGICAL OBSC,PDNINE
      IF (INT.LT.0) JNT=0
      IF (INT.GE.0) JNT=INT
      NCAL=0
      DDT=DT
      IF (DABS(DDT).LT.1.0D-4) DDT=STPSZE(X,0.032D0)
      TSTEP=DDT
      OBSC=.FALSE.
      PDNINE=.FALSE.
      IF (INT.GE.0) CALL PAPER(X,TSTART)
      CALL NINE(X,TSTART,TSTOP,JNT,DDT)
      IF (INT.GE.0) CALL PAPER(X,TSTOP)
      RETURN
      END

      SUBROUTINE FOUR(X,TSTART,TSTOP,INT,DT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2/SHADO/KKEEYY
      COMMON/TSTEP/TSTEP/NCAL/NCAL
      DIMENSION X(7),XLOAD(7)
      LOGICAL LPRINT
      NCAL=0
      LCON=1
      DDT=DT
      LPRINT=.TRUE.
      IF (INT.EQ.-1) LPRINT=.FALSE.
      IF (DABS(DDT).LT.1.0D-4) DDT=STPSZE(X,0.032D0)
      XMSTEP=(TSTOP-TSTART)/DDT
      YMSTEP=DABS(XMSTEP)+0.0001D0
      NMSTEP=IDINT(YMSTEP)
      IF (NMSTEP.LT.1) NMSTEP=1
      TMSTEP=(TSTOP-TSTART)/NMSTEP
      TSTEP=TMSTEP
      X(7)=TSTART
      IF (LPRINT) CALL PAPER(X,X(7))
   10 CALL RUK(X)
      IF (KKEEYY.EQ.0) GO TO 15
      TLOAD=X(7)
      DO 12 K=1,6
   12 XLOAD(K)=X(K)
      CALL SHADOW(TLOAD,XLOAD)
   15 IF (DABS(TSTOP-X(7)).LE.1.0D0) GO TO 30
      IF (TSTOP.GT.TSTART.AND.X(7).GE.TSTOP) GO TO 30
      IF (TSTOP.LT.TSTART.AND.X(7).LE.TSTOP) GO TO 30
      IF (INT.EQ.0) GO TO 20
      IF (LCON/INT*INT.NE.LCON) GO TO 20
      IF (LPRINT) CALL PAPER(X,X(7))
   20 LCON=LCON+1
      GO TO 10
   30 IF (LPRINT) CALL PAPER(X,X(7))
      RETURN
      END
```

```
      SUBROUTINE DFQ(X,DX)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/TSTEP/TSTEP
      DIMENSION X(7),DX(7)
      CALL DIFFEQ(X(7),X(1),X(2),X(3),X(4),
     *            X(5),X(6),XDD,YDD,ZDD)
      DX(1)=X(4)
      DX(2)=X(5)
      DX(3)=X(6)
      DX(4)=XDD
      DX(5)=YDD
      DX(6)=ZDD
      DX(7)=1.0D0
      DO 10 K=1,7
   10 DX(K)=TSTEP*DX(K)
      RETURN
      END

      SUBROUTINE RUK(X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(7),D(7),F(7),X(7)
      N=7
      CALL DFQ(X,D)
      DO 10 K=1,N
   10 U(K)=X(K)+0.5D0*D(K)
      CALL DFQ(U,F)
      DO 20 K=1,N
      D(K)=D(K)+2.0D0*F(K)
   20 U(K)=X(K)+0.5D0*F(K)
      CALL DFQ(U,F)
      DO 30 K=1,N
      D(K)=D(K)+2.0D0*F(K)
   30 U(K)=X(K)+F(K)
      CALL DFQ(U,F)
      DO 40 K=1,N
   40 X(K)=X(K)+(D(K)+F(K))/6.0D0
      RETURN
      END

      DOUBLE PRECISION FUNCTION STPSZE(X,STEPCN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION X(7),ELD(7)
      CALL STATEL(X,ELD)
      RMIN=ELD(1)*(1.0D0-ELD(2))
      ANGM=DSQRT(XMU*ELD(1)*(1.0D0-ELD(2)**2))
      STPSZE=STEPCN*RMIN**2/ANGM
      RETURN
      END

      SUBROUTINE DIFFEQ(D,U,V,W,UD,VD,WD,UDD,VDD,WDD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL R4,R5,R6,R7,CA,CB,CC,CD,CE,CF,CG,CH,CI
```

```
      REAL DA1,DA2,DA3,DA4,DC,ZA,ZB,ZC,ZD,ZE,ZF
      REAL RS,DRS,XSUN,DXSUN,RM,DRM,XMOON,DXMOON
      REAL V1,V2,RHO,RHON,SF
      COMMON/LOG/LJ2,LJ3,LJ4,LDRAG,LSUN,LMOON
      COMMON/ATMR/ATMR/NCAL/NCAL
      DIMENSION XS(6),XM(6)
      LOGICAL LJ2,LJ3,LJ4,LDRAG,LSUN,LMOON
      COMMON/FAC/FAC2,FAC3,FAC4,DXMU,XMUS,XMUM

      T=D
      R2=U*U+V*V+W*W
      R1=DSQRT(R2)
      R3=R1*R2
      R4=R2*R2
      R5=R2*R3
      DA1=W/R1
      DA2=DA1*DA1
      XEARTH=DXMU/R3
      UDD=-XEARTH*U
      VDD=-XEARTH*V
      WDD=-XEARTH*W
      NCAL=NCAL+1

      IF (LJ2) GO TO 10
      CA=FAC2/R5
      CB=7.5E0*DA2-1.5E0
      CC=CB-3.0E0
      UDD=UDD+CA*CB*U
      VDD=VDD+CA*CB*V
      WDD=WDD+CA*CC*W

   10 IF (LJ3) GO TO 20
      R6=R3*R3
      DA3=DA1*DA2
      CD=FAC3/R6
      CE=17.5E0*DA3-7.50E0*DA1
      CF=17.5E0*DA3-15.0E0*DA1
      UDD=UDD+CD*CE*U
      VDD=VDD+CD*CE*V
      WDD=WDD+CD*CF*W+1.5E0*FAC3/R5

   20 IF (LJ4) GO TO 30
      R7=R3*R4
      DA4=DA2*DA2
      CG=FAC4/R7
      CH=39.375E0*DA4-26.25E0*DA2+1.875E0
      CI=39.375E0*DA4-43.75E0*DA2+9.375E0
      UDD=UDD+CG*CH*U
      VDD=VDD+CG*CH*V
      WDD=WDD+CG*CI*W

