Reports of the Department of Geodetic Science

Report No. 197

NASA CR-
14175

SELENODETIC COMPUTER PROGRAM LIBRARY

by

F. A. Fajemirokun
F. D. Hotter
H. B. Papo

Prepared for

National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas

Contract Nos. NAS 9-9695 & NAS 9-13093
OSURF Project No. 2841
Interim Report No. 8

The Ohio State University
Research Foundation
Columbus, Ohio 43212

February, 1973

(NASA-CR-141750) SELENODETIC COMPUTER PROGRAM LIBRARY Interim Report (Ohio State Univ. Research Foundation) 185 p
N75-73005
Unclas
00/98 12170

Reports of the Department of Geodetic Science

Report No. 197

SELENODETIC COMPUTER PROGRAM LIBRARY

By

F. A. Fajemirokun

F. D. Hotter

H. B. Papo

Prepared for

National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas

Contract Nos. NAS 9-9695 & NAS 9-13093
OSURF Project No. 2841
Interim Report No. 8

The Ohio State University
Research Foundation
Columbus, Ohio 43212

February, 1973

PREFACE

This project is under the supervision of Ivan I. Mueller, Professor of the Department of Geodetic Science at The Ohio State University, and under the technical direction of Mr. Richard L. Nance, Code TF541, Mapping Sciences Branch, Earth Observation Division, NASA/JSC, Houston, Texas. The contract is administered by the Facility and Laboratory Support Branch, Code BB631/B4, NASA/JSC, Houston, Texas.

TABLE OF CONTENTS

General Index to Documentation

Preceding page blank

INTRODUCTION

The purpose of this report is to document the main programs and subprograms developed and used in The Ohio State University Reports of the Department of Geodetic Science Nos. 156 and 157 [Papo, 1971 and Fajemirokun, 1971]. The reader is assumed to have a basic knowledge of the reports and terminology common to the reports (e.g., coordinate system definition, etc.) so that this documentation may serve as an aid for further research.

Basically, this report is divided into two main sections. The first section deals with subroutines developed for this project. These subroutines constitute the mainstay of the research. The second section describes the main programs used to exercise these subroutines to create the output given in Reports Nos. 156 and 157.

Program/subprogram description is limited to formulation created for the reports. Other documentation mentioned in the references includes:

A. Jet Propulsion Laboratory Documentation which is described in [O'Handley, et al., 1969].

B. Fortran Scientific Subroutine Documentation which is available in IBM publications.

C. Ohio State University Subroutines which are general matrix manipulation routines (addition, subtraction, etc.) whose operations are easily identified by their names (MADD, MSUB, etc.).

Subroutine description consists of the following elements where applicable:

SUBROUTINE
CALL STATEMENT
SUBROUTINE PURPOSE
INPUT PARAMETERS
OUTPUT PARAMETERS
COMMON AREA PARAMETERS
PROGRAM DESCRIPTION
SUBROUTINES REQUIRED
REFERENCES

Program documentation includes the above elements as well as a flowchart of the program logic and samples of card input.

Extensive cross-references are used to eliminate repetition in descriptions of programs/subprograms using identical parameters. To assist in locating referenced programs an index in included in each major section.

Throughout the report floating 8 byte IBM 360 computer words are referred to as "extended precision" and floating/integer 4 byte computer words are referred to as "single precision variables/integers" respectively.

GENERAL REFERENCES :

A. Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

B. O'Handley, Douglas A. et al., (1969). "JPL Development Ephemeris Number 69," Technical Report 32-1456.

C. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

DESCRIPTION OF SUBPROGRAMS
(SUBROUTINES)

SUBROUTINE INDEX

NAME	PURPOSE	PAGE
APPLIB	Calculation of the apparent libration of the moon for an observer on earth.	9
CROSVE	Forms the vector cross product, $A \times B = C$.	11
DMSRAD	Conversion of an angle from degrees, minutes and seconds of arc into radians.	13
DVDPF1	Administers the numerical integration of the simulated ephemeris of the moon through the DVDQ routine.	15
DVDPF2	Slightly modified version of DVDPF1 which allows usage of the output subroutine OUTGAJ.	19
EKHARD	Evaluation of the Eulerian angles of the moon and the physical librations of the moon, including their time rates for a given epoch.	23
EPHITL	Interpolates the 18 tabulated quantities of the simulated ephemeris created by the subprogram FUNEPH and adds the result of the interpolation to the reference case.	27
EVERAL	Interpolation by the use of Everett's fifth order modified method.	31
FUNEPH	Computation of the time derivatives of the "perturbations" of the state vector of the moon, the Eulerian angles and their time rates for the moon and the earth in the simulated earth-moon environment.	35
FUNPL5	Computation of the second time derivatives of the physical libration angles of the real moon, the state transition matrix and the parameter sensitivity matrix used in the numerical integration of the physical librations of the moon. The selenodetic positions of the earth and sun are obtained from the JPL DE-69 tape.	41
FUNPL6	Accomplishes the same purposes as FUNPL5, but the simulated environment is used in lieu of the DE-69 tape.	47

NAME	PURPOSE	PAGE
FUNST3	Evaluates the time derivatives of the perturbations of the 18 elements of the simulated ephemeris (provided by FUNEPH); evaluates the time derivatives of the state vector of a satellite in the simulated earth-moon environment.	49
GASTIM	Computes the Greenwich apparent siderial time at a given epoch, given the nutation in obliquity and longitude for the epoch.	55
GEULAN	Computes the three Eulerian angles (θ, ψ, ω) between the mean ecliptic system of 1950.0 and the average terrestrial system.	57
INVSPE	Copies a layer of a three dimensional input matrix (A) into a two dimensional matrix (B).	59
LADIS	Computes the distance between an earth observatory and a lunar reflector at a given epoch and the zenith distance of the ray at the earth observatory.	61
LASOLV	This is the basic routine for adjusting simulated laser ranging.	65
LUCA	Creates special skew matrices for differentiation of a rotation matrix as outlined by Lucas.	73
MATPA	Forms the product $A'PA$ where A is an IA × JA matrix and P is an IA vector representing a diagonal IA × IA matrix and the product is dimensioned JA × JA.	75
MCROSS	Computes the cross product, R, of two input vectors, X and Y.	77
MEANAN	Calculates elements of the mean orbits of the sun and the moon for a given epoch.	79
NUTATE	Computes the nutation matrix for transformation from the mean to the true celestial Cartesian coordinate system of date.	83
ONEM	Multiplies a matrix A by a vector B, resulting in a vector C.	85
OPTOBS	Generates a bundle of rays simulating an optical observation from a point in space exterior to the lunar surface (e.g., the earth or a satellite) to an array of 30 points.	87

NAME	PURPOSE	PAGE
OUTADJ	Adjusts the numerically integrated physical libration angles to Eckhardt's theory.	93
OUTEPH	Prints routine for output of the intermediate results of the simulated ephemeris.	97
OUTGAJ	Processes simulated earth based optical observation of the moon for solution of lunar coordinates and physical parameters of the moon (initial values of the physical libration parameters and C_{22}, β, C_{20}). This subroutine is called by the DVDPF2 routine at epochs where optical bundles have been observed.	99
PLADIS	Computation of partial derivatives of simulated laser distances with respect to the considered parameters.	107
PMAT	Computation of the transformation matrix necessary to rotate an earth fixed or lunar fixed coordinate system to an inertially oriented system.	113
POLE	Computation of the coordinates of the true pole from the pole of the Conventional International Origin using values from the International Polar Motion Service (IMPS) from 1958 to 1970.5.	115
PRECSS	Calculates the precession matrix P and its time derivative $\dot{P}$. P is an orthogonal matrix that transforms a vector from the mean equatorial system of 1950.0 into the mean equatorial system of date.	119
PREITR	Prepares elements for STVITR subroutine for the Keplerian motion of a satellite about a primary body.	121
PRESTV	Generation of series of constants and a transformation matrix for the calculation of the state vector of a satellite in a Keplerian orbit about a primary body. (The state vector is created by the use of a companion subroutine STVKEP).	123
REDPI	To reduce a given angle θ to the interval $0 < \theta < 2\pi$.	127
ROTATE	Forms a 3 × 3 rotation matrix, RNA, representing a rotation of angle ANG (in radians) about an axis xyz, designated 1, 2 and 3, respectively.	129
SEFODI	Generation of second and fourth modified differences for Everett interpolation of L sets of tabulated quantities.	131

NAME	PURPOSE	PAGE
SKEPTR	Transformation of a satellite state vector into instantaneous Keplerian elements.	133
SPEQU	Copies a two dimensional matrix (B) into Layer L of a three dimensional matrix (A).	137
STRANS	Computation of the rotation matrix S to rotate coordinates from the true celestial system to the average terrestrial system.	139
STVITR	Calculation of a state vector of a satellite in a Keplerian orbit for a given epoch and in components of a fixed Cartesian coordinate system.	141
STVKEP	Calculation of a state vector in a Keplerian orbit. (This subroutine must be used in conjunction with subroutine PRESTV which generates a series of coefficients in common storage used in the state vector computation).	145
TRIK	Used with either FUNPL5 or FUNPL6. Provides the second time derivatives of the physical libration parameters and the Θ, $\bar{\Phi}$ partial derivative matrices.	149
TRIM	Multiplies two 3 × 3 matrices, A and B, to form a 3 × 3 product C.	157

SUBROUTINE : APPLIB

CALL STATEMENT : APPLIB (OBS, EPH, PNT, OMM, OME, AL, JOB).

SUBROUTINE PURPOSE : Calculation of the apparent libration of the moon for an observer on earth.

INPUT PARAMETERS :

A. Vectors (extended precision).
   1. OBS(3). Cartesian coordinates of the observer in the average terrestrial system (in km).
   2. EPH(3). Cartesian coordinates of the moon in the geocentric mean ecliptic system (in km).
   3. PNT(3). Cartesian coordinates of the lunar point in the selenodetic system for which the position angle is evaluated (in km).

B. Matrices (extended precision).
   1. OMM(3, 3). The orthogonal transformation matrix from the selenodetic to ecliptic system.
   2. OME(3, 3). The orthogonal transformation matrix from the average terrestrial system to the ecliptic system.

C. Scalar (integer, single precision).
   1. JOB. An integer set to zero or non zero. If set to zero only the apparent librations in longitude and latitude are computed and the libration angles in radians are returned in the first two components of the AL vector. If set to a non zero integer the subroutine returns (in the three terms of the AL vector) the total apparent libration in longitude, latitude and position angle.

OUTPUT PARAMETERS :

A. Vector (extended precision).
   1. AL(3). A vector containing the apparent librations in longitude, latitude and (depending on the value of the integer JOB) the position angle (in radians).

PROGRAM DESCRIPTION : The statements in APPLIB have been programmed to follow the expressions given in [Papo, 1971], Section 4.5.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. ONEM
2. CROSVE

REFERENCES: Papo, Haim B. (1971). "Optimal Selenodetic Control," Reports of the Department of Geodetic Science, No. 156, The Ohio State University, Columbus.

APPLIB

```
      SUBROUTINE APPLIB (OBS,EPH,PNT,OMM,OME,AL)
C     SUBROUTINE FOR CALCULATING THE APPARENT LIBRATION OF THE MOON
C     OBS - CARTESIAN COORDINATES OF OBSERVER IN AVERAGE TERESTRIAL SYSTEM
C     EPH - CARTESIAN GEOCENTRIC COORDINATES OF THE MOON IN MEAN ECLIPTIC SYSTEM
C     PNT - CARTESIAN SELENOGRAPHIC COORDINATES OF POINT ON THE MOON
C     OBS,EPH,PNT IN KILOMETERS
C     OMM - ORIENTATION MATRIX FROM SELENOGRAPHIC TO ECLIPTIC SYSTEM
C     OME - ORIENTATION MATRIX FROM AVERAGE TERESTRIAL TO ECLIPTIC SYSTEM
C     AL - APPARENT LIBRATION IN RADIANS
C           (1) LONGITUDE   (2) LATITUDE   (3) POSITION ANGLE
      IMPLICIT REAL * 8 (A-H,O-Z)
      DIMENSION OBS(3),EPH(3),OMM(3,3),OME(3,3),AL(3),T1(3),T2(3),T3(3),
     PAV(3),BV(3),CV(3),PNT(3),TMM(3,3)
      CALL ONEM (OME,OBS,T2)
      DO 1 I=1,3
    1 T1(I)=T2 (I)-EPH(I)
      CALL DGMTRA (OMM,TMM,3,3)
      CALL ONEM (TMM,T1,T2)
      R=DSQRT(T2(1)**2+T2(2)**2+T2(3)**2)
      AL(1)=DATAN2(T2(2),T2(1))
      AL(2)=DARSIN(T2(3)/R)
      CALL ONEM (OMM,PNT,T3)
      DO 2 I=1,3
      AV(I)=OME(I,3)
      BV(I)=OMM(I,3)
    2 CV(I)=-T1(I)+T3(I)
      CALL CROSVE (CV,AV,T1)
      CALL CROSVE (CV,BV,T2)
      CALL CROSVE (AV,BV,T3)
      SGN=0.D0
      PA=0.D0
      DO 3 I=1,3
      PA=PA+T1(I)*T2(I)
    3 SGN=SGN+CV(I)*T3(I)
      T=1.D0/DSQRT((T1(1)**2+T1(2)**2+T1(3)**2)*(T2(1)**2+T2(2)**2+T2(3)
     P**2))
      AL(3)  =DARCOS(PA*T)
      IF(SGN.LE.1.D-14)GOTO 9
      AL(3)=-AL(3)
    9 RETURN
      END
```

SUBROUTINE : CROSVE

CALL STATEMENT : CROSVE (A, B, C)

SUBROUTINE PURPOSE : Forms the vector cross product

$$A \times B = C$$

INPUT PARAMETERS :

- A. Vectors (extended precision)
  - 1. A(3)
  - 2. B(3)

OUTPUT PARAMETERS :

- A. Vector (extended precision)
  - 1. C(3)

PROGRAM DESCRIPTION :

A. The diagonal elements of the D matrix are initialized to be zero

$$\underset{i=j}{d_{ij}} = 0$$

B. The D matrix (non diagonal terms) are then formed explicitly:

$$d_{12} = -a_3$$

$$d_{21} = a_3$$

$$d_{13} = a_2$$

$$d_{31} = -a_2$$

$$d_{23} = -a_1$$

$$d_{32} = a_1$$

C. The D matrix is then multiplied by the B vector using the ONEM subroutine and the product is stored in the C vector.

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. ONEM

REFERENCES : None

CROSVE

```
      SUBROUTINECROSVE(A,B,C)
C     VECTOR CROSS PRODUCT  A X B = C
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONA(3),B(3),C(3),D(3,3)
      DATA D(1,1),D(2,2),D(3,3)/3*0.D0/
      D(1,2)=-A(3)
      D(2,1)=A(3)
      D(1,3)=A(2)
      D(3,1)=-A(2)
      D(2,3)=-A(1)
      D(3,2)=A(1)
      CALL ONEM(D,B,C)
      RETURN
      END
```

SUBROUTINE : DMSRAD

CALL STATEMENT : DMSRAD (N, KDP, K, ANG)

SUBROUTINE PURPOSE : Conversion of an angle from degrees, minutes and seconds of arc into radians.

INPUT PARAMETERS :

A. Integers (single precision).

1. N contains the sign and the number of degrees, minutes and whole seconds. For example:

   Angle = $-15^{\circ}\ 3'\ 43.1305''$, then N = -150343

2. KDP indicates the number of significant digits in the seconds fraction. For the above example KDP = 4.
3. K is the fraction part of seconds expressed as a positive integer. For the above example K = 1305.

NOTE : The subroutine is limited to $N \neq 0$.

OUTPUT PARAMETERS :

A. Scalars (extended precision).

1. ANG is the angle in radians.

PROGRAM DESCRIPTION : The input angle is converted to seconds of arc and converted to radians.

SUBROUTINES REQUIRED : None

REFERENCES : None

DMSRAD

```
      SUBROUTINE DMSRAD (N,KDP,K,ANG)
C     CONVERSION FROM DEG/MIN/SEC TO RADIANS
C     IMPORTANT : N SHOULD BE DIFFERENT FROM ZERO
C     N  -  SIGN , DEGREES MINUTES AND SECONDS (INTEGERS)
C     K  -  FRACTION OF A SECOND WITH KDP DIGITS
      IMPLICIT REAL *8 (A-H,O-Z)
      IF(N.LT.0)K=-K
      NDM=N/100
      ND=NDM/100
      NM=NDM-ND*100
      ANG=(DFLOAT(ND*3600+NM*60+N-NDM*100)+DFLOAT(K)/(10.D0**KDP))*.4848
     P136811095360-05
      RETURN
      END
```

SUBROUTINE : DVDPF1

CALL STATEMENT : DVDPF1 (Parameters as in DVDQ, RAT, RAT2, IHL, PAR, FUN, OUT, JOB)

SUBROUTINE PURPOSE : To administer the numerical integration of the simulated ephemeris through the use of the DVDQ numerical integrator and customer supplied subroutines.

INPUT PARAMETERS :

A. Parameters input to the DVDQ routine are described in [Krogh, 1969].

B. Scalars (extended precision).
   1. RAT1 is relative accuracy required in integration for the ephemeris of the moon (e.g. the geocentric state vector).
   2. RAT2 is the relative accuracy required in integration of the Eulerian angles of the earth-moon system (in radians).

C. Integers (single precision).
   1. IHL is a dummy variable.
   2. JOB is a control integer for use by the subroutine. If set to zero, the subroutine will perform integration only. If set to 1, the subroutine will also allow the accumulation of normal equations and a constant vector for a least squares adjustment.

D. Vector (extended precision).
   1. PAR is a vector of parameters input from the main program and used for generating the simulated ephemeris and describ in the documentation of the main program.

E. External subroutine.
   1. FUN is the name of the subroutine which generates time derivatives for a given set of functions (FUNEPH for example).

OUTPUT PARAMETERS :

A. External subroutine.
   1. OUT is the name of the output subroutine for passing control over to DVDPF1 at selected input intervals (the parameter DELT).

B. Common area DVOUT is used only if JOB = 1, and contains the following:
   1. Scalar (extended precision).

    a. ETOB is the <u>next</u> epoch at which an observation has been made and program control is to be passed to OUT.

2. Integers (single precision).
    a. NUT is a flag which is set to zero if the integration is proceeding normally and to one if errors are detected in FUN or OUT. If set to one, the integration is terminated.
    b. ICW is the serial number of the next observation.

PROGRAM DESCRIPTION : The program is the primary "integrator"of the parameters in the simulated environment [Papo, 1971]. The basic integrator is DVDQ and the calculation of time derivatives is provided by FUNEPH. Description of the DVDQ is contained in [Krogh, 1969] and the equations assumed for integration in [Papo, 1971] Chapter 4.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. DVDQ (JPL subroutine)
2. DVDQ1 (JPL subroutine altered for input after initial call of DVDQ)
3. External subroutines given in input-output section.

REFERENCES :

A. Krogh, F. T. (1969). "VODQ/SVDQ/DVDQ--Variable Order Integrators for the Numerical Solution of Ordinary Differential Equations," Section 3.14, JPL, Pasadena, California.

B. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

DVDPF1

```
      SUBROUTINE DVDPF1(NQ    ,Y,YP,KD,EP,RAT1,RAT2  ,HMIN,HMAX,DELT
     P     ,KST,IHL,PAR,FUN,OUT,KQ,YN,DT,JOB)
C     SUBROUTINE FOR INTEGRATING DIFERENTIAL EQUATIONS BY THE DVDQ
C.    NUMERICAL INTEGRATOR . THE NUMBER OF INTEGRANDS SHUULD BE AT
C     LEAST SIX
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 EP(NQ ),HMIN,HMAX,EMX,RAT1,RAT2
      DIMENSION Y( NQ),YP( NQ),YN( NQ),DT(20, NQ),KD( NQ),KQ( NQ),PAR(5)
      COMMON/DVOUT/ETOB,NUT,ICW
      MXSTP=500000
      KD(1)=1
      ET=PAR(1)
      ETFN=PAR(2)
      H=PAR(3)
      NUT=0
      ICW=0
      WRITE(6,891)Y,ET,ETFN,H,DELT
      WRITE(6,892)HMIN,HMAX,RAT1,RAT2,NQ,KD(1),MXSTP
      KLU=-1
      GOTO 881
  879 CALL DVDQ (NQ,ET,Y,YP,KD,EP,IFL,H,HMIN,HMAX,DELT,ETFN,MXSTP,
     PKST,K MX,EMX,KQ,YN,DT)
      GOTO 884
  883 CALL DVDQ1(NQ,ET,Y,YP,KD,EP,IFL,H,HMIN,HMAX,DELT,ETFN,MXSTP,
     PKST,K MX,EMX,KQ,YN,DT)
  884 GOTO (881,882,885,885,885,886,887,888) , IFL
  885 CALL FUN (ET,Y,YP)
      CALL OUT(ET,Y,YP,IHL,NQ,PAR)
      IF(JOB.NE.1) GOTO 883
      DELT=ETOB-ET
      ICW=ICW+1
      IF(NUT.EQ.1) GOTO 888
      GOTO 883
  886 EP(KMX)=32.*EMX*EP(KMX)
      IF (EP(KMX).GT.0.) EP(KMX)=-EP(KMX)
      GOTO 883
  881 DO 871 I=1,6
      EP(I)=Y(I)*RAT1
      IF(ABS(EP(I)).LT.5.E-07) EP(I)=-5.E-07
  871 IF(EP(I).GT.0.) EP(I)=-EP(I)
      IF(NQ-6 ) 874,874,872
  872 DO 873 I=7,12
      EP(I)=Y(I)*RAT2
      EP(I+6)=Y(I+6)*RAT2*10000.
      IF(ABS(EP(I)).LT.1.E-07) EP(I)=-1.E-07
      IF(ABS(EP(I+6)).LT.1.E-09) EP(I+6)=-1.E-09
  873 IF(EP(I).GT.0.) EP(I)=-EP(I)
  874 CONTINUE
      KLU=KLU+1
```

DVDPF1 (Cont)

```
882 CALL FUN (ET,Y,YP)
    IF(KLU) 879,879,883
887 WRITE(6,889) ET,H
    GOTO 890
888 WRITE(6,880) KST,ICW
890 RETURN
880 FORMAT(//5X,' CONGRATULATIONS , YOU DID IT  ',2I7)
889 FORMAT(//5X,' STEPSIZE TOO SMALL , YOU ARE IN TROUBLE ',2D20.12)
891 FORMAT(//(5X,6D19.11))
892 FORMAT(5X,4E19.7/5X,3I19//)
    END
```

SUBROUTINE : DVDPF2

CALL STATEMENT : DVDPF2 (Same parameters as DVDPF1)

DISCUSSION : This subroutine is a slightly modified version of DVDPF1 which allows usage of the output subroutine OUTGAJ. The main differences are:

A. Additional matrices are defined to be used as calling parameters for OUTGAJ.
B. The DVOUT common area is extended to be used in OUTGAJ.
C. The ALL common area is defined for use in OUTGAJ.
D. An integer N is defined and read in to identify the number of non-zero optical rays in the next bundle of rays.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. DVDQ (JPL routine)
2. FUN (representing FUNPL6)
3. OUT (representing OUTGAJ)

REFERENCES:

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

DVDPF2

```
      SUBROUTINE DVDPF2(NQ   ,Y,YP,KD,EP,RAT1,RAT2  ,HMIN,HMAX,DELT
     P      ,KST,IHL,PAR,FUN,OUT,KQ,YN,DT,JOB)
C     SUBROUTINE FOR INTEGRATING DIFERENTIAL EQUATIONS BY THE DVDQ
C     NUMERICAL INTEGRATOR . THE NUMBER OF INTEGRANDS SHOULD BE AT
C     LEAST SIX
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 EP(NQ ),HMIN,HMAX,EMX,RAT1,RAT2
      DIMENSION Y( NQ),YP( NQ),YN( NQ),DT(20, NQ),KD( NQ),KQ( NQ),PAR(5)
     P,EM3(44,44),A1(44,75),A1M3(75,44),A3(44,3),AZ(44,2),N(31),SEV(6)
      COMMON/DVOUT/ETOB,KM,ICW,N ,ISKIP/ALL/SEV,ROM,NUT,IA,IN2,IM,IISK
      READ(5,61)N
      N2=IN2
      KM=IM
C     KM IS THE NUMBER OF BUNDLES TO BE PROCESSED
      ISKIP=IISK
      MXSTP=500000
      KD(1)=1
      ET=PAR(1)
      ETFN=PAR(2)
      H=PAR(3)
      ICW=0
      WRITE(6,891)Y,ET,ETFN,H,DELT
      WRITE(6,892)HMIN,HMAX,RAT1,RAT2,NQ,KD(1),MXSTP
      KLU=-1
      GOTO 881
  879 CALL DVDQ (NQ,ET,Y,YP,KD,EP,IFL,H,HMIN,HMAX,DELT,ETFN,MXSTP,
     PKST,K MX,EMX,KQ,YN,DT)
      GOTO 884
  883 CALL DVDQ1(NQ,ET,Y,YP,KD,EP,IFL,H,HMIN,HMAX,DELT,ETFN,MXSTP,
     PKST,K MX,EMX,KQ,YN,DT)
  884 GOTO (881,882,885,885,885,886,887,888) , IFL
  885 CALL FUN (ET,Y,YP)
      CALL OUT (ET,Y,N2,EM3,A1,A1M3,A3,AZ)
      N2=N(ICW+1)*2
      IF(JOB.NE.1) GOTO 883
      DELT=ETOB-ET
      IF(ICW.EQ.KM) DELT=DELT+.1D0
      ICW=ICW+1
      IF(NUT.EQ.1) GOTO 888
      GOTO 883
  886 EP(KMX)=32.*EMX*EP(KMX)
      IF (EP(KMX).GT.0.) EP(KMX)=-EP(KMX)
      GOTO 883
  881 DO 871 I=1,6
      EP(I)=Y(I)*RAT1
      IF(ABS(EP(I)).LT.1.E-15) EP(I)=-1.E-10
  871 IF(EP(I).GT.0.) EP(I)=-EP(I)
      IF(NQ-6 ) 874,874,872
  872 DO 873 I=7 ,NQ
      EP(I)=Y(I)*RAT2
      IF(ABS(EP(I)).LT.1.E-15) EP(I)=-1.E-10
```

DVDPF2 (Cont)

```
873 IF(EP(I).GT.0.) EP(I)=-EP(I)
874 CONTINUE
    KLU=KLU+1
882 CALL FUN (ET,Y,YP)
    IF(KLU) 879,879,883
887 WRITE(6,889) ET,H
    GOTO 890
888 WRITE(6,880) KST,ICW
890 RETURN
 61 FORMAT(25I3)
880 FORMAT(//5X,' CONGRATULATIONS , YOU DID IT  ',2I7)
889 FORMAT(//5X,' STEPSIZE TOO SMALL , YOU ARE IN TROUBLE ',2D20.12)
891 FORMAT(//(5X,6D19.11))
892 FORMAT(5X,4E19.7/5X,3I19//)
    END
```

SUBROUTINE : EKHARD

CALL STATEMENT : EKHARD (ET, YTH, PHILA)

SUBROUTINE PURPOSE : Evaluation of the Eulerian angles of the moon and the physical librations of the moon including their time rates for a given epoch.

INPUT PARAMETERS :

- A. Scaler (extended precision).
  - 1. ET is the epoch in Julian days.

OUTPUT PARAMETERS :

- A. Vectors (extended precision).
  - 1. YTH is a 6-element vector containing the Eulerian angles of the moon and their time rates in the following order (Papo's notation): ω, ψ, θ, $\dot{\omega}$, $\dot{\psi}$, $\dot{\theta}$ in radians and radians per day.
  - 2. PHILA is a 3-element vector containing the physical libration angles of the moon in the following order (Papo's notation): τ, σ, ρ in radians.

PROGRAM DESCRIPTION : Physical libration theory as developed by Eckhardt [Eckhardt, 1970] is used. The basic purpose of the subroutine is achieved by the following steps.

- A. The subroutine MEANAN is used to compute angles of the mean orbits of the sun and the moon for the epoch ET. Then the sine/cosine combination of the Delauney variables are evaluated.
- B. Eckhardt's coefficients as given in [Eckhardt, 1970] are used to evaluate the physical libration angles and time rates of the angles at epoch ET.
- C. Combinations of the appropriate angles computed above are used to obtain the Eulerian angles of the moon.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. MEANAN

REFERENCES :

A. Eckhardt, D. H. (1970). "Lunar Libration Tables." The Moon, Vol. 1, No. 2, February.

B. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

```
      SUBROUTINE EKHARD (ET,YTH,PHLA)
C     CALCULATION OF EULER ANGLES USING ECKHARDT'S CONSTANTS
      IMPLICITREAL*8(A-H,O-Z)
C     ET - EPOCH IN JULIAN DAYS
C     YTH - EULERIAN ANGLES FOR THE MOON AND THEIR TIME DERIVATIVES
C     PHI , PSI , TETA
C     PHLA - PHYSICAL LIBRATION ANGLES IN LONGITUDE , NODE AND INCLINATION
      DIMENSION PHLA(3),              ANM(6),DNM(6),TAU(9),CTAU(9),DTAU(9
     P),SIG(5),CSIG(5),DSIG(5),ROO(5),CROO(5),DROO(5),YTH(6)
      DATA CTAU,CSIG,CROO/1.7,91.6,-1.4,4.2,-3.5,-16.9,1.0,15.3,10.0,-3.
     P0,-10.6,-23.8,2.5,-100.6,-3.1,-10.8,23.8,-1.9,-98.4/
     P,     RADSE,PI/                          206264.806247096355D0,3.14159
     P26535897930D0/,EIMO/.026769D0/
      CALL MEANAN (ET,ANM,DNM)
C     DELAUNEY ANGLES AND THEIR TIME RATES
      ANMM=ANM(2)-ANM(4)
      DNMM=DNM(2)-DNM(4)
      ANMS=ANM(1)-ANM(3)
      DNMS=DNM(1)-DNM(3)
      ARLA=ANM(2)-ANM(5)
      DRLA=DNM(2)-DNM(5)
      ELON=ANM(2)-ANM(1)
      DLON=DNM(2)-DNM(1)
      SLM=DSIN(ANMM)
      CLM=DCOS(ANMM)
      SLS=DSIN(ANMS)
      CLS=DCOS(ANMS)
      SF=DSIN(ARLA)
      CF=DCOS(ARLA)
      SD=DSIN(ELON)
      CD=DCOS(ELON)
C     SINES AND COSINES OF DELAUNEY ANGLE COMBINATIONS
      S1=2.D0*SF*CF                                                     2F
      C1=CF*CF -SF*SF
      D1=2.D0*DRLA
      S2=SLM*C1-S1*CLM                                                  L-2F
      C2=CLM*C1+SLM*S1
      D2=DNMM-D1
      S3=SLM*C2+S2*CLM                                                  2L-2F
      C3=CLM*C2-SLM*S2
      D3=DNMM+D2
      S4=SLM*CD-SD*CLM                                                  L-D
      C4=CLM*CD+SLM*SD
      D4=DNMM-DLON
      S5=2.D0*S4*C4                                                     2L-2D
      C5=C4*C4-S4*S4
      D5=2.D0*D4
      S6=S5*CLM-SLM*C5                                                  L-2D
      C6=C5*CLM+SLM*S5
      D6=D5-DNMM
      S7=S4*CLS-SLS*C4                                                  L-L'-D
```

```
      C7=C4*CLS+SLS*S4
      D7=D4-DNMS
      S8=S5*C3-S3*C5                                                    2F-2D
      C6=C3*C5+S3*S5
      D8=D5-D3
      S9=S5*CLS-SLS*C5                                                  2L-L'-2D
      C9=C5*CLS+SLS*S5
      D9=D5-DNMS
C     CREATION OF VECTORS FOR PHYSICAL LIBRATION ANGLE AND RATE CALCULATIONS
      TAU(1)=S8
      TAU(2)=SLS
      TAU(3)=S7
      TAU(4)=S6
      TAU(5)=S4
      TAU(6)=SLM
      TAU(7)=S9
      TAU(8)=S3
      TAU(9)=S5
      DTAU(1)=C8 *D8
      DTAU(2)=CLS*DNMS
      DTAU(3)=C7 *D7
      DTAU(4)=C6 *D6
      DTAU(5)=C4 *D4
      DTAU(6)=CLM*DNMM
      DTAU(7)=C9 *D9
      DTAU(8)=C3 *D3
      DTAU(9)=C5 *D5
      SIG(1)=S8
      SIG(2)=S1
      SIG(3)=S2
      SIG(4)=S6
      SIG(5)=SLM
      DSIG(1)=DTAU(1)
      DSIG(2)=C1*D1
      DSIG(3)=C2*D2
      DSIG(4)=DTAU(4)
      DSIG(5)=DTAU(6)
      ROO(1)=C8
      ROO(2)=C1
      ROO(3)=C2
      ROO(4)=C6
      ROO(5)=CLM
      DROO(1)=-S8*D8
      DROO(2)=-S1*D1
      DROO(3)=-S2*D2
      DROO(4)=-S6*D6
      DROO(5)=-SLM*DNMM
C     CALCULATION OF PHYSICAL LIBRATION ANGLES ( TAU SIGMA ROO )
      ATAU=0.D0
      ASIG=0.D0
      AROO=0.D0
```

```
      DATAU=0.D0
      DASIG=0.D0
      DAROO=0.D0
      DO   7I=1,9
      ATAU=ATAU+TAU(I)*CTAU(I)
    7 DATAU=DATAU+DTAU(I)*CTAU(I)
      DO   8I=1,5
      ASIG=ASIG+SIG(I)*CSIG(I)
      DASIG=DASIG+DSIG(I)*CSIG(I)
      AROO=AROO+ROO(I)*CROO(I)
    8 DAROO=DAROO+DROO(I)*CROO(I)
      ASIG=(ASIG/EIMQ      )/RADSE
      DASIG=(DASIG/EIMQ       )/RADSE
      ATAU=ATAU/RADSE
      DATAU=DATAU/RADSE
      AROO=AROO/RADSE
      DAROO=DAROO/RADSE
      PHLA(1)=ATAU*RADSE
      PHLA(2)=ASIG*RADSE
      PHLA(3)=AROO*RADSE
C     CALCULATION OF EULER ANGLES (FI PSI TETA ) AND THEIR TIME DERIVATIVES
      YTH(2)=ANM(5)+ASIG
      YTH(5)=DNM(5)+DASIG
      YTH(1)=PI+ANM(2)+ATAU-YTH(2)
      YTH(4)=DNM(2)+DATAU-YTH(5)
      YTH(3)=EIMQ+AROO
      YTH(6)=DAROO
      DO   9I=1,3
      CALL REDPI (YTH(I))
    9 CONTINUE
      RETURN
      END
```

SUBROUTINE : EPHITL

CALL STATEMENT : EPHITL (ET, N, Q)

SUBROUTINE PURPOSE : The subroutine is designed to interpolate the 18 tabulated quantities of the simulated ephemeris and to add the result of the interpolation to the reference case.

INPUT PARAMETERS :

A. Scalars (extended precision).
   1. ET is the epoch in Julian days minus 2440000.0 for which the ephemerides are required.
B. Integer (single precision).
   1. N is a computer unit identifier. The computer unit is the device on which the tabulated ephemerides are stored. For example, for //FT04F001 DD ..., N = 4.
C. Common Block Input (EPHEM), extended precision.
   1. ETE is a scaler which contains the epoch in Julian days minus 2440000.0 of the first input data.
   2. FH is a 3-dimensional matrix (3 x 18 x 15) which contains ephemeris data (including second and fourth differences) for 14 half-day intervals beginning with ETE. Note that when initially calling EPHITL, EPHEM should contain an arbitrary set of ETE, FH data (normally the first block in the tabulated data.)
D. Data block input (extended precision).
   1. ENCKE and PM are 7- and 6-element vectors respectively which contain constants necessary to perform transformation of the tabulated ephemeris data from perturbations to the reference case into the ephemeris output. AMM is a 3 x 3 orthogonal transformation matrix also used in the transformation. These data block variables are discussed further in the FUNEPH subroutine.

OUTPUT PARAMETERS :

A. Vector (extended precision).
   1. Q is a vector of 18 parameters which are interpolat

to the desired input epoch. The variables contained in Q are as follows:

a. The state vector of the lunar mass center with respect to the geocentric inertially oriented system in km and km/day, (six quantities).
b. The Eulerian angles of the moon and their time rates in radians and radians/day, φ, ψ, θ and $\dot{\varphi}$, $\dot{\psi}$, $\dot{\theta}$, respectively (six quantities).
c. The Eulerian angles of the earth and their time rates in the same units as above, denoted as η, λ, ϵ and $\dot{\eta}$, $\dot{\lambda}$, $\dot{\epsilon}$ [Papo, 1971] Chapter 4.

PROGRAM DESCRIPTION : The data contained on unit N is interpolated for the epoch input using second and fourth differences of the tabulated quantities and is added to the reference cases of the translatory and rotational motio of the moon and the earth.

SUBROUTINES REQUIRED :

O.S.U. Project Library:
1. EVERAL
2. STVITR
3. REDPI

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

EPHITL

```
      SUBROUTINE EPHITL (ET,N,Q).
C     SUBROUTINE FOR INTERPOLATING THE SIMULATED EPHEMERIS
C     ET - EPOCH FOR WHICH THE EPHEMERIS IS REQUIRED
C     N - UNIT ON WHICH THE EPHEMERIS RESIDES
C     Q - THE INTERPOLATED VALUES FOR  (1 - 6) - GEOCENTRIC POSITION
C     AND VELOCITY OF THE MOON  (7 - 12) - EULERIAN ANGLES OF THE MOON .
C     INCLUDING TIME RATES  (13 - 18) - EULERIAN ANGLES OF THE EARTH
C     INCLUDING TIME RATES
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION Y(18),FH(3,18,15),ENCKE(7),AMM(3,3),PM(6),Q(18),STV(6)
      COMMON /EPHEM/ ETE,FH
      DATAENCKE/ 0.2205000000000000D 03     ,     0.3796009972382398D 01         1
     P    , 0.8236118784910790D-01     ,     0.2300000000000000D 00  ,          2
     P      0.4860686866954950D 01     ,     0.4091600000000000D 00  ,          3
     P      0.6300388698000000D 01/,AMM/    -0.1435883157358185D 00  ,          4
     P     -0.9857708968030405D 00     ,    -0.8739642212516620D-01  ,          5
     P      0.9896100062034347D 00     ,    -0.1423649647081624D 00  ,          6
     P     -0.2010603008243764D-01     ,     0.7377750754042759D-02  ,          7
     P     -0.8937536463711470D-01     ,     0.9959706888026594D 00 /,          8
     PPM/   0.9999999999999999D-12     ,     0.2205000000000000D 03  ,          9
     P      0.3847482493894471D 06     ,     0.5489999999999999D-01  ,         10
     P      0.2451334771583748D 01     ,     0.2299715021951530D 00 /          11
       GOTO 27
   20  READ(N) ETE,FH
   27  DIF=ET-ETE
       IF(DIF) 21,22,23
   21  REWIND N
       GOTO 20
   23  IF(DIF.LT.7.D0) GOTO 22
       IST=IDINT(DIF/7.D0)-1
       IF(IST.EQ.0) GOTO 20
       DO 24 I=1,IST
   24  READ(N) ETE
       GOTO 20
   22  LP=IDINT(DIF*2.D0)+1
       SS=2.D0*DMOD(DIF,.5D0)
       CALL EVERAL (SS,LP,FH,Y,18,3,15)
       CALL STVITR (ET,STV,AMM,PM)
       DO 1 I=1,6
       Q(I)=Y(I)+STV(I)
       Q(I+6)=Y(I+6)
    1  Q(I+12)=Y(I+12)
       TT=ET-ENCKE(1)
       Q(7)=Q(7)+ENCKE(2)+ENCKE(4)*TT
       CALL REDPI(Q(7))
       Q(8)=Q(8)+ENCKE(3)
       Q(10)=Q(10)+ENCKE(4)
       Q(13)=Q(13)+ENCKE(5)+ENCKE(7)*TT
       CALL REDPI(Q(13))
       Q(15)=Q(15)+ENCKE(6)
       Q(16)=Q(16)+ENCKE(7)
       RETURN
       END
```

SUBROUTINE : EVERAL

CALL STATEMENT : EVERAL (SS, LP, FDX, STV, N, K1, K2)

SUBROUTINE PURPOSE : Interpolation by the use of Everett's fifth order modified method.

INPUT PARAMETERS :

- A. Matrices (extended precision).
  - 1. FDX(K1, N, K2), where K1, N, K2 subscripts are integers (single precision):
    - a. The first integer, K1, represents the number of quantities needed for the interpolation of each function, i.e., the function value and its second and fourth modified differences (see Program Description). K1 must be set to 3.
    - b. The second integer, N, indicates the number of functions to be interpolated.
    - c. The third integer, K2, indicates the number of layers (see Program Description contained in FDX).
- B. Scalar (extended precision).
  - 1. SS is the interpolation factor (i.e., it must be a decimal fraction between 0.0 and 1.0).
- C. Integer (single precision).
  - 1. LP is an integer denoting the sequential number of the layer (out of the total of K2), preceding the point at which interpolation is to be performed.

OUTPUT PARAMETERS :

- A. Vector (extended precision):
  - 1. STV is an N dimension vector containing the interpolated functions.

PROGRAM DESCRIPTION : Following the notation of Brouwer and Clemence on pages 144 and 145 [Brouwer and Clemence, 1961], the interpolation of value f_n is given by (see text for notations):

$$f_n = f_0 + n\delta_{\frac{1}{2}} + E_0^2\delta_0^2 + E_1^2\delta_1^2 + E_0^4\delta_0^4 + E_1^4\delta_1^4 + \cdots$$

To create this function the following identities are formed:

Computer Notation	Variable
FS(1)	n
FS(2)	E_1^2
FS(3)	E_1^4
FP(1)	$n-1$
FP(2)	$-E_0^2$
FP(4)	$-E_0^4$

The level is then found from LS = LP+1 and

Computer Notation	Variable
FDX(1, N, LS)	f_1
FDX(2, N, LS)	δ_1^2
FDX(3, N, LS)	δ_1^4
FDX(1, N, LP)	f_0
FDX(2, N, LP)	δ_0^4
FDX(3, N, LP)	δ_0^4

The interpolated variable is then (in computer format using variable notation) :

$$\begin{aligned} f_n = \quad & f_1 n - f_0 (n-1) \\ & + E_1^2 \delta_1^2 - (-E_0^2)\,\delta_0^2 \\ & + E_1^4 \delta_1^4 - (-E_0^4)\,\delta_0^4 \end{aligned}$$

which reduces to the Brouwer and Clemence formula.

SUBROUTINES REQUIRED : None

REFERENCES :

A. Brouwer, D. and Clemence, G. (1961). "Methods of Celestial Mechanics," Academic Press, New York.

B. O'Handley, Douglas A. et. al., (1969). "JPL Development Ephemeris Number 69," Technical Report 32-1465.

EVERAL

```
      SUBROUTINE EVERAL (SS,LP,FOX,STV,N,K1,K2)
C     EVERET FIFTH ORDER MODIFIED INTERPOLATION (JPL)
C     FOX - COORDINATES TO BE INTERPOLATED INCLUDING SECOND AND FORTH
C     MODIFIED DIFFERENCES
C       FIRST SUBSCRIPT  1 - F   2 - DEL2  3 - DEL4
C     SECOND SUBSCRIPT QUANTITIES TO BE INTERPOLATED ( N OF THEM )
C       THIRD SUBSCRIPT  SEQUENTIAL DATA SET ( 3 X N LAYER )
C     STV - INTERPOLATED QUANTITIES
C     SS  - INTERPOLATION FACTOR
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION FS(3),FP(3),STV(N),FOX(K1,N,K2)
      LS=LP+1
      SM2=SS-2.D0
      SM1=SS-1.D0
      SP1=SS+1.D0
      SP2=SS+2.D0
      FS(1)=SS
      FS(2)=SM1*SS*SP1/6.D0
      FS(3)=SM2*SM1*SS*SP1*SP2/120.D0
      FP(1)=SM1
      FP(2)=SS*SM1*SM2/6.D0
      FP(3)=SP1*SS*SM1*SM2*(SM2-1.D0)/120.D0
      DO401K=1,N
      STV(K)=0.D0
      DO402I=1,3
  402 STV(K)=STV(K)+FOX(I,K,LS)*FS(I)-FOX(I,K,LP )*FP(I)
  401 CONTINUE
      RETURN
      END
```

SUBROUTINE : FUNEPH

CALL STATEMENT : FUNEPH (ET, Y, YP)

SUBROUTINE PURPOSE : Computation of the time derivatives of the perturbations of the state vector of the moon, the Eulerian angles and their time rates for the moon and the earth in the simulated earth-moon environment described in [Papo, 1971] Chapter 4.

INPUT PARAMETERS :

A. Scalar (extended precision).
  1. ET is the epoch for which the time derivatives are required in Julian days minus 2440000.0.

B. Vector (extended precision).
  1. Y is a vector of 18 parameters which are the "perturbations" to the reference case of the simulated environment as developed in [Papo, 1971]. The order of the parameters is the same as the Q output vector explained in the EPHITL subroutine description.

C. Common block input (extended precision).
  1. MAINFUN is a common area containing:
    a. ENCKE is a vector of 7 elements containing in order (see Papo, Pages 131-135, 154-159):
      (1). The standard or zero epoch of integration in Julian days minus 2440000.
      (2). The mean longitude of the moon.
      (3). The longitude of the node of the lunar orbit.
      (4). The mean motion of the moon.
      (5). The mean longitude of the U axis of the average terrestrial coordinate system.
      (6). The mean inclination of the equatorial plane of the earth with respect to the ecliptic system.
      (7). The mean rotational velocity of the earth about the W axis.

      NOTE: (2) through (7) pertain to epoch ENCKE(1) and are given in radians or in radians per day.

        b. AMM is a 3 × 3 matrix which is an orthogonal transformation matrix from a coordinate system defined by the perigee and the plane of the reference orbit of the moon into the ecliptic system.
        c. PM is a 6-element vector containing Keplerian orbital parameters of the moon. The elements are the precision required in the solution of Kepler's equation, the standard epoch in Julian days (minus 244. 0000. 0), the major semi-axis (in km), the eccentricity (in radians), the mean anomaly at the standard epoch (in radians), and the mean motion (in radians per day).
    2. ALL is a common area containing the vector Q.
        a. Q is an 18-element vector containing the total magnitudes (e.g., reference case plus perturbations) of the ephemeris. The quantities are in the same order and units as listed in the Q output vector described in the EPHITL subroutine.

<u>OUTPUT PARAMETERS</u> :

A. Vector (extended precision).
    1. YP is an 18-element vector containing the time derivatives of the input Y vector. It is based on the equations of motion for the simulated environment developed in [Papo, 1971] Pages 138-150 and 154-159.

<u>PROGRAM DESCRIPTION</u> : This subroutine is essentially the generator of the simulated earth-moon environment as outlined by [Papo, 1971] Pages 119-160.

<u>SUBROUTINES REQUIRED</u> :

A. O.S.U. Project Library:
    1. STVITR
    2. REDPI
    3. ROTATE
    4. TRIM
    5. ONEM
B. Fortran Scientific Subroutine Package:
    1. DGMTRA
    2. DGMADD

<u>REFERENCES</u> :

Papo, Haim B. (1971). "Optimal Selenodetic Control." The Ohio State University, <u>Department of Geodetic Science, Report No. 156</u>

FUNEPH

```
      SUBROUTINE FUNEPH (ET,Y,YP)
C     GENERATOR OF SIMULATED EPHEMERIS (MOON AND EARTH)
C     Y( 1- 6)  MOON  STATE VECTOR GEOCENTRIC ECLIPTIC
C     Y( 7-12)  MOON EULERIAN ANGLES AND TIME RATES
C     Y(13-18) EARTH EULERIAN ANGLES AND TIME RATES
C     POSITION  :  X , Y , Z
C     EULERIAN ANGLES  :  LONGITUDE , NODE , INCLINATION
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION Y(18),YP(18),Q(18),STV(6),ENCKE(7),XYZ(3)    ,X(3),T1(3,3    2*P
     P),T2(3,3),T3(3,3),T4(3,3),TM(3,3),TMT(3,3),TE(3,3),TET(3,3),VX(3),       1
     PVX3(3),XCR(3,3),AMM(3,3),      GS(3,3),QS(3,3),PS(3,3),UVW(3),PM(6)      2
      COMMON/MAIFUN/ENCKE,AMM,PM,NUMB/ALL/Q
      DATA ZERO,ONE     ,CPHL,ALF,BET,GAM,EMU,CEAR,CONM,ES,HS    ,GS          3*P
     P /0.D0,1.D0        ,.892668D16,.4194213D-3,.629D-3,.2095788D-3,.327      1
     P802D-2,35.99241768D10,30121597539975
40.D0       ,66067.85625,536904      2
     P6.636,3578363.863,3*0.D0,3579114.284,3*0.D0,3580615.126/                3
      NUMB=NUMB+1
      CALL STV1TR (ET,STV,AMM,PM)
      DO 1 I=1,6
      Q(I)=Y(I)+STV(1)
      Q(I+6)=Y(I+6)
    1 Q(I+12)=Y(I+12)
      TT=ET-ENCKE(1)
      Q(7)=Q(7)+ENCKE(2)+ENCKE(4)*TT
      CALL REDPI(Q(7))
      Q(8)=Q(8)+ENCKE(3)
      Q(10)=Q(10)+ENCKE(4)
      Q(13)=Q(13)+ENCKE(5)+ENCKE(7)*T
      CALL REDPI(Q(13))
      Q(15)=Q(15)+ENCKE(6)
      Q(16)=Q(16)+ENCKE(7)
C
      XX=ZERO
      XO=ZERO
      DO 2 I=1,3
      X(I)=Q(I)
      XX=XX+X(I)*X(I)
    2 XO=XO+STV(I)*STV(I)
      X7=ONE/XX**3.5D0
      X5=X7*XX
      X3=X5*XX
C
      CALL ROTATE (3,Q(7),T1)
      CALL ROTATE(1,-Q(9),T2)
      CALL ROTATE(3,Q(8),T3)
      CALL TRIM (T1,T2,T4)
      CALL TRIM (T4,T3,TM)
      CALL DGMTRA (TM,TMT,3,3)
      CALL TRIM (TMT,GS,T3)
      CALL TRIM (T3,TM,PS)
      CALL ONEM (TM,X,XYZ)
```

```
C
      OM1=T1(2,1)*T2(3,2)*Q(11)-T1(1,1)*Q(12)
      OM2=T1(1,1)*T2(2,3)*Q(11)+T1(1,2)*Q(12)
      OM3=         T2(3,3)*Q(11)+                Q(10)
      TRE1= ALF*(CPHL*XYZ(2)*XYZ(3)*X5-OM2*OM3)+T1(1,2)*T2(3,3)*Q(11)*Q(
     P12)+T1(1,1)*T2(3,2)*Q(11)*Q(10)+T1(2,1)*Q(12)*Q(10)
      TRE2=-BET*(CPHL*XYZ(1)*XYZ(3)*X5-OM1*OM3)+T1(1,1)*T2(3,3)*Q(11)*Q(
     P12)+T1(2,1)*T2(3,2)*Q(11)*Q(10)-T1(1,1)*Q(12)*Q(10)
      TRE3= GAM*(CPHL*XYZ(1)*XYZ(2)*X5-OM1*OM2)+T2(3,2)*Q(11)*Q(12)
      YP(10)=( T1(1,2)*T2(3,3)*TRE1+T1(1,1)*T2(3,3)*TRE2)/T2(3,2)+TRE3
      YP(11)=(-T1(1,2)         *TRE1-T1(1,1)         *TRE2)/T2(3,2)
      YP(12)= -T1(1,1)         *TRE1+T1(1,2)         *TRE2
C
      CALL ROTATE (3,Q(13),T1)
      CALL ROTATE (1,-Q(15),T2)
      CALL ROTATE (3,Q(14),T3)
      CALL TRIM (T1,T2,T4)
      CALL TRIM (T4,T3,TE)
      CALL DGMTRA (TE,TET,3,3)
      DO 23 I=1,3
      DO 23 J=1,3
   23 QS(I,J)=132135.7125D0*TE(3,I)*TE(3,J)
      CALL ONEM (TE,X,UVW)
C
      HEL=ENCKE(7)*(ONE+EMU)-T2(3,3)*Q(17)
      TWO1=T1(1,2)*T2(3,3)*Q(17)*Q(18)+T1(1,1)*T2(3,2)*Q(17)*HEL+T1(2,1)
     P*Q(18)*HEL+CEAR*UVW(2)*UVW(3)*X5
      TWO2=T1(1,1)*T2(3,3)*Q(17)*Q(18)+T1(2,1)*T2(3,2)*Q(17)*HEL-T1(1,1)
     P*Q(18)*HEL-CEAR*UVW(1)*UVW(3)*X5
      YP(17)=(T1(1,2)*TWO1+T1(1,1)*TWO2)/T2(2,3)
      YP(18)=-T1(1,1)*TWO1+T1(1,2)*TWO2
      YP(16)=T2(3,2)*Q(17)*Q(18)-T2(3,3)*YP(17)
C
      QL=ZERO
      DO 3 I=1,3
    3 QL=QL+Y(I)*(STV(I)+Y(I)*.5D0)
      EFO=ONE-ONE/(ONE+2.D0*QL/X0)**1.5D0
      X0=ONE/X0**1.5D0
      DO 4 I=1,3
    4 VX3(I)=X0*(Y(I)-X(I)*EFO)
C
      DO 5 I=1,3
      DO 5 J=1,3
      XCR(I,J)=-2.5D0*X(I)*X(J)*X7
      IF(I.NE.J) GOTO 5
      XCR(I,J)=XCR(I,J)+XX*X7
    5 CONTINUE
      CALL DGMADD (PS,QS,T1,3,3)
      CALL TRIM (XCR,T1,T2)
      DO 6 I=1,3
```

```
      DO 6 J=1,3
      IF(I.NE.J) GOTO 6
      T2(I,J)=T2(I,J)+X5*(ES+HS)
    6 CONTINUE
      CALL ONEM (T2,X,VX)
      DO 7 I=1,3
      YP(I+3)=-CONM*(VX(I)+VX3(I))
      YP(I   )=Y(I+3 )
      YP(I+6 )=Y(I+9 )
    7 YP(I+12)=Y(I+15)
   99 RETURN
      END
```

SUBROUTINE : FUNPL5

CALL STATEMENT : FUNPL5 (ET, Y, YP)

SUBROUTINE PURPOSE : The subroutine calculates the second time derivatives of the physical libration angles of the real moon and the state transition matrix and the parameter sensitivity matrix used in the numerical integration of the physical librations of the moon as described in [Papo, 1971] Chapter 3.32. The selenodetic positions of the earth and the sun are obtained from the JPL DE-69 tape [O'Handley, 1969].

INPUT PARAMETERS :

- A. Scalars (extended precision). Also note the Common Area Parameters section.
  - 1. ET is the epoch in Julian days for which the derivative are needed.
- B. Vector (extended precision).
  - 1. Y is a 60-element vector containing, in order (supplie by the calling program DVDPF):
    - a. The physical libration angles and time rates $(\tau, \sigma, \rho, \dot{\tau}, \dot{\sigma}, \dot{\rho})$.
    - b. The 36-elements of the state transition matr stored columnwize.
    - c. The 18 elements of the parameter sensitivi matrix stored columnwise.

OUTPUT PARAMETERS :

- A. Vector (extended precision). Also note the Common Area Parameters section.
  - 1. YP is a 60-element vector containing the time derivativ of the input Y vector:

$$YP \equiv \dot{Y}$$

COMMON AREA PARAMETERS : The program extensively uses common area input/output, listed separately in this section. The common areas are MAIF, EXE, ALL, CETBL1, CETBL2, CETBL4.

- A. Area /MAIF/ALF, BET, GAM, TEQ.
  - 1. ALF, BET, GAM are the rations between the moments of inertia of the moon:

$$\alpha = \frac{C-B}{A}, \quad \beta = \frac{C-A}{B}, \quad \gamma = \frac{B-A}{C}$$

[Papo, 1971]. They are extended precision scalars evaluated in the main program and passed through the subroutine calling FUNPL5 (usually DVDPF).

2. TEQ is the mean inclination of the lunar equator with respect to the mean ecliptic of date (I).

B. /EXE/TO, B, SUN, NUMB.

1. TO is an extended precision scalar containing the epoch of the previous step in the integration at which FUNPL5 was called.
2. B is an extended precision 3-element vector containing the geocentric position vector of the moon in kilometers.
3. SUN is a vector similar to B, containing the geocentric position vector of the sun at epoch TO.
4. NUMB is a dummy integer.

C. /ALL/SEV, A, NUT, KLU.

1. SEV is an extended precision 6-element vector containing the mean longitude and longitude of node of the moon's orbit and their first and second time derivatives respectively, evaluated at the epoch ET.
2. A is an extended precision dummy scalar.
3. NUT is a single precision integer which is normally set to zero. The integer is assigned the value one and the integration is terminated if abnormal program operation is detected.
4. KLU is a single precision integer which is used to control the program actions. If set to one, the partial derivatives of the YP elements 7-60 are integrated with the physical libration angles. If set to an integer greater than one, only the physical libration angles (elements 1-6 of the Y vector) are integrated.

D. Common areas /CETBL1/CETBL2/CETBL4/ are required for input from the JPL DE-69 tape [O'Handley et al., 1969].

PROGRAM DESCRIPTION : The FUNPL5 program is designed to find the second time derivatives of the real moon's (e.g. as described by DE-69) motion [Papo, 1971] Chapter 3 . The routine requires the use of subroutine TRIK and the logic used in the program is given below.

A. The geocentric positions of the moon and sun are read in from the DE-69 tape.

B. The position vectors from (A) are transformed into selenocentric positions in the mean ecliptic system of date (ET).

C. The Eulerian angles of the moon and their time derivatives are evaluated from combinations of the Y and SEV vector elements.

D. The TRIK subroutine is called to evaluate the time derivatives of the physical libration angles and to create (in Papo's notation) the θ and Φ matrices need for the evaluation of $\dot{U}$ and $\dot{Q}$ [Papo, 1971] Page 95.

E. The output of the TRIK subroutine is used to form the output YP vector.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
    1. MEANAN
    2. PRECSS
    3. TRIM
    4. ONEM
    5. REDPI
    6. TRIK
    7. ROTATE

B. JPL Programs:
    1. READE

REFERENCES :

A. O'Handley, Douglas A. et al. (1969). "JPL Development Ephemeris Number 69," Technical Report 32-1465.

B. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science Report No. 156.

FUNPL5

```
      SUBROUTINE FUNPL5 (ET,Y,YP)
C     SUBROUTINE FUNPL5 FOR READING THE DE-69 TAPE , CALLING THE TRIK SUBROUTINE
C     AND ADMINISTERING THE FORMATION OF DERIVATIVES OF THE PHYSICAL LIBRATION
C     ANGLES AND THE PARTIAL DERIVATIVES MATRICES (TRANSITION AND SENSITIVITY)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION Y(60),YP(60),SEV(6)       ,Z(6),ZP(3),D(3),B(3),SUN(3),
     POME(3),F(4),TE1(6),TE2(6),IRE(13),S(6,12),T1(3,3),T2(3,3),T3(3,3)
     P,TETA(6,6),FETA(6,3)
C     Y(1-6) - PHYSICAL LIBRATIONS ANGLES
C     Y(7-42)- STATE TRANSITION MATRIX IN COLUMNS (TAU ...RODOT)
C     Y(43-60) - PARAMETERS SENSITIVITY IN COLUMNS (C22,BETA,C20)
      COMMON/MAIF/ALF,BET,GAM,TEQ  /EXE/TO,B,SUN,NUMB/ALL/  SEV,A,NUT,
     PKLU/CETBL1/AU,REA,TPU,EMR/CETBL2/ICW,ICE,IRE/CETBL4/S,F
      DATA PI,TSE /3.14159265358979D0,0.D0/
C     ET - EPOCH IN (JD-2440000.)
      IF(ET.EQ.TO) GOTO 2
      TO=ET
      UJ=ET+2440000.D0
      CALL READE (UJ,TSE,IER)
C     READING FROM THE DE-69 TAPE
      IF(IER.NE.0) GOTO 6
      DO 1 I=1,3
      D(I)=S(I,11)
    1 OME(I)=S(I,10)
C     POSITIONS OF EARTH(D) AND SUN(OME) IN MEAN EQUATORIAL OF 1950
      CALL MEANAN (UJ,TE1,TE2)
      CALL PRECSS (UJ,T2,T3,1)
      CALL ROTATE (1,TE1(6),T1)
      CALL TRIM (T1,T2,T3)
      CALL ONEM (T3,D,B)
      CALL ONEM (T3,OME,SUN)
C     POSITIONS OF EARTH(B) AND SUN(SUN) IN MEAN OF DATE ECLIPTIC
      SEV(1)=TE1(2)
      SEV(2)=TE1(5)
      SEV(3)=TE2(2)
      SEV(4)=TE2(5)
      SEV(5)=-0.29654088D-13+0.405942343D-20*(ET+24980.D0)
      SEV(6)= 0.543658286D-13+0.477579229D-20*(ET+24980.D0)
C     SEV  VECTOR OF MEAN LONGITUDE , NODE AND THEIR FIRST AND SECOND DERIVATIVS
    2 CONTINUE
      Z(1)=Y(1)-Y(2)+SEV(1)-SEV(2)+PI
      CALL REDPI (Z(1))
      Z(2)=Y(2)+SEV(2)
      CALL REDPI (Z(2))
      Z(3)=Y(3)+TEQ
      Z(4)=Y(4)-Y(5)+SEV(3)-SEV(4)
      Z(5)=Y(5)+SEV(4)
      Z(6)=Y(6)
C     Z - EULERIAN ANGLES CALCULATED FROM Y
      CALL TRIK (Z,ZP,TETA,FETA)
      DO 3 K=1,3
```

FUNPI5 (Cont)

```
      YP(K)=Y(3+K)
    3 YP(3+K)=ZP(K)
C     YP(1-6) ASSIGNMENT OF DERIVATIVES FOR THE PHYSICAL LIBRATION ANGLES
      IF(KLU.GT.1) GOTO 17
C     KLU IS SET GREATER THAN 1 FOR THE CASE WHEN NO PARTIALS ARE GENERATED
      DO 16 I=1,6
      DO 13 J=1,6
      IJ=J*6+I
      YP(IJ)=0.D0
      DO 13 L=1,6
   13 YP(IJ)=YP(IJ)+TETA(I,L)*Y(J*6+L)
C     CREATION OF U DOT IN VECTOR FORM YP(7-42)
      DO 15 J=1,3
      IJ=36+I+6*J
      YP(IJ)=0.D0
      DO 14 L=1,6
   14 YP(IJ)=YP(IJ)+TETA(I,L)*Y(36+6*J+L)
   15 YP(IJ)=YP(IJ)+FETA(I,J)
C     CREATION OF Q DOT IN VECTOR FORM YP(43-60)
   16 CONTINUE
   17 NUMB=NUMB+1
      RETURN
    6 WRITE(6,70)ET,NUMB
      NUT=1
      RETURN
   70 FORMAT(5X,'SOMETHING IS WRONG IN THE DATA',F20.8,I10)
      END
```

SUBROUTINE : FUNPL6

CALL STATEMENT : FUNPL6 (ET, Y, YP)

SUBROUTINE PURPOSE : The subroutine accomplishes the same purposes FUNPL5 except that the simulated environment is used in lieu of the JPL ephemeris DE-69.

INPUT PARAMETERS : Input parameters are exactly the same as for FUNPL5.

OUTPUT PARAMETERS : Output parameters are exactly the same as for FUNPL5.

COMMON AREA PARAMETERS : Common storage is the same as FUNPL5 except that the areas CETBL1/CETBL2/CETBL4 are not required.

PROGRAM DESCRIPTION : The computation performed by the subroutine is exactly the same as FUNPL5 except that the simulated ephemeris data is read from a data set No. 4 through the use of EPHITL.

SUBROUTINES REQUIRED :

- A. O.S.U. Project Library:
  1. REDPI
  2. TRIK
  3. EPHITL

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science Report No. 156.

FUNPL6

```
      SUBROUTINE FUNPL6 (ET,Y,YP)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION Y(60),YP(60),SEV(6)     ,Z(6),ZP(3),D(3),B(3),SUN(3),
     POME(3),F(4),TE1(6),TE2(6),1RE(13),S(6,12),T1(3,3),T2(3,3),T3(3,3)
     P,TETA(6,6),FETA(6,3),QU(18)
C     Y(1-6) - PHYSICAL LIBRATIONS ANGLES
C     Y(7-42)- STATE TRANSITION MATRIX IN COLUMNS (TAU ...RODOT)
C     Y(43-60) - PARAMETERS SENSITIVITY IN COLUMNS (C22,BETA,C20)
      COMMON/MAIF/ALF,BET,GAM,TEQ  /EXE/TO,B,SUN,NUMB/ALL/  SEV,A,NUT,
     PKLU
      DATA PI,TSE /3.14159265358979D0,0.D0/
C     ET - EPOCH IN (JD-2440000.)
      IF(ET.EQ.TO) GOTO 2
      TO=ET
      CALL EPHITL (ET,4,QU)
      DO 1 I=1,3
      B(I)=QU(I)
    1 SUN(I)=1.D 05
    2 CONTINUE
      Z(1)=Y(1)-Y(2)+SEV(1)-SEV(2)+PI+(SEV(3)-SEV(4))*(ET-222.5D0)
      CALL REDPI (Z(1))
      Z(2)=Y(2)+SEV(2)+SEV(4)*(ET-222.5D0)
      CALL REDPI (Z(2))
      Z(3)=Y(3)+TEQ
      Z(4)=Y(4)-Y(5)+SEV(3)-SEV(4)
      Z(5)=Y(5)+SEV(4)
      Z(6)=Y(6)
C     Z - EULERIAN ANGLES CALCULATED FROM Y
      CALL TRIK (Z,ZP,TETA,FETA)
      DO 3 K=1,3
      YP(K)=Y(3+K)
    3 YP(3+K)=ZP(K)
C     YP(1-6) ASSIGNMENT OF DERIVATIVES FOR THE PHYSICAL LIBRATION ANGLES
      IF(KLU.GT.1) GOTO 17
C     KLU IS SET GREATER THAN 1 FOR THE CASE WHEN NO PARTIALS ARE GENERATED
      DO 16 I=1,6
      DO 13 J=1,6
      IJ=J*6+I
      YP(IJ)=0.D0
      DO 13 L=1,6
   13 YP(IJ)=YP(IJ)+TETA(I,L)*Y(J*6+L)
C     CREATION OF U DOT IN VECTOR FORM YP(7-42)
      DO 15 J=1,3
      IJ=36+I+6*J
      YP(IJ)=0.D0
      DO 14 L=1,6
   14 YP(IJ)=YP(IJ)+TETA(I,L)*Y(36+6*J+L)
   15 YP(IJ)=YP(IJ)+FETA(I,J)
C     CREATION OF Q DOT IN VECTOR FORM YP(43-60
   16 CONTINUE
   17 CONTINUE
      RETURN
      END
```

SUBROUTINE : FUNST3

CALL STATEMENT : FUNST3 (ET, Y, YP)

SUBROUTINE PURPOSE : Basically the subroutine accomplishes two purposes:

A. The subroutine evaluates the time derivatives of the perturbations of the 18 elements of the simulated ephemeris (similar to FUNEPH).

B. The subroutine evaluates the time derivatives of the state vector of a satellite in the simulated earth moon environment.

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. ET is the epoch in Julian days at which the time derivative are to be evaluated.

B. Vector (extended precision).
   1. Y is a 24-element vector containing, sequentially:
      a. The 18 perturbation of the elements of the simulated ephemeris as defined by comment cards in the statement listings. Units of the parameter sets are kilometers, kilometers per day, radians and radians per day respectively.
      b. Elements 19-24 contain the state vector (kilometers and kilometers per day) of the satellite in the selenocentric system.

OUTPUT PARAMETERS :

A. Vector (extended precision).
   1. YP is a 24-element vector returned from the subroutine containing the time derivatives of the Y vector described in the above paragraph. Units are per day time derivatives of the original vector.

COMMON AREA PARAMETERS : Common block storage is used to transmit information between the subroutine and the main program or calling subroutine. Two common areas are used:

A. Area /MAIFUN/ENCKE, AMM, PM, NUMB.
   1. The extended precision variables ENCKE, AMM and PM are quantities which are described in the FUNEPH Program Description.

2. NUMB is a single precision integer which serves as a counter to indicate how many times the FUNST3 subroutine has been called.

B. Area ALL/Q.

1. The extended precision vector Q contains the 18 elements of the simulated ephemeris in the same order as in the Y vector. To obtain the Q vector the perturbed elements of the Y vector are added to the "reference" case.

PROGRAM DESCRIPTION : The theoretical basis of FUNST3 formulation is outlined in [Papo, 1971] Chapter 4, Section 4.4.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. STVITR
   2. REDPI
   3. ROTATE
   4. TRIM
   5. ONEM

B. Fortran Scientific Subroutine Package:
   1. DGMPRD
   2. DGMTRA
   3. DGMADD

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

FUNST3

```
      SUBROUTINE FUNST3 (ET,Y,YP)
C     SIMULATED ENVIRONMENT
C     SATELLITE TRAJECTORY GENERATOR COWELL EQUATIONS OF MOTION FOR SATELLI E
C     NO RANGE OR RRATE DATA GENERATED
C     Y( 1- 6)  MOON  STATE VECTOR GEOCENTRIC ECLIPTIC
C     Y( 7-12)  MOON EULERIAN ANGLES AND TIME RATES
C     Y(13-18) EARTH EULERIAN ANGLES AND TIME RATES
C     Y(19-24) SATELLITE STATE VECTOR SELENOCENTRIC ECLIPTIC
C     POSITION  :  X , Y , Z
C     EULERIAN ANGLES  :  LONGITUDE , NODE , INCLINATION
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(24),YP(24),Q(18),STV(6),ENCKE(7),S(3),R(3),X(3),T1(3,3     5*P
     P),T2(3,3),T3(3,3),T4(3,3),TM(3,3),TMT(3,3),TE(3,3),TET(3,3),VX(3),       1
     PVX3(3),VS(3),EMS(12),XM(12,3),SM(12,3),SSM(12),SCO(3),OMS(12),PM(6       2
     P ),RRRR(3,4               )  ,GS(3,3),QS(3,3),PS(3,3),XYZ(3),UVW(3       4
     P),XCR(3,3),SCR(3,3),RCR(3,3),VR(3),AMM(3,3),VR3(3)                       5
      COMMON /MAIFUN/ENCKE,AMM,PM,NUMB /ALL/ Q
      DATA ZERO,ONE,EMM,CPHL,ALF,BET,GAM,EMU,CEAR,CONS,ES,HS                 5*P
     P            /0.D0,1.D0,81.3D0,.892668D 16,.4194213D-03,.629D-03
     P,.2095788D-03,.3278C2D-02,359924176800.D0,.297556D 16,66067.85625D       1
     P0,5369046.636D0/
      DATA GS,CONM /      3578363.863D0,3*0.D0,3579114.284D0,3*0.D0,3580
     P615.126D0,3012159753997540.D0/                                           5
      DATAEMS,XM/.234556D-06,-.246487D-06,.189182D-06                          3
     P, -.146261D-06, .144376D-06,  -.15725D-06,.34727D-07, -.74331D-07,       4
     P.123304D-06, -.99066D-07,  .99814D-07,  -.102563D-06,                    5
     P1619.177, 1532.288,1497.018,1630.787, 938.990,1110.483,750.123,          6
     P836.942,715.644,582.937,2*750.123,-433.527,559.617,866.747,1.0,          7
     P-940.490,-1325.401, -1302.581,-223.928, 856.851, 1609.474, 1304.         8
     P581, 435.527, 449.355,593.968,-151.986,-594.968, -1117.734, -151.        9
     P986, 868.554, 1504.745, 1330.968, 301.319,-869.554,-1505.745/           10
      NUMB=NUMB+1
      CALL STVITR (ET,STV,AMM,PM)
      DO 1 I=1,6
      Q(I)=Y(I)+STV(I)
      Q(I+6)=Y(I+6)
    1 Q(I+12)=Y(I+12)
      TT=ET-ENCKE(1)
      Q(7)=Q(7)+ENCKE(2)+ENCKE(4)*TT
      CALL REDPI(Q(7))
      Q(8)=Q(8)+ENCKE(3)
      Q(10)=Q(10)+ENCKE(4)
      Q(13)=Q(13)+ENCKE(5)+ENCKE(7)*TT
      CALL REDPI(Q(13))
      Q(15)=Q(15)+ENCKE(6)
      Q(16)=Q(16)+ENCKE(7)
      DO 22 I=1,12
   22 SSM(I)=ZERO
C
      XX=ZERO
```

```
      XO=ZERO
      SS=ZERO
      RR=ZERO
      DO 2 I=1,3
      X(I)=Q(I)
      S(I)=Y(I+18)
      R(I)=X(I)+S(I)
      SCO(I)=ZERO
      XX=XX+X(I)*X(I)
      XO=XO+STV(I)*STV(I)
      SS=SS+S(I)*S(I)
    2 RR=RR+R(I)*R(I)
      X7=ONE/XX**3.5DO
      S7=ONE/(EMM*SS**3.5DO)
      R7=ONE/RR**3.5DO
      X5=X7*XX
      S5=S7*SS
      R5=R7*RR
      X3=X5*XX
      S3=S5*SS
C
      CALL ROTATE (3,Q(7),T1)
      CALL ROTATE(1,-Q(9),T2)
      CALL ROTATE(3,Q(8),T3)
      CALL TRIM (T1,T2,T4)
      CALL TRIM (T4,T3,TM)
      CALL DGMTRA (TM,TMT,3,3)
      CALL TRIM (TMT,GS,T3)
      CALL TRIM (T3,TM,PS)
      CALL ONEM (TM,X,XYZ)
      CALL DGMPRO (XM,TM,SM,12,3,3)
C
      OM1=T1(2,1)*T2(3,2)*Q(11)-T1(1,1)*Q(12)
      OM2=T1(1,1)*T2(2,3)*Q(11)+T1(1,2)*Q(12)
      OM3=       T2(3,3)*Q(11)+            Q(10)
      TRE1= ALF*(CPHL*XYZ(2)*XYZ(3)*X5-OM2*OM3)+T1(1,2)*T2(3,3)*Q(11)*Q(
     P12)+T1(1,1)*T2(3,2)*Q(11)*Q(10)+T1(2,1)*Q(12)*Q(10)
      TRE2=-BET*(CPHL*XYZ(1)*XYZ(3)*X5-OM1*OM3)+T1(1,1)*T2(3,3)*Q(11)*Q(
     P12)+T1(2,1)*T2(3,2)*Q(11)*Q(10)-T1(1,1)*Q(12)*Q(10)
      TRE3= GAM*(CPHL*XYZ(1)*XYZ(2)*X5-OM1*OM2)+T2(3,2)*Q(11)*Q(12)
      YP(10)=( T1(1,2)*T2(3,3)*TRE1+T1(1,1)*T2(3,3)*TRE2)/T2(3,2)+TRE3
      YP(11)=(-T1(1,2)      *TRE1-T1(1,1)      *TRE2)/T2(3,2)
      YP(12)= -T1(1,1)      *TRE1+T1(1,2)      *TRE2
C
      CALL ROTATE (3,Q(13),T1)
      CALL ROTATE (1,-Q(15),T2
      CALL ROTATE (3,Q(14),T3)
      CALL TRIM (T1,T2,T4)
      CALL TRIM (T4,T3,TE)
      CALL DGMTRA (TE,TET,3,3)
      DO 23 I=1,3
```

FUNST3 (Cont)

```
      DO 23 J=1,3
   23 QS(I,J)=132135.7125D0*TE(3,I)*TE(3,J)
      CALL ONEM (TE,X,UVW)
C
      HEL=ENCKE(7)*(ONE+EMU)-T2(3,3)*Q(17)
      TWO1=T1(1,2)*T2(3,3)*Q(17)*Q(18)+T1(1,1)*T2(3,2)*Q(17)*HEL+T1(2,1)
     P*Q(18)*HEL+CEAR*UVW(2)*UVW(3)*X5
      TWO2=T1(1,1)*T2(3,3)*Q(17)*Q(18)+T1(2,1)*T2(3,2)*Q(17)*HEL-T1(1,1)
     P*Q(18)*HEL-CEAR*UVW(1)*UVW(3)*X5
      YP(17)=(T1(1,2)*TWO1+T1(1,1)*TWO2)/T2(2,3)
      YP(18)=-T1(1,1)*TWO1+T1(1,2)*TWO2
      YP(16)=T2(3,2)*Q(17)*Q(18)-T2(3,3)*YP(17)
C
      QL=ZERO
      QR=ZERO
      DO 3 I=1,3
      DO 10 J=1,12
      SM(J,I)=S(I)-SM(J,I)
   10 SSM(J)=SSM(J)+SM(J,I)**2
      QR=QR+S(I)*(X(I)+S(I)*0.5D0)
    3 QL=QL+Y(I)*(STV(I)+Y(I)*.5D0)
      EFR=ONE-ONE/(ONE+2.D0*QR/XX)**
      EFO=ONE-ONE/(ONE+2.D0*QL/XO)**
      XO=ONE/XO**1.5D0
      DO 4 I=1,3
      VR3(I)=X3*(S(I)-EFR*R(I))
    4 VX3(I)=XO*(Y(I)-X(I)*EFO)
      DO 15 I=1,12
      DMS(I)=EMS(I)/SSM(I)**1.5D0
      DO 15 J=1,3
   15 SCO(J)=SCO(J)+DMS(I)*SM(I,J)
C
      DO 5 I=1,3
      DO 5 J=1,3
      XCR(I,J)=-2.5D0*X(I)*X(J)*X7
      SCR(I,J)=-2.5D0*S(I)*S(J)*S7
      RCR(I,J)=-2.5D0*R(I)*R(J)*R7
      IF(I.NE.J) GOTO 5
      XCR(I,J)=XCR(I,J)+XX*X7
      SCR(I,J)=SCR(I,J)+SS*S7
      RCR(I,J)=RCR(I,J)+RR*R7
    5 CONTINUE
      CALL DGMADD (PS,QS,T1,3,3)
      CALL TRIM (XCR,T1,T2)
      CALL TRIM (SCR,PS,T3)
      CALL TRIM (RCR,QS,T4)
      DO 6 I=1,3
      DO 6 J=1,3
      IF(I.NE.J) GOTO 6
      T2(I,J)=T2(I,J)+X5*(ES+HS)
      T3(I,J)=T3(I,J)+S5*HS+S3
```

FUNST3 (Cont)

```
      T4(I,J)=T4(I,J)+R5*ES
    6 CONTINUE
      CALL ONEM (T2,X,VX)
      CALL ONEM (T3,S,VS)
      CALL ONEM (T4,R,VR)
      DO 7 I=1,3
      YP(I+3)=-CONM*(VX(I)+VX3(I))
      YP(I+21)=-CONS*(VR(I)+VS(I)+SCO(I)-VX(I)+VR3(I))
      YP(I    )=Y(I+3 )
      YP(I+6 )=Y(I+9 )
      YP(I+12)=Y(I+15)
    7 YP(I+18)=Y(I+21)
   99 RETURN
      END
```

SUBROUTINE : GASTIM

CALL STATEMENT : GASTIM (JD, DOBLQ, DLONG, GAST)

SUBROUTINE PURPOSE : The subroutine computes the Greenwich apparent sidereal time at a given epoch, given the nutation in obliquity and longitude for the epoch.

INPUT PARAMETERS :

- A. Scalars (extended precision).
  1. JD is the epoch in Julian days minus 2437000.0.
  2. DOBLQ is the nutation in obliquity in radians.
  3. DLONG is the nutation in longitude in radians.

OUTPUT PARAMETERS :

- A. Scalars (extended precision).
  1. GAST is the Greenwich apparent siderial time for the epoch specified by JD.

PROGRAM DESCRIPTION : [Fajemirokun, 1971] Page 51.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. MEANAN

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

<u>GASTIM</u>

```
      SUBROUTINE GASTIM(JD,DOBLQ,DLONG,GAST)
      IMPLICITREAL*8(A-H,O-Z)
      DOUBLE PRECISION JD
      DIMENSION ANM(6),DNM(6)
      DATA PI2/6.2831853071795860D0/,RADDEG/57.295779513082320D0/
C     THIS ROUTINE COMPUTES THE G.A.S.T.
C     IMPUT PARAM.--JD,NUTATION(IN RADIANS) IN LONG & OBLQ
C     OUTPUT IS GAST IN RADIANS
      FJD=JD+2437000.0D0
      CALLMEANAN(FJD,ANM,DNM)
      OBLQ=ANM(6)
      OBLQ=OBLQ+DOBLQ
      EQE=DLONG*DCOS(OBLQ)
      IJD=IDINT(JD)
      DJD=DFLOAT(IJD)+0.5D0
      FRAC=JD-DJD
      IF(FRAC.LT.0.D0)FRAC=1.D0+FRAC
      UT=FRAC*24.D0
      TM=(FJD-2415020.D0)/36525.D0
      GMST=UT+(6.D0+38.D0/60.D0+45.836D0/3600.D0)+(8640184.542D0/3600.D0
     *)*TM+(0.0929D0/3600.D0)*TM*TM
      GMST=DMOD(GMST,24.D0)
      GMST=GMST*15.D0/RADDEG
      GAST=GMST+EQE
      GAST=DMOD(GAST,PI2)
      RETURN
   60 FORMAT(//,10X,D25.16)
      END
```

SUBROUTINE : GEULAN

CALL STATEMENT : GEULAN (JD, THETA, PHI, PSI, DOBLQ, DLONG)

SUBROUTINE PURPOSE : Computes the three Eulerian angles (θ, ψ, ω) between the mean ecliptic system of 1950.0 and the average terrestrial system.

INPUT PARAMETERS :

A. Scalars (extended precision).
   1. JD is the Julian date minus 2437000.0.
   2. DOBLQ is the nutation in obliquity in radians.
   3. DLONG is the nutation in longitude in radians.

OUTPUT PARAMETERS :

A. Scalars (extended precision).
   1. THETA is the Eulerian angle θ in radians.
   2. PHI is the Eulerian angle ω in radians.
   3. PSI is the Eulerian angle ψ in radians.

PROGRAM DESCRIPTION : The development of the earth's Eulerian angles is outlined in Report 157, Section 2.35.

SUBROUTINES REQUIRED :

O.S.U. Project Library:
1. PRECES
2. NUTATE
3. GASTIM
4. STRANS
5. MEANAN
6. ROTATE
7. TRIM

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

GEULAN

```
      SUBROUTINE GEULAN(JD,THETA,PHI,PSI,DOBLQ,DLONG)
      IMPLICITREAL*8(A-H,O-Z)
      DOUBLEPRECISIONJD
      DIMENSION PRE(6),PRECM(3,3),DNUTA(3,3),SMAT(3,3),ANM(6).DNM(6),
     *RE1(3,3),T1(3,3),T2(3,3),A(3,3)
      DATA                              ET/2433282.423D0/
      DATA PI2/6.2831853071795860D0/
C     THIS ROUTINE COMPUTES THE 3 EULERIAN ANGLES BETWEEN THE MEAN ECLIP
C     TIC SYSTEM (1950.0) AND THE MEAN TERRESTIAL ROTATING SYSTEM
C     EXTERNAL ROUTINES REQD.-PRECES,NUTATE,GASTIM,STRANS,MEANAN,ROTATE,
C     TRIM. INPUT PARAM.--JUL. DATE,NUTAT.IN OBLQ. AND LONGITUDE
C
      FJD=JD+2437000.0D0
      CALLPRECES(FJD,PRE,PRECM)
      CALLNUTATE(FJD,DOBLQ,DLONG,DNUTA)
      CALLGASTIM(JD,DOBLQ,DLONG,GAST)
      CALLSTