CT-82-100-112LR

FAA TECHNICAL CENTER LETTER REPORT

PROGRAM TO COMPUTE THE

USER ERRORS OF THE

GLOBAL POSITIONING SYSTEM (GPS)

by

Erhard Buchberger

September 1982

U. S. DEPARTMENT OF TRANSPORTATION

FEDERAL AVIATION ADMINISTRATION

TECHNICAL CENTER

Atlantic City Airport, N.J. 08405

TABLE OF CONTENTS

	Page No.
Abstract	i
Introduction	1
Background	1
Scope	1
Discussion	1
User Manual	1
Program EB00	1
Program FILE	3
Program PLSIG/PLCOR	3
Program PLAEL	3
Program PL5FI	3
Subroutine Description	4
Data Files	4
References	8
Appendix	
A - Transfer Files	
B - Flowcharts	
C - Program Listings	
D - Examples of Program Outputs	

LIST OF TABLES

TABLE		Page
1	Time Parameters Used in Programs	6
2	Order of Keplerian Parameters	7

INTRODUCTION

Background

C.S. Wong at Texas A&M University developed a program on a Texas Instruments 990/4 minicomputer, which computes the following parameters from the GPS satellite almanac: the satellite rise and fall times, the satellite azimuth and elevation, and the geometric dilution of precision (Reference 1). This program was modified by M. Magrogan for execution on a Digital Equipment Corporation PDP-11/34 minicomputer to analyze FAA test results (Reference 2). Additional changes and refinements to these routines allowed computation of the dilution of precision parameters on a Hewlett Packard (HP) 1000 minicomputer not only for the case of four satellites, but also for three satellites plus altitude aiding (Reference 3). The derivation of the GPS dilution of precision parameters assume that the standard deviations of the pseudorange measurement errors are equal and independent. R.J. Esposito at the FAA Technical Center has recently discussed the user error behavior for the more general case when the measurement errors are independent but not necessarily equal (Reference 4).

Scope

This report documents the programs which compute and plot the GPS user errors from pseudorange errors (biases and standard deviations which are independent but not necessarily equal) on the HP 1000 for different satellite/configurations with and without an altimeter input.

DISCUSSION

The main program EB00 and its associated routines described in this report stem from the program HD00 and routines contained in reference 3. In order to be executed, the programs require inputs which are requested with explaining comments. The algorithms and equations used in these programs are given in references 4 through 9.

Table 1 displays the input time parameters. Special attention must be given to the fact that different units (day and second) and different origins are used.

The Keplerian parameters, which are included in the almanac message and used to compute satellite trajectories, are listed in table 2. Note that parameters 9 through 14 are computed in the program EB00, with μ as the earth's universal gravitional constant.

USER MANUAL

Appendix A contains the transfer files for programs EB00, PLCOR, PLSIG, PLAEL, and PL5FI. Appendix B gives the flowcharts on programs EB00 (pages B-1 through B-10), FILE (page B-11), PLCOR/PLSIG (pages B-12 and B-13), PLAEL (pages B-14 through B-16), PL5FI (pages B-17 through B-19), and subroutines EB06 (pages B-20 through B-23), PSUB1/PLSUB (pages B-24 and B-25), INPUT (page B-26). Appendix C provides listings of these programs and all subroutines. Each program is called on the HP computer with its name, and user prompting explicity asks for the required inputs. Appendix D gives examples of the program outputs.

Program EB00 (Pages C-1 through C-10)

This program computes and prints out the position of three or four satellites in earth-centered earth-fixed (ECEF) coordinates and in azimuth and elevation relative

to the user, the user-satellite direction cosine matrix including time (G-matrix), the covariance matrix of the user error (sigma matrix), the square root of the diagonal elements of the sigma matrix (standard deviations), the multiple standard deviations in two, three, and four dimensions, the correlation coefficients (related to the off diagonal elements of the sigma matrix), and the user bias errors. The multiple dimensional user standard deviations are proportional to the dilution of precision parameters when the standard deviation of the pseudorange measurement errors are equal and independent.

All input data and relevant information are printed out. The data output for every point in time, for which computations are made, consists of one page. Since the program is frequently run just to generate plot files, the data printout is often not needed and may be suppressed by setting the sense switch number Ø (ISSW(Ø)) to the "ON" position. In this configuration, only one page with the input data is printed. The start sequence and parts of the input data of the program EBØØ are the same as for the program HDØØ, and portions of the later user manual are repeated here for completeness.

The user position must be entered in geographical coordinates (latitude (LAT) and longitude (LON)) and altitude (ALT) in feet. If the location is the FAA Technical Center, this information is already stored in the program. After the program has received user position, the geographical coordinates are converted into ECEF and printed out. The next data requested are the start (TSTART) and stop (TFIN) times, the time increment (TDEL), the time of collection (TC) of the almanac, and the date of use (DTU) (see table 1).

The choice must be then be made whether the computations should be made with four satellites or three satellites. When the navigation mode utilizes three satellites plus altitude aiding, the altimeter is modeled as the fourth space vehicle located in the center of the earth. The elements in the fourth row of the G-matrix are given by 0, 0, -1, 0. If an altimeter input is not available, the program assumes that the receiver clock bias is extrapolated and the fourth row of the G-matrix is given by 0, 0, 0, 1.

The next decision concerns the input of the almanac for the satellites:

(1) If the almanac is on a magnetic tape, it must be stated so and the entire almanac file is read from the tape. Then, the PRN numbers of the satellites are requested, and tne program checks whether the data is on the tape.

(2) The other choice is either to provide the program with the names of the data files containing an almanac for each space vehicle, or to enter the data by the keyboard.

If the psuedorange variance matrix V is identical to the unit matrix, the values are provided by the program, otherwise, the diagonal elements must be entered via the keyboard. Since each pseudorange variance may have unique value, it must be associated with the proper satellite PRN number. The time tagged psuedorange bias errors are then read from the appropriate file and stored in a array (see program FILE). Finally, the names of the three plot files are entered, and the program prints the relevant data and generates the data files for the plot programs.

PROGRAM FILE (Page C-11)

This program was developed to generate files with time tagged psuedorange bias errors (URE) for the program EBØØ. The program FILE requests the number of files to be produced, the number of records for each file, the file name, and the data to be stored. Finally, it generates a URE file in the correct format to be compatible with the URE input in EBØØ.

Program PLSIG/PLCOR (Pages C-12 through C-19)

Each program plots data stored in the sigma-matrix files which are described under the section entitled Data Files. Each program makes five subplots on one figure.

PLCOR plots the user error parameter versus Greenwich mean time (GMT): the four single dimensional standard deviations, the six correlation coefficients in two subplots, and the four user bias errors.

PLSIG plots the user error parameters versus GMT in a slightly different form: the upper triangular elements of the sigma matrix in three subplots, the single dimensional user standard deviations, and the user bias errors.

Both programs need the same input data: the name of the plot file, the logical unit of the output device, and the ID of the device-subroutine (Monitor: 1, 1; Printer/Plotter: 20,5). In order to scale the time axis and to select a time slot from the data file, the user enters the start and stop times and, as a way of distinguishing several plots, the date. The last entry may have any six alphanumeric characters. The plot is then generated and displayed on the selected output device.

Program PLAEL (Pages C-20 through C-22)

The program PLAEL requires the following input information: the logical unit of the output device, the ID of the device subroutine, the start and stop times, the name of the file containing the satellite azimuth and elevation angles relative to the user, and the number of satellites whose data are stored in the file. The program generates a figure with subplots of the satellite elevation and azimuth angles versus GMT. A dotted line is drawn at an elevation angle of 10 degrees above the horizon.

Program PL5FI (Pages C-23 and C-24)

This program is designed to plot the two dimensional user standard deviations versus GMT for up to five data files and in the same figure. However, any file may be plotted which contains a record of the y-value, a time in hours and minutes as the x-value. The only thing which should be changed are the titles of the axes.

PL5FI requires the same type of input information as the other plot routines: the logical unit, ID, start and stop times, and names of the files. The program generates plots with a legend for each curve since each data file is plotted in a different line style.

Subroutine Description

EBØ2: computes the rotation matrix, given on page C-25.
EBØ3: converts user geographical coordinates into the ECEF system, given on page 26.
EBØ5: calls EBØ6 and EBØ7 and computes satellite azimuth and elevation given, on page C-27.
EBØ6: computes satellite position in ECEF, given on page C-28.
EBØ7: computes user to satellite distance, given on page C-29.
EBØ8. computes the G-matrix and its transpose, and calls EBØ9B and MINVD, given on pages C-30 through C-32.
EBØ9B: multiplies two matrices, given on page C-33.
MINVD: inverts the matrices from the "Scientific Subroutine" package from DEC (Reference 6), given on pages C-34 through C-36.
EB11: multiplies a matrix and a vector, given on page C-37.
EB12: converts seconds into hours, minutes and seconds, given on page C-38.
EB14: unpacks the almanac data (with assembly language subroutine DINTF), given on pages C-39 through C-41.
DINTF: assembly language subroutine, unpacks the almanac data (with subrouinte EB14), given on page C-42.
PLSUB: plots user sigma matrix elements, single dimensional standard deviations, correlation coefficients, and bias errors, given on page C-43.
PSUB1: plots user multi dimensional standard deviations, given on page C-44.
INPUT: reads data files in plot programs, given on page C-45.

Data Files

Five types of data files are utilized, two as input for EBØØ, and three with data points for the plot programs. The files are:

1. Almanac Files: Each contains the date of collection of the almanac, the satellite PRN number (first record), and, in the next two lines, the eight almanac parameters in the order described in table 2.

2. URE Files: These files are generated by the program FILE for each satellite, and include the number of records and the pseudorange bias errors as a function of time per record.

3. Sigma Files: These are the files for the programs PLCOR and PLSIG. The order of the record values stored in a data block is the following:
   a. time in hours and minutes,
   b. upper triangular sigma matrix (10 values),
   c. square root of the diagonal elements of the sigma matrix (4 values),
   d. correlation coefficients (6 values),
   e. user bias errors (4 values).

This data block is then repeated for each time increment.

4. HDOP Files: Each record contains the time versus the two dimensional standard deviation in meters for the plot program PL5FI. The standard deviation and time are set equal to zero in the last record.

5. Angle Files: These are generated for the plot program PLAEL, and include the time in hours and minutes and the satellite records per data block. Each record contains the program internal satellite number and the elevation and azimuth for this satellite.

Parameter	Start of Count	Unit
DTU - Date of use	Beginning of Year	Day
DTC - Date of collection of Almanac	Beginning of Year	Day
T - GPS - Time	Beginning of Day	Second
TC - Collection Time of Almanac	Beginning of week (0000 hours Sunday)	Second

The used time zone is Greenwich Mean Time (GMT)

Table 1: TIME PARAMETERS USED IN PROGRAMS

No. in Almanac	Keplerian Parameter	Name in Program and (GPS) - Name)
1	Eccentricity of ellipsoid	E (e)
2	Reference time for almanac	TOA (toa)
3	Orbit inclination	Di (i)
4	Drift rate of right ascension	OMD ($\dot{\Omega}$)
5	Square root of semi-major axis	RA ($\sqrt{A}$)
6	Right ascension of the orbit	OMO (Ω_o)
7	Argument of perigee	W (ω)
8	Mean anomaly	MO (Mo)
9	Semi-major axis	A
10	Cosine of orbit inclination angle	cos(i)
11	Sine of orbit inclination angle	sin(i)
12	Cosine of argument of perigee	cos(ω)
13	Sine of argument of perigee	sin(ω)
14	Mean motion	$\sqrt{\mu/A^3}$

Table 2: ORDER OF KEPLERIAN PARAMETERS

REFERENCES

1. A GPS Almanac, Wong, C.S., Texas A&M University.

2. Global Positioning System (GPS) Satellite Elevation and Azimuth Angles, Magrogan, M., FAA Technical Center Report Number CT-82-100-41LR.

3. A Program to Compute the Dilution of Precision Parameters for the GPS System, Buchberger, E., FAA Technical Center Report Number CT-82-100-106LR.

4. Derivation of Global Positioning System (GPS) User Errors with Representative Examples, Esposito, R.J., FAA Technical Center Report Number CT-82-100-113LR.

5. Computer Program Development Specification for the Z-User-Set, Specification No. GP-US-305.

6. The Scientific Subroutine Package, Digital Equipment Corporation.

7. Principle of Operation of NAVSTAR and System Characteristics, Milliken, R.J., and Zoller, C.J., Navigation, Vol. 25, pages 95-106, 1978.

8. Using the Global Positioning Satellite System for Positioning and Navigation, Rhyne, T., Noe, P., and Painter, J., Altair Corporation.

Transfer Files - These files are used to load programs which consist of several subroutines.

```
TEB00  T=00004 IS ON CR00002 USING 00002 BLKS R=0000

0001   REL, %EB00
0002   SEA, %EB02
0003   SEA, %EB03
0004   SEA, %EB05
0005   SEA, %EB06
0006   SEA, %EB07
0007   SEA, %EB08
0008   SEA, %EB06B
0009   SEA, %EB09B
0010   SEA, %MINVP
0011   SEA, %EB11
0012   SEA, %EB12
0013   SEA, %EB14
0014   SEA, %DINTF
```

```
TPLCOR T=00004 IS ON CR00002 USING 00002 BLKS R=0000

0001   REL,%PLCOR
0002   SEA,%PLSUB
0003   SEA,%PSUBI
0004   RE,%DLTBL
0005   SEA,%INPUT
0006   SEA,%GPS
```

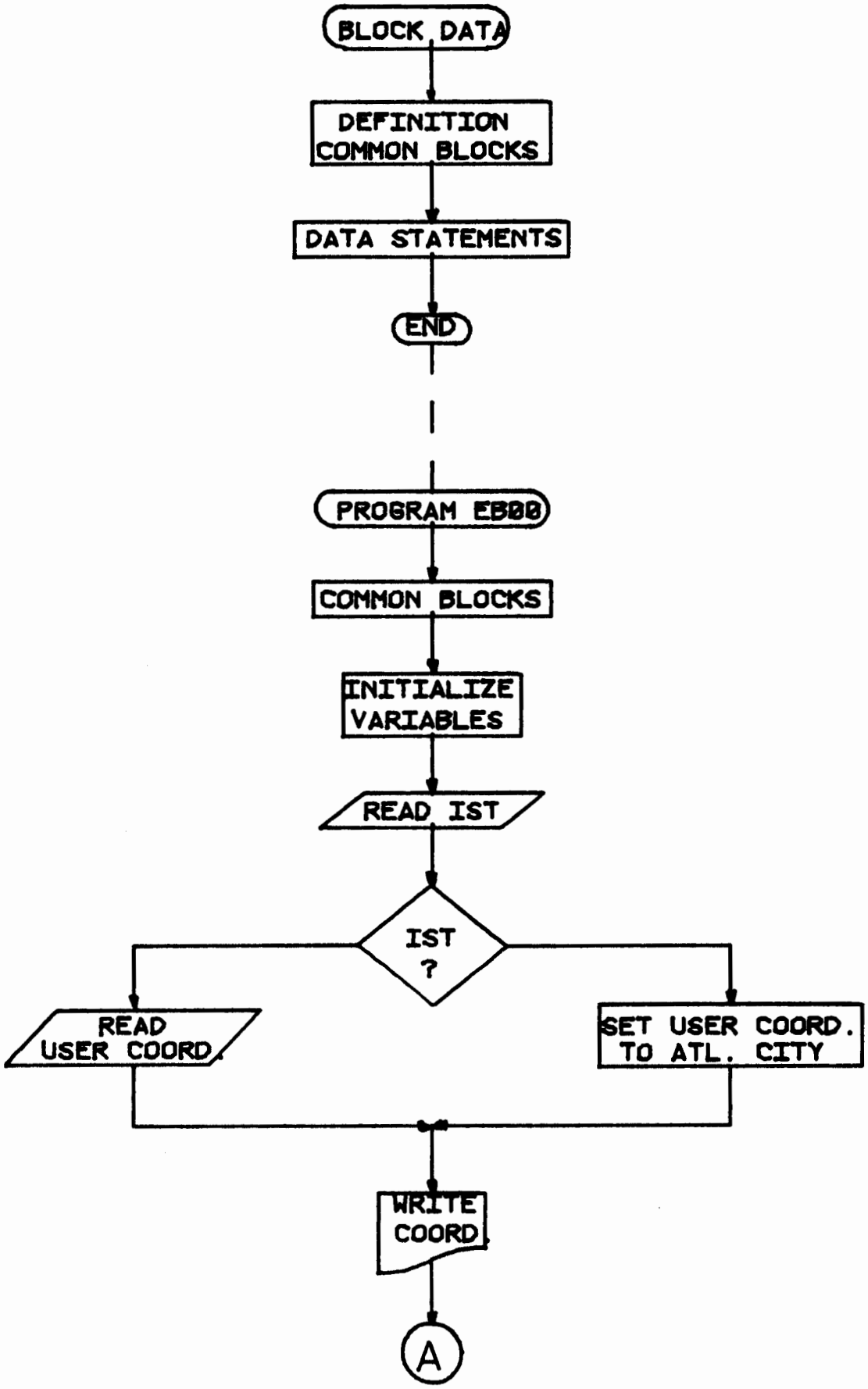
```
TPLSIG T=00004 IS ON CR00011 USING 00002 BLKS R=0000

0001   REL,%PLSIG
0002   SEA,%PLSUB
0003   RE,%DLTBL
0004   SEA,%INPUT
0005   SEA,%GPS
```

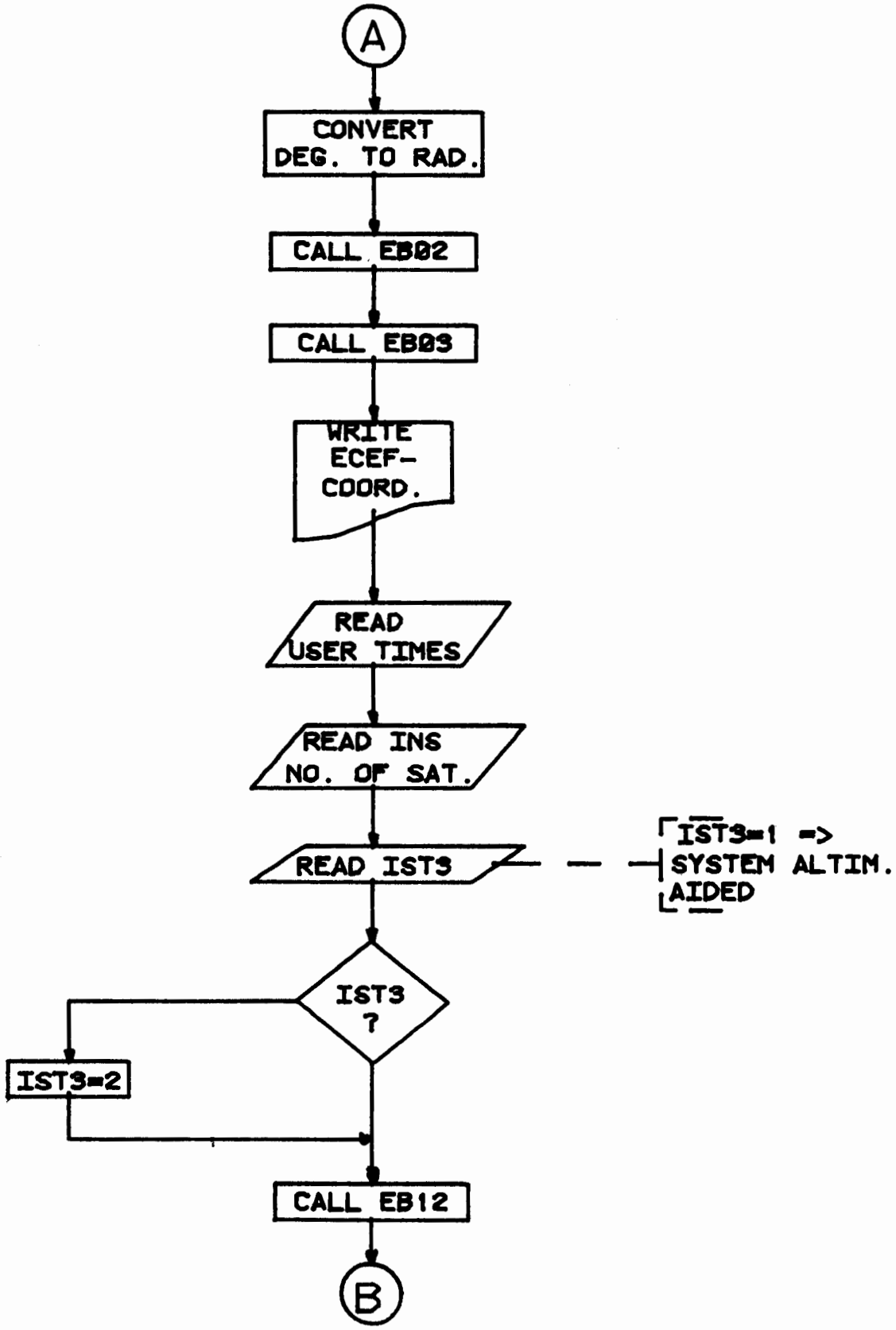
```
TPLAEL T=00004 IS ON CR00011 USING 00002 BLKS R=0000

0001   RE, %PLAEL
0002   RE, %DLTBL
0003   SEA, %GPS
```

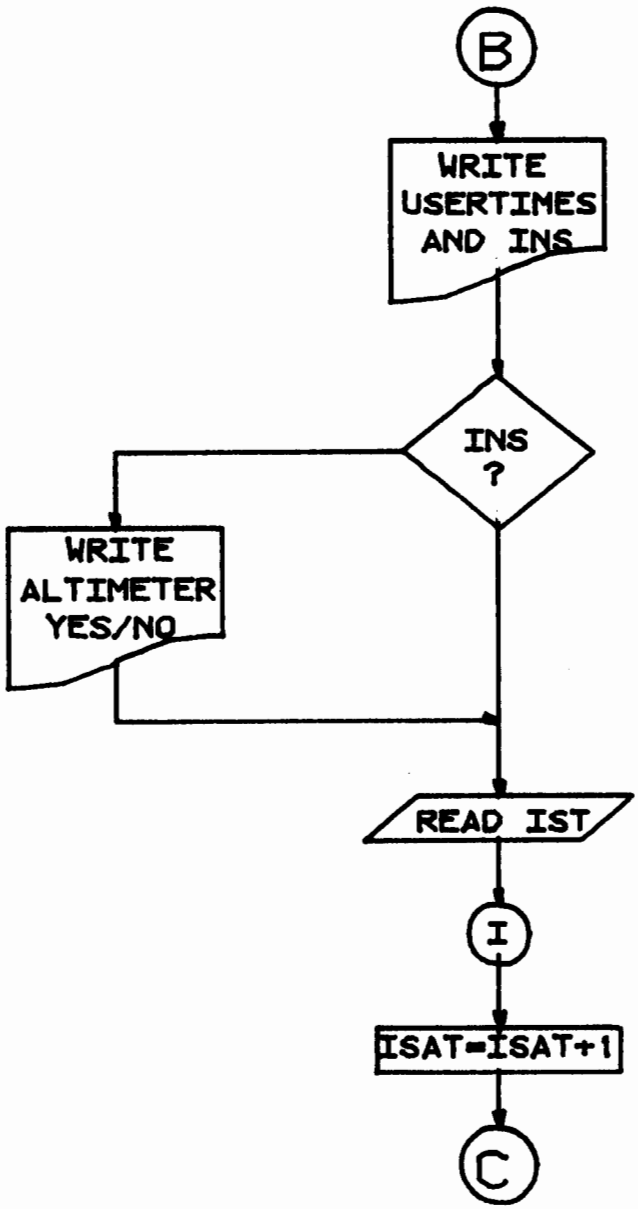
```
TPL5FI T=00004 IS ON CR00011 USING 00002 BLKS R=0000

0001   RE, %PL5FI
0002   RE, %DLTBL
0003   SEA, %GPS
```

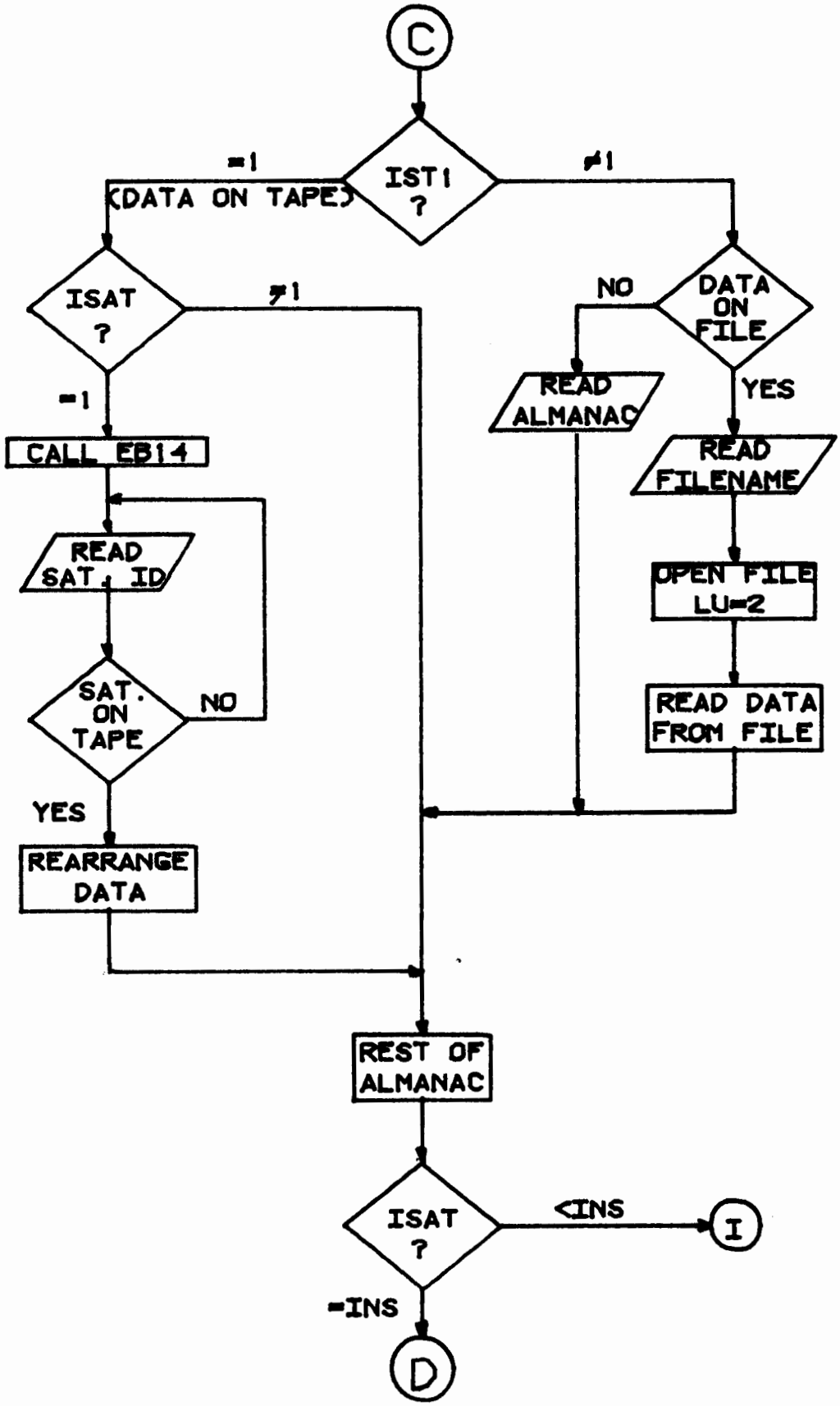
APPENDIX B

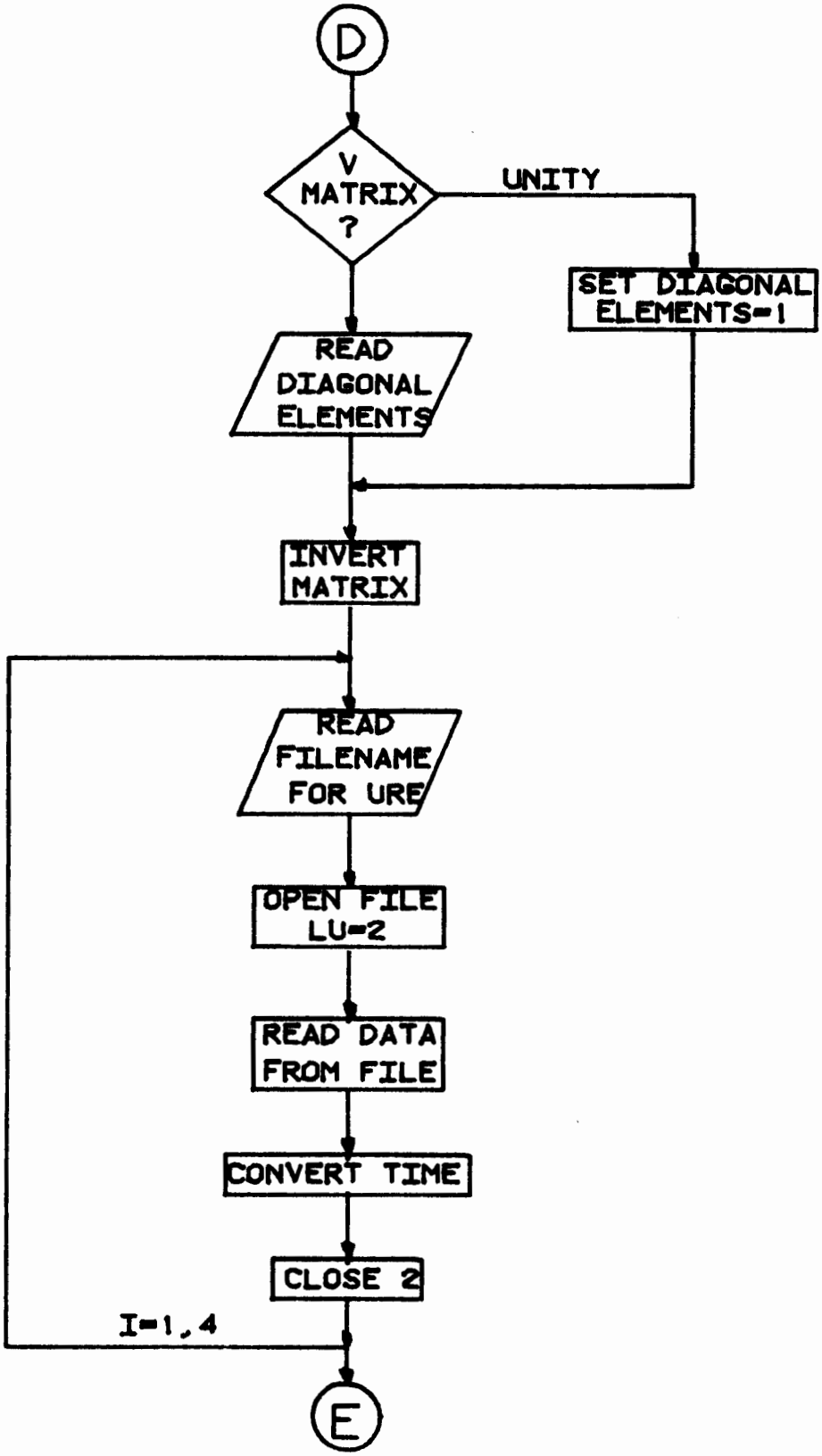
FLOWCHART

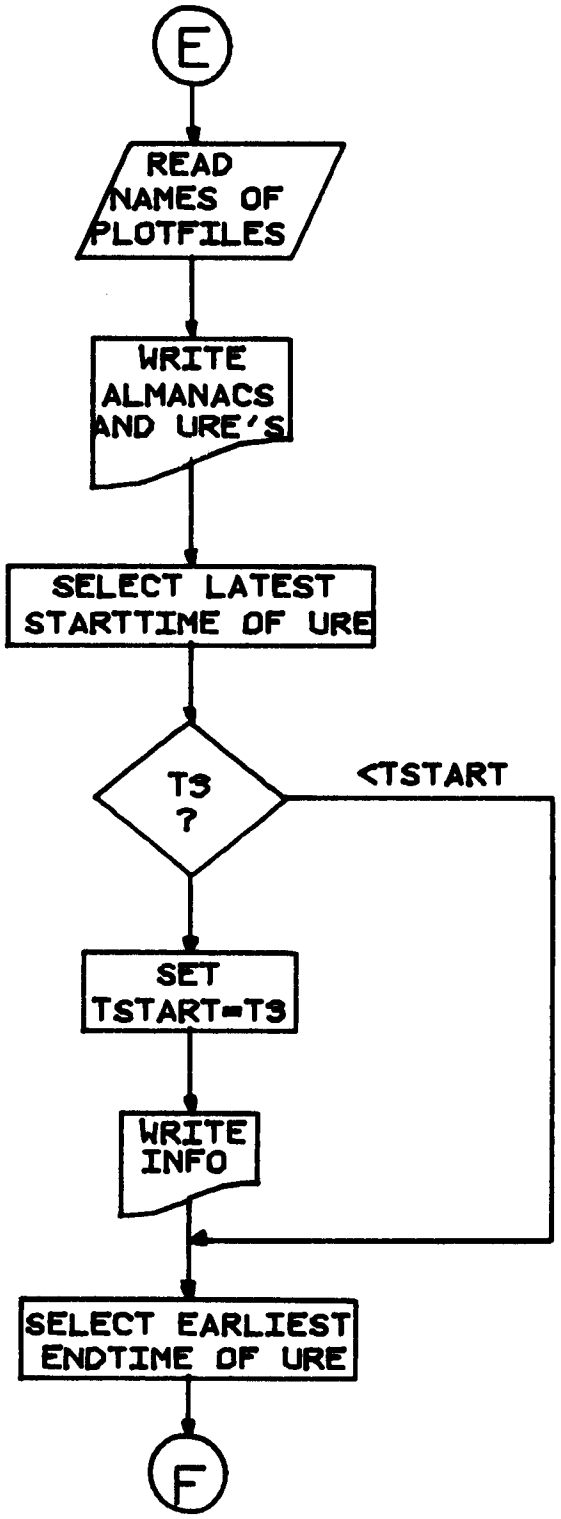


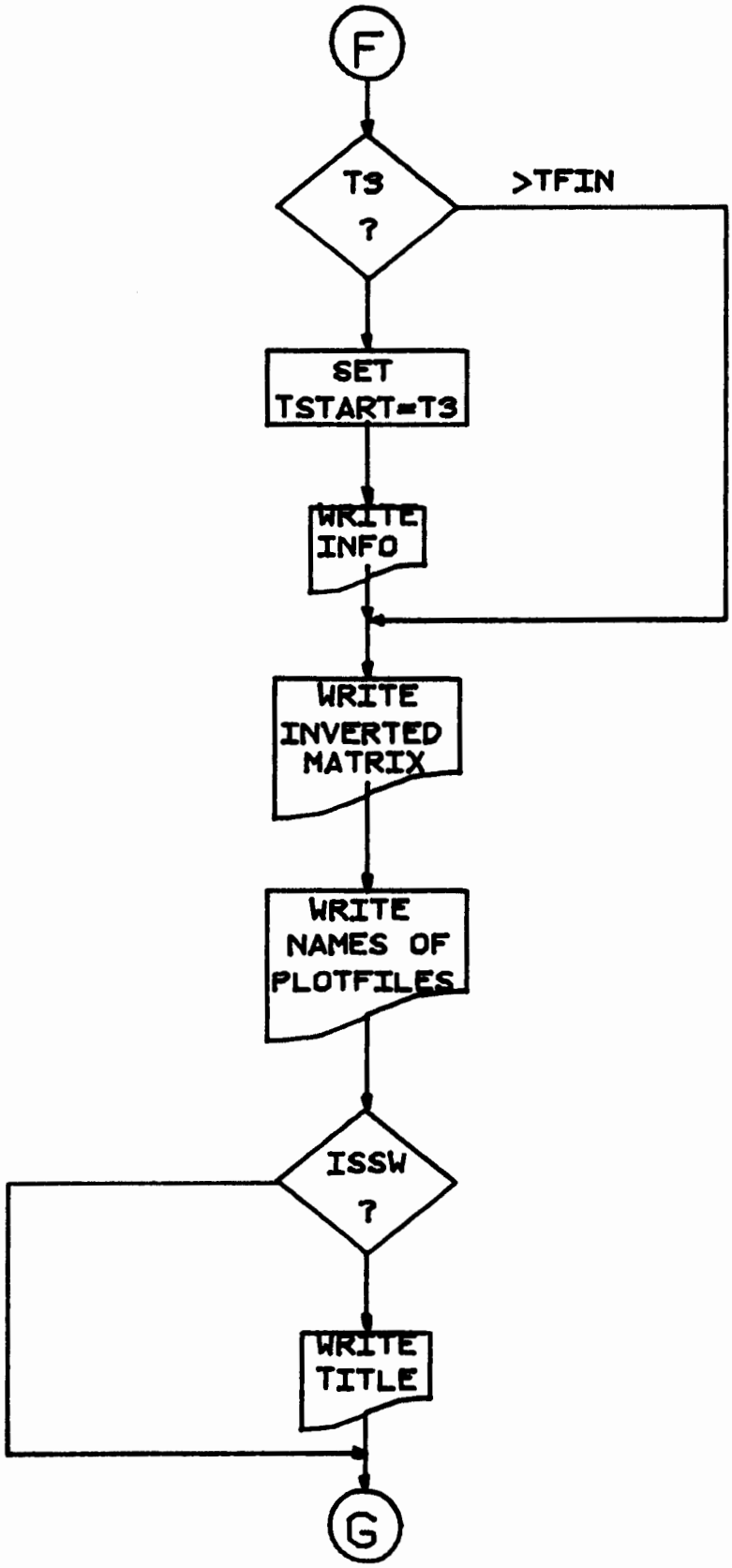

A
CONVERT DEG. TO RAD.
CALL EB02
CALL EB03
WRITE ECEF-COORD.
READ USER TIMES
READ INS NO. OF SAT.
READ IST3
IST3=1 => SYSTEM ALTIM. AIDED
IST3 ?
IST3=2
CALL EB12
B

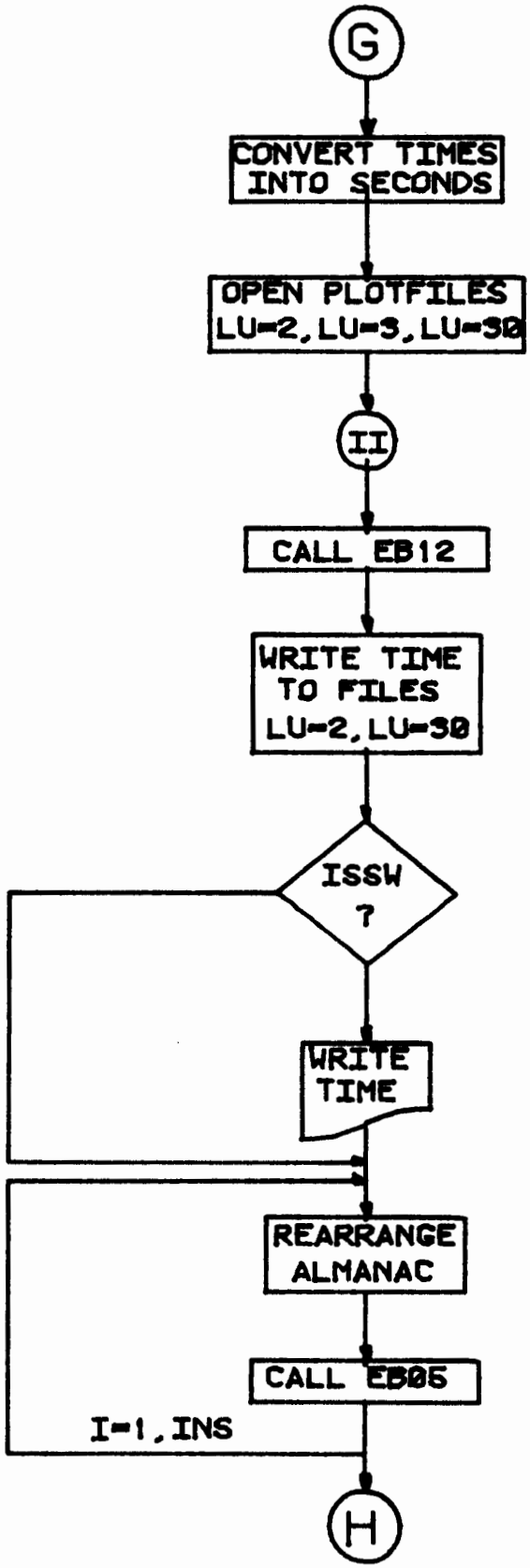

B
WRITE USERTIMES AND INS
INS ?
WRITE ALTIMETER YES/NO
READ IST
I
ISAT=ISAT+1
C

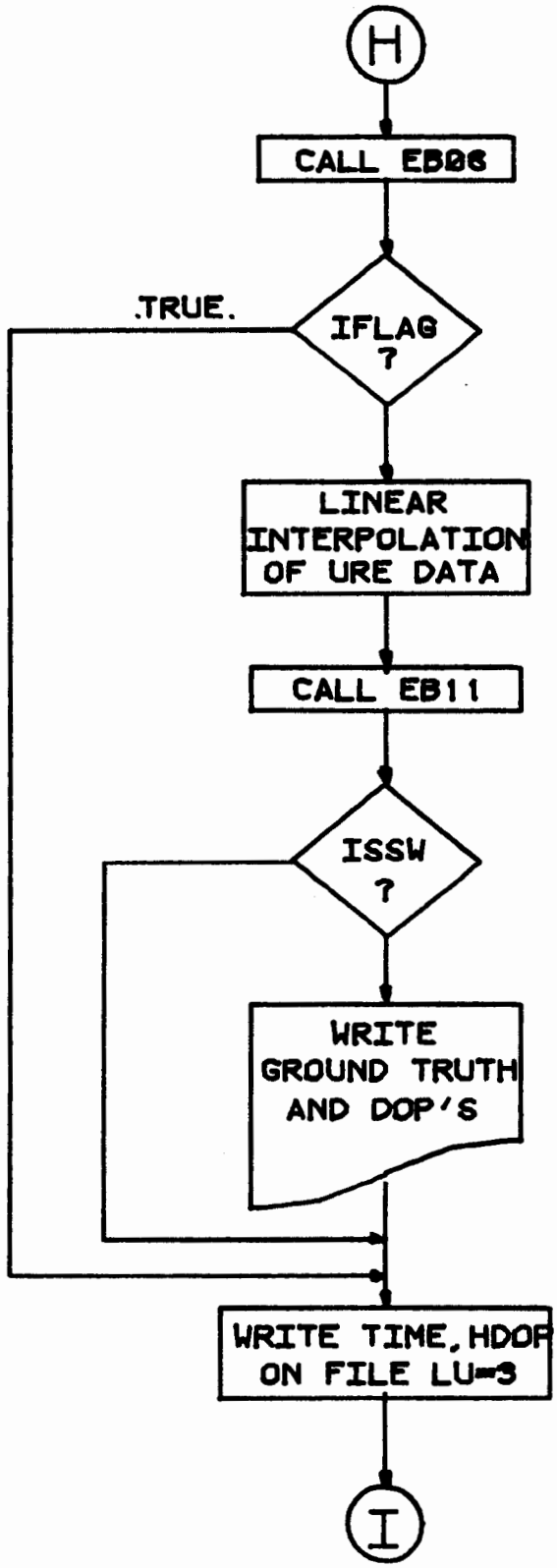

C
IST1 ?
=1
(DATA ON TAPE)
≠1
ISAT ?
≠1
=1
CALL EB14
READ SAT. ID
SAT. ON TAPE
NO
YES
REARRANGE DATA
DATA ON FILE
NO
YES
READ ALMANAC
READ FILENAME
OPEN FILE LU=2
READ DATA FROM FILE
REST OF ALMANAC
ISAT ?
<INS
I
=INS
D

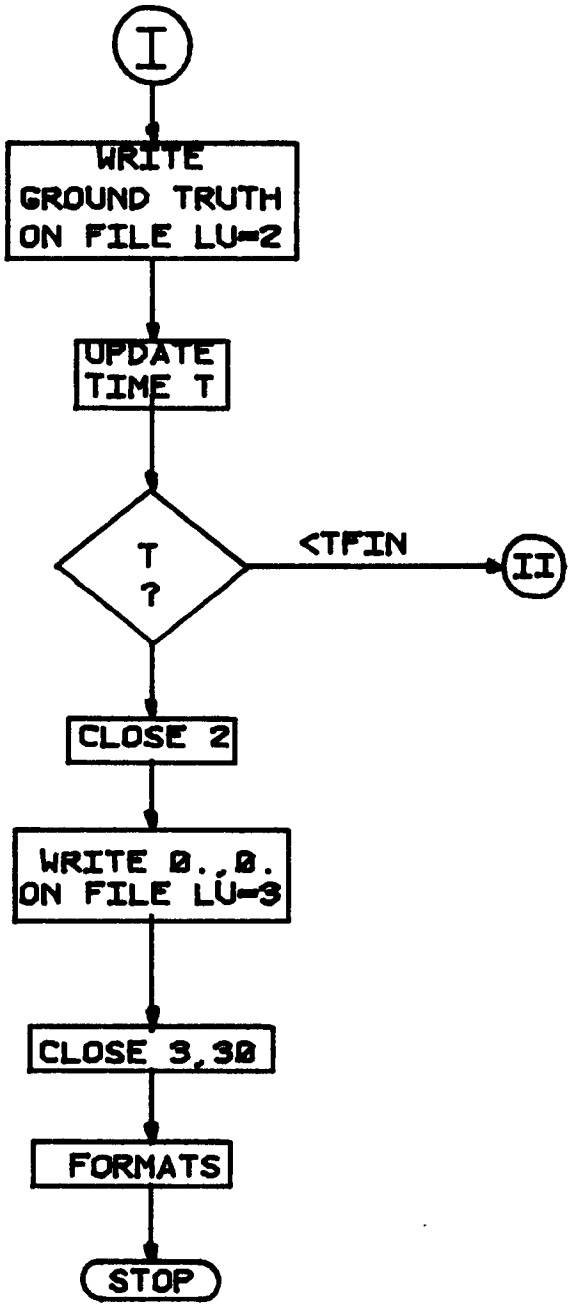

D
V MATRIX ?
UNITY
SET DIAGONAL ELEMENTS=1
READ DIAGONAL ELEMENTS
INVERT MATRIX
READ FILENAME FOR URE
OPEN FILE LU=2
READ DATA FROM FILE
CONVERT TIME
CLOSE 2
I=1,4
E

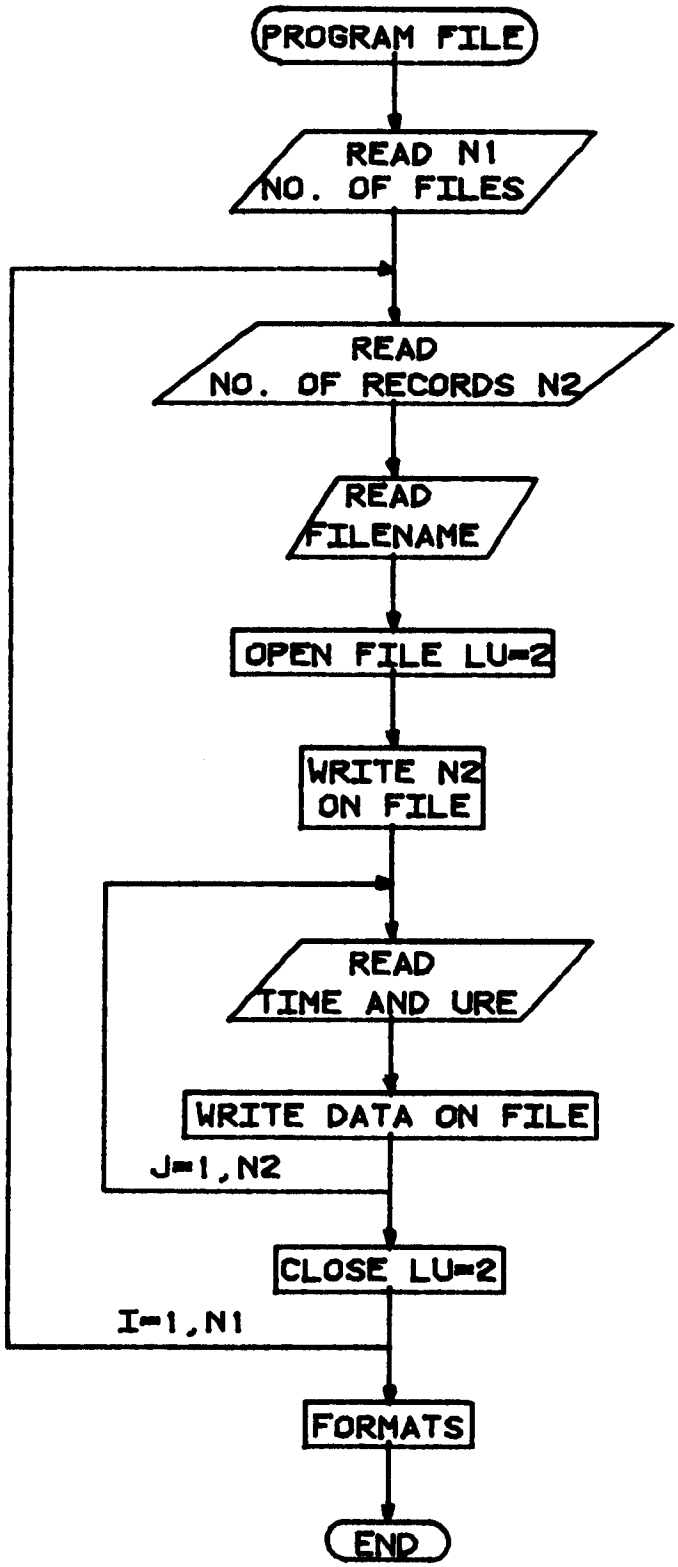

E
READ NAMES OF PLOTFILES
WRITE ALMANACS AND URE'S
SELECT LATEST STARTTIME OF URE
T3 ?
<TSTART
SET TSTART=T3
WRITE INFO
SELECT EARLIEST ENDTIME OF URE
F

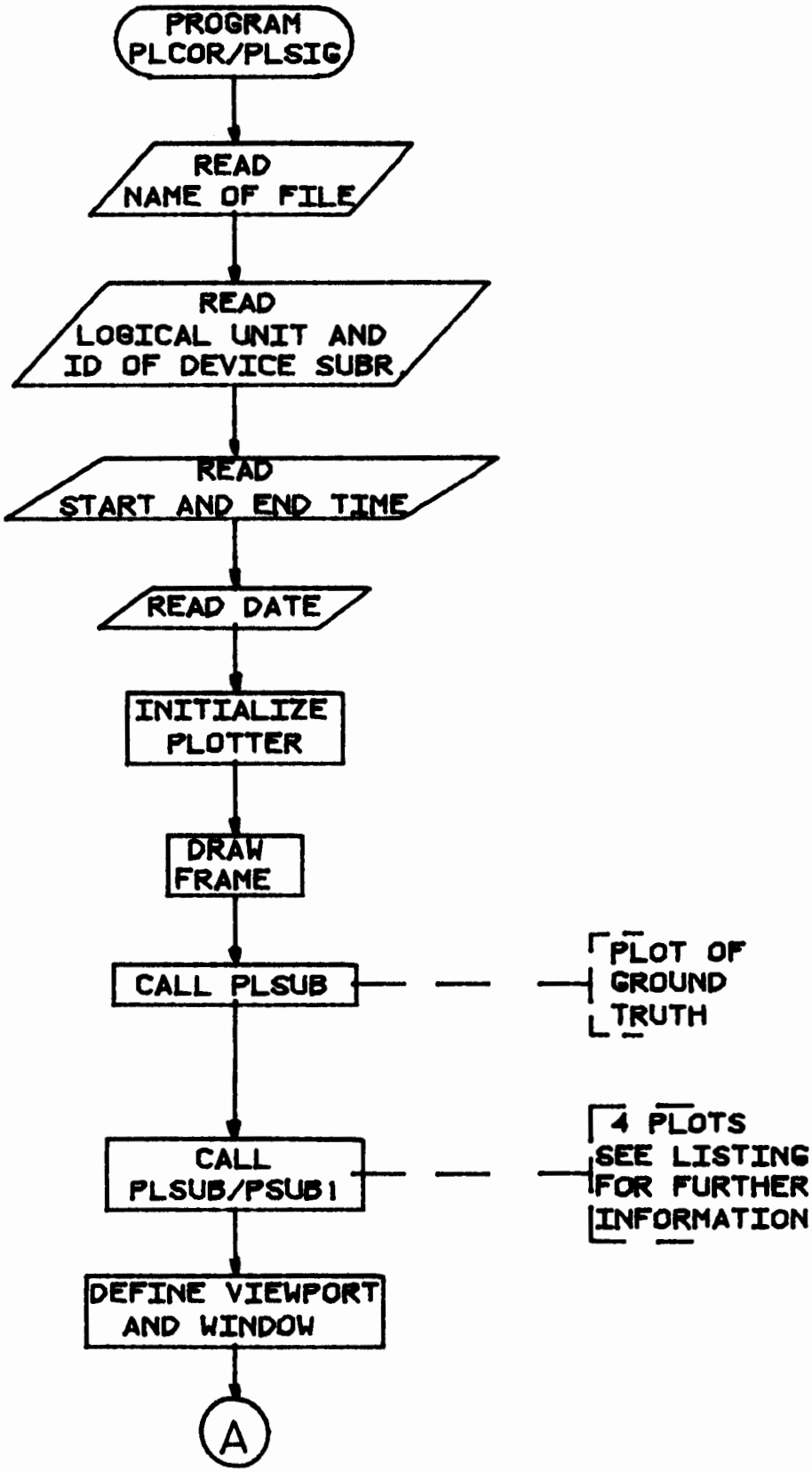

F
T3
?
>TFIN
SET
TSTART=T3
WRITE
INFO
WRITE
INVERTED
MATRIX
WRITE
NAMES OF
PLOTFILES
ISSW
?
WRITE
TITLE
G

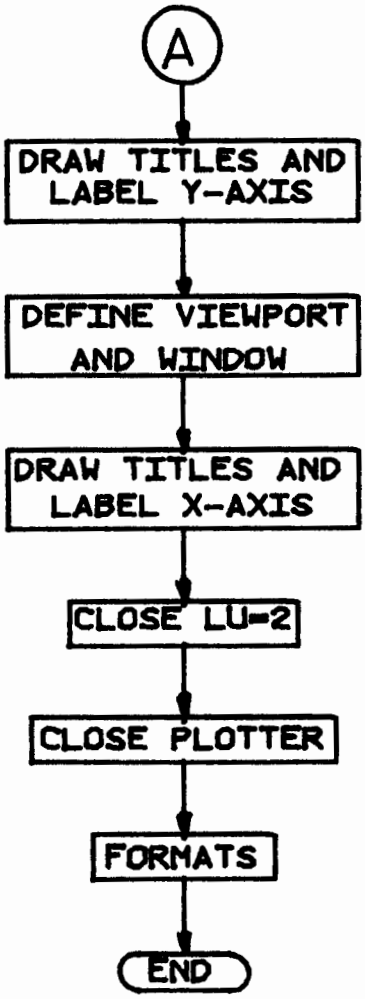

G
CONVERT TIMES INTO SECONDS
OPEN PLOTFILES LU=2, LU=3, LU=30
II
CALL EB12
WRITE TIME TO FILES LU=2, LU=30
ISSW ?
WRITE TIME
REARRANGE ALMANAC
CALL EB05
I=1, INS
H

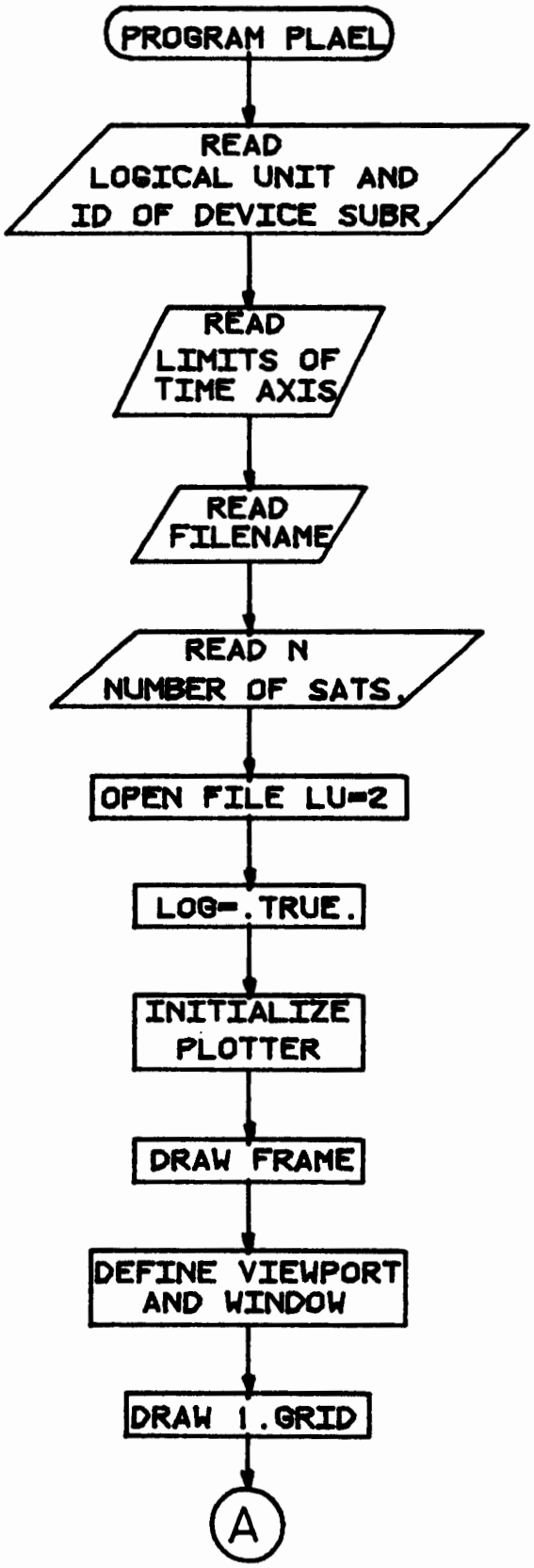

H
CALL EB06
.TRUE.
IFLAG
?
LINEAR
INTERPOLATION
OF URE DATA
CALL EB11
ISSW
?
WRITE
GROUND TRUTH
AND DOP'S
WRITE TIME, HDOP
ON FILE LU=3
I

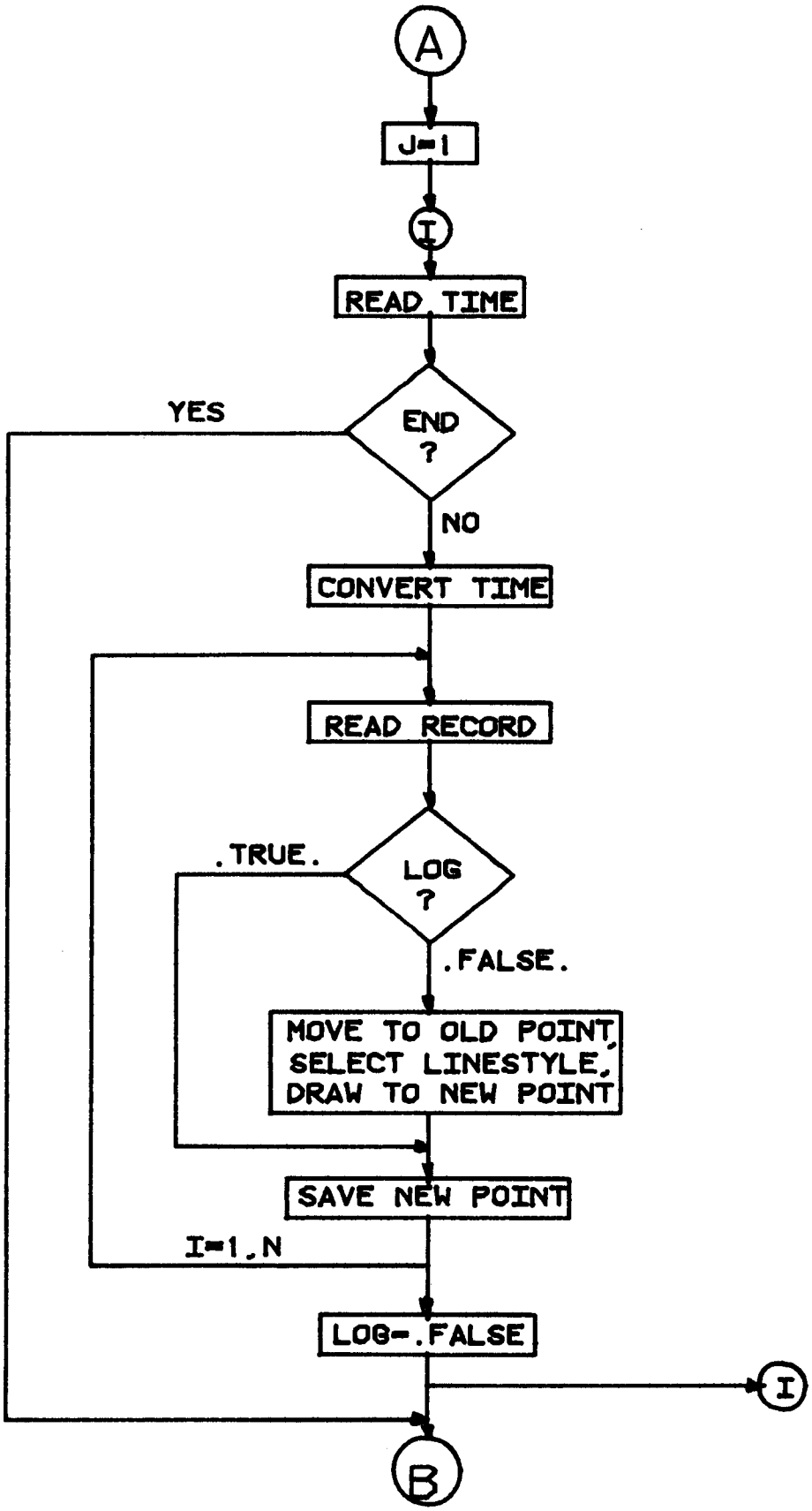

I
WRITE
GROUND TRUTH
ON FILE LU=2
UPDATE
TIME T
T
?
<TFIN
II
CLOSE 2
WRITE Ø.,Ø.
ON FILE LU=3
CLOSE 3,3Ø
FORMATS
STOP

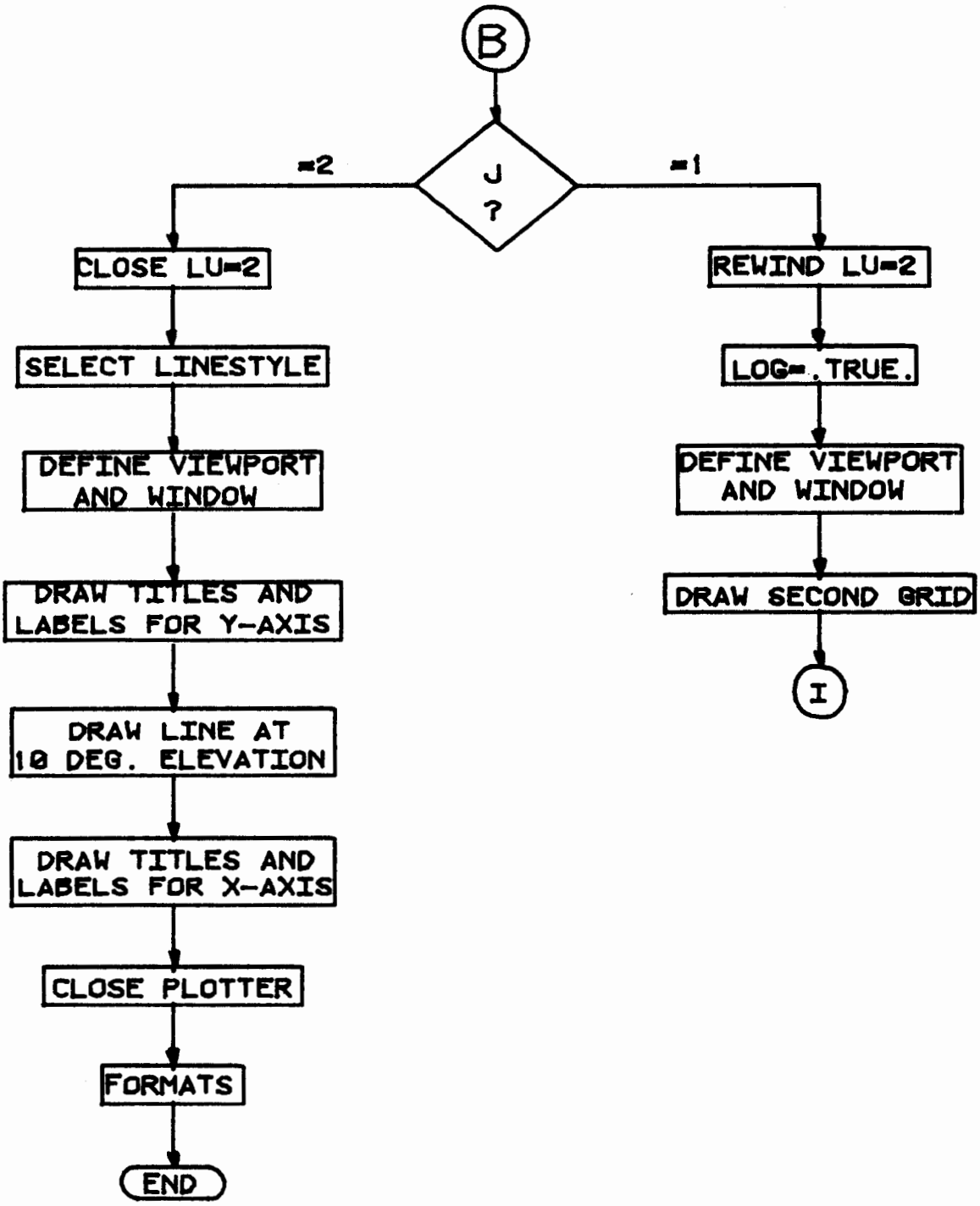

PROGRAM FILE
READ N1
NO. OF FILES
READ
NO. OF RECORDS N2
READ
FILENAME
OPEN FILE LU=2
WRITE N2
ON FILE
READ
TIME AND URE
WRITE DATA ON FILE
J=1,N2
CLOSE LU=2
I=1,N1
FORMATS
END

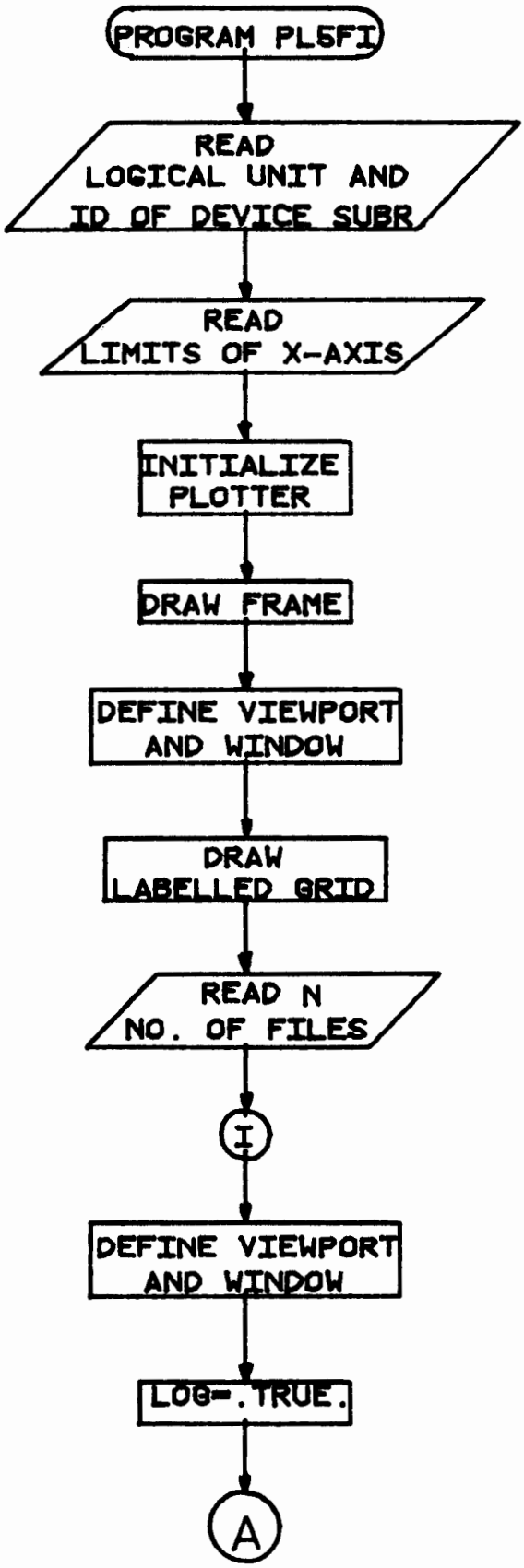

PROGRAM PLCOR/PLSIG
READ NAME OF FILE
READ LOGICAL UNIT AND ID OF DEVICE SUBR
READ START AND END TIME
READ DATE
INITIALIZE PLOTTER
DRAW FRAME
CALL PLSUB
PLOT OF GROUND TRUTH
CALL PLSUB/PSUB1
4 PLOTS SEE LISTING FOR FURTHER INFORMATION
DEFINE VIEWPORT AND WINDOW
A

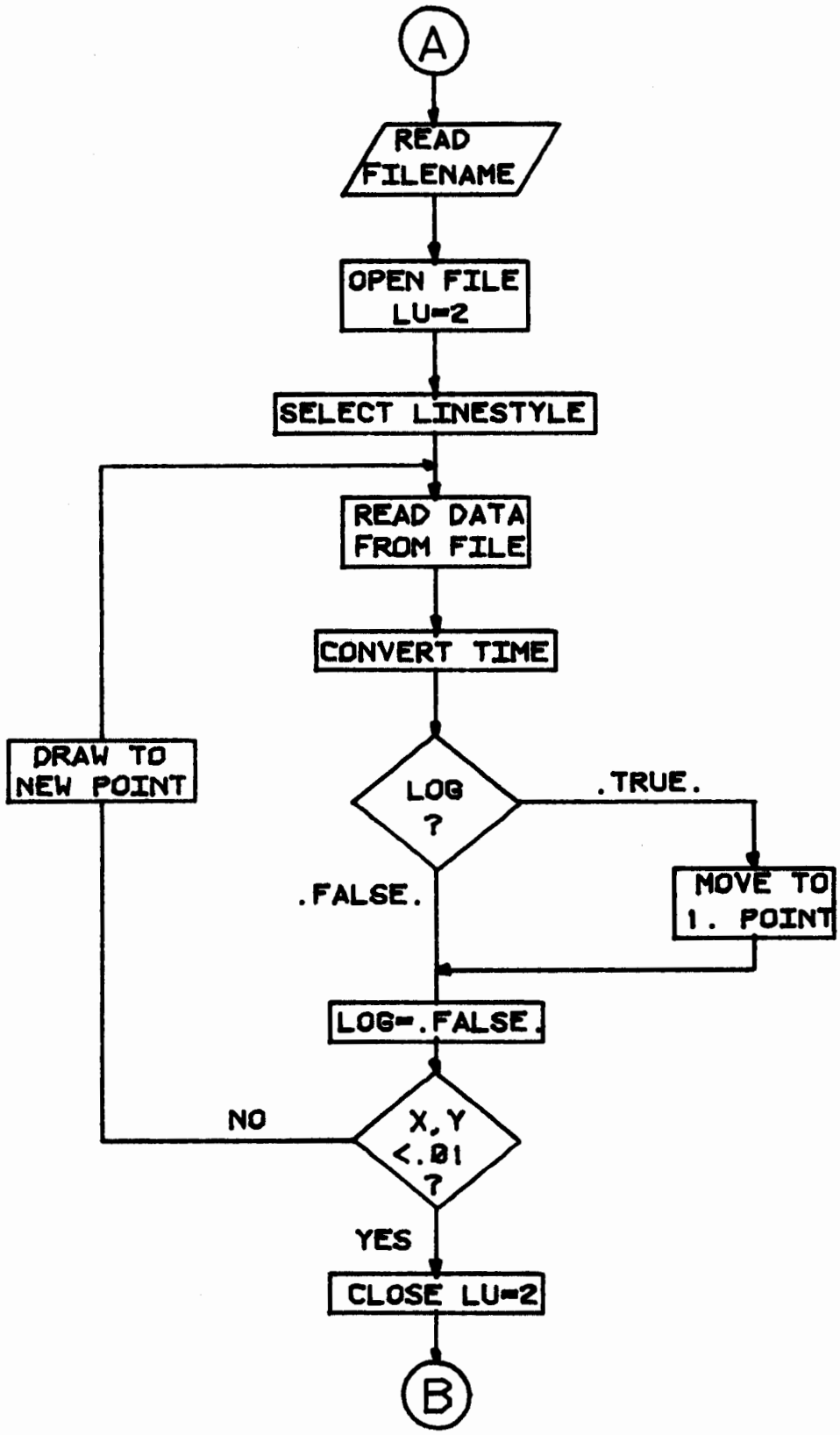

A
DRAW TITLES AND LABEL Y-AXIS
DEFINE VIEWPORT AND WINDOW
DRAW TITLES AND LABEL X-AXIS
CLOSE LU=2
CLOSE PLOTTER
FORMATS
END

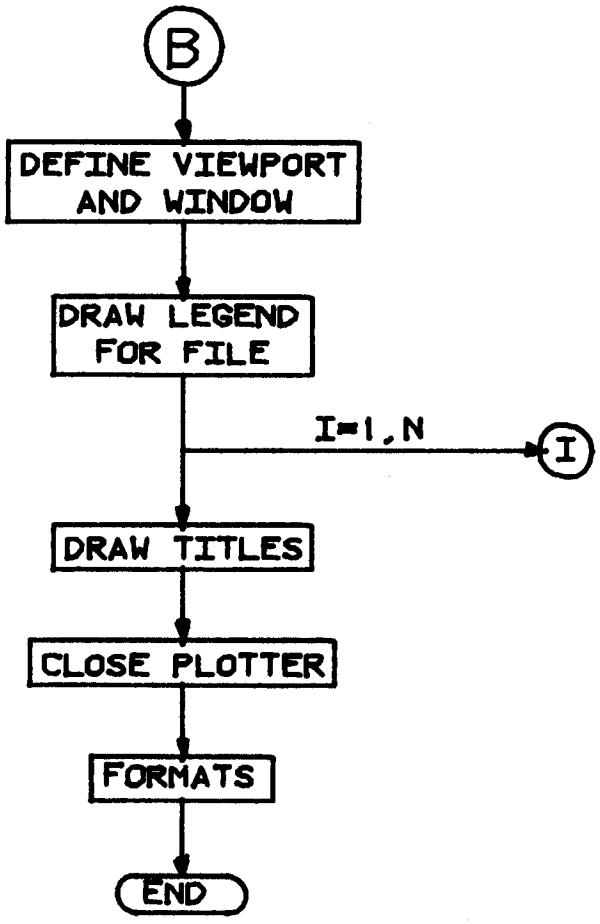

PROGRAM PLAEL
READ
LOGICAL UNIT AND
ID OF DEVICE SUBR.
READ
LIMITS OF
TIME AXIS
READ
FILENAME
READ N
NUMBER OF SATS.
OPEN FILE LU=2
LOG=.TRUE.
INITIALIZE
PLOTTER
DRAW FRAME
DEFINE VIEWPORT
AND WINDOW
DRAW 1.GRID
A

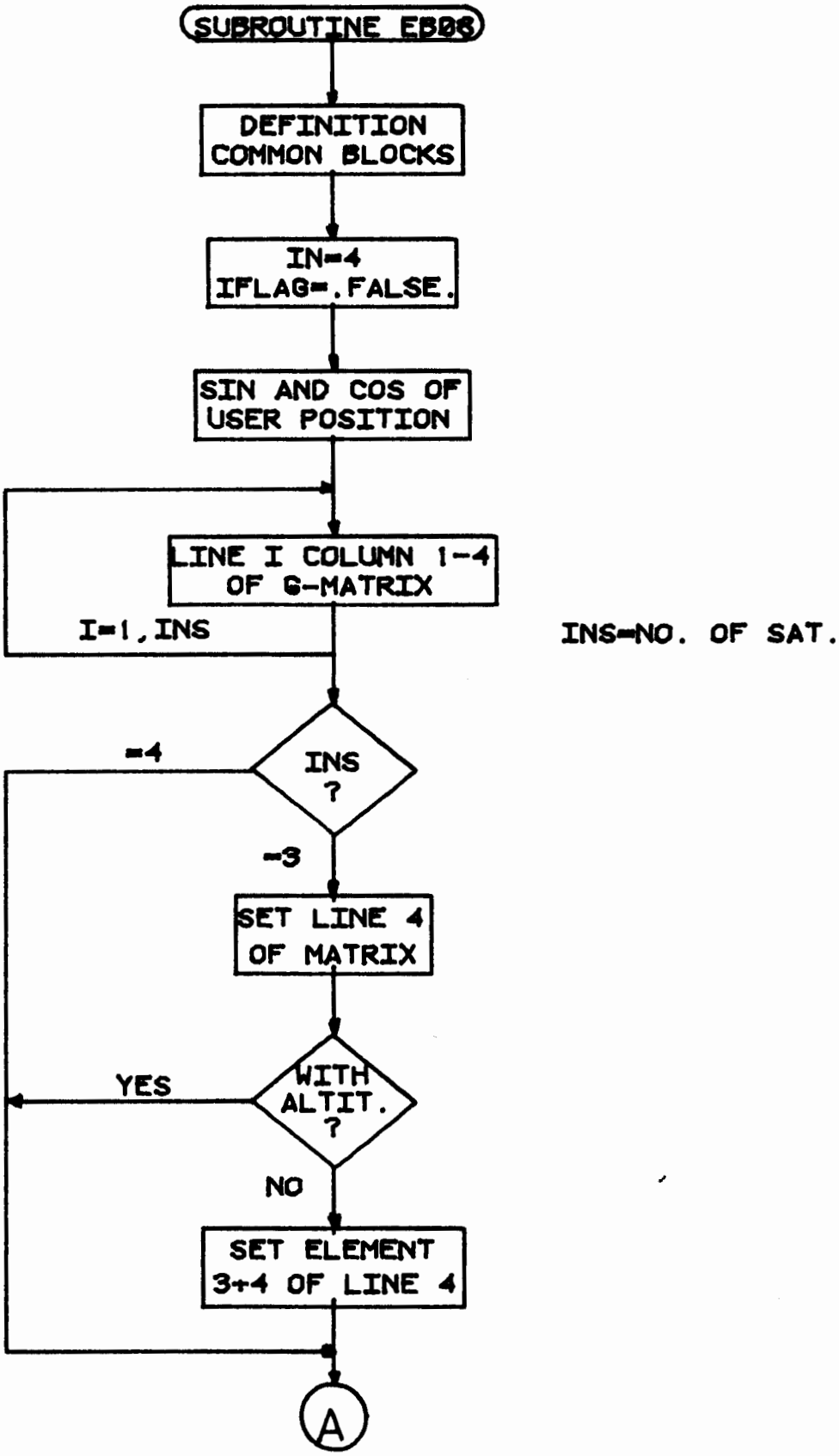

A
J=1
I
READ TIME
YES
END ?
NO
CONVERT TIME
READ RECORD
.TRUE.
LOG ?
.FALSE.
MOVE TO OLD POINT, SELECT LINESTYLE, DRAW TO NEW POINT
SAVE NEW POINT
I=1,N
LOG=.FALSE
I
B

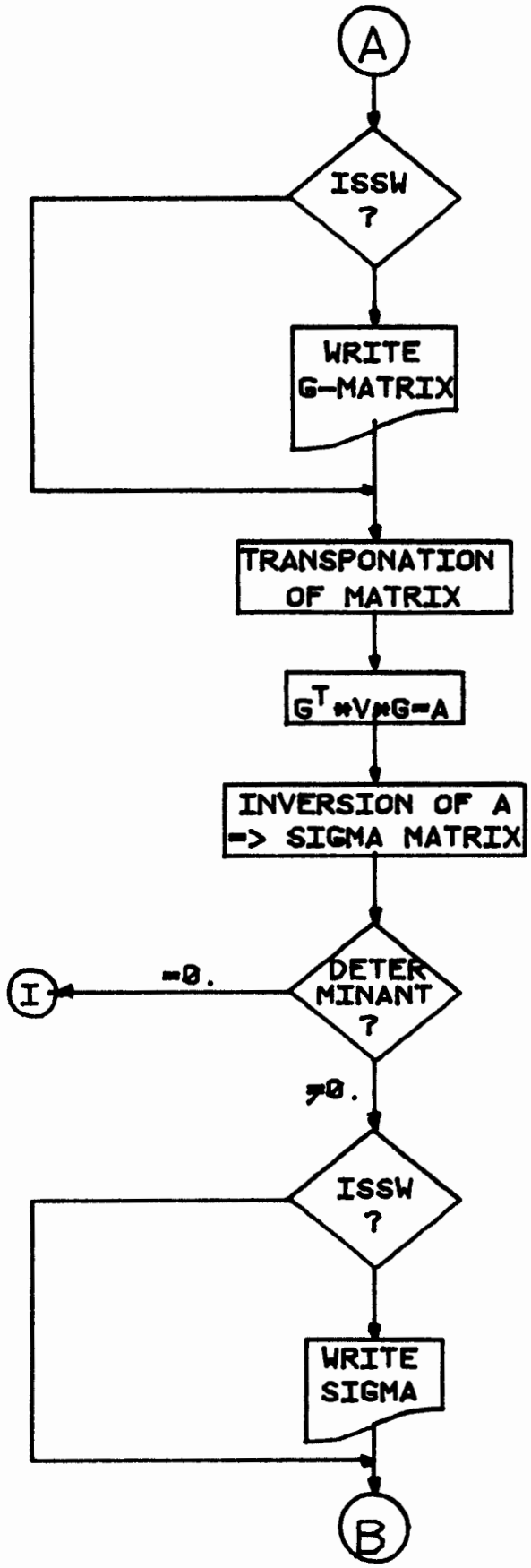

B
J
?
=2
=1
CLOSE LU=2
SELECT LINESTYLE
DEFINE VIEWPORT AND WINDOW
DRAW TITLES AND LABELS FOR Y-AXIS
DRAW LINE AT 10 DEG. ELEVATION
DRAW TITLES AND LABELS FOR X-AXIS
CLOSE PLOTTER
FORMATS
END
REWIND LU=2
LOG=.TRUE.
DEFINE VIEWPORT AND WINDOW
DRAW SECOND GRID
I

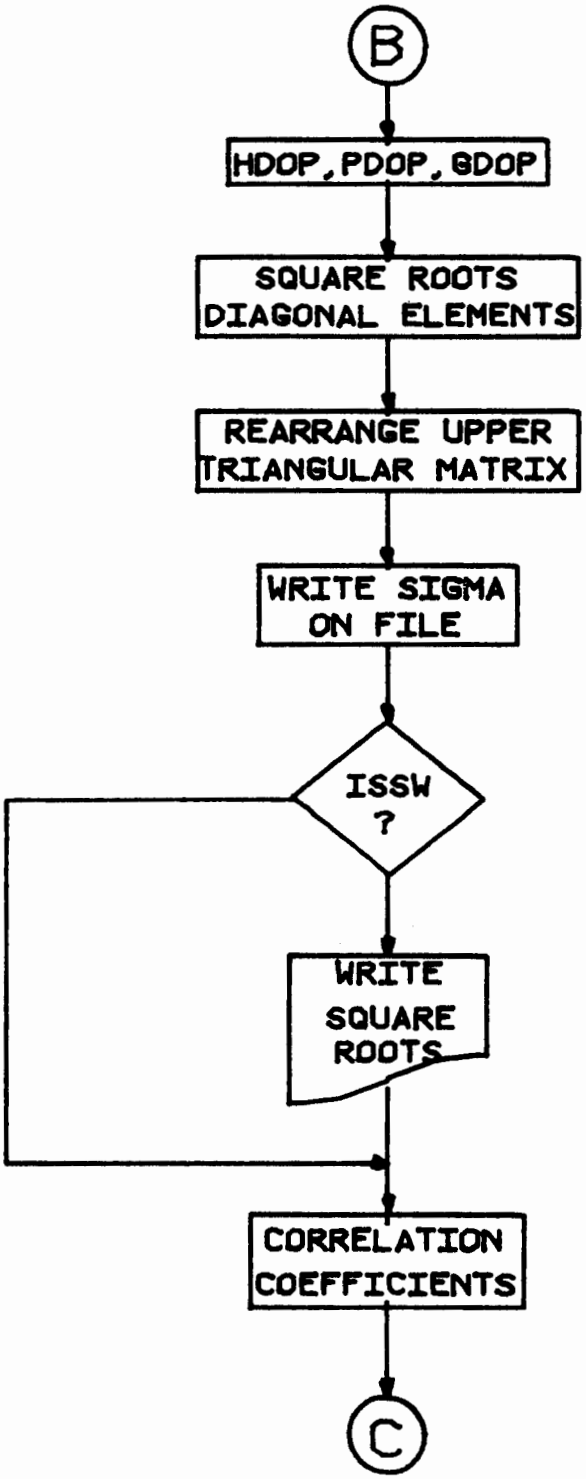

PROGRAM PL5FI
READ
LOGICAL UNIT AND
ID OF DEVICE SUBR
READ
LIMITS OF X-AXIS
INITIALIZE
PLOTTER
DRAW FRAME
DEFINE VIEWPORT
AND WINDOW
DRAW
LABELLED GRID
READ N
NO. OF FILES
I
DEFINE VIEWPORT
AND WINDOW
LOG=.TRUE.
A

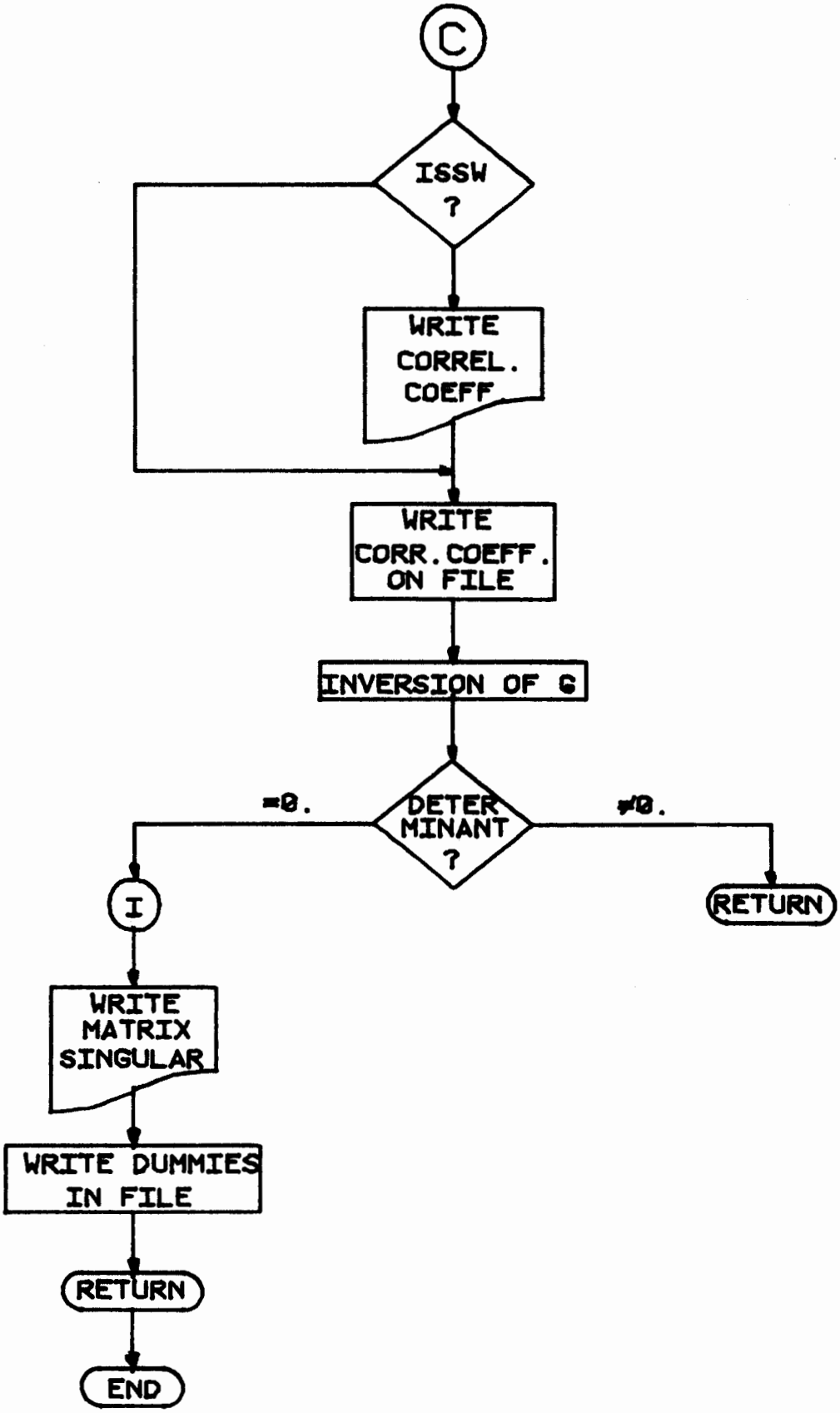

A
READ FILENAME
OPEN FILE LU=2
SELECT LINESTYLE
READ DATA FROM FILE
CONVERT TIME
LOG ?
.TRUE.
MOVE TO 1. POINT
.FALSE.
LOG=.FALSE.
X,Y <.01 ?
NO
DRAW TO NEW POINT
YES
CLOSE LU=2
B

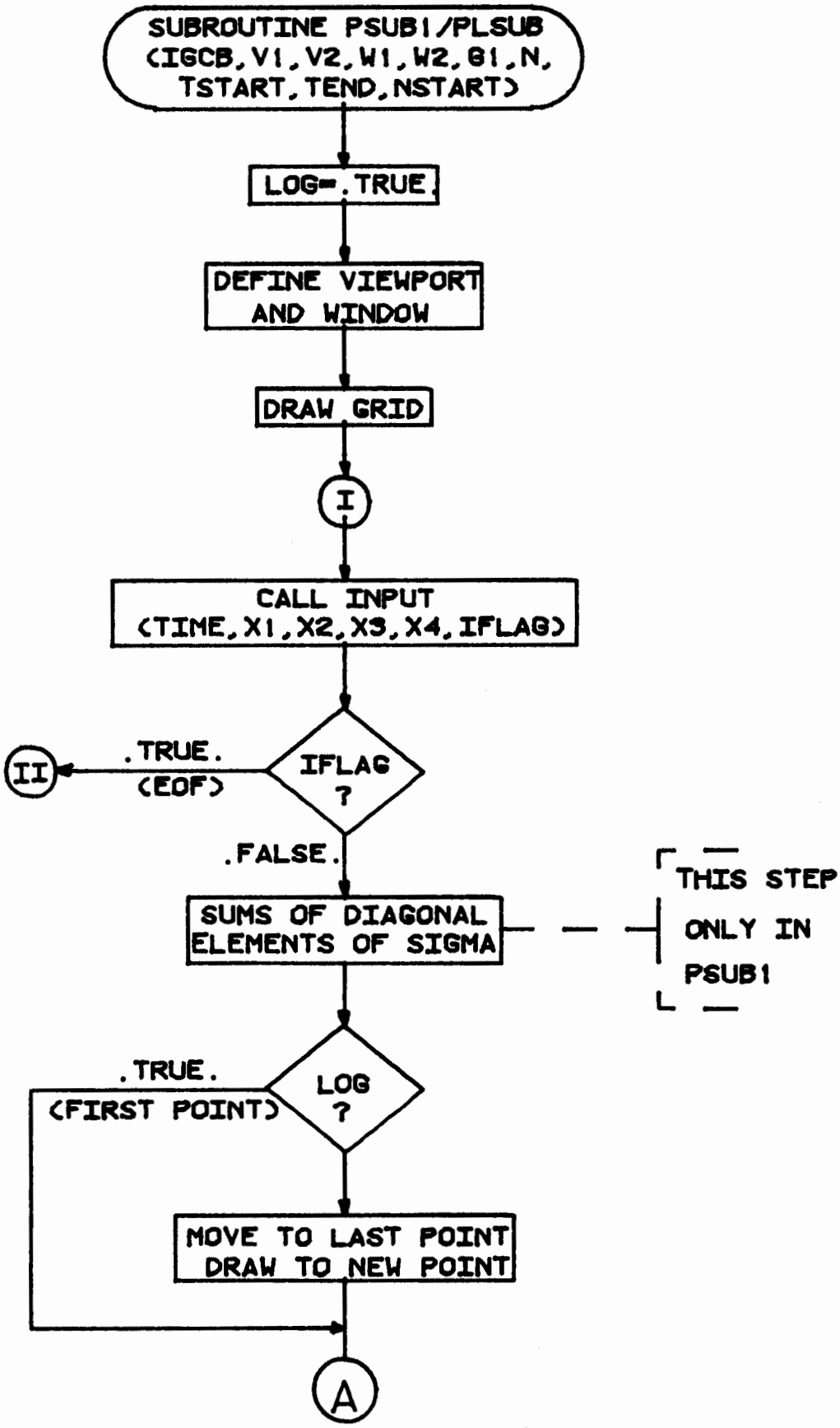

B
DEFINE VIEWPORT AND WINDOW
DRAW LEGEND FOR FILE
I=1,N
I
DRAW TITLES
CLOSE PLOTTER
FORMATS
END

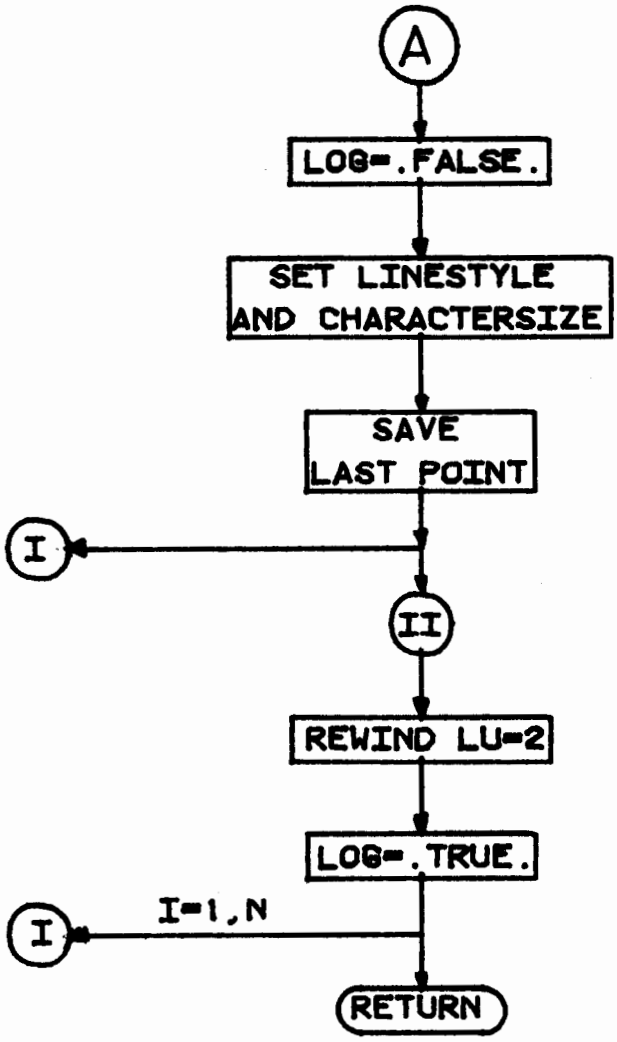

SUBROUTINE EB06
DEFINITION COMMON BLOCKS
IN=4
IFLAG=.FALSE.
SIN AND COS OF USER POSITION
LINE I COLUMN 1-4 OF G-MATRIX
I=1,INS
INS=NO. OF SAT.
INS ?
=4
=3
SET LINE 4 OF MATRIX
WITH ALTIT. ?
YES
NO
SET ELEMENT 3+4 OF LINE 4
A

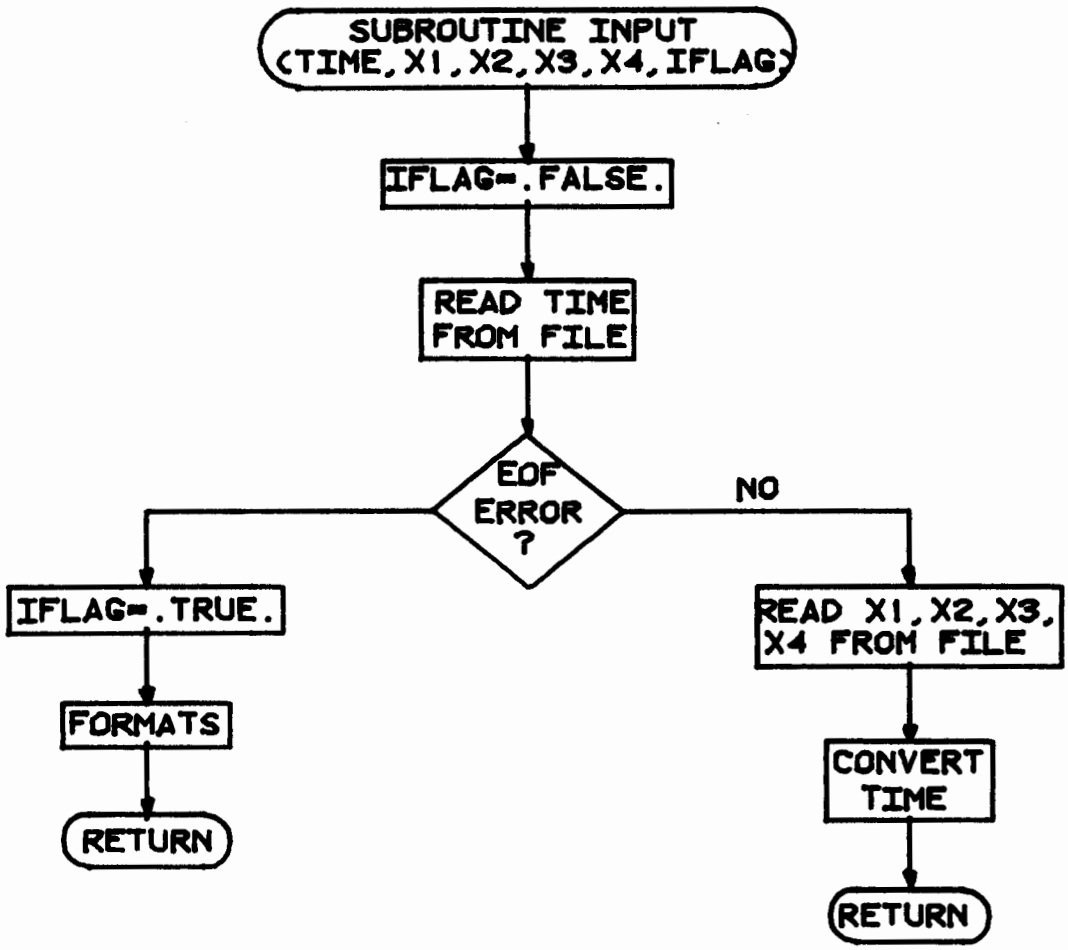

A
ISSW
?
WRITE
G-MATRIX
TRANSPONATION
OF MATRIX
G^T*V*G=A
INVERSION OF A
=> SIGMA MATRIX
DETER
MINANT
?
=0.
I
≠0.
ISSW
?
WRITE
SIGMA
B


B
HDOP, PDOP, GDOP
SQUARE ROOTS DIAGONAL ELEMENTS
REARRANGE UPPER TRIANGULAR MATRIX
WRITE SIGMA ON FILE
ISSW ?
WRITE SQUARE ROOTS
CORRELATION COEFFICIENTS
C


C
ISSW ?
WRITE CORREL. COEFF
WRITE CORR. COEFF. ON FILE
INVERSION OF G
DETER MINANT ?
=0.
≠0.
I
RETURN
WRITE MATRIX SINGULAR
WRITE DUMMIES IN FILE
RETURN
END


SUBROUTINE PSUB1/PLSUB
(IGCB,V1,V2,W1,W2,G1,N,
TSTART,TEND,NSTART)
LOG=.TRUE.
DEFINE VIEWPORT
AND WINDOW
DRAW GRID
I
CALL INPUT
(TIME,X1,X2,X3,X4,IFLAG)
IFLAG
?
.TRUE.
(EOF)
II
.FALSE.
SUMS OF DIAGONAL
ELEMENTS OF SIGMA
THIS STEP
ONLY IN
PSUB1
LOG
?
.TRUE.
(FIRST POINT)
MOVE TO LAST POINT
DRAW TO NEW POINT
A


A
LOG=.FALSE.
SET LINESTYLE
AND CHARACTERSIZE
SAVE
LAST POINT
I
II
REWIND LU=2
LOG=.TRUE.
I=1,N
I
RETURN


SUBROUTINE INPUT
(TIME,X1,X2,X3,X4,IFLAG)
IFLAG=.FALSE.
READ TIME
FROM FILE
EOF
ERROR
?
NO
IFLAG=.TRUE.
FORMATS
RETURN
READ X1,X2,X3,
X4 FROM FILE
CONVERT
TIME
RETURN

```
%EB00  T=00004 IS ON CR00011 USING 00114 BLKS R=0000

      FTN4X,L,Y
      C ************************
      C *   FILENAME : %EB00    *
      C ************************
            BLOCK DATA
            IMPLICIT DOUBLE PRECISION (A-H,K-Z)
            INTEGER N
            LOGICAL IFLAG
            COMMON /CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2
            COMMON/TIME/TDEL,TFIN
            COMMON/AZM/ELV(4),AZM(4)
            COMMON/POS/LAT,LON,ALT,R
            COMMON/NAC/TUY,TCY,TKC,DTC(4),DTU,TC,T
            COMMON/DB2/AD1(14)
            COMMON/DTX/U(4),D(4,4),RBAR(4)
            COMMON/SAT/S(4)
            COMMON/URE/URE1(4),DX(4)
            COMMON/GDOP/GDOP,PDOP,HDOP,INS,IFLAG,V(4,4),IST3,AP,AB
            COMMON/ELEV/E(3,3)
            COMMON/BLG/G(4,4),N
            COMMON/ALM/AD2(14,24),ID1(24)
            DATA PI/3.14159265D0/,RTD/57.2957795D0/,PI2/6.2831853D0/
            DATA OMF/.996647220D0/,AE/6.378135D6/
            DATA E2/6.694317778D-3/,OMF2/.993305683D0/
            DATA SGC/1.996499D7/,ERATE/7.292115147D-5/
            END
      $FILES (3,3)
            PROGRAM EB00
      C ***************************************************************
      C *  ROUTINE COMPUTES FOR A GIVEN DAY AND 3 OR 4 GIVEN SATELLITES  *
      C * FROM THE ALMANAC-DATA THE POSITION OF THE SATELLITES,THE       *
      C * DISTANCES TO THE USER,THE DILUTION-OF-PRECISION-PARAMETERS AND *
      C * WITH THE PSEUDO-RANGE-ERRORS (URE) THE GROUND-TRUTH.           *
      C *  THE SATELLITE-POSITIONS ARE COMPUTED BY A MODIFIED PROGRAM OF *
      C * THE TEXAS A&M-UNIVERSITY.                                      *
      C *  PLOTFILES WITH A TIMETAGGED HDOP ON LU=3 WITH THE SIGMA -     *
      C * MATRIX,SQUARE-ROOT OF DIAGONAL ELEMENTS AND THE GROUND-TRUTH   *
      C * ON LU=2 AND WITH THE ELEVATION AND AZIMUTH ON LU=30 ARE CREATED*
      C *                                                                *
      C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
      C *                                                                *
      C *   USED SUBROUTINES:     EB02                                   *
      C *                         EB03                                   *
      C *                         EB05                                   *
      C *                         EB06                                   *
      C *                         EB07                                   *
      C *                         EB08                                   *
      C *                         EB09B                                  *
      C *                         MINVD                                  *
      C *                         EB11                                   *
      C *                         EB12                                   *
      C *                         EB14                                   *
      C *                         DINTF                                  *
      C *                                                                *
      C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
      C *                                                                *
      C *  USED EARTH-REFERENCE-ELLIPSOID : WGS-72                       *
      C *                                                                *
```

```
C *   IMPORTANT VARIABLES AND CONSTANTS :                                *
C *   INTEGER : IST,IST1,IST2,IST3 --> CONTROL-PARAMETERS                *
C *             I,II,J,I1,I2,I3    --> INDEX FOR DO-LOOPS                *
C *             ICOUNT             --> COUNTER FOR OUTPUT                *
C *             N                  --> SIZE OF MATRIX (USED IN           *
C *                                    INVERSION-SUBROUTINE)             *
C *             INS                --> NUMBER OF SATELLITES FOR          *
C *                                    HDOP COMPUTATION                  *
C *             ISAT               --> INDEX OF SATELLITES               *
C *             ID (4)             --> PRN-NO. OF USED SATELLITES        *
C *             ID1 (24)           --> PRN-NO. OF ALL SATELLITES         *
C *             INAM (3)           --> NAME OF INPUT- AND PLOTFILE       *
C *             INUM               --> NUMBER OF POINTS IN URE-FILE*
C *   LOGICAL : IFLAG              --> INDICATES SUCCESS OF MATRIX-*
C *                                    INVERSION                         *
C *                                                                      *
C *   REAL    : A1,A2,T1,T2,TOUT,  --> ONLY FOR OUTPUT,THESE REAL-       *
C *             TMAX,TMIN              VARIABLES ARE NOT INVOLVED        *
C *                                    IN ANY SIGNIFICANT COMPUTAT.*
C *             ARR (15,4)         --> TIME OF DATA-POINT                *
C *             URE (15,4)         --> URE-VALUE AT POINT OF TIME        *
C *                                                                      *
C *   DOUBLE PRECISION :                                                 *
C *             AD (14,4)          --> ALMANAC FOR 4 SATELLITES          *
C *             AD1 (14)           --> ALMANAC JUST USED                 *
C *             AD2 (14,24)        --> ALMANAC FOR 24 SATELLITES         *
C *             ALT                --> ALTITUDE                          *
C *             AE                 --> RADIUS OF EARTH                   *
C *             AZM (4)            --> AZIMUTHANGLE OF SATELLITE         *
C *             AP,AB              --> VARIANCE OF SATELLITES URE-       *
C *                                    AND CLOCK-BIAS (3 SAT.,WITH-*
C *                                    OUT ALTIMETER-AIDING)             *
C *             A(4,4),AT(4,4),    --> MATRIX,INVERTED MATRIX AND        *
C *             AAT(4,4)               MULTIPLICATED MATRIX.             *
C *                                    A CONTAINS THE DIRECTION-         *
C *                                    COSINES IN ENU-COORDINATES        *
C *             D(4,4)             --> DISTANCES FROM USER TO 4          *
C *                                    SATELLITES IN ECEF-COORD.         *
C *             DTC(4)             --> DATE OF COLLECTION IN DAYS        *
C *                                    FROM BEGINNING OF THE YEAR        *
C *             DTU                --> DATE OF USE IN DAYS (CONVER-*
C *                                    TED INTO SECONDS) FROM            *
C *                                    BEGINNING OF THE YEAR             *
C *             DX(4)              --> USER-ERRORS DUE TO SPACE-         *
C *                                    SEGMENT OF GPS                    *
C *             E (3,3)            --> MATRIX CONTAINING SINES AND *
C *                                    COSINES OF USER POSITION          *
C *             E2                 --> SQUARE-ROOT OF THE EXCENTRI-*
C *                                    CITY OF THE EARTH-ELLIPSOID *
C *             ELV (4)            --> ELEVATIONANGLE OF SATELLITES*
C *             ERATE              --> EARTH-ROTATION-RATE               *
C *             G (4,4)            --> MATRIX FOR INVERSION-SUBROU.*
C *             GDOP               --> GEOMETRIC DILUTION OF PRECI-*
C *                                    SION                              *
C *             HDOP               --> HORIZONTAL DILUTION OF            *
C *                                    PRECISION                         *
C *             LAT                --> LATITUDE                          *
C *             LON                --> LONGITUDE                         *
C *             PI,PI2             --> PI(3.1415...) AND 2*PI            *
C *             PDOP               --> POSITION,DILUTION OF              *
```

```
C *                                          PRECISION                    *
C *              RTD                     --> RADIAN TO DEGREE CONVERSION *
C *                                          FACTOR (180/PI)              *
C *              RBAR                    --> LENGTH OF VECTOR USER-SATEL.*
C *              S (4)                   --> SATELLITE-POSITION IN ECEF  *
C *              SGC                     --> SQUARE ROOT OF GRAVITATIONAL*
C *                                          CONSTANT OF THE EARTH        *
C *                      ALL TIMES IN GMT AND IN SECONDS                  *
C *              T                       --> GPS-TIME                     *
C *              TC                      --> COLLECTION TIME,START OF     *
C *                                          COUNT AT BEGINNING OF EPOCH *
C *              TCY                     --> TIME OF COLLECTIONS STARTING*
C *                                          AT BEGINNING OF YEAR         *
C *              TDEL                    --> TIMEINCREMENT                *
C *              TFIN                    --> ENDTIME                      *
C *              TKC                     --> DIFFERENCE BETWEEN COLLEC-   *
C *                                          TIONTIME AND REFERENCETIME, *
C *                                          WRAPAROUNDS CONSIDERED       *
C *              TUY                     --> TIME OF USE                  *
C *              U (4)                   --> USER POSITION IN ECEF-COORD.*
C *              URE1(4)                 --> PSEUDO-RANGE-ERRORS          *
C *              V(4,4)                  --> KOVARIANCE-MATRIX            *
C *                                                                       *
C ***********************************************************************
C
C
      IMPLICIT DOUBLE PRECISION(A-H,K-Z)
      REAL A1,A2,URE(15,4),ARR(15,4),T2,T1,TMAX,TMIN,TOUT,TEXT(2)
      LOGICAL IFLAG
      DIMENSION AD(14,4),ID(4),INAM(3),INAM1(3),INAM2(3)
      COMMON/TIME/TDEL,TFIN
      COMMON/AZM/ELV(4),AZM(4)
      COMMON/POS/LAT,LON,ALT,R
      COMMON/NAC/TUY,TCY,TKC,DTC(4),DTU,TC,T
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2
      COMMON/DB2/AD1(14)
      COMMON/DTX/U(4),D(4,4)
      COMMON/SAT/S(4)
      COMMON/URE/URE1(4),DX(4)
      COMMON/ALM/AD2(14,24),ID1(24)
      COMMON/GDOP/GDOP,PDOP,HDOP,INS,IFLAG,V(4,4),IST3,AP,AB
      DATA TEXT/'YES ','NO'/
      ISAT=0
C!!   AB=0.0
C!!   AP=0.0
      TC=0.
      ICOUNT=0
      WRITE (6,90)
C .......................................................................
C
C                          INPUT
C
C ********** INPUT THE LATITUDE,LONGITUDE AND THE ALTITUDE **********
C          LON WEST AND LAT SOUTH ARE DENOTED BY A NEGATIVE VALUE
C
      WRITE (16,1)
1     FORMAT (/'    INPUT USER POSITION'
     1/          " ATLANTIC CITY (=1) OTHER POSITION (=2)"
     2/          " ANSWER :_")
      READ (16,98) IST
```

```
0179        IF(IST.EQ.2) GOTO 1001
0180  C ----------------------
0181        LAT=0.3944955D2
0182        LON=-0.74566777D2
0183        ALT=0.4626D2
0184        GOTO 1002
0185  C ----------------------
0186  1001  CONTINUE
0187        WRITE (16,2)
0188  2     FORMAT (/"   USER POSITION IN DEGREES (FORMAT :G15.8)"
0189       1/         " LATITUDE          :_")
0190        READ (16,97) LAT
0191        WRITE (16,3)
0192  3     FORMAT (" LONGITUDE         :_")
0193        READ (16,97) LON
0194        WRITE (16,4)
0195  4     FORMAT (" ALTITUDE IN FEET :_")
0196        READ (16,97) ALT
0197  1002  CONTINUE
0198  C --------------------
0199        WRITE (6,50) LAT,LON,ALT
0200  50    FORMAT (" PROGRAM   * E B 0 0 *"
0201       1/         " ---------------------"
0202       2//        " USER-POSITION :  "
0203       3/         "   LATITUDE :",G15.8," DEGREE"
0204       4/         "   LONGITUDE:",G15.8," DEGREE"
0205       5/         "   ALTITUDE :",G15.8," FEET")
0206        LAT=LAT/RTD
0207        LON=LON/RTD
0208        ALT=ALT/0.32809D1
0209  C
0210  C ********** EB02 COMPUTES E-MATRIX **********
0211  C
0212        CALL EB02
0213  C
0214  C ********** EB03 TAKES LAT, LON, AND ALT; **********
0215  C             IT RETURNS X,Y,Z IN U-VECTOR
0216  C
0217        CALL EB03
0218        WRITE (6,51) (U(I),I=1,3)
0219  51    FORMAT ( " ECEF-COORDINATES :  X = ",G16.8,"  Y = ",G16.8,"  Z =
0220       1 ",G16.8)
0221        U(4)=0.
0222  C
0223  C ********** READ USER TIME **********
0224  C
0225        WRITE (16,5)
0226  5     FORMAT (/" USER TIME (FORMAT : G15.8)"
0227       1/         " STARTTIME(TSTART),ENDTIME(TFIN),TIME INCREMENT(TDEL)"
0228       2/         " AND ALMANAC-COLLECTION-TIME (TC) IN SECONDS."
0229       3/         " START OF COUNT:TSTART,TFIN,TDEL-->EVERY DAY AT 00:00"
0230       4/         "                   TC                -->BEGINNING OF EPOCH"
0231       5/         " TSTART=_")
0232        READ (16,97) T
0233        WRITE (16,6)
0234  6     FORMAT (" TFIN  =_")
0235        READ (16,97) TFIN
0236        WRITE (16,7)
0237  7     FORMAT (" TDEL  =_")
0238        READ (16,97) TDEL
```

```
0239          WRITE (16,8)
0240   8      FORMAT ( " TC     =_")
0241          READ (16,97) TC
0242          WRITE (16,10)
0243   10     FORMAT (" DATE OF USE IN DAYS FROM JANUARY,FIRST :_")
0244          READ (16,97) DTU
0245          WRITE (16,9)
0246   9      FORMAT ( " NUMBER OF SATELLITES (3 OR 4) =_")
0247          READ (16,98) INS
0248          IF(INS.EQ.4) GOTO 1016
0249          WRITE (16,34)
0250   34     FORMAT ( " SYSTEM ALTIMETER AIDED (=1) :_")
0251          READ (16,98) IST3
0252          IF(IST3.NE.1) IST3=2
0253   C!!    IF(IST3.EQ.1) GOTO 1016
0254   C!!    IST3=2
0255   C!!    WRITE (16,35)
0256   C!!35     FORMAT ( " SATELLITE URE- (AP) AND CLOCK-VARIANCE (AB)"
0257   C!!   1/          " AP (SQUARED) :_")
0258   C!!    READ (16,97) AP
0259   C!!    WRITE (16,36)
0260   C!!36     FORMAT ( " AB (SQUARED) :_")
0261   C!!    READ (16,97) AB
0262   C ---------------
0263   1016   CONTINUE
0264          CALL EB12(T,T1)
0265          CALL EB12(TFIN,T2)
0266          WRITE (6,52) T,T1,TFIN,T2,TDEL,DTU,INS
0267   52     FORMAT (/" USERTIMES :"
0268         1/          "    START (TSTART): ",G15.8," SEC. = ",F5.2," HOURS"
0269         2/          "    END (TFIN)    : ",G15.8," SEC. = ",F5.2," HOURS"
0270         3/          "    INCR. (TDEL)  : ",G15.8," SEC"
0271         4/          "    DATE (DTU)    : ",G15.8," DAYS"
0272         5//         " THE COMPUTATIONS ARE MADE WITH",I2," SATELLITES")
0273          IF(INS.EQ.3) WRITE (6,66) TEXT(IST3)   !!,AP,AB
0274   66     FORMAT (/" SYSTEM ALTIMETER-AIDED : ",A4)
0275   C!!   1/          " AP-SQUARED :",G8.4,"    AB-SQUARED :",G8.4,
0276   C!!   2"    (APPLICABLE : 3 SATELLITES WITHOUT ALTIMETER)")
0277   C
0278   C ********** READ ALMANAC - DATA **********
0279   C
0280          WRITE (16,12)
0281   12     FORMAT ( " ALMANAC-DATA FROM TAPE (=1) ? _")
0282          READ (16,98) IST1
0283   1000   CONTINUE
0284          ISAT=ISAT+1
0285          IF(IST1.EQ.1) GOTO 1009
0286          WRITE (16,13) ISAT
0287   13     FORMAT (/" OWN ALMANAC =1 OR DATA FROM FILE =2 FOR SV NO.",I2,"?"
0288         1/          " ANSWER :_")
0289          READ (16,98) IST
0290          IF(IST.EQ.1) GOTO 1005
0291   C -----------------
0292          WRITE (16,14)
0293   14     FORMAT (/" FILENAME :_")
0294          READ(16,92) INAM
0295          OPEN (2,FILE=INAM,FORM='FORMATTED')
0296          READ (2,96) DTC(ISAT),ID(ISAT)
0297          READ (2,91) (AD(I,ISAT),I=1,4)
0298          READ (2,91) (AD(I,ISAT),I=5,8)
```

```
0299          CLOSE(2)
0300          WRITE (16,15) ID(ISAT)
0301   15     FORMAT ( " SATELLITE ID = ",I2)
0302          GOTO 1010
0303   C ----------------
0304   1005   CONTINUE
0305          WRITE (16,16) ISAT
0306   16     FORMAT ( " ####### ALMANAC-DATA FOR SATELLITE ",I2,"#######"
0307         1/          " FORMAT :D12.4"
0308         2//         " DATE OF COLLECTION                    :_")
0309          READ (16,99) DTC(ISAT)
0310          WRITE (16,17)
0311   17     FORMAT (" EXCENTRICITY                        :_")
0312          READ (16,99) AD(1,ISAT)
0313          WRITE (16,18)
0314   18     FORMAT (" REFERENCE TIME                      :_")
0315          READ (16,99) AD(2,ISAT)
0316          WRITE (16,19)
0317   19     FORMAT (" ORBIT INCLINATION (IO)              :_")
0318          READ (16,99) AD(3,ISAT)
0319          WRITE (16,20)
0320   20     FORMAT (" DRIFT RATE OF (IO)                  :_")
0321          READ (16,99) AD(4,ISAT)
0322          WRITE (16,21)
0323   21     FORMAT (" SQARE ROOT OF GREAT HALF-AXIS       :_")
0324          READ (16,99) AD(5,ISAT)
0325          WRITE (16,22)
0326   22     FORMAT (" RIGHT ASCENSION OF ORBIT            :_")
0327          READ (16,99) AD(6,ISAT)
0328          WRITE (16,23)
0329   23     FORMAT (" ARGUMENT OF PERIGEE                 :_")
0330          READ (16,99) AD(7,ISAT)
0331          WRITE (16,24)
0332   24     FORMAT (" MEAN ANOMALY                        :_")
0333          READ (16,99) AD(8,ISAT)
0334          WRITE (16,25)
0335   25     FORMAT (" SATELLITE ID                        :_")
0336          READ (16,98) ID(ISAT)
0337          GOTO 1010
0338   C ------------
0339   1009   CONTINUE
0340          IF(ISAT.NE.1) GOTO 1010
0341          CALL EB14
0342          DO 107 I1=1,INS
0343   1011   CONTINUE
0344          WRITE (16,26) I1
0345   26     FORMAT ( " ID OF SATELLITE (",I2,") : _")
0346          READ (16,98) ID(I1)
0347          DO 108 I2=1,24
0348          IF(ID(I1).NE.ID1(I2).AND.I2.LT.24) GOTO 108
0349          IF(ID(I1).EQ.ID1(I2)) GOTO 1012
0350          WRITE (16,27) ID(I1)
0351   27     FORMAT ( " =====> SATELLITE-ALMANAC OF SAT.(",I2,") NOT ON"
0352         1" TAPE !! <=====")
0353          GOTO 1011
0354   108    CONTINUE
0355   1012   CONTINUE
0356          DO 107 I3=1,8
0357          AD(I3,I1)=AD2(I3,I2)
0358          ID(I1)=ID1(I2)
```

```
107    CONTINUE
C -----------------
1010   CONTINUE
       AD(9,ISAT)=AD(5,ISAT)**2
       AD(10,ISAT)=COS(AD(3,ISAT))
       AD(11,ISAT)=SIN(AD(3,ISAT))
       AD(12,ISAT)=COS(AD(7,ISAT))
       AD(13,ISAT)=SIN(AD(7,ISAT))
       AD(14,ISAT)=SGC/(AD(5,ISAT)*AD(9,ISAT))
       IF(ISAT.LT.INS) GOTO 1000
C
C ********** INPUT V-MATRIX AND INVERSION **********
C
       WRITE (16,28)
28     FORMAT (/" V-MATRIX =UNITY-MATRIX (=1) ? "
      1/          " ANSWER :_")
       READ (16,98) IST
       IF(IST.EQ.1) GOTO 1014
       WRITE (16,29)
29     FORMAT ( " INPUT DIAGONAL-ELEMENTS OF MATRIX  "
      1/          " !! PARAMETERS HAVE TO CORRESPOND TO SATELLITES !!")
       DO 110 I=1,4
       WRITE (16,30) I,I
30     FORMAT ( " ELEMENT (",I2,",",I2,") :_")
       READ (16,97) V(I,I)
       V(I,I)=1./V(I,I)
110    CONTINUE
       GOTO 1015
1014   CONTINUE
       DO 111 I=1,4
       V(I,I)=1.
111    CONTINUE
1015   CONTINUE
       DO 112 I=1,4
       DO 112 J=1,4
       IF(I.EQ.J) GOTO 112
       V(I,J)=0.
112    CONTINUE
C
C ********** INPUT USER-      -ERROR      **********
C
       DO 109 II=1,4
       WRITE (16,31) ID(II)
31     FORMAT ( "  FILENAME FOR URE OF SAT.(ID=",I2,") :_")
       READ (16,92) INAM
       OPEN(2,FILE=INAM,FORM='FORMATTED')
       READ(2,98) INUM
       DO 103 I=1,INUM
       READ (2,*) ARR(I,II),URE(I,II)
       ARR(I,II)=ARR(I,II)*3600.
103    CONTINUE
       CLOSE (2)
109    CONTINUE
C
C ********** INPUT NAME OF PLOTFILE **********
C
       WRITE (16,32)
32     FORMAT (/" NAME OF PLOTFILE FOR SIGMA-MATRIX :_")
       READ (16,92) INAM
       WRITE (16,33)
```

```
0419  33    FORMAT ( " NAME OF PLOTFILE FOR HDOP :_")
0420        READ (16,92) INAM1
0421        WRITE (16,37)
0422  37    FORMAT ( " NAME OF PLOTFILE FOR AZIMUTH AND ELEVATION :_")
0423        READ (16,92) INAM2
0424  C ..................................................................
0425  C
0426  C ********** OUTPUT ALMANAC AND URE **********
0427  C
0428        A1=SNGL(DTC(1))
0429        A2=SNGL(TC)
0430        WRITE (6,53) A1,A2,(ID(I),I=1,4)
0431  53    FORMAT (/"     USED ALMANACS            DTC= ",F5.0,"    TC= ",F6.0,
0432       1/          "     -------------"
0433       2/          " SAT. ID : ",I3,19X,I3,19X,I3,19X,I3)
0434        WRITE (6,54) ((AD(I,J),J=1,4),I=1,14)
0435  54    FORMAT (4(10X,D12.4,))
0436        WRITE (6,55) (ID(I),I=1,4)
0437  55    FORMAT (/"     USED URES"
0438       1/          "     ---------"
0439       2/          " SAT. ID : ",I3,25X,I3,25X,I3,25X,I3)
0440        WRITE (6,56) (((ARR(I,J),URE(I,J)),J=1,4),I=1,INUM)
0441  56    FORMAT (4(2E12.4,4X))
0442  C
0443  C ********** START- AND ENDTIMECONTROL **********
0444  C
0445        T3=DBLE(AMAX1(ARR(1,1),ARR(1,2),ARR(1,3),ARR(1,4)))
0446        IF(T3.LE.T) GOTO 1003
0447        T=T3
0448        CALL EB12(T,TMAX)
0449        WRITE (6,57) TMAX,T
0450  57    FORMAT (/" ---> URE-DATA STARTS AT ",F7.4," HOURS."
0451       1/          " ==> NEW TSTART = ",D12.4," SEC.")
0452  1003  CONTINUE
0453        T3=DBLE(AMIN1(ARR(INUM,1),ARR(INUM,2),ARR(INUM,3),ARR(INUM,4)))
0454        IF(T3.GE.TFIN) GOTO 1007
0455        TFIN=T3
0456        CALL EB12(TFIN,TMIN)
0457        WRITE (6,58) TMIN,TFIN
0458  58    FORMAT (/" ---> URE-DATA ENDS AT ",F7.4," HOURS."
0459       1/          " ==> NEW TFIN   = ",D12.4," SEC.")
0460  1007  CONTINUE
0461        WRITE (6,59)
0462  59    FORMAT (/" INVERTED V-MATRIX ")
0463        WRITE (6,91) ((V(I,J),J=1,4),I=1,4)
0464  C ..................................................................
0465  C
0466  C ********** WRITE TITLE **********
0467  C
0468        WRITE (6,60) INAM,INAM1,INAM2
0469  60    FORMAT (/" NAME OF PLOTFILE FOR SIGMA-MATRIX            : ",3A2,
0470       1/          " NAME OF PLOTFILE FOR HDOP                  : ",3A2,
0471       2/          " NAME OF PLOTFILE FOR AZIMUTH AND ELEVATION : ",3A2)
0472        IF(ISSW(0).GE.0) WRITE (6,61)
0473  61    FORMAT ( "1OUTPUT OF COMPUTATIONS "
0474       1/          " --------------------- ")
0475        DTU=(DTU-1.0)*0.8640D5
0476        TCY=(DTC(1)-1.0)*0.864D5+DMOD(TC,0.8640D5)
0477        OPEN (2,FILE=INAM,FORM='FORMATTED')
0478        OPEN (3,FILE=INAM1,FORM='FORMATTED')
```

```
      OPEN (30,FILE=INAM2,FORM='FORMATTED')
1004  CONTINUE
      ICOUNT=ICOUNT+1
      CALL EB12 (T,TOUT)
      WRITE (2,97) TOUT
      WRITE (30,97) TOUT
      IF(ISSW(0).GE.0) WRITE (6,62) TOUT
62    FORMAT (/" TIME : ",F12.4," HOURS "
     1/,15X," SATELLITE POSITIONS"
     2/,11X,"X",17X,"Y",17X,"Z",16X,"ELV",15X,"AZM")
      DO 102 I=1,INS
      DO 101 J=1,14
      AD1(J)=AD(J,I)
101   CONTINUE
C
C ********** COMPUTATION OF POSITION,DISTANCE TO USER **********
C              AND AZIMUTH AND ELEVATION OF SATELLITE (I)
C
      II=I
      CALL EB05(II)
102   CONTINUE
C
C ********** COMPUTATION OF G-MATRIX,HDOP AND GDOP **********
C
C!!   IF((INS.EQ.4).OR.(INS.EQ.3.AND.IST3.EQ.1)) CALL EB08
C!!   IF(INS.EQ.3.AND.IST3.EQ.2) CALL EB08B
      CALL EB08
      IF(IFLAG) GOTO 1013
C
C ********** INTERPOLATION OF URE-DATA **********
C
      DO 104 I=1,4
      DO 105 J=1,INUM-1
      IF(T-ARR(J,I).GE.0.AND.ARR(J+1,I)-T.GT.0) GOTO 1008
105   CONTINUE
1008  CONTINUE
      B=DBLE((URE(J+1,I)-URE(J,I))/(ARR(J+1,I)-ARR(J,I)))
      A=DBLE(URE(J,I))-DBLE(ARR(J,I))*B
      URE1(I)=A+B*T
104   CONTINUE
C
C ********** COMPUTATION OF GROUND-TRUTH **********
C
      CALL EB11
      IF(ISSW(0).GE.0) WRITE (6,63)
63    FORMAT (/" GROUND-TRUTH AND FIGURES OF MERIT ")
      IF(ISSW(0).GE.0) WRITE (6,64)
64    FORMAT (10X,"DX",16X,"DY",16X,"DZ",15X,"-DB",13X,"HDOP",8X,
     1"PDOP",8X,"GDOP")
      IF(ISSW(0).GE.0) WRITE (6,65) (DX(I),I=1,4),HDOP,PDOP,GDOP
65    FORMAT (4D18.8,4X,3(2X,G10.4))
      A1=SNGL(HDOP)
      WRITE (3,89) TOUT,HDOP
1013  CONTINUE
      WRITE (2,94) DX
      IF(ISSW(0).GE.0) WRITE (6,90)
      T=T+TDEL
      IF(T.LE.TFIN) GOTO 1004
      CLOSE (2)
      WRITE (3,89) 0.,0.
```

```
      CLOSE (3)
      CLOSE (30)
      STOP
C
C ********** FORMATS **********
C
89    FORMAT (2F12.6)
90    FORMAT (1H1)
91    FORMAT (4D18.8)
92    FORMAT (3A2)
93    FORMAT (I4)
94    FORMAT (4E12.4)
95    FORMAT (A4)
96    FORMAT (G15.8,I3)
97    FORMAT(G15.8)
98    FORMAT(I2)
99    FORMAT(D12.4)
      END
```

```
&FILE  T=00004 IS ON CR00011 USING 00006 BLKS R=0000

FTN4X,L,Y
$FILES (1,1)
      PROGRAM FILE
C ***********************************************************
C *          FILENAME  :  &FILE                             *
C *     PROGRAM TO READ URE-DATA AND CREATE FILES.          *
C ***********************************************************
C
      DIMENSION URE(15),TIME(15),INAME(3)
      WRITE (16,1)
1     FORMAT (" NUMBER OF FILES :_")
      READ (16,99) N1
      DO 100 I=1,N1
      WRITE (16,2) I
2     FORMAT ("NUMBER OF RECORDS IN FILE ",I2," :_")
      READ (16,99) N2
      WRITE (16,3) I
3     FORMAT ("NAME OF FILE ",I2," :_")
      READ (16,98) INAME
      OPEN (2,FILE=INAME,FORM='FORMATTED')
      WRITE (2,99) N2
C -------------------
      DO 101 J=1,N2
      WRITE (16,4) J
4     FORMAT (" TIME(",I2,") =_")
      READ (16,97) TIME (J)
      WRITE (16,5) J
5     FORMAT ("  URE(",I2,") :_")
      READ (16,97) URE(J)
      WRITE (2,96) TIME(J),URE(J)
101   CONTINUE
      CLOSE(2)
100   CONTINUE
C -------------------
96    FORMAT (2E12.4)
97    FORMAT (E12.4)
98    FORMAT (3A2)
99    FORMAT (I2)
      STOP
      END
```

```
&PLCOR T=00004 IS ON CR00002 USING 00030 BLKS R=0000

FTN4X,L,Y
$FILES (1,1)
      PROGRAM PLCOR
C ***************************************************************
C *                FILENAME :   &PLCOR                          *
C *                ^^^^^^^^^^^^^^^^^^^                          *
C * PROGRAM PLOTS DATA FILES GENERATED BY THE PROGRAM "EB00"    *
C * THE FILES CONTAIN VALUES IN THE FOLLOWING ORDER :           *
C *     TIME IN HOURS AND MINUTES                               *
C *     UPPER-TRIANGULAR SIGMA-MATRIX (X1)                      *
C *     SQUARE-ROOTS OF DIAGONAL ELEMENTS (X2)                  *
C *     CORRELATION-COEFFICIENT (X3)                            *
C *     GROUND-TRUTH (X4)                                       *
C *                                                             *
C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
C * PLOTTED VALUES ARE :                                        *
C *    SQUARE-ROOT OF DIAGONAL ELEMENTS                         *
C *    SQUARE-ROOT OF THE SUM OF THE DIAGONAL ELEMENTS          *
C *    GROUND-TRUTH                                             *
C *    CORRELATION-COEFFICIENT                                  *
C *                                                             *
C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
C *   USED SUBROUTINES : DLTBL,GPS --> PLOTLIBRARY              *
C *                      INPUT,PLSUB                            *
C *                                                             *
C ***************************************************************
C
      DIMENSION IGCB(192),INAM(3),IDATE(3)
C
C ********** READ NAME OF FILE,LOGICAL UNIT AND ID OF **********
C                        DEVICE SUB
C
      WRITE (16,1)
1     FORMAT (" NAME OF PLOTFILE :_")
      READ (16,92) INAM
      OPEN (2,FILE=INAM,FORM='FORMATTED')
      WRITE (16,3)
3     FORMAT ( " LOGICAL UNIT (LU) AND ID OF DEVICE-SUBROUTINE"
     1/         "   LU =_")
      READ (16,93) LU
      WRITE (16,4)
4     FORMAT ( "   ID =_")
      READ (16,93) ID
C
C ********** INITIALIZE PLOTTER, GET LIMITS OF X-AXIS **********
C            AND LABELS AND DRAW VERTICAL GRID
C
      WRITE (16,5)
5     FORMAT ( " START-AND ENDTIME IN HOURS."
     1/         " TSTART :_")
      READ (16,94) TSTART
      WRITE (16,6)
6     FORMAT ( " TEND   :_")
      READ (16,94) TEND
      WRITE (16,19)
19    FORMAT ( " DATE (MMDDYY) :_")
      READ (16,92) IDATE
      CALL PLOTR (IGCB,ID,1,LU)
```

```
      CALL SETAR (IGCB,1.1)
      CALL FRAME (IGCB)
C ..................................................................
C                        PLOTTING OF DATA
C
C ********** GROUND TRUTH **********
C
      CALL PLSUB (IGCB,10.,28.,-20.,20.,20.,4,TSTART,TEND,21)
C
C ********** CORRELATION-COEFFICIENTS **********
C
      CALL PLSUB (IGCB,28.,46.,-1.,1.,1.,3,TSTART,TEND,15)
      CALL PLSUB (IGCB,46.,64.,-1.,1.,1.,3,TSTART,TEND,18)
C
C ********** SQUARE-ROOTS OF THE SUM OF THE DIAGONAL ELEMENTS ********
C
      CALL PSUB1 (IGCB,64.,82.,0.,40.,20.,3,TSTART,TEND,11)
C
C ********** SQUARE-ROOT OF DIAGONAL ELEMENTS **********
C
      CALL PLSUB (IGCB,82.,100.,0.,20.,10.,4,TSTART,TEND,11)
C
C ********** LABELS **********
C
      CALL VIEWP (IGCB,0.,100.,0.,100.)
      CALL WINDW (IGCB,0.,100.,0.,100.)
      CALL CSIZE (IGCB,1.,1.,0.)
C -----------------
      CALL MOVE  (IGCB,5.,98.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 20.
      CALL MOVE (IGCB,5.,83.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 0.
      CALL MOVE (IGCB,1.,91.)
      CALL LABEL (IGCB)
      WRITE (LU,7)
7     FORMAT ( "SQUARE")
      CALL MOVE (IGCB,1.,89.)
      CALL LABEL (IGCB)
      WRITE (LU,8)
8     FORMAT ( "ROOTS")
      CALL MOVE (IGCB,10.,82.)
      CALL LINE (IGCB,0)
      CALL DRAW (IGCB,0.,82.)
C -----------------------
      CALL MOVE (IGCB,5.,80.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 40.
      CALL MOVE (IGCB,1.,73.)
      CALL LABEL (IGCB)
      WRITE (LU,21)
21    FORMAT ( "SQUARE-ROOTS" )
      CALL MOVE (IGCB,1.,71.)
      CALL LABEL (IGCB)
      WRITE (LU,9)
9     FORMAT ( "OF SUMS")
      CALL MOVE (IGCB,5.,65.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 0.
```

```
0119          CALL MOVE (IGCB,10.,64.)
0120          CALL DRAW (IGCB,0.,64.)
0121    C ---------------------
0122          CALL MOVE (IGCB,5.,62.)
0123          CALL LABEL (IGCB)
0124          WRITE (LU,91) 1.
0125          CALL MOVE (IGCB,1.,55.)
0126          CALL LABEL (IGCB)
0127          WRITE (LU,20)
0128    20    FORMAT ( "CORRELATION" )
0129          CALL MOVE (IGCB,1.,53.)
0130          CALL LABEL (IGCB)
0131          WRITE (LU,10)
0132    10    FORMAT ( "COEFFICIENTS")
0133          CALL MOVE (IGCB,5.,47.)
0134          CALL LABEL (IGCB)
0135          WRITE (LU,91) -1.
0136          CALL MOVE (IGCB,10.,46.)
0137          CALL DRAW (IGCB,0.,46.)
0138    C ---------------------
0139          CALL MOVE (IGCB,5.,44.)
0140          CALL LABEL (IGCB)
0141          WRITE (LU,91) 1.
0142          CALL MOVE (IGCB,1.,37.)
0143          CALL LABEL (IGCB)
0144          WRITE (LU,20)
0145          CALL MOVE (IGCB,1.,35.)
0146          CALL LABEL (IGCB)
0147          WRITE (LU,10)
0148          CALL MOVE (IGCB,5.,29.)
0149          CALL LABEL (IGCB)
0150          WRITE (LU,91) -1.
0151          CALL MOVE (IGCB,10.,28.)
0152          CALL DRAW (IGCB,0.,28.)
0153    C ---------------------
0154          CALL MOVE (IGCB,5.,26.)
0155          CALL LABEL (IGCB)
0156          WRITE (LU,91) 20.
0157          CALL MOVE (IGCB,1.,19.)
0158          CALL LABEL (IGCB)
0159          WRITE (LU,12)
0160    12    FORMAT ( "GROUND-")
0161          CALL MOVE (IGCB,1.,17.)
0162          CALL LABEL (IGCB)
0163          WRITE (LU,13)
0164    13    FORMAT ( "TRUTH")
0165          CALL MOVE (IGCB,5.,11.)
0166          CALL LABEL (IGCB)
0167          WRITE (LU,91) -20.
0168          CALL MOVE (IGCB,0.,10.)
0169          CALL DRAW (IGCB,10.,10.)
0170    C
0171    C ********** TITLES **********
0172    C
0173          CALL VIEWP (IGCB,0.,110.,0.,100.)
0174          CALL WINDW (IGCB,0.,110.,0.,100.)
0175          CALL CSIZE (IGCB,2.,1.,0.)
0176          CALL MOVE (IGCB,1.,7.)
0177          CALL LABEL (IGCB)
0178          WRITE (LU,14) TSTART
```

```
14      FORMAT ( " TSTART = ",F5.2)
        CALL MOVE (IGCB,90.,7.)
        CALL LABEL (IGCB)
        WRITE (LU,15) TEND
15      FORMAT ( " TEND = ",F5.2)
C ---------------------
        CALL MOVE (IGCB,30.,4.)
        CALL LABEL (IGCB)
        WRITE (LU,16)
16      FORMAT ( "GROUND TRUTH COMPUTATIONS")
        CALL MOVE (IGCB,40.,7.5)
        CALL LABEL (IGCB)
        WRITE (LU,17)
17      FORMAT ( "TIME IN HOURS")
        CALL CSIZE (IGCB,1.,1.,0.)
        CALL MOVE (IGCB,80.,1.)
        CALL LABEL (IGCB)
        WRITE (LU,18) IDATE
18      FORMAT ( "PROGRAM PLCOR --> DATE : ",A2,"-",A2,"-",A2)
        CALL MOVE (IGCB,100.,10.)
        CALL DRAW (IGCB,110.,10.)
C ...  ..................................................................
        CLOSE (2)
        CALL PLOTR (IGCB,ID,0)
C
C ********** FORMATS **********
C
91      FORMAT (F5.0)
92      FORMAT (3A2)
93      FORMAT (I2)
94      FORMAT (F8.6)
95      FORMAT (1X,3F12.6)
        STOP
        END
```

```
&PLSIG T=00004 IS ON CR20011 USING 00018 BLKS R=0000

FTN4X,L,Y
#FILES (1,1)
      PROGRAM PLSIG
C ***************************************************************
C *                FILENAME :   &PLSIG                          *
C *                ******************                           *
C * PROGRAM PLOTS DATA FILES GENERATED BY THE PROGRAM "EB00"    *
C * THE FILES CONTAIN VALUES IN THE FOLLOWING ORDER :           *
C *     TIME IN HOURS AND MINUTES                               *
C *     UPPER-TRIANGULAR SIGMA-MATRIX (X1)                      *
C *     SQUARE-ROOTS OF DIAGONAL ELEMENTS (X2)                  *
C *     CORRELATION-COEFFICIENT (X3)                            *
C *     GROUND-TRUTH (X4)                                       *
C *                                                             *
C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
C * PLOTTED VALUES ARE :                                        *
C *    ELEMENTS OF SIGMA-MATRIX                                 *
C *    SQUARE-ROOTS                                             *
C *    GROUND TRUTH                                             *
C *                                                             *
C * - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
C *   USED SUBROUTINES : DLTBL,GPS --> PLOTLIBRARY              *
C *                      INPUT,PLSUB                            *
C *                                                             *
C ***************************************************************
C
      DIMENSION IGCB(192),INAM(3),IDATE(3)
C
C ********** READ NAME OF FILE,LOGICAL UNIT AND ID OF **********
C                        DEVICE SUB
C
      WRITE (16,1)
1     FORMAT (" NAME OF PLOTFILE :_")
      READ (16,92) INAM
      OPEN (2,FILE=INAM,FORM='FORMATTED')
      WRITE (16,3)
3     FORMAT ( " LOGICAL UNIT (LU) AND ID OF DEVICE-SUBROUTINE"
     1/          "   LU =_")
      READ (16,93) LU
      WRITE (16,4)
4     FORMAT ( "   ID =_")
      READ (16,93) ID
C
C ********** INITIALIZE PLOTTER, GET LIMITS OF X-AXIS **********
C             AND LABELS AND DRAW VERTICAL GRID
C
      WRITE (16,5)
5     FORMAT ( " START-AND ENDTIME IN HOURS."
     1/          " TSTART :_")
      READ (16,94) TSTART
      WRITE (16,6)
6     FORMAT ( " TEND   :_")
      READ (16,94) TEND
      WRITE (16,19)
19    FORMAT ( " DATE (MMDDYY) :_")
      READ (16,92) IDATE
      CALL PLOTR (IGCB,ID,1,LU)
      CALL SETAR (IGCB)
```

```
      CALL FRAME (IGCB)
C ..................................................................
C                        PLOTTING OF DATA
C
C ********** GROUND TRUTH **********
C
      CALL PLSUB (IGCB,10.,28.,-10.,10.,10.,4,TSTART,TEND,21)
C
C ********** SQUARE-ROOT OF DIAGONAL ELEMENTS **********
C
      CALL PLSUB (IGCB,28.,46.,0.,20.,10.,4,TSTART,TEND,11)
C
C ********** SIGMA MATRIX --> 3 PLOTS **********
C
      CALL PLSUB (IGCB,46.,64.,-100.,100.,100.,4,TSTART,TEND,1)
      CALL PLSUB (IGCB,64.,82.,-100.,100.,100.,3,TSTART,TEND,5)
      CALL PLSUB (IGCB,82.,100.,-100.,100.,100.,3,TSTART,TEND,8)
C
C ********** LABELS **********
C
      CALL VIEWP (IGCB,0.,100.,0.,100.)
      CALL WINDW (IGCB,0.,100.,0.,100.)
      CALL CSIZE (IGCB,1.,1.,0.)
C -----------------
      CALL MOVE  (IGCB,5.,98.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 100.
      CALL MOVE (IGCB,5.,83.)
      CALL LABEL (IGCB)
      WRITE (LU,91) -100.
      CALL MOVE (IGCB,1.,91.)
      CALL LABEL (IGCB)
      WRITE (LU,7)
7     FORMAT ( "SIGMA")
      CALL MOVE (IGCB,1.,89.)
      CALL LABEL (IGCB)
      WRITE (LU,8)
8     FORMAT ( "LINE 3+4")
      CALL MOVE (IGCB,10.,82.)
      CALL LINE (IGCB,0)
      CALL DRAW (IGCB,0.,82.)
C ---------------------
      CALL MOVE (IGCB,5.,80.)
      CALL LABEL (IGCB)
      WRITE (LU,91) 100.
      CALL MOVE (IGCB,1.,73.)
      CALL LABEL (IGCB)
      WRITE (LU,7)
      CALL MOVE (IGCB,1.,71.)
      CALL LABEL (IGCB)
      WRITE (LU,9)
9     FORMAT ( "LINE 2")
      CALL MOVE (IGCB,5.,65.)
      CALL LABEL (IGCB)
      WRITE (LU,91) -100.
      CALL MOVE (IGCB,10.,64.)
      CALL DRAW (IGCB,0.,64.)
C ---------------------
      CALL MOVE (IGCB,5.,62.)
      CALL LABEL (IGCB)
```

```
0119          WRITE (LU,91) 100.
0120          CALL MOVE (IGCB,1.,55.)
0121          CALL LABEL (IGCB)
0122          WRITE (LU,7)
0123          CALL MOVE (IGCB,1.,53.)
0124          CALL LABEL (IGCB)
0125          WRITE (LU,10)
0126    10    FORMAT ( "LINE 1")
0127          CALL MOVE (IGCB,5.,47.)
0128          CALL LABEL (IGCB)
0129          WRITE (LU,91) -100.
0130          CALL MOVE (IGCB,10.,46.)
0131          CALL DRAW (IGCB,0.,46.)
0132    C ---------------------
0133          CALL MOVE (IGCB,5.,44.)
0134          CALL LABEL (IGCB)
0135          WRITE (LU,91) 20.
0136          CALL MOVE (IGCB,1.,37.)
0137          CALL LABEL (IGCB)
0138          WRITE (LU,11)
0139    11    FORMAT ( "SQUARE")
0140          CALL MOVE (IGCB,1.,35.,)
0141          CALL LABEL (IGCB)
0142          WRITE (LU,20)
0143    20    FORMAT ( "ROOTS" )
0144          CALL MOVE (IGCB,5.,29.)
0145          CALL LABEL (IGCB)
0146          WRITE (LU,91) 0.
0147          CALL MOVE (IGCB,10.,28.)
0148          CALL DRAW (IGCB,0.,28.)
0149    C ---------------------
0150          CALL MOVE (IGCB,5.,26.)
0151          CALL LABEL (IGCB)
0152          WRITE (LU,91) 10.
0153          CALL MOVE (IGCB,1.,19.)
0154          CALL LABEL (IGCB)
0155          WRITE (LU,12)
0156    12    FORMAT ( "GROUND-")
0157          CALL MOVE (IGCB,1.,17.)
0158          CALL LABEL (IGCB)
0159          WRITE (LU,13)
0160    13    FORMAT ( "TRUTH")
0161          CALL MOVE (IGCB,5.,11.)
0162          CALL LABEL (IGCB)
0163          WRITE (LU,91) -10.
0164          CALL MOVE (IGCB,10.,10.)
0165          CALL DRAW (IGCB,0.,10.)
0166    C ---------------------
0167          CALL CSIZE (IGCB,2.,1.,0.)
0168          CALL MOVE (IGCB,1.,7.)
0169          CALL LABEL (IGCB)
0170          WRITE (LU,14) TSTART
0171    14    FORMAT ( " TSTART = ",F5.2)
0172          CALL MOVE (IGCB,1.,2.)
0173          CALL LABEL (IGCB)
0174          WRITE (LU,15) TEND
0175    15    FORMAT ( " TEND   = ",F5.2)
0176    C ---------------------
0177          CALL MOVE (IGCB,30.,4.)
0178          CALL LABEL (IGCB)
```

```
0179          WRITE (LU,16)
0180   16     FORMAT ( "GROUND TRUTH COMPUTATIONS --> PLOT-VERSION 1")
0181          CALL MOVE (IGCB,40.,7.5)
0182          CALL LABEL (IGCB)
0183          WRITE (LU,17)
0184   17     FORMAT ( "TIME IN HOURS")
0185          CALL MOVE (IGCB,75.,1.)
0186          CALL LABEL (IGCB)
0187          WRITE (LU,18) IDATE
0188   18     FORMAT ( " DATE : ",A2,"-",A2,"-",A2)
0189   C ..................................................................
0190          CLOSE (2)
0191          CALL PLOTR (IGCB,ID,0)
0192   C
0193   C ********** FORMATS **********
0194   C
0195   91     FORMAT (F5.0)
0196   92     FORMAT (3A2)
0197   93     FORMAT (I2)
0198   94     FORMAT (F8.6)
0199   95     FORMAT (1X,3F12.6)
0200          STOP
0201          END
```

&PLAEL T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
FTN4X,L,Y
$FILES (1,1)
      PROGRAM PLAEL
C ***************************************************
C *            FILENAME  :  &PLAEL                  *
C *            ^^^^^^^^^^^^^^^^^^                   *
C *  PROGRAM TO PLOT AZIMUTH AND ELEVATION FROM     *
C * A DATAFILE GENERATED BY THE PROGRAM EB00.       *
C ***************************************************
C
      DIMENSION IGCB(192),INAM(3),X(2),XO(4)
      DOUBLE PRECISION DX(2)
      LOGICAL LOG
C
C ********** GET INFORMATION TO INITIALIZE PLOTTER **********
C            AND LIMITS OF TIME-AXIS AND DRAW GRID
C
      WRITE (16,1)
1     FORMAT ( " LOGICAL UNIT OF PLOT AND ID OF DEVICE-SUBROUTINE"
     1/         " LU,ID :_")
      READ (16,*) LU,ID
      WRITE (16,2)
2     FORMAT ( " LIMITS OF TIME-AXIS "
     1/         " TSTART,TEND :_")
      READ (16,*) TSTART,TEND
      WRITE (16,3)
3     FORMAT ( " FILENAME :_")
      READ (16,90) INAM
      OPEN (2,FILE=INAM,FORM='FORMATTED')
      WRITE (16,4)
4     FORMAT ( " NUMBER OF SATELLITES IN FILE :_")
      READ (16,*) N
      LOG=.TRUE.
      CALL PLOTR (IGCB,ID,1,LU)
      CALL SETAR (IGCB,1.3)
      CALL FRAME (IGCB)
      CALL VIEWP (IGCB,20.,130.,20.,60.)
      CALL WINDW (IGCB,TSTART,TEND,0.,90.)
      CALL GRID (IGCB,0.25,30.,TSTART,0.,2.,1.)
C
C ********** PLOT DATA **********
C
      J=1
1000  CONTINUE
      READ (2,*,END=2000) T
      IT=IFIX(T)
      T1=T-IT
      T=IT+T1/0.6
      DO 100 I=1,N
      READ (2,*) M,DX
      X(1)=SNGL(DX(1))
      X(2)=SNGL(DX(2))
      IF(LOG) GOTO 1001
      CALL MOVE (IGCB,TO,XO(I))
      CALL LINE (IGCB,I-1)
      CALL DRAW (IGCB,T,X(J))
1001  CONTINUE
      XO(I)=X(J)
```

```
0059  100    CONTINUE
0060         LOG=.FALSE.
0061         TO=T
0062         GOTO 1000
0063  2000   CONTINUE
0064         IF(J.EQ.2) GOTO 2001
0065         REWIND (2)
0066         LOG=.TRUE.
0067         CALL VIEWP (IGCB,20.,130.,60.,100.)
0068         CALL WINDW (IGCB,TSTART,TEND,0.,360.)
0069         CALL GRID (IGCB,0.25,30.,TSTART,0.,2.,2.)
0070         J=2
0071         GOTO 1000
0072  2001   CONTINUE
0073         CLOSE (2)
0074  C
0075  C ********** PLOT TITLES AND LABELS **********
0076  C
0077         CALL LINE (IGCB,0)
0078         CALL VIEWP (IGCB,0.,130.,0.,100.)
0079         CALL WINDW (IGCB,0.,130.,0.,100.)
0080         CALL MOVE (IGCB,13.,97.)
0081         CALL CSIZE (IGCB,2.,1.,0.)
0082         CALL LABEL (IGCB)
0083         WRITE (LU,91) 360.
0084         CALL MOVE (IGCB,1.,81.)
0085         CALL LABEL (IGCB)
0086         WRITE (LU,5)
0087  5      FORMAT ("AZIMUTH")
0088         CALL MOVE (IGBC,1.,78.)
0089         CALL LABEL (IGCB)
0090         WRITE (LU,6)
0091  6      FORMAT ("IN DEGREES")
0092         CALL MOVE (IGCB,13.,61.)
0093         CALL LABEL (IGCB)
0094         WRITE (LU,91) 0.
0095         CALL MOVE (IGCB,0.,60.)
0096         CALL DRAW (IGCB,20.,60.)
0097         CALL MOVE (IGCB,13.,57.)
0098         CALL LABEL (IGCB)
0099         WRITE (LU,91) 90.
0100         CALL MOVE (IGCB,1.,41.)
0101         CALL LABEL (IGCB)
0102         WRITE (LU,7)
0103  7      FORMAT ("ELEVATION")
0104         CALL MOVE (IGCB,1.,38.)
0105         CALL LABEL (IGCB)
0106         WRITE (LU,6)
0107         CALL MOVE (IGCB,13.,21.)
0108         CALL LABEL (IGCB)
0109         WRITE (LU,91) 0.
0110  C ----------------
0111         CALL MOVE (IGCB,0.,20.)
0112         CALL DRAW (IGCB,20.,20.)
0113         CALL LINE (IGCB,1)
0114         CALL MOVE (IGCB,20.,24.444)
0115         CALL DRAW (IGCB,130.,24.444)
0116         CALL MOVE (IGCB,16.,17.)
0117         CALL LABEL (IGCB)
0118         WRITE (LU,92) TSTART
```

```
0119          CALL MOVE (IGCB,54.,12.)
0120          CALL LABEL (IGCB)
0121          WRITE (LU,8)
0122    8     FORMAT ( "GMT-TIME IN HOURS")
0123          CALL MOVE (IGCB,54.,5.)
0124          CALL LABEL (IGCB)
0125          WRITE (LU,9)
0126    9     FORMAT ( "PLOTPROGRAM PLAEL")
0127          CALL MOVE (IGCB,120.,17.)
0128          CALL LABEL (IGCB)
0129          WRITE (LU,92) TEND
0130          CALL MOVE (IGCB,0.,0.)
0131          CALL PLOTR (IGCB,ID,0)
0132    C ----------------
0133    90    FORMAT (3A2)
0134    91    FORMAT (F4.0)
0135    92    FORMAT (F5.2)
0136          STOP
0137          END
```

```
&PL5FI T=00004 IS ON CR00011 USING 00018 BLKS R=0000

      FTN4X,L,Y
      $FILES (1,1)
            PROGRAM PL5FI
      C ***************************************************
      C *              FILENAME  :  &PL5FI                *
      C *              ^^^^^^^^^^^^^^^^^^                 *
      C *  PROGRAM TO PLOT UP TO 5 FILES CONTAINING       *
      C * TIME-TAGGED TWO-DIMENSIONAL ACCURACIES.         *
      C *  FILES ARE GENERATED BY THE PROGRAM EB00 AND    *
      C * TERMINATED BY AN END-CRITERIA : X=Y=0.          *
      C ***************************************************
      C
            DIMENSION IGCB(192),INAM(3)
            LOGICAL LOG
      C
      C ********** GET INFORMATION TO INITIALIZE PLOTTER **********
      C              AND LIMITS OF TIME-AXIS AND DRAW GRID
      C
            WRITE (16,1)
      1     FORMAT ( " LOGICAL UNIT OF PLOT AND ID OF DEVICE-SUBROUTINE"
           1/          " LU,ID :_")
            READ (16,*) LU,ID
            WRITE (16,2)
      2     FORMAT ( " LIMITS OF TIME-AXIS "
           1/          " TSTART,TEND :_")
            READ (16,*) TSTART,TEND
            CALL PLOTR (IGCB,ID,1,LU)
            CALL SETAR (IGCB,1,3)
            CALL FRAME (IGCB)
            CALL VIEWP (IGCB,20.,112.,20.,80.)
            CALL WINDW (IGCB,TSTART,TEND,0.,100.)
            CALL FXD (IGCB,2)
            CALL LGRID (IGCB,-0.25,-20.,TSTART,0.,2.,2.)
      C
      C ********** PLOT DATA **********
      C
            CALL CSIZE (IGCB,2.,1.,0.)
            WRITE (16,3)
      3     FORMAT ( " NUMBER OF FILES :_")
            READ (16,*) N
            DO 100 I=1,N
            CALL VIEWP (IGCB,20.,112.,20.,80.)
            CALL WINDW (IGCB,TSTART,TEND,0.,100.)
            LOG=.TRUE.
            WRITE (16,4) I
      4     FORMAT ( " FILENAME ",I2," :_")
            READ (16,90) INAM
            OPEN (2,FILE=INAM,FORM='FORMATTED')
            CALL LINE (IGCB,I-1)
      1000  CONTINUE
            READ (2,*) X,Y
            IT=IFIX(X)
            T1=X-IT
            X=IT+T1/0.6
            IF(LOG) CALL MOVE (IGCB,X,Y)
            LOG=.FALSE.
            IF(X.LT.0.01.AND.Y.LT.0.01) GOTO 1001
            CALL DRAW (IGCB,X,Y)
```

```
      GOTO 1000
1001  CONTINUE
      CLOSE (2)
      CALL VIEWP (IGCB,0.,130.,0.,100.)
      CALL WINDW (IGCB,0.,130.,0.,100.)
      CALL MOVE (IGCB,I*20.,90.)
      CALL DRAW (IGCB,I*20.+10.,90.)
      CALL MOVE (IGCB,I*20.+12.,89.)
      CALL LABEL (IGCB)
      WRITE (LU,90) INAM
100   CONTINUE
C
C ********** PLOT TITLES **********
C
      CALL MOVE (IGCB,10.,25.)
      CALL LDIR (IGCB,1.57)
      CALL LABEL (IGCB)
      WRITE (LU,5)
5     FORMAT ( "TWO-DIMENSIONAL ACCURACIES")
      CALL LDIR (IGCB,0.)
      CALL MOVE (IGCB,54.,12.)
      CALL LABEL (IGCB)
      WRITE (LU,6)
6     FORMAT ( "GMT-TIME IN HOURS")
      CALL MOVE (IGCB,54.,5.)
      CALL LABEL (IGCB)
      WRITE (LU,9)
9     FORMAT ( "PLOTPROGRAM PL5FI")
      CALL MOVE (IGCB,0.,0.)
      CALL PLOTR (IGCB,ID,0)
C ----------------
90    FORMAT (3A2)
      STOP
      END
```

&EB02 T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
FTN4X,L,Y
C ***********************************************
C *          FILENAME : &EB02                   *
C *                                             *
C *        PROGRAM COMPUTES THE E-MATRIX        *
C ***********************************************
C
      SUBROUTINE EB02
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      COMMON/ELEV/E(3,3)
      COMMON/POS/LAT,LON,ALT,R
C ------------------
      SN=DSIN(LON)
      CN=DCOS(LON)
      ST=DSIN(LAT)
      CT=DCOS(LAT)
      E(1,1)=-SN
      E(1,2)=CN
      E(1,3)=0.0
      E(2,1)=-CN*ST
      E(2,2)=-SN*ST
      E(2,3)=CT
      E(3,1)=CN*CT
      E(3,2)=SN*CT
      E(3,3)=ST
      RETURN
      END
```

```
&EB03  T=00004 IS ON CR00011 USING 00012 BLKS R=0000

FTN4X,L,Y
C ***************************************************
C *                FILENAME :  &EB03                *
C *                                                 *
C * COMPUTES THE USER-POSITION IN ECEF-COORDINATES *
C * (TU-VECTOR=U-VECTOR (IN MAINPROGRAM) )          *
C *                                                 *
C ***************************************************
C
      SUBROUTINE EB03
      IMPLICIT DOUBLE PRECISION(A-Z)
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE
      COMMON/DTX/TU(4)
      COMMON/POS/LAT,LON,ALT,XY
C ------------------
      SLAT=DSIN(LAT)
      XY=AE/DSQRT(1.E0-E2*SLAT*SLAT)
      TU(3)=(XY*OMF2+ALT)*SLAT
      XY=(XY+ALT)*DCOS(LAT)
      TU(1)=XY*DCOS(LON)
      TU(2)=XY*DSIN(LON)
      RETURN
      END
```

&EB05 T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
FTN4X,L,Y
C ***************************************************************
C *                FILENAME :  &EB05                            *
C *                                                             *
C *      COMPUTES ELEVATION AND AZIMUTH OF SATELLITES AND IN    *
C *   SUB. "EB06" THE POSITION AND IN SUB. "EB07" THE DISTANCE *
C *   USER-SATELLITE.                                           *
C *                                                             *
C ***************************************************************
C
      SUBROUTINE EB05 (I)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      DIMENSION Y(4,3)
      COMMON/ELEV/E(3,3)
      COMMON/SAT/S(4)
      COMMON/AZM/ELV(4),AZM(4)
      COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE
      COMMON/DTX/U(4),D(4,4),RBAR(4)
C ---------------------
      CALL EB06
      CALL EB07(I)
      DO 100 II=1,3
      Y(I,II)=0.D0
      DO 100 III=1,3
      Y(I,II)=Y(I,II)+S(III)*E(II,III)
100   CONTINUE
C
C ********** COMPUTATION OF ELEVATION AND AZIMUTH OF **********
C                      SATELLITES AND OUTPUT
C
      Y(I,3)=Y(I,3)-AE
      SY=DSQRT(Y(I,1)*Y(I,1)+Y(I,2)*Y(I,2))
      ELV(I)=DATAN2(Y(I,3),SY)*RTD
      AZM(I)=DATAN2(Y(I,1),Y(I,2))*RTD
      IF(AZM(I).LT.0.) AZM(I)=AZM(I)+360.0
      IF(ISSW(0).GE.0) WRITE (6,1) (S(J),J=1,3),ELV(I),AZM(I)
      WRITE (30,2) I,ELV(I),AZM(I)
2     FORMAT (I2,2D18.8)
1     FORMAT (5D18.8)
      RETURN
      END
```

```
&EB06  T=00004 IS ON CR00011 USING 00018 BLKS R=0000

      FTN4X,L,Y
      C ***************************************************
      C *                  FILENAME :  &EB06              *
      C *                                                 *
      C * COMPUTES THE ACTUAL POSITION OF THE SATELLITE   *
      C * IN ECEF-COORDINATES ( S-VECTOR = X,Y,Z )        *
      C *                                                 *
      C ***************************************************
      C
            SUBROUTINE EB06
            IMPLICIT DOUBLE PRECISION (A-Z)
            INTEGER I
            COMMON/NAC/TUY,TCY,TKC,DTC(4),DTU,TC,T
            COMMON/CNST/PI,RTD,OMF,AE,E2,OMF2,SGC,ERATE,PI2
            COMMON/SAT/X,Y,Z,MK
            COMMON/DB2/ECC,TOE,INOT,NRATE,SRAD,OMEO,OME,MNOT,R,CINO,
           *SINO,COME,SOME,C
      C ---------------------------
            A=TC-TOE
            TUY=DTU+T
            IF(A.GT.-0.3024D6.AND.A.LT.3024D6) TKC=A
            IF(A.LT.-0.3024D6) TKC=A+0.6048D6
            IF(A.GT.0.3024D6) TKC=A-0.6048D6
            TK=TUY-TCY+TKC
            OGA=OMEO+(NRATE-ERATE)*TK-ERATE*TOE
            MK=MNOT+C*TK
          6 IF(MK.GE.-PI2)GO TO 7
            MK=MK+PI2
            GO TO 6
          7 IF(MK.LE.PI2)GO TO 8
            MK=MK-PI2
            GO TO 7
          8 EK=MK+ECC*DSIN(MK)
            DO 100 I=1,4
            COSE=DCOS(EK)
            SINE=DSIN(EK)
            TK=1.-ECC*COSE
      100   EK=EK-(EK-MK-ECC*SINE)/TK
            COSE=(COSE-ECC)/TK
            SINE=DSQRT(1.-ECC*ECC)*SINE/TK
            EK=COSE*COME-SINE*SOME
            MK=SINE*COME+COSE*SOME
            SINE=2.*MK*EK
            COSE=2.*EK*EK-1.
            RK=TK*R
            X=RK*EK
            Y=RK*MK
            Z=Y*SINO
            EK=CINO*Y
            SINE=DSIN(OGA)
            COSE=DCOS(OGA)
            Y=X*SINE+EK*COSE
            X=X*COSE-EK*SINE
            RETURN
            END
```

```
FTN4X,L,Y
C ***************************************************
C *              FILENAME : &EB07                   *
C *                                                 *
C *     COMPUTES DISTANCES TO SATTELLITE            *
C *                                                 *
C ***************************************************
C
      SUBROUTINE EB07(I)
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      COMMON/SAT/S(4)
      COMMON/DTX/U(4),D(4,4),R(4)
C -----------------------
      R(I)=0.
      DO 100 J=1,3
      D(I,J)=S(J)-U(J)
      R(I)=D(I,J)*D(I,J)+R(I)
100   CONTINUE
      R(I)=DSQRT(R(I))
      D(I,4)=1.0
      RETURN
      END
```

```
&EB08  T=00004 IS ON CR00011 USING 00018 BLKS R=0000

      FTN4X,L,Y
      C ***********************************************
      C *              FILENAME :  &EB08              *
      C *                                             *
      C * COMPUTES THE G-MATRIX AND DOES THE MATRIX-  *
      C * TRANSFORMATION TO DELIVER AS A RESULT THE   *
      C * DILUTION-OF-PRECISION-PARAMETERS FOR A CONFI- *
      C * GURATION WITH 4 SATELLITES OR WITH 3 SATEL- *
      C * LITES WITH ALTITUDE-AIDING.                 *
      C *                                             *
      C ***********************************************
      C
            SUBROUTINE EB08
            IMPLICIT DOUBLE PRECISION (A-H,K-M,O-Z)
            DIMENSION A(4,4),AT(4,4),AAT(4,4),SSIG(4),K(4,3)
            LOGICAL IFLAG
            REAL X1(10),X3(4)
            DIMENSION IL(4),IM(4)
            COMMON/POS/LAT,LON,ALT
            COMMON/DTX/U(4),D(4,4)
            COMMON/CNST/PI,RTD
            COMMON/BLG/G(4,4)
            COMMON/GDOP/GDOP,PDOP,HDOP,INS,IFLAG,V(4,4),IST3
            IN=4
            IFLAG=.FALSE.
      C
      C ********** COMPUTATION OF A(=G) **********
      C
            A1=DSIN(LON)
            A2=DCOS(LON)
            A3=DSIN(LAT)
            A4=DCOS(LAT)
            DO 100 I=1,INS
            A5=1./DSQRT(D(I,1)*D(I,1)+D(I,2)*D(I,2)+D(I,3)*D(I,3))
            A(I,1)=(-D(I,1)*A1+D(I,2)*A2)*A5
            A(I,2)=(-D(I,1)*A3*A2-D(I,2)*A3*A1+D(I,3)*A4)*A5
            A(I,3)=(D(I,1)*A4*A2+D(I,2)*A4*A1+D(I,3)*A3)*A5
            A(I,4)=1.
      100   CONTINUE
            IF(INS.EQ.4) GOTO 1000
            A(4,1)=0.
            A(4,2)=0.
            A(4,3)=-1.
            A(4,4)=0.
            IF(IST3.EQ.1) GOTO 1000
            A(4,3)=0.
            A(4,4)=1.
      1000  CONTINUE
            IF(ISSW(0).GE.0) WRITE (6,8)
      8     FORMAT (/" G-MATRIX ")
            IF(ISSW(0).GE.0) WRITE (6,2) ((A(I,J),J=1,4),I=1,4)
      C
      C ********** MATRIX-TRANSPONATION **********
      C
            DO 101 I=1,4
            DO 101 J=1,4
            AT(I,J)=A(J,I)
      101   CONTINUE
```

```
C
C ********** MULTIPLICATION OF MATRIZES (AT*A)=AAT **********
C
      CALL EB09B (AT,V,AAT,IN)
      CALL EB09B (AAT,A,G,IN)
C
C ********** INVERSION OF AT (=SIGMA-MATRIX) **********
C             AND COMPUTATION OF HDOP AND GDOP
C
      CALL MINVD (G,IN,DET,IL,IM)
      ICOUNT=1
      IF(DET.EQ.0.0D0) GOTO 1010
      IF(ISSW(0).GE.0) WRITE (6,1)
1     FORMAT (/" SIGMA-MATRIX ")
      IF(ISSW(0).GE.0) WRITE (6,2) (((G(I,J)),J=1,4),I=1,4)
2     FORMAT (4D18.8)
      GDOP=DSQRT(G(1,1)+G(2,2)+G(3,3)+G(4,4))
      HDOP=DSQRT(G(1,1)+G(2,2))
      PDOP=DSQRT(G(1,1)+G(2,2)+G(3,3))
C
C ********** COMPUTATION OF SQUARE-ROOTS OF DIAGONAL **********
C              ELEMENTS AND CORRELATION COEFFICIENT
C
      II=0
      DO 105 I=1,4
      SSIG(I)=DSQRT(G(I,I))
      X3(I)=SNGL(SSIG(I))
      DO 105 J=1,4
      IF(I.GT.J) GOTO 105
      II=II+1
      X1(II)=SNGL(G(I,J))
105   CONTINUE
      WRITE (2,9) X1
9     FORMAT (10E12.4)
      WRITE (2,10) X3
10    FORMAT (4E12.4)
      IF(ISSW(0).GE.0) WRITE (6,3) SSIG
3     FORMAT (/" SQUARE-ROOTS OF DIAGONAL ELEMENTS "
     1/,4D18.8)
      DO 106 I=1,4
      JJ=0
      DO 106 J=1,4
      IF(I.EQ.J) GOTO 106
      AT(I,J)=0.
      IF(DABS(G(I,I)).LT.0.D-10.OR.DABS(G(J,J)).LT.0.D-10) GOTO 1001
      AT(I,J)=G(I,J)/DSQRT(G(I,I)*G(J,J))
1001  CONTINUE
      JJ=JJ+1
      K(I,JJ)=AT(I,J)
106   CONTINUE
      IF(ISSW(0).GE.0) WRITE (6,4)
4     FORMAT (/" CORRELATION COEFFICIENTS "
     1/,4X,"12",8X,"13",8X,"14",8X,"21",8X,"23",8X,"24",8X,"31",8X,
     2"32",8X,"34",8X,"41",8X,"42",8X,"43")
      IF(ISSW(0).GE.0) WRITE (6,11) ((K(I,J),J=1,3),I=1,4)
11    FORMAT (1H ,12G10.5)
      WRITE (2,5) K(1,1),K(1,2),K(1,3),K(2,2),K(2,3),K(3,3)
5     FORMAT (1H ,6G10.5)
C
C ********** INVERSION OF A (=G) **********
```

```
C
      DO 102 I=1,4
      DO 102 J=1,4
      G(I,J)=A(I,J)
102   CONTINUE
      CALL MINVD (G,IN,DET,IL,IM)
      ICOUNT=2
      IF(DET.EQ.0.0D0) GOTO 1010
      RETURN
1010  CONTINUE
      IFLAG=.TRUE.
      IF(ICOUNT.EQ.2) GOTO 1011
      WRITE (2,9) X1
      WRITE (2,10) X3
      WRITE (2,5) K(1,1),K(1,2),K(1,3),K(2,2),K(2,3),K(3,3)
      RETURN
1011  CONTINUE
      WRITE (6,7)
6     FORMAT ( " *************************** "
     1/         " *PRODUCT-MATRIX SINGULAR* "
     2/         " *************************** ")
7     FORMAT ( " ******************* "
     1/         " *G-MATRIX SINGULAR* "
     2/         " ******************* ")
      RETURN
      END
```

&EB09B T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
FTN4X,L,Y
C ***********************************************************
C *                FILENAME  : &EB09B                       *
C *     MULTIPLICATES TWO MATRICES :  H*G=A                 *
C ***********************************************************
C
      SUBROUTINE EB09B (H,G,A,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION H(N,N),G(N,N),A(N,N)
C ------------------
      DO 100 I=1,N
      DO 100 J=1,N
      TEMP=0.0D0
      DO 101 K=1,N
      TEMP=TEMP+H(I,K)*G(K,J)
101   CONTINUE
      A(I,J)=TEMP
100   CONTINUE
      RETURN
      END
```

```
&MINVD T=00003 IS ON CR00011 USING 00024 BLKS R=0000

0001  FTN4X,L,Y
0002  C
0003  C     ..................................................................
0004  C
0005  C        SUBROUTINE MINVD          HP-VERSION
0006  C
0007  C        PURPOSE
0008  C           INVERT A MATRIX
0009  C
0010  C        USAGE
0011  C           CALL MINVD(A,N,D,L,M)
0012  C
0013  C        DESCRIPTION OF PARAMETERS
0014  C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
0015  C               RESULTANT INVERSE.
0016  C           N - ORDER OF MATRIX A
0017  C           D - RESULTANT DETERMINANT
0018  C           L - WORK VECTOR OF LENGTH N
0019  C           M - WORK VECTOR OF LENGTH N
0020  C
0021  C        REMARKS
0022  C           MATRIX A MUST BE A GENERAL MATRIX
0023  C
0024  C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
0025  C           NONE
0026  C
0027  C        METHOD
0028  C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
0029  C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
0030  C           THE MATRIX IS SINGULAR.
0031  C
0032  C     ..................................................................
0033  C
0034        SUBROUTINE MINVD(A,N,D,L,M)
0035        IMPLICIT DOUBLE PRECISION (A-H,O-T,V-Z)
0036        IMPLICIT REAL*4 (U)
0037        DIMENSION A(1),L(1),M(1)
0038  C
0039  C        SEARCH FOR LARGEST ELEMENT
0040  C
0041        D=1.0D0
0042        NK=-N
0043        DO 80 K=1,N
0044        NK=NK+N
0045        L(K)=K
0046        M(K)=K
0047        KK=NK+K
0048        BIGA=A(KK)
0049        DO 20 J=K,N
0050        IZ=N*(J-1)
0051        DO 20 I=K,N
0052        IJ=IZ+I
0053     10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
0054     15 BIGA=A(IJ)
0055        L(K)=I
0056        M(K)=J
0057     20 CONTINUE
0058  C
```

```
C          INTERCHANGE ROWS
C
      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
C
C          INTERCHANGE COLUMNS
C
   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
C
C          DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C          CONTAINED IN BIGA)
C
   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
C
C          REDUCE MATRIX
C
      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
C
C          DIVIDE ROW BY PIVOT
C
      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
C
C          PRODUCT OF PIVOTS
```

```
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0D0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  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
```

```
&EB11  J=00004 IS ON CR00011 USING 00004 BLKS R=0000

FTN4X,L,Y
C ***************************************************
C *            FILENAME  :  &EB11                   *
C *                                                 *
C *  MULTIPLICATES A VECTOR (URE) AND A MATRIX (G) *
C *                                                 *
C ***************************************************
C
      SUBROUTINE EB11
      IMPLICIT DOUBLE PRECISION (A-H,K-Z)
      COMMON/URE/URE(4),DX(4)
      COMMON/BLG/G(4,4)
C -------------------
      DO 100 J=1,4
      DX(J)=0.
      DO 101 I=1,4
      DX(J)=DX(J)+G(J,I)*URE(I)
101   CONTINUE
      DX(J)=-DX(J)
100   CONTINUE
      RETURN
      END
```

&EB12 T=00004 IS ON CR00011 USING 00012 BLKS R=0000

```
FTN4X,L,Y
C ***************************************************
C *                FILENAME : &EB12                 *
C *                                                 *
C * ROUTINE TO CONVERT SECONDS TO HOURS MIN SEC.    *
C *                                                 *
C ***************************************************
C
      SUBROUTINE EB12 (TMIN,THR)
      IMPLICIT DOUBLE PRECISION (A-Z)
      REAL THR
      INTEGER*4 HR,MIN,SEC
C ---------------------
      THR=(SNGL(TMIN)/3600.0)+.00005
      HR=INT(THR)
      THR=(THR-HR)*60
      MIN=INT(THR)
      THR=(THR-MIN)*60
      SEC=INT(THR)
      THR=HR+(MIN*.01)+(SEC*.0001)+.00005
      RETURN
      END
```

```
&EB14  T=00004 IS ON CR00011 USING 00024 BLKS R=0000

FTN4,L,Y
C ***************************************************
C *              FILENAME : &EB14                   *
C * READS ALMANAC-DATA FROM DATA-TAPE,CONVERTS IT   *
C * INTO DOUBLE PRECISION AND CREATES ARRAY WITH    *
C * DATA FOR MAINPROGRAM.                           *
C *                                                 *
C ***************************************************
C
      SUBROUTINE EB14
      DOUBLE PRECISION AD,A,B,C,DTC
      COMMON/ALM/AD(14,24),ID1(24)
      COMMON/NAC/A,B,C,DTC
      DIMENSION IALMAC(12,24),ISVPRN(24),ID(24)
      INTEGER ALMAC(24,24),IAR(2),IBUF(310),IDARY(24)
      EQUIVALENCE(IBUF(16),IALMAC)
      REAL OMD,OMO,MO
      INTEGER H
      DATA PI/3.14159265/,ISVPRN/4,7,6,8,5,9,18*0/
C
C  ISTAD IS THE STARTING ADDRESS OF ALMANAC DATA AND IWCNT IS THE NUMBER
C  OF WORDS IN THE ALMANAC DATA.
C
      DATA ISTAD/172700B/,IWCNT/310/,LU8/8/
C
C  CALL EXEC TO READ IN ALMANAC DATA.
C
100   CALL EXEC(1,LU8,IBUF,IWCNT)
      CALL ABREG(IERR,NWRD)
      IF(NWRD .NE. 288) GO TO 100
      DO 40 J=1,24
          DO 50 I=1,12
          I2=2*I
          I1=I2-1
          ITMP=IALMAC(I,J)
          ALMAC(I1,J)=IAND(ITMP,377B)
          ALMAC(I2,J)=ISHFT(ITMP,-8)
50        CONTINUE
40    CONTINUE
      WRITE(16,120)
      READ(16,*)DTC
C
C  GET SATELLITE ID(PRN NUMBER) FROM ALMANAC DATA.
C
      DO 10 J=1,24
          DO 20 I=1,24
          ID(I)=ALMAC(1,I)
C
C  TEST TO SEE IF SATELLITE IS IN PRN NUMBER ORDER.
C
          IF(ID(I).EQ. 0)GO TO 20
          IF(ID(I).EQ. ISVPRN(J))GO TO 30
20        CONTINUE
      GO TO 10
C
C  GET ECENTRICITY OF SATELLITE ORBIT FROM ALMANAC DATA.
C
30    IAR(2)=IOR(ISHFT(ALMAC(2,I),8),ALMAC(3,I))
```

```
0059          IDARY(J)=ID
0060          IAR(1)=0
0061          ID1(J)=ID(I)
0062    C
0063    C  ECENTRICITY IS SCALED BY A FACTOR OF 2**-21.
0064    C
0065          E=DINTF(IAR)*4.7683716E-7
0066    C
0067    C  GET TIME OF APPLICABILITY OF THE ALMANAC PARAMETERS.
0068    C
0069          IAR(2)=ALMAC(4,I)
0070          IAR(1)=0
0071    C
0072    C  TOA(SECONDS) IS SCALED BY A FACTOR OF 2**12.
0073    C
0074          TOA=DINTF(IAR)*4096.0
0075    C
0076    C  GET SATELLITE ORBIT INCLINATION FROM ALMANAC DATA.
0077    C
0078          IAR(2)=IOR(ISHFT(ALMAC(5,I),8),ALMAC(6,I))
0079          IAR(1)=0
0080          IF(IAR(2) .LT. 0) IAR(1)=-1
0081    C
0082    C  DI(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-19 INCREMENTED
0083    C  BY 1/3 AND MULTIPLIED BY PI TO YIELD RADIANS.
0084    C
0085          DI=((DINTF(IAR)*1.9073486E-6)+(1.0/3.0))*PI
0086    C
0087    C  GET DRIFT RATE OF LONGITUDE OF ASCENDING NODE OF THE ORBIT
0088    C  FROM ALMANAC DATA.
0089    C
0090          IAR(1)=0
0091          IAR(2)=IOR(ISHFT(ALMAC(7,I),8),ALMAC(8,I))
0092          IF(IAR(2) .LT. 0) IAR(1)=-1
0093    C
0094    C  OMD (SEMICIRCLES/SEC) IS SCALED BY A FACTOR OF 2**-38 AND
0095    C  MULTIPLIED BY PI TO YIELD RADIANS/SEC.
0096    C
0097          OMD=DINTF(IAR)*PI*3.6379788E-12
0098    C
0099    C  GET HEALTH WORD FROM ALMANAC DATA.
0100    C
0101          H=ALMAC(9,I)
0102    C
0103    C  GET THE SQUARE ROOT OF THE SEMI-MAJOR AXIS OF THE ORBIT
0104    C  FROM ALMANAC DATA.
0105    C
0106          IAR(1)=ALMAC(10,I)
0107          IAR(2)=IOR(ISHFT(ALMAC(11,I),8),ALMAC(12,I))
0108    C
0109    C  RA(METERS) IS SCALED BY A FACTOR OF 2**-11.
0110    C
0111          RA=DINTF(IAR)*4.8828125E-4
0112    C
0113    C  GET THE RIGHT ASCENSION OF THE ORBIT FROM ALMANAC DATA.
0114    C
0115          IAR(2)=IOR(ISHFT(ALMAC(14,I),8),ALMAC(15,I))
0116          IAR(1)=ALMAC(13,I)
0117          IF(IAND(ALMAC(13,I),200B).NE.0)IAR(1)=IOR(IAR(1),177400B)
0118    C
```

```
C  OMO(SEMICIRCLES)IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
C
      OMO=DINTF(IAR)*PI*1.1920929E-7
C
C  GET THE ARGUMENT OF PERIGEE FROM ALMANAC DATA.
C
      IAR(1)=ALMAC(16,I)
      IAR(2)=IOR(ISHFT(ALMAC(17,I),8),ALMAC(18,I))
      IF(IAND(IAR(1),200B).NE.0)IAR(1)=IOR(IAR(1),177400B)
C
C  W(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
C
      W=DINTF(IAR)*PI*1.1920929E-7
C
C  GET MEAN ANOMALY AT TOA FROM ALMANAC DATA.
C
      IAR(1)=ALMAC(19,I)
      IF(IAND(IAR(1),200B).NE.0)IAR(1)=IOR(IAR(1),177400B)
      IAR(2)=IOR(ISHFT(ALMAC(20,I),8),ALMAC(21,I))
C
C  MO(SEMICIRCLES) IS SCALED BY A FACTOR OF 2**-23 AND
C  MULTIPLIED BY PI TO YIELD RADIANS.
C
      MO=DINTF(IAR)*PI*1.1920929E-7
      AD(1,J)=DBLE(E)
      AD(2,J)=DBLE(TOA)
      AD(3,J)=DBLE(DI)
      AD(4,J)=DBLE(OMD)
      AD(5,J)=DBLE(RA)
      AD(6,J)=DBLE(OMO)
      AD(7,J)=DBLE(W)
      AD(8,J)=DBLE(MO)
10    CONTINUE
      RETURN
120   FORMAT(" ENTER COLLECTION DATE FOR ALMANAC DATA(NTH DAY)_")
      END
```

&DINTF T=00004 IS ON CR00013 USING 00003 BLKS R=0000

```
0001  ASMB,L
0002        NAM   DINTF
0003        ENT   DINTF
0004        EXT   .FLTD
0005  DINTF NOP
0006        ISZ   DINTF
0007        LDA   DINTF,I
0008        STA   IBF
0009        ISZ   DINTF
0010        DLD   IBF,I
0011        AND   B377
0012        JSB   .FLTD
0013        JMP   DINTF,I
0014  IBF   BSS   1
0015  B377  OCT   377
0016        END
```

```
&PLSUB T=00004 IS ON CR00011 USING 00018 BLKS R=0000

0001  FTN4X,L,Y
0002  C ***********************************************************
0003  C *                FILENAME :   &PLSUB                      *
0004  C *                ^^^^^^^^^^^^^^^^^^^                      *
0005  C * ROUTINE PLOTS THE DATA FOR MAINPROGRAM PLSIG.            *
0006  C ***********************************************************
0007  C
0008        SUBROUTINE PLSUB (IGCB,V1,V2,W1,W2,G1,N,TSTART,TEND,NSTART)
0009        DIMENSION IGCB(192),X1(10),X2(4),X3(6),X4(4),BUFF(24)
0010        EQUIVALENCE (BUFF(1),X1(1)),(BUFF(11),X2(1)),(BUFF(15),X3(1)),
0011       1(BUFF(21),X4(1))
0012        LOGICAL IFLAG,LOG
0013  C ----------------
0014        LOG=.TRUE.
0015        CALL VIEWP (IGCB,10.,100.,V1,V2)
0016        CALL WINDW (IGCB,TSTART,TEND,W1,W2)
0017        CALL GRID (IGCB,0.25,G1)
0018        DO 100 I=1,N
0019  1000  CONTINUE
0020        CALL INPUT (TIME,X1,X2,X3,X4,IFLAG)
0021        IF(IFLAG) GOTO 1002
0022        IF(LOG) GOTO 1001
0023        CALL MOVE (IGCB,TIME1,X)
0024        CALL DRAW (IGCB,TIME,BUFF(NSTART+I-1))
0025  1001  CONTINUE
0026        LOG=.FALSE.
0027        CALL LINE (IGCB,I-1)
0028        IF(I.EQ.4) CALL LINE (IGCB,6)
0029        TIME1=TIME
0030        X=BUFF(NSTART+I-1)
0031        GOTO 1000
0032  1002  CONTINUE
0033        REWIND (2)
0034        LOG=.TRUE.
0035  100   CONTINUE
0036        RETURN
0037        END
```

&PSUB1 T=00004 IS ON CR00011 USING 00006 BLKS R=0000

```
FTN4X,L,Y
C ***********************************************************
C *                FILENAME :   &PSUB1                      *
C *                ^^^^^^^^^^^^^^^^^^^                      *
C * ROUTINE PLOTS THE DATA FOR MAINPROGRAM PLSIG AND COMPUTES *
C * THE SQUARE-ROOT OF THE SUMS OF THE DIAGONAL SIGMA-MATRIX  *
C * ELEMENTS.                                               *
C ***********************************************************
C
      SUBROUTINE PSUB1 (IGCB,V1,V2,W1,W2,G1,N,TSTART,TEND,NSTART)
      DIMENSION IGCB(192),X1(10),X2(4),X3(6),X4(4),BUFF(24)
      EQUIVALENCE (BUFF(1),X1(1)),(BUFF(11),X2(1)),(BUFF(15),X3(1)),
     1(BUFF(21),X4(1))
      LOGICAL IFLAG,LOG
C ----------------
      LOG=.TRUE.
      CALL VIEWP (IGCB,10.,100.,V1,V2)
      CALL WINDW (IGCB,TSTART,TEND,W1,W2)
      CALL GRID (IGCB,0.25,G1)
      DO 100 I=1,N
 1000 CONTINUE
      CALL INPUT (TIME,X1,X2,X3,X4,IFLAG)
      IF(IFLAG) GOTO 1002
      IF(I.EQ.1) X2(1)=SQRT(X2(1)*X2(1)+X2(2)*X2(2))
      IF(I.EQ.2) X2(2)=SQRT(X2(1)*X2(1)+X2(2)*X2(2)+X2(3)*X2(3))
      IF(I.EQ.3) X2(3)=SQRT(X2(1)*X2(1)+X2(2)*X2(2)+X2(3)*X2(3)+X2(4)*
     1X2(4))
      IF(LOG) GOTO 1001
      CALL MOVE (IGCB,TIME1,X)
      CALL DRAW (IGCB,TIME,BUFF(NSTART+I-1))
 1001 CONTINUE
      LOG=.FALSE.
      CALL LINE (IGCB,I-1)
      IF(I.EQ.4.) CALL LINE (IGCB,6)
      CALL CSIZE (IGCB,1.,1.,0.)
      TIME1=TIME
      X=BUFF(NSTART+I-1)
      GOTO 1000
 1002 CONTINUE
      REWIND (2)
      LOG=.TRUE.
 100  CONTINUE
      RETURN
      END
```

&INPUT T=00004 IS ON CR00011 USING 00002 BLKS R=0000

```
FTN4X,L,Y
C ***********************************************
C *              FILENAME  :  &INPUT              *
C *                                               *
C *    ROUTINE READS DATAFILE ON LU=2 OPENED IN   *
C * MAINPROGRAM "PLSIG" RECORD BY RECORD AND      *
C * CONVERTS THE TIME.IF END IS ENCOUNTERED A FLAG*
C * IS SET.                                       *
C ***********************************************
C
      SUBROUTINE INPUT (TIME,X1,X2,X3,X4,IFLAG)
      DIMENSION X1(10),X2(4),X3(6),X4(4)
      LOGICAL IFLAG
      IFLAG=.FALSE.
      READ (2,1,ERR=1000,END=1000) TIME
C     IF(TIME.EQ.0.E0) GOTO 1000
      READ (2,2,ERR=1000) X1
      READ (2,3,ERR=1000) X2
      READ (2,4,ERR=1000) X3
      READ (2,3,ERR=1000) X4
      IT=IFIX(TIME)
      T1=TIME-IT
      TIME=IT+T1/0.6
      RETURN
1000  CONTINUE
      IFLAG=.TRUE.
      RETURN
C
C ********** FORMATS **********
C
1     FORMAT (G15.8)
2     FORMAT (10E12.4)
3     FORMAT (4E12.4)
4     FORMAT (1H ,6G10.5)
      END
```

APPENDIX D EXAMPLES OF PROGRAM OUTPUTS

```
PROGRAM   * E B 0 0 *
---------------------

USER-POSITION :
  LATITUDE :  39.449555     DEGREE
  LONGITUDE: -74.566777     DEGREE
  ALTITUDE :  46.260000     FEET
ECEF-COORDINATES :  X =    1312420.0      Y =   -4753942.1      Z =      4030988.3

USERTIMES :
  START (TSTART):   36000.000     SEC. = 10.00 HOURS
  END (TFIN)    :   40000.000     SEC. = 11.06 HOURS
  INCR. (TDEL)  :   300.00000     SEC
  DATE (DTU)    :   57.000000     DAYS

THE COMPUTATIONS ARE MADE WITH 4 SATELLITES

   USED ALMANACS          DTC=   57.   TC=   300.
   -------------
SAT. ID :   5                      6                      8                      9
            .2305D-02              .1551D-02              .2207D-02              .8108D-02
            .3318D+06              .3318D+06              .3318D+06              .3318D+06
            .1112D+01              .1107D+01              .1103D+01              .1104D+01
           -.6195D-08             -.6457D-08             -.6343D-08             -.6503D-08
            .5154D+04              .5154D+04              .5154D+04              .5154D+04
            .5694D+00             -.1550D+01              .5653D+00             -.1546D+01
           -.2092D+01              .2049D+01             -.1983D+00              .1352D+01
            .6950D+00             -.1074D+01              .3566D+00             -.1142D+01
            .2656D+08              .2656D+08              .2656D+08              .2656D+08
            .4425D+00              .4472D+00              .4505D+00              .4503D+00
            .8968D+00              .8944D+00              .8928D+00              .8929D+00
           -.4977D+00             -.4606D+00              .9804D+00              .2167D+00
           -.8674D+00              .8876D+00             -.1970D+00              .9762D+00
            .1459D-03              .1459D-03              .1459D-03              .1459D-03

   USED URES
   ---------
SAT. ID :   5                          6                          8                          9
  .3600E+05   .1000E+01      .3600E+05   .2000E+01      .3600E+05   .1000E+02      .3600E+05   .3750E+01
  .3960E+05   .1000E+01      .3960E+05   .2000E+01      .3960E+05   .1000E+02      .3960E+05   .3750E+01
  .4320E+05   .1000E+01      .4320E+05   .2000E+01      .4320E+05   .1000E+02      .4320E+05   .3750E+01
  .4680E+05   .1000E+01      .4680E+05   .2000E+01      .4680E+05   .1000E+02      .4680E+05   .3750E+01
  .5040E+05   .1000E+01      .5040E+05   .2000E+01      .5040E+05   .1000E+02      .5040E+05   .3750E+01

INVERTED V-MATRIX
    .10000000D+01     .00000000D+00     .00000000D+00     .00000000D+00
    .00000000D+00     .10000000D+01     .00000000D+00     .00000000D+00
    .00000000D+00     .00000000D+00     .10000000D+01     .00000000D+00
    .00000000D+00     .00000000D+00     .00000000D+00     .10000000D+01

NAME OF PLOTFILE FOR SIGMA-MATRIX          : PLS
NAME OF PLOTFILE FOR HDOP                  : PLH
NAME OF PLOTFILE FOR AZIMUTH AND ELEVATION : PLA
```

SUMMARY OF COMPUTATIONS

TIME 10.0000 HOURS

SATELLITE POSITIONS

X	Y	Z	ELV	AZM
-.17346533D+08	-.18847444D+08	-.70716165D+07	-.90063132D+00	.22708284D+03
-.44207278D+07	-.15734795D+08	.20870653D+08	.57712265D+02	.31078137D+03
.45499720D+07	-.13309344D+08	.22521572D+08	.65611427D+02	.50217224D+01
-.13309804D+08	.17834258D+07	.22665828D+08	.92841590D+01	.32934396D+03

G-MATRIX

-.83972717D+00	-.54279034D+00	-.15392680D-01	.10000000D+01
-.40658235D+00	.34688300D+00	.84519997D+00	.10000000D+01
.40916471D-01	.41157621D+00	.91045641D+00	.10000000D+01
-.50281055D+00	.84916514D+00	.16155533D+00	.10000000D+01

COVARIANCE-MATRIX

.12550939D+02	-.22447338D+01	-.72155331D+01	.93881167D+01
-.22447338D+01	.15376285D+01	.76601251D+00	-.17321497D+01
-.72155331D+01	.76601251D+00	.53931775D+01	-.60872582D+01
.93881167D+01	-.17321497D+01	-.60872582D+01	.76145325D+01

SQUARE-ROOTS OF DIAGONAL ELEMENTS

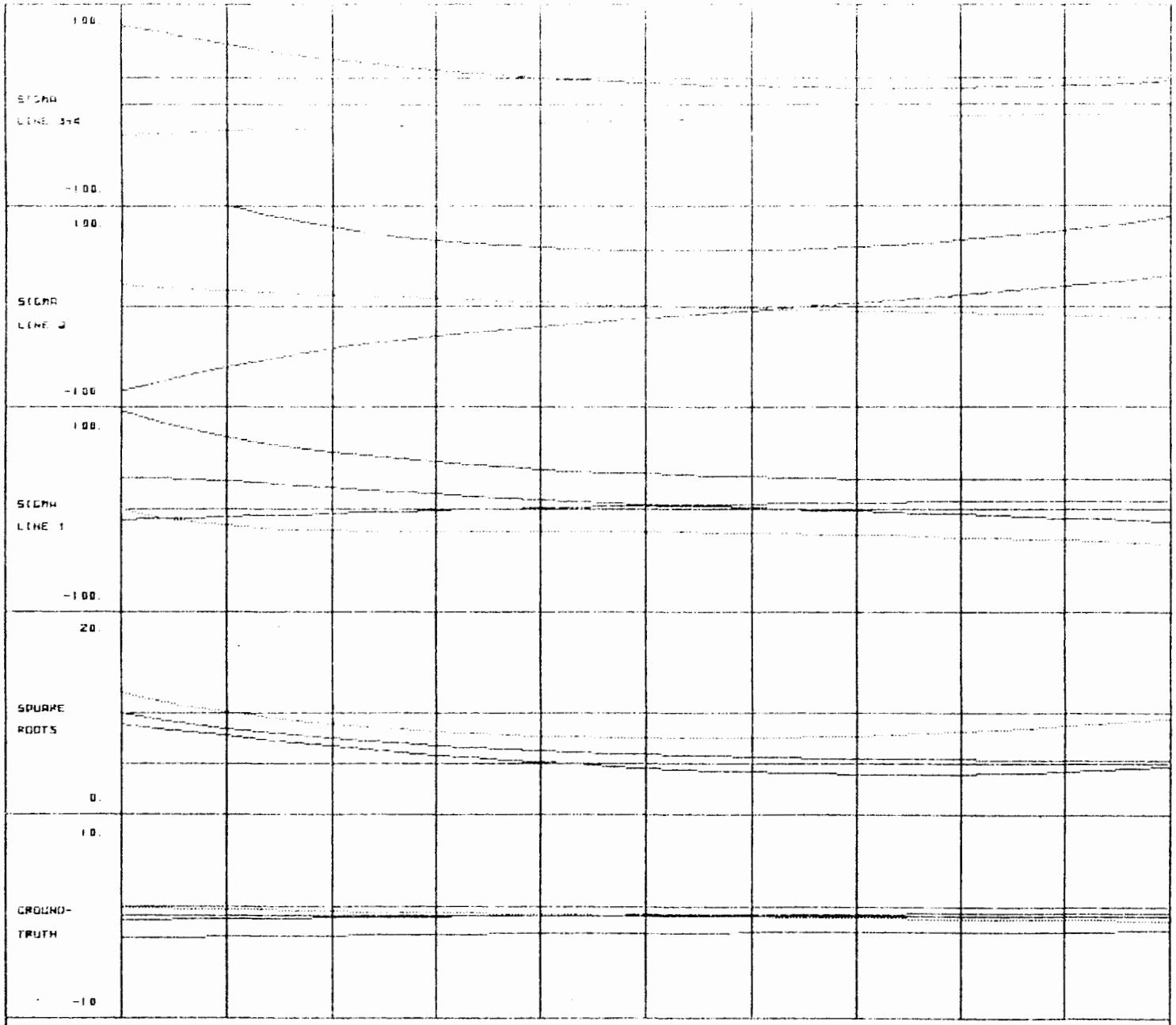
.35427305D+01	.12400115D+01	.24275868D+01	.27594442D+01

CORRELATION COEFFICIENTS

12	13	14	21	23	24	31	32	34	41	42	43
-.5110	-.8390	.96033	-.5110	.25447	-.5062	-.8390	.25447	-.9087	.96033	-.5062	-.9087

GROUND-TRUTH AND FIGURES OF MERIT

DX	DY	DZ	-DB	HDOP	PDOP	GDOP
-.19098208D+02	.18107395D+01	.65784006D+01	-.15953173D+02	3.753	4.470	5.253



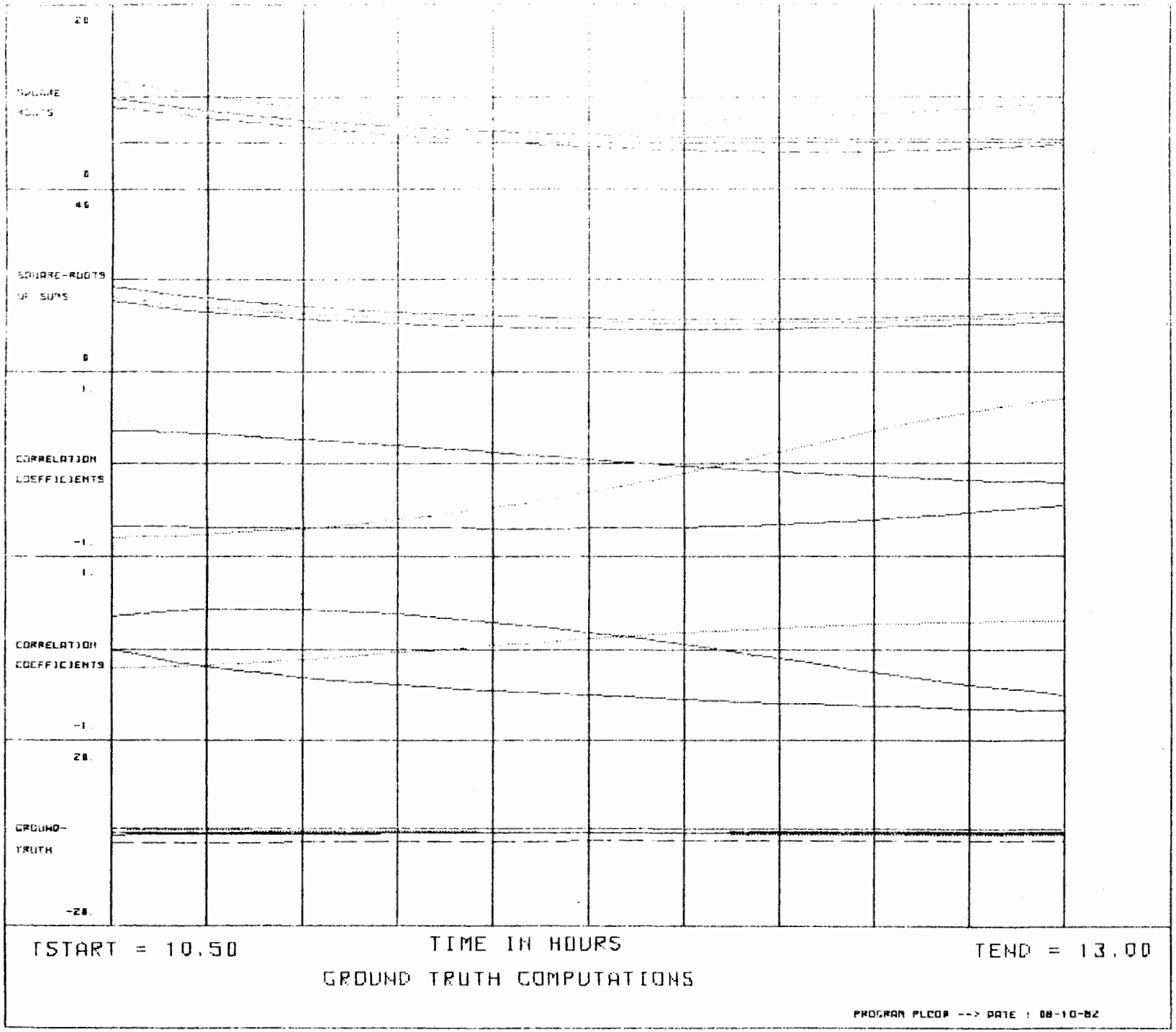
TSTART = 10.50

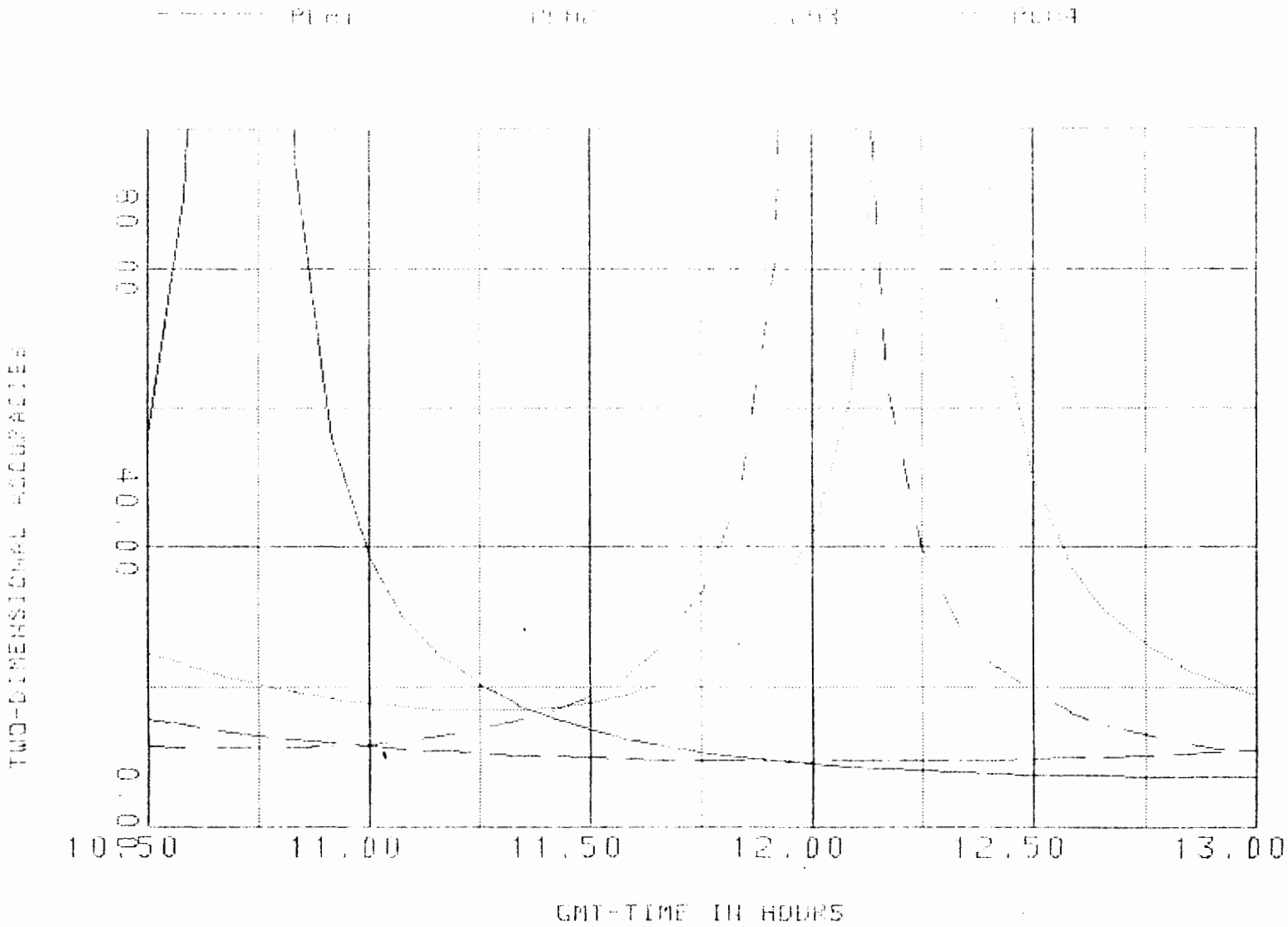
TIME IN HOURS

GROUND TRUTH COMPUTATIONS --> PLOT-VERSION 1

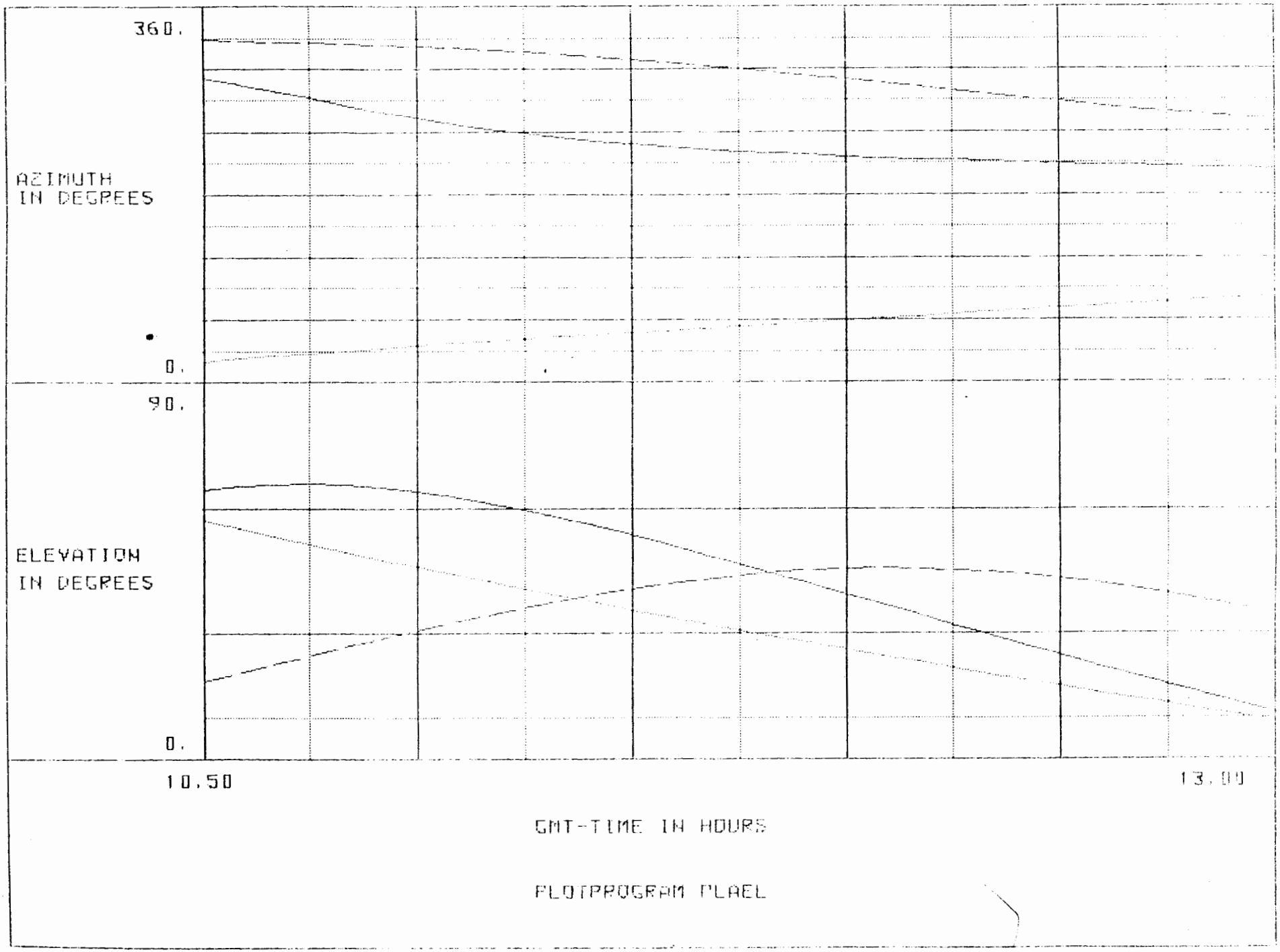
TEND = 13.00

DATE : 08-10-82


20
SQUARE
ROOTS
0
40
SQUARE-ROOTS
OF SUMS
0
1.
CORRELATION
COEFFICIENTS
-1.
1.
CORRELATION
COEFFICIENTS
-1.
20.
GROUND-
TRUTH
-20.
TSTART = 10.50
TIME IN HOURS
TEND = 13.00
GROUND TRUTH COMPUTATIONS
PROGRAM PLCOP --> DATE : 08-10-82



PLOTPROGRAM PL5F1


360.
AZIMUTH
IN DEGREES
0.
90.
ELEVATION
IN DEGREES
0.
10.50
13.00
GMT-TIME IN HOURS
PLOTPROGRAM PLAEL