   30 IF (LDRAG) GO TO 35
      IF (R1.GE.7000.0E0) GO TO 35
```

```
      IF (R1.LE.6498.0E0) GO TO 32
      DC=2.0E0
      RHON=35.36E0
      ZA=-0.738571E+4
      ZB=-0.645263E+4
      ZC=+0.402901E+1
      ZD=-0.887715E+3
      ZE=+0.453687E+2
      ZF=(R1+ZA)*ZE/((R1+ZB)*ZD)
      RHO=RHON*ZF**ZC
      GO TO 34
32    RHO=RHON*EXP((6498.0-R1)/11.0)
34    V2=UD*UD+VD*VD+WD*WD
      V1=SQRT(V2)
      SF=DC*0.5E0*RHO*V1*ATMR
      UDD=UDD-SF*UD
      VDD=VDD-SF*VD
      WDD=WDD-SF*WD

35    IF (LSUN) GO TO 40
      CALL SUN (T,XS)
      RS=XS(1)**2+XS(2)**2+XS(3)**2
      RS=SQRT(RS)
      DRS=(XS(1)-U)**2+(XS(2)-V)**2+(XS(3)-W)**2
      DRS=SQRT(DRS)
      XSUN =XMUS/RS**3
      DXSUN =XMUS/DRS**3
      UDD=UDD+(DXSUN-XSUN-6.65D-19)*XS(1)-DXSUN*U
      VDD=VDD+(DXSUN-XSUN-6.65D-19)*XS(2)-DXSUN*V
      WDD=WDD+(DXSUN-XSUN-6.65D-19)*XS(3)-DXSUN*W

40    IF (LMOON) GO TO 50
      CALL MOON(T,XM)
      RM=XM(1)**2+XM(2)**2+XM(3)**2
      RM=SQRT(RM)
      DRM=(XM(1)-U)**2+(XM(2)-V)**2+(XM(3)-W)**2
      DRM=SQRT(DRM)
      XMOON=XMUM/RM**3
      DXMOON=XMUM/DRM**3
      UDD=UDD+(DXMOON-XMOON)*XM(1)-DXMOON*U
      VDD=VDD+(DXMOON-XMOON)*XM(2)-DXMOON*V
      WDD=WDD+(DXMOON-XMOON)*XM(3)-DXMOON*W

50    RETURN
      END
      SUBROUTINE ORBIT (ELD,X,T,TEPOCH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      REAL S1PHI,C1PHI,S2PHI,C2PHI,S3PHI,C3PHI
      REAL SI,CI,SI2,XC2,U,UN,XINC,BMEAN
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION ELD(7),ELR(7),ELN(7),X(7)
      CALL DEGRAD(ELD,ELR)
      CALL ELRELN(ELR,ELN)
```

```
XN=ELN(1)
AMEAN=XN*(T-TEPOCH)
XC2=-XJ2*(RE/ELD(1))**2
BMEAN=AMEAN
UN=ELN(6)
U=UN+BMEAN
XINC=ELN(4)
SI=SIN(XINC)
CI=COS(XINC)
SI2=SI**2
S1PHI=SIN(U) - SIN(UN)
C1PHI=COS(U) - COS(UN)
S2PHI=SIN(2*U)-SIN(2*UN)
C2PHI=COS(2*U)-COS(2*UN)
S3PHI=SIN(3*U)-SIN(3*UN)
C3PHI=COS(3*U)-COS(3*UN)
ELN(1)=ELN(1)+2.25*ELN(1)*XC2*SI2*C2PHI
ELN(2)=ELN(2)+XC2*((-1.5+1.875*SI2)*C1PHI-0.875*SI2*C3PHI)
ELN(3)=ELN(3)+XC2*((-1.5+2.625*SI2)*S1PHI-0.875*SI2*S3PHI)
ELN(4)=ELN(4)-0.75*XC2*SI*CI*C2PHI
ELN(5)=ELN(5)+1.5*XC2*CI*(BMEAN-0.5*S2PHI)
ELN(6)=ELN(6)-2.25*XC2*SI2*COS(2*UN)*BMEAN
ELN(6)=ELN(6)+1.50*XC2*(4.0*SI2-3.0)*BMEAN
ELN(6)=ELN(6)+0.375*XC2*(2.0-5.0*SI2)*S2PHI
ELN(6)=ELN(6)+AMEAN
CALL ELNELR(ELN,ELR)
CALL RADDEG(ELR,ELD)
CALL ELSTAT(ELD,X)
RETURN
END

SUBROUTINE STATEL (X,ELD)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/XMU/XMU,RE,XJ2
DIMENSION X(7), ELD(7), B(3)
DATA  PI/3.141592653589790O/
DATA  TPI/6.2831853071795D0/
DATA  RTD/57.2957795130820O/
B(1)=X(2)*X(6)-X(3)*X(5)
B(2)=X(3)*X(4)-X(1)*X(6)
B(3)=X(1)*X(5)-X(2)*X(4)
R2=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)
V2=X(4)*X(4)+X(5)*X(5)+X(6)*X(6)
B2=B(1)*B(1)+B(2)*B(2)+B(3)*B(3)
AA=X(1)*X(4)+X(2)*X(5)+X(3)*X(6)
R1=DSQRT(R2)
B1=DSQRT(B2)
AA=AA/XMU
P1=B2/XMU
C3=V2-2.0D0*XMU/R1
SMA=-XMU/C3
ECC=DSQRT(DABS(1.0D0+C3*P1/XMU))
XINC=ARKTNS(180,B(3),DSQRT(B(1)**2+B(2)**2))
```

```
      XNODE=ARKTNS(360,-B(2),B(1))
      THETA=ARKTNS(360,(P1-R1),B1*AA)
      ARGLAT=ARKTNS(360,X(2)*B(1)-X(1)*B(2),X(3)*B1)
      PERI=ARGLAT-THETA
      IF (PERI.LT.0.0D0) PERI=PERI+TPI
      F1=AA*XMU/DSQRT(XMU*SMA)
      F2=1.0D0-R1/SMA
      IF (DABS(ECC).GE.1.0D-8) GO TO 10
      COSE=1.0D0
      SINE=0.0D0
      GO TO 20
   10 SINE=F1/ECC
      COSE=F2/ECC
   20 E=ARKTNS(360,COSE,SINE)
      XMEAN=(E-ECC*SINE)
      IF (ECC.EQ.0.0D0) XMEAN=THETA
      ELD(1)=SMA
      ELD(2)=ECC
      ELD(3)=XINC*RTD
      ELD(4)=XNODE*RTD
      ELD(5)=PERI*RTD
      ELD(6)=XMEAN*RTD
      RETURN
      END

      SUBROUTINE ELSTAT (ELD,X)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      DIMENSION X(7), ELD(7), A(3,2)
      DATA  DTR/0.01745329251994 3D0/
      SNI=DSIN(ELD(3)*DTR)
      CNI=DCOS(ELD(3)*DTR)
      SOM=DSIN(ELD(4)*DTR)
      COM=DCOS(ELD(4)*DTR)
      XM=DMOD(ELD(6),360.0D0)*DTR
      ECC=ELD(2)
      E=XKEP(ECC,XM,1.0D-10)
      SINE=DSIN(E)
      COSE=DCOS(E)
      STA=DSQRT(1.0D0-ECC**2)*SINE/(1.0D0-ECC*COSE)
      CTA=(COSE-ECC)/(1.0D0-ECC*COSE)
      TAA=ARKTNS(180,CTA,STA)
      TBB=TAA+DTR*ELD(5)
      CBA=DCOS(TBB)
      SBA=DSIN(TBB)
      A(1,1)=+COM*CBA-SOM*CNI*SBA
      A(2,1)=+SOM*CBA+COM*CNI*SBA
      A(3,1)=+SNI*SBA
      A(1,2)=-COM*SBA-SOM*CNI*CBA
      A(2,2)=-SOM*SBA+COM*CNI*CBA
      A(3,2)=+SNI*CBA
      P=ELD(1)*(1.0D0-ECC**2)
      R=P/(1.0D0+ECC*CTA)
```

```
      VR=ECC*STA*DSQRT(XMU/P)
      VT=DSQRT(XMU*(2.0D0/R-1.0D0/ELD(1))-VR*VR)
      DO 10 K=1,3
      X(K)=R*A(K,1)
   10 X(K+3)=VR*A(K,1)+VT*A(K,2)
      RETURN
      END

      FUNCTION ARKTNS (N,X,Y)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA  PI/3.14159265358979D0/
      DATA  TPI/6.283185307179586D0/
      IF (X.NE.0.0D0) GO TO 10
      IF (Y.GT.0.0D0) T=0.5D0*PI
      IF (Y.LT.0.0D0) T=1.5D0*PI
      IF (Y.EQ.0.0D0) T=0.0D0
      GO TO 20
   10 T=DATAN(Y/X)
      IF (X.LT.0.0D0) T=T+PI
      IF (T.LT.0.0D0) T=T+TPI
   20 IF (N.EQ.360) GO TO 30
      IF (T.GT.PI) T=T-TPI
   30 ARKTNS=T
      RETURN
      END

      FUNCTION SKEP (ECC,XM,TOL)
  100 FORMAT (10X,40H ** KEPLERS EQUATION DID NOT CONVERGE **)
      EOLD=XM
      DO 10 K=1,20
      SEC=SIN(EOLD)*ECC
      CEC=COS(EOLD)*ECC
      ENEW=(XM+SEC-EOLD*CEC)/(1.0E0-CEC)
      DE=ABS(ENEW-EOLD)
      IF (DE.LE.TOL) GO TO 20
   10 EOLD=ENEW
      TYPE 100
      STOP
   20 SKEP=ENEW
      RETURN
      END

      DOUBLE PRECISION FUNCTION XKEP (ECC,XM,TOL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (10X,40H ** KEPLERS EQUATION DID NOT CONVERGE **)
      EOLD=XM
      DO 10 K=1,100
      SEC=DSIN(EOLD)*ECC
      CEC=DCOS(EOLD)*ECC
      ENEW=(XM+SEC-EOLD*CEC)/(1.0D0-CEC)
      DE=DABS(ENEW-EOLD)
      IF (DE.LE.TOL) GO TO 20
   10 EOLD=ENEW
```

```
      TYPE 100
      STOP
   20 XKEP=ENEW
      RETURN
      END

      SUBROUTINE POSION(T,X1,Y1,Z1)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/XMU/XMU,RE,XJ2
      COMMON/LOC/XLAT,XLON,ALT,LSTA1
      COMMON/INIT/INIT/XYZ/XE,YE,ZE
      DATA DTR/0.01745329251994 3D0/
      DATA ECCN/0.0818188108D0/
      DATA INIT/0/
      T1=IDINT(T/86400.0D0)
      T2=DMOD (T,86400.0D0)
      CALL GHA70(T2,T1,GHAN,0.0D0,OMEGA)
      GHAN=GHAN*DTR
      SINT=DSIN(GHAN)
      COST=DCOS(GHAN)
      IF (INIT.GT.0) GO TO 10
      SLAT=DSIN(XLAT*DTR)
      CLAT=DCOS(XLAT*DTR)
      SLON=DSIN(XLON*DTR)
      CLON=DCOS(XLON*DTR)
      FACT=1.0D0/DSQRT(1.0D0-ECCN**2*SLAT**2)
      XE=(RE+ALT)*CLAT*CLON*FACT
      YE=(RE+ALT)*CLAT*SLON*FACT
      ZE=(RE+ALT)*SLAT*FACT*(1.0D0-ECCN**2)
   10 X1=+XE*COST-YE*SINT
      Y1=+XE*SINT+YE*COST
      Z1=+ZE
      INIT=INIT+1
      RETURN
      END

      SUBROUTINE PLACE (T,X1,Y1,Z1,XLAT,XLON)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA  DTR/0.01745329251994 3D0/
      T1=IDINT(T/86400.0D0)
      T2=DMOD (T,86400.0D0)
      CALL GHA70 (T2,T1,GHAN,0.0D0,OMEGA)
      GHAN=GHAN*DTR
      SINT=DSIN(GHAN)
      COST=DCOS(GHAN)
      X= X1*COST+Y1*SINT
      Y=-X1*SINT+Y1*COST
      Z=+Z1
      XMAG=DSQRT(X*X+Y*Y)
      XLAT=DATAN(Z/XMAG)/DTR
      XLON=ARKTNS(360,X,Y)/DTR
      RETURN
      END
```

```
      SUBROUTINE GHA70 (TSEC,TDAY,GHAN,DA,OMEGA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DATA  RTD/57.295779513082D0/
      D=TDAY+7305.0D0
      OMEGA=0.00417807462D0/(1.0D0+5.21D-13*D)
      DF=DMOD(0.98564735D0*D,360.0D0)
      TEMP=100.07554D0+DF+2.9015D-13*D**2+OMEGA*TSEC
      GHAN=DMOD(TEMP+DA*RTD,360.0D0)
      IF (GHAN.LT.0.0D0) GHAN=GHAN+360.0D0
      RETURN
      END

      SUBROUTINE YDHMS(T,KY,KDAY,KH,MN,SEC)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION MYST(20)
      DATA MYST/40951,41316,41682,42047,42412,
     *          42777,43143,43508,43873,44238,
     *          44604,44969,45334,45699,46065,
     *          0,0,0,0,0/
      MJD=40587+T/86400.0D0
      DO 10 K=1,20
      IF (MJD.LE.MYST(K).OR.MJD.GT.MYST(K+1)) GO TO 10
      KDAY=MJD-MYST(K)
      KY=70+K
      GO TO 20
   10 CONTINUE
      STOP
   20 TSEC=DMOD(T,86400.0D0)
      KH=TSEC/3600.0D0
      TSEC=TSEC-KH*3600.0D0
      MN=TSEC/60.0D0
      SEC=TSEC-MN*60.0D0
      RETURN
      END
      SUBROUTINE SUN(T,XP)
      DOUBLE PRECISION T,TB,TM,TF,XP,XB,XM,XF,XJB,XJM,XJF
      DIMENSION XP(6),XB(6),XM(6),XF(6)
      COMMON/SSAVE/TB,TM,TF,AX,BX,CX,AY,BY,CY,AZ,BZ,CZ
      COMMON/KEY/KEYS,KEYM
      DATA KEYS/0/
      IF (T.LE.TF.AND.T.GE.TB.AND.KEYS.EQ.1) GO TO 20
      TM=T
      TF=T+432000.0
      TB=T-432000.0
      KEYS=1
      CALL SUNPS(TB/86400.0D0,XB)
      CALL SUNPS(TM/86400.0D0,XM)
      CALL SUNPS(TF/86400.0D0,XF)
      T1=TB-TM
      T2=TF-TB
      T3=TF-TM
      X1=XF(1)-XM(1)
```

```
      Y1=XF(2)-XM(2)
      Z1=XF(3)-XM(3)
      X2=XM(1)-XB(1)
      Y2=XM(2)-XB(2)
      Z2=XM(3)-XB(3)
      T123=1.0/(T1*T2*T3)
      T12=T1**2
      T32=T3**2
      AX=XM(1)
      AY=XM(2)
      AZ=XM(3)
      BX=X1*T12+X2*T32
      BY=Y1*T12+Y2*T32
      BZ=Z1*T12+Z2*T32
      CX=X1*T1+X2*T3
      CY=Y1*T1+Y2*T3
      CZ=Z1*T1+Z2*T3
      BX=-BX*T123
      BY=-BY*T123
      BZ=-BZ*T123
      CX=+CX*T123
      CY=+CY*T123
      CZ=+CZ*T123
   20 TT1=T -TM
      TT2=TT1**2
      XP(1)=AX+BX*TT1+CX*TT2
      XP(2)=AY+BY*TT1+CY*TT2
      XP(3)=AZ+BZ*TT1+CZ*TT2
      RETURN
      END

      SUBROUTINE MOON(T,XP)
      DOUBLE PRECISION T,TB,TM,TF,XP,XB,XM,XF,XJB,XJM,XJF
      DIMENSION XP(6),XB(6),XM(6),XF(6)
      COMMON/MSAVE/TB,TM,TF,AX,BX,CX,AY,BY,CY,AZ,BZ,CZ
      COMMON/KEY/KEYS,KEYM
      DATA KEYM/0/
      IF (T.LE.TF.AND.T.GE.TB.AND.KEYM.EQ.1) GO TO 20
      TM=T
      TF=T+43200.0
      TB=T-43200.0
      KEYM=1
      CALL MOONPS(TB/86400.0D0,XB)
      CALL MOONPS(TM/86400.0D0,XM)
      CALL MOONPS(TF/86400.0D0,XF)
      T1=TB-TM
      T2=TF-TB
      T3=TF-TM
      X1=XF(1)-XM(1)
      Y1=XF(2)-XM(2)
      Z1=XF(3)-XM(3)
      X2=XM(1)-XB(1)
      Y2=XM(2)-XB(2)
```

```
      Z2=XM(3)-XB(3)
      T123=1.0/(T1*T2*T3)
      T12=T1**2
      T32=T3**2
      AX=XM(1)
      AY=XM(2)
      AZ=XM(3)
      BX=X1*T12+X2*T32
      BY=Y1*T12+Y2*T32
      BZ=Z1*T12+Z2*T32
      CX=X1*T1+X2*T3
      CY=Y1*T1+Y2*T3
      CZ=Z1*T1+Z2*T3
      BX=-BX*T123
      BY=-BY*T123
      BZ=-BZ*T123
      CX=+CX*T123
      CY=+CY*T123
      CZ=+CZ*T123
   20 TT1=T -TM
      TT2=TT1**2
      XP(1)=AX+BX*TT1+CX*TT2
      XP(2)=AY+BY*TT1+CY*TT2
      XP(3)=AZ+BZ*TT1+CZ*TT2
      RETURN
      END

      SUBROUTINE SUNPS(XJD,XS)
      DOUBLE PRECISION XJD,XS
      DIMENSION XS(6),XMS(4),XWS(4),XES(3),XIS(4),XT(5)
      DATA XES/0.01675104E0,-0.00004180E0,+0.000000125E0/
      DATA XIS/23.4522944E0,-0.0130125E0,-0.00000164E0,+5.03E-7/
      DATA XWS/281.2208333E0,+0.000047068 4E0,+4.53E-4,+3.33E-6/
      DATA XMS/358.4758300E0,+0.9856002670E0,-1.50E-4,-3.33E-6/
      DATA DTR/0.01745329251994 3E0/
      DATA PI/3.14159265358979E0/
      DATA TPI/6.28318530717 95E0/
      D1=XJD+25567.5E0
      T1=D1/36525.0E0
      T2=T1*T1
      T3=T2*T1
      ZA=149600000.0E0
      ZE=XES(1)+XES(2)*T1+XES(3)*T2
      ZI=XIS(1)+XIS(2)*T1+XIS(3)*T2+XIS(4)*T3
      ZP=XWS(1)+XWS(2)*D1+XWS(3)*T2+XWS(4)*T3
      ZM=XMS(1)+XMS(2)*D1+XMS(3)*T2+XMS(4)*T3
      ZI=ZI*DTR
      ZP=ZP*DTR
      ZMEAN=AMOD(ZM,360.0E0)*DTR
      ECCANM=SKEP(ZE,ZMEAN,1.0E-5)
      SINE=SIN(ECCANM)
      COSE=COS(ECCANM)
      SINFF=SINE*SQRT(1.0E0-ZE**2)/(1.0E0-ZE*COSE)
```

```
      COSFF=(COSE-ZE)/(1.0E0-ZE*COSE)
      UP=ZP+ATAN2(SINFF,COSFF)
      RS=ZA*(1.0E0-ZE*ZE)/(1.0E0+ZE*COSFF)
      XS(1)=+RS*COS(UP)
      XS(2)=+RS*SIN(UP)*COS(ZI)
      XS(3)=+RS*SIN(UP)*SIN(ZI)
      RETURN
      END

      SUBROUTINE MOONPS(XJD,XM)
      DOUBLE PRECISION XJD,XM
      DIMENSION A(4),B(4),C(4),AP(6),R(6),E(4),TEMP(3),XM(6)
      DATA A/248.77099E0,13.064992446498E0,6.890E-12,+0.295E-18/
      DATA B/317.28125E0, +0.1643580025E0,-9.297E-12,-0.302E-18/
      DATA C/-14.688635E0,-.052953922199E0,+1.557E-12,+0.05E-18/
      DATA E/23.4431852E0,-.0130125E0,-.16389E-5,+0.503E-6/
      DATA  DTR/0.01745329251994 3E0/
      DATA  XI/0.08980411316E0/
      DATA  TPI/6.2831853071795E0/
      DATA  ECC/0.054900489E0/
      DATA  AM/384400.0E0/
      D=XJD+0.5E0
      T=(XJD+25567.5E0)/36525.0E0
C     D IN THE NUMBER OF DAYS FROM JAN 0  12 HRS  1970
C     T IS THE JULIAN CENTURIES FROM 1900  JAN 0  12 HR
C     XJD IS THE JULIAN DAYS FROM JAN 1  0HRS  1970
      D2=D*D
      D3=D2*D
      T2=T*T
      T3=T2*T
      TEMP(1)=A(1)+A(2)*D+A(3)*D2+A(4)*D3
      TEMP(2)=B(1)+B(2)*D+B(3)*D2+B(4)*D3
      TEMP(3)=C(1)+C(2)*D+C(3)*D2+C(4)*D3
      DO 10 K=1,3
   10 TEMP(K)=AMOD(TEMP(K),360.0E0)*DTR
      XL=TEMP(1)
      XLOM=TEMP(2)
      OM=TEMP(3)
      EP=(E(1)+E(2)*T+E(3)*T2+E(4)*T3)*DTR
      ENEXT=SKEP(ECC,XL,1.0E-5)
      SE=SIN(ENEXT)
      CE=COS(ENEXT)
      DEN=1.0E0-ECC*CE
      RMAG=AM*DEN
      SI=SIN(XI)
      CI=COS(XI)
      SF=SQRT(1.0E0-ECC*ECC)*SE/DEN
      CF=(CE-ECC)/DEN
      SLO=SIN(XLOM)
      CLO=COS(XLOM)
      SO=SIN(OM)
      CO=COS(OM)
      SS=SF*CLO+CF*SLO
```

```
      CS=CF*CLO-SF*SLO
      RP(1)=RMAG*(CS*CO-SS*CI*SO)
      RP(2)=RMAG*(CS*SO+SS*CI*CO)
      RP(3)=RMAG*SS*SI
      SE=SIN(EP)
      CE=COS(EP)
      XM(1)=RP(1)
      XM(2)=RP(2)*CE-RP(3)*SE
      XM(3)=RP(2)*SE+RP(3)*CE
      RETURN
      END
      SUBROUTINE DIVR(A,B,C,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1),C(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
   10 C(K1)=A(K1)/B(KC)
      RETURN
      END

      SUBROUTINE BIGLIT(XMB,XML)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/KOUNT/KOUNT(7)
      DIMENSION XML(1),XMB(1)
      LB=0
      LS=0
      DO 10 K=1,7
      DO 10 J=1,7
      LB=LB+1
      IF (KOUNT(K).EQ.0.OR.KOUNT(J).EQ.0) GO TO 10
      LS=LS+1
      XML(LS)=XMB(LB)
   10 CONTINUE
      RETURN
      END

      SUBROUTINE LITBIG(XML,XMB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/KOUNT/KOUNT(7)
      DIMENSION XML(1),XMB(1)
      LB=0
      LS=0
      DO 10 K=1,7
      DO 10 J=1,7
      LB=LB+1
      XMB(LB)=0.0D0
      IF ((K-J).EQ.0) XMB(LB)=1.0D-8
      IF ((K-J).EQ.0.AND.K.EQ.1) XMB(LB)=1.0D-16
      IF ((K-J).EQ.0.AND.K.EQ.7) XMB(LB)=1.0D-02
      IF (KOUNT(K).EQ.0.OR.KOUNT(J).EQ.0) GO TO 10
      LS=LS+1
      XMB(LB)=XML(LS)
```

```
10 CONTINUE
   RETURN
   END

   FUNCTION FASTRG(ELN,T,TN)
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   COMMON/BIAS/BIAS
   DIMENSION ELN(7),ELR(7),RX(7),ELNA(7)
   REAL R2
   DO 10 K=1,6
10 ELNA(K)=ELN(K)
   ELNA(6)=ELNA(6)+ELNA(1)*(T-TN)
   CALL ELNELR(ELNA,ELR)
   CALL FLSTAT(ELR,RX)
   CALL POSION(T,XSTAT,YSTAT,ZSTAT)
   R2=(XSTAT-RX(1))**2+(YSTAT-RX(2))**2+(ZSTAT-RX(3))**2
   FASTRG=SQRT(R2)+BIAS
   RETURN
   END

   SUBROUTINE FLSTAT (ELR,X)
   DOUBLE PRECISION ELR,X,XMU,RE,XJ2
   COMMON/XMU/XMU,RE,XJ2
   DIMENSION X(7), ELR(7), A(3,2), ELRS(7)
   XMUS=XMU
   DO 5 K=1,6
 5 ELRS(K)=ELR(K)
   SNI=SIN(ELRS(3))
   CNI=COS(ELRS(3))
   SOM=SIN(ELRS(4))
   COM=COS(ELRS(4))
   XM=AMOD(ELRS(6),6.283185307179)
   ECC=ELRS(2)
   E=SKEP(ECC,XM,1.0E-6)
   SINE=SIN(E)
   COSE=COS(E)
   STA=SQRT(1.0E0-ECC**2)*SINE/(1.0E0-ECC*COSE)
   CTA=(COSE-ECC)/(1.0E0-ECC*COSE)
   TAA=ATAN2(STA,CTA)
   TBB=TAA+ELRS(5)
   CBA=COS(TBB)
   SBA=SIN(TBB)
   A(1,1)=+COM*CBA-SOM*CNI*SBA
   A(2,1)=+SOM*CBA+COM*CNI*SBA
   A(3,1)=+SNI*SBA
   A(1,2)=-COM*SBA-SOM*CNI*CBA
   A(2,2)=-SOM*SBA+COM*CNI*CBA
   A(3,2)=+SNI*CBA
   P=ELRS(1)*(1.0E0-ECC**2)
   R=P/(1.0E0+ECC*CTA)
   VR=ECC*STA*SQRT(XMUS/P)
   VT=SQRT(XMUS*(2.0E0/R-1.0E0/ELRS(1))-VR*VR)
   DO 10 K=1,3
```

```
      X(K)=R*A(K,1)
   10 X(K+3)=VR*A(K,1)+VT*A(K,2)
      RETURN
      END

      SUBROUTINE DERIV(ELN,TIME,TN,A,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/KOUNT/KOUNT(7)
      DIMENSION ELN(7),TIME(1),A(1),DX(7),XPDX(7),XMDX(7)
      DATA DX/1.0D-9,1.0D-4,1.0D-4,1.0D-4,1.0D-4,1.0D-4,1.0D0/
      DX(1)=ELN(1)*1.0D-4
      KU=0
      DO 20 KC=1,NC
      IF (KOUNT(KC).EQ.0) GO TO 20
      KU=KU+1
      DO 10 KR=1,NR
      IF (KC.EQ.7) GO TO 5
      T=TIME(KR)
      CALL TRAFER(ELN,XPDX,NC,1)
      CALL TRAFER(ELN,XMDX,NC,1)
      XPDX(KC)=XPDX(KC)+DX(KC)/2.0D0
      XMDX(KC)=XMDX(KC)-DX(KC)/2.0D0
    5 K1=KR+(KU-1)*NR
      IF (KC.LE.6) A(K1)=(FASTRG(XPDX,T,TN)-FASTRG(XMDX,T,TN))
     *                 /DX(KC)
   10 IF (KC.EQ.7) A(K1)=1.0D0
   20 CONTINUE
      RETURN
      END

      SUBROUTINE REDUCE(KA,KB,KC,KD,KE,KF,KG,NC,NUMBER)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/KOUNT/KOUNT(7)
      KOUNT(1)=KA
      KOUNT(2)=KB
      KOUNT(3)=KC
      KOUNT(4)=KD
      KOUNT(5)=KE
      KOUNT(6)=KF
      KOUNT(7)=KG
      NUMBER=0
      DO 10 K=1,NC
      IF (KOUNT(K).NE.0) NUMBER=NUMBER+1
   10 CONTINUE
      RETURN
      END

      SUBROUTINE ERASE(DX,DE,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/KOUNT/KOUNT(7)
      DIMENSION DX(7),DE(7)
      KU=0
      DO 10 KC=1,NC
```

```
      DE(KC)=0
      IF (KOUNT(KC).EQ.0) GO TO 10
      KU=KU+1
      DE(KC)=DX(KU)
   10 CONTINUE
      RETURN
      END

      DOUBLE PRECISION FUNCTION RANGE(ELN,T,TN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BIAS/BIAS
      DIMENSION ELN(7),ELR(7),ELD(7),RX(7)
      CALL ELNELR(ELN,ELR)
      CALL RADDEG(ELR,ELD)
      CALL POSION(T,XSTAT,YSTAT,ZSTAT)
      CALL ORBIT(ELD,RX,T,TN)
      R2=(XSTAT-RX(1))**2+(YSTAT-RX(2))**2+(ZSTAT-RX(3))**2
      RANGE=DSQRT(R2)+BIAS
      RETURN
      END

      SUBROUTINE ORDER(Y,TIME,KMAX,TMAX)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (/,24H RANGE DATA OUT OF ORDER)
  101 FORMAT (/,20H RANGE DATA IN ORDER)
      DIMENSION Y(1),TIME(1)
      DO 10 K=2,KMAX
      TMAX=TIME(K)
      TMIN=TIME(K-1)
      IF (TMAX.LT.TMIN) GO TO 20
   10 CONTINUE
      TYPE 101
      RETURN
   20 TYPE 100
      TMAX=TIME(1)
      DO 40 K=1,KMAX
      IF (K.GE.KMAX) GO TO 50
      JS=K+1
      DO 30 J=JS,KMAX
      TK=TIME(K)
      TJ=TIME(J)
      IF (TK.LE.TJ) GO TO 30
      YK=Y(K)
      YJ=Y(J)
      TIME(K)=TJ
      TIME(J)=TK
      Y(K)=YJ
      Y(J)=YK
   30 CONTINUE
   40 CONTINUE
   50 TMAX=TIME(KMAX)
      RETURN
      END
```

```
      SUBROUTINE OUT(A,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  100 FORMAT (1X,12(1PE10.2))
  101 FORMAT (1H )
      DIMENSION A(1),B(12)
      TYPE 101
      DO 10 KR=1,NR
      DO  5 KC=1,NC
      K1=KR+(KC-1)*NR
    5 B(KC)=A(K1)
   10 TYPE 100,(B(K),K=1,NC)
      RETURN
      END

      SUBROUTINE TRAFER(A,B,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
   10 B(K1)=A(K1)
      RETURN
      END

      SUBROUTINE TRNPSE(A,B,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
      K2=KC+(KR-1)*NC
   10 B(K2)=A(K1)
      RETURN
      END

      SUBROUTINE ADD(A,B,C,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1),C(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
   10 C(K1)=A(K1)+B(K1)
      RETURN
      END

      SUBROUTINE SUB(A,B,C,NR,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1),C(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
   10 C(K1)=A(K1)-B(K1)
```

```
      RETURN
      END

      SUBROUTINE MULT(A,B,C,NR,NS,NC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1),C(1)
      DO 10 KC=1,NC
      DO 10 KR=1,NR
      K1=KR+(KC-1)*NR
      C(K1)=0.0D0
      DO 10 KS=1,NS
      K2=KR+(KS-1)*NR
      K3=KS+(KC-1)*NS
   10 C(K1)=C(K1)+A(K2)*B(K3)
      RETURN
      END

      SUBROUTINE IVERSE(BSAVE,B,NX)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION BSAVE(1),B(1)
      CALL TRAFER(BSAVE,B,NX,NX)
      SCALE=0.0D0
      DO 10 K=1,NX
      K1=K+(K-1)*NX
   10 SCALE=SCALE+DLOG(B(K1))
      SCALE=DEXP(SCALE/NX)
      N2=NX*NX
      DO 20 K=1,N2
   20 B(K)=B(K)/SCALE
      CALL JVERSE(B,NX)
      DO 30 K=1,N2
   30 B(K)=B(K)/SCALE
      RETURN
      END

      SUBROUTINE JVERSE(A,NX)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),L(100),M(100)
      D=1.0D0
      NN=NX
      N=NN
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
      IF (DABS(BIGA).GT.DABS(A(IJ))) GO TO 20
```

```
      BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
   45 IF(BIGA) 48,46,48
   46 D=0.0D0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ) +A(IJ)
   65 CONTINUE
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.0D0/BIGA
   80 CONTINUE
      K=NN
  100 K=(K-1)
      IF(K) 150,150,105
```

```
105 I=L(K)
    IF(I-K) 120,120,108
108 JQ=N*(K-1)
    JR=N*(I-1)
    DO 110 J=1,N
    JK=JQ+J
    HOLD=A(JK)
    JI=JR+J
    A(JK)=-A(JI)
110 A(JI) =HOLD
120 J=M(K)
    IF(J-K) 100,100,125
125 KI=K-N
    DO 130 I=1,N
    KI=KI+N
    HOLD=A(KI)
    JI=KI-K+J
    A(KI)=-A(JI)
130 A(JI) =HOLD
    GO TO 100
150 RETURN
    END

    SUBROUTINE SHADOW(T,X)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION X(7),R(3,8),V(3,8),XM(6),XS(6)
    DATA TOLD/0.D0/
    CALL SUN(T,XS)
    STEP=T-TOLD
    CALL MOON(T,XM)
    TOLD=T
    DO 1 I=1,3
    R(I,1)=X(I)
    V(I,1)=X(I+3)
    R(I,2)=X(I)-XM(I)
  1 R(I,3)=X(I)-XS(I)
    CALL SHAD(R,V,T,STEP)
    RETURN
    END

    DOUBLE PRECISION FUNCTION ADOT(X,Y)
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    DIMENSION  X(3),     Y(3)
    A=FNORM(X)
    B=FNORM(Y)
    ANG=DOT(X,Y)/A/B
    ADOT=ARKTNS(180,ANG,DSQRT(1.0D0-ANG*ANG))*57.295779D0
    RETURN
    END

    DOUBLE PRECISION FUNCTION DOT(X,Y)
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    DIMENSION X(3),Y(3)
```

```
      DOT = X(1)*Y(1) + X(2)*Y(2) + X(3)*Y(3)
      RETURN
      END

      DOUBLE PRECISION FUNCTION FNORM(X)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION X(3)
      FNORM=DSQRT(X(1)**2+X(2)**2+X(3)**2)
    3 RETURN
      END

      SUBROUTINE ITRATE(STEP,TIME,R1,V1,F,K,KK,DT,R,V)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION X1(3),R1(3,8),R(3,8),V1(3,8),V(3,8),R3(3),VVV(3),CB(20)
      DIMENSION XS(6),XM(6)
      COMMON/XMU/XMU,RE,XJ2
      COUNT=0.D0
      C=XMU
      CM=3476.D0
      ZRS=6.965D5
      XK=1
      F=0.D0
      DT=0.D0
      T=0.D0
      LK=1
      TOLD=0.D0
      FOLD=0.D0
      TOLDD=0.D0
      FOLDD=0.D0
      NN=1
      IF (KK.EQ.2) XK=-1
      DO 2000 I=1,8
      DO 2000 J=1,3
      R(J,I)=R1(J,I)
 2000 V(J,I)=V1(J,I)
      GO TO (1,2),K
    1 DO 3 I=1,3
    3 X1(I)=R(I,1)-R(I,3)
      SEVA= ADOT(X1,R(1,1))
      RFS=FNORM(R(1,3))
      RFE= FNORM(R(1,1))
      DELS= DARSIN(6500.D0/RFE)*57.2957795D0
      ZRE=6500.D0
      GO TO 4
    2 DO 5 I=1,3
    5 X1(I)=R(I,2)-R(I,3)
      SEVA= ADOT(X1,R(1,2))
      RFS= FNORM(R(1,3))
      RFE= FNORM(R(1,2))
      DELS=DARSIN(3476.D0/RFE)*57.2957795D0
      ZRE=3476.D0
      GO TO 4
    4 CONTINUE
```

```
      ZRE=DSIGN(ZRE,XK)
      SIG=DARSIN((ZRS+ZRE)/RFS)*57.2957795D0
      F=SEVA+DELS+XK*SIG-180.D0
      IF (DABS(F).LE.0.0005D0) RETURN
      IF (COUNT.GT.100.D0) GO TO 11
      GO TO (6,7),LK
    6 LK=2
      NN=1
      DT=-STEP
      GO TO 8
    7 IF (FOLD*F.GT.0.D0) GO TO 21
    9 DT=(TOLD*F-T*FOLD)/(F-FOLD)
    8 FOLDD=FOLD
      TOLDD=TOLD
      FOLD=F
      TOLD=T
      T=DT
      CALL FG(R1(1,1),V1(1,1),DT,RR,VVV,NN)
      NN=2
      TNEW=TIME+T
      CALL SUN(TNEW,XS)
      CALL MOON(TNEW,XM)
      DO 10 I=1,3
      R(I,1)=RR(I)
      V(I,1)=VVV(I)
      R(I,2)=RR(I)-XM(I)
      V(I,2)=VVV(I)-XM(I+3)
      R(I,3)=RR(I)-XS(I)
   10 V(I,3)=VVV(I)-XS(I+3)
      COUNT=COUNT+1.D0
      GO TO (1,2),K
   11 TYPE 12
   12 FORMAT(1H0,22HMAX ITERATIONS IN ITER)
      RETURN
  21  FOLD=FOLDD
      TOLD=TOLDD
      GO TO 9
      END

      SUBROUTINE SHAD(R1,V1,X,  STEP)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION R(3,8),V(3,8),R1(3,8),V1(3,8),X1(3)
      DATA IPFLAG/1/,IUFLAG/1/,IMFLAG/1/,IMUFLG/1/, FLAG/0./,FLAG1/0./,
     1FLAG2/0./,FLAG3/0./
      DATA IVFLAG/1/,IVUFLG/1/,PFLAG/0./,PUFLAG/0./
      IF (STEP.EQ.0.D0)RETURN
   41 CONTINUE
      DO 4 I=1,3
    4 X1(I)=R1(I,1)-R1(I,3)
      SEVA=ADOT(X1,R1(1,1))
      RFS=FNORM(R1(1,3))
      RFE=FNORM(R1(1,1))
      DELS=DARSIN(6500.D0/RFE)*57.2957795D0
```

```
      IF (SEVA-90.D0) 300,300,301
 301  ZRS=6.965D5
      ZRE=6500.D0
      ZPSIG=DARSIN((ZRS+ZRE)/RFS)*57.2957795D0
      FS=SEVA+DELS+ZPSIG-180.D0
      IF (FS) 300,302,302
 302  FLAG=1.D0
      IF (IPFLAG.NE.1) GO TO 1000
      K=1
      N=1
      IPFLAG=2
      KK=1
      TYPE 5
   5  FORMAT(/,1X,'SATELLITE ENTERED EARTH PENUMBRA')
      GO TO 20
1000  CONTINUE
      ZSIGMA=DARSIN((ZRS-ZRE)/RFS)*57.2957795D0
      FS=DELS-ZSIGMA+SEVA-180.D0
      IF (FS)303,304,304
 304  FLAG1=1.D0
      IF (IUFLAG.NE.1) GO TO 310
      K=1
      IUFLAG=2
      N=2
      KK=2
      TYPE 7
   7  FORMAT(1X,'SATELLITE ENTERED EARTH UMBRA')
      GO TO 20
 300  FLAG=0.D0
 303  FLAG1=0.D0
 310  DO 30 I=1,3
  30  X1(I)=R1(I,2)-R1(I,3)
      SEVA=ADOT(X1,R1(1,2))
      IF (SEVA-90.D0) 2000,2000,901
 901  ZRS=6.965D5
      ZRE=3476.D0
      ZPSIG=DARSIN((ZRS+ZRE)/RFS)*57.2957795D0
      DELM=DARSIN(3476.D0/FNORM(R1(1,2)))*57.2957795D0
      FS=SEVA+DELM+ZPSIG-180.D0
      IF (FS) 2000,902,902
 902  FLAG2=1.D0
      IF (IMFLAG.NE.1) GO TO 3000
      K=2
      IMFLAG=2
      N=3
      KK=1
      TYPE 8
   8  FORMAT(/,1X,'SATELLITE ENTERED MOON PENUMBRA')
      GO TO 20
3000  CONTINUE
      ZSIGMA=DARSIN((ZRS-ZRE)/RFS)*57.2957795D0
      FS=DELM-ZSIGMA+SEVA-180.D0
      IF (FS)2001,904,904
```

```
  904 FLAG3=1.D0
      IF (IMUFLG.NE.1) GO TO 2010
      K=2
      IMUFLG=2
      N=4
      KK=2
      TYPE 9
    9 FORMAT(1X,'SATELLITE ENTERED MOON UMBRA')
      GO TO 20
 2000 FLAG2=0.D0
 2001 FLAG3=0.D0
 2010 CONTINUE
      IF (IPFLAG.NE.2.OR.FLAG.NE.0.D0) GO TO 11
      TYPE 12
   12 FORMAT(1X,'SATELLITE LEFT EARTH PENUMBRA')
      K=1
      N=5
      KK=1
      IPFLAG=1
      GO TO 20
   11 IF (IUFLAG.NE.2.OR.FLAG1.NE.0.D0) GO TO 14
      TYPE 13
   13 FORMAT(1X,'SATELLITE LEFT EARTH UMBRA')
      KK=2
      N=6
      K=1
      IUFLAG=1
      GO TO 20
   14 IF (IMFLAG.NE.2.OR.FLAG2.NE.0.D0) GO TO 15
      TYPE 16
   16 FORMAT(1X,'SATELLITE LEFT MOON PENUMBRA')
      KK=1
      K=2
      N=7
      IMFLAG=1
      GO TO 20
   15 IF (IMUFLG.NE.2.OR.FLAG3.NE.0.D0) GO TO 505
      TYPE 18
   18 FORMAT(1X,'SATELLITE LEFT MOON UMBRA')
      K=2
      N=8
      KK=2
      IMUFLG=1
   20 CALL ITRATE(STEP,X,R1,V1,FS,K,KK,DT,R,V)
      TIM=X+DT
      MJD=IDINT(TIM/86400.0D0)+40587
      TSEC=DMOD(TIM,86400.0D0)
      MH=TSEC/3600.0D0
      MM=DMOD(TSEC/60.0D0,60.0D0)
      SC=DMOD(TSEC,60.0D0)
    6 FORMAT(6H MJD =,I6,4X,12H HR/MIN/SEC=,I2,1H/,I2,1H/,F6.3)
      TYPE 6,MJD,MH,MM,SC
      GO TO (1000,310,3000,2010,11,14,15,505 ),N
```

```
505 CONTINUE
    RETURN
    END

    DOUBLE PRECISION FUNCTION DARSIN(X)
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    C=DSQRT(1.D0-X*X)
    DARSIN=DATAN(X/C)
    RETURN
    END

    SUBROUTINE FG(X,V,DT,XT,VT,K)
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    DIMENSION X(3),V(3),XT(3),VT(3)
    IF (K.GT.1) GO TO 1
    U=398601.5D0
    R2=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)
    V2=V(1)*V(1)+V(2)*V(2)+V(3)*V(3)
    R=DSQRT(R2)
    R3=R2*R
    U0=U/R3
    P0=(X(1)*V(1)+X(2)*V(2)+X(3)*V(3))/R2
    Q0=(V2-R2*U0)/R2
    F2=-U0/2.D0
    F3=U0*P0/2.D0
    F4=(3.D0*U0*Q0-15.D0*U0*P0*P0+U0*U0)/24.D0
    G2=-U0/6.D0
    G3=U0*P0
  1 CONTINUE
    DT2=DT*DT
    DT3=DT2*DT
    DO 2 I=1,3
    XT(I)=(1.D0+F2*DT2+F3*DT3)*X(I)+(DT+G2*DT3)*V(I)
  2 VT(I)=(2.D0*F2*DT+3.D0*F3*DT2+4.D0*F4*DT3)*X(I)
   1+(1.D0+3.D0*G2*DT2+G3*DT3)*V(I)
    RETURN
    END
```

REFERENCES

1. Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

2. Willey, R. E. and Pisacane, U. L., "The Motion of an Artificial Satellite in a Non-spherical Gravitational Field With a Quadratic Scale Height," Technical Memorandum, TG1236, Johns Hopkins University, Applied Physics Laboratory, Silver Spring, MD 20910, January 1974.

3. Kaufman, B., TRIP (TRajectory Integration Program), NRL Report No. 7436, July 18, 1972.

*U.S. GOVERNMENT PRINTING OFFICE: 1978 261-250/105 1-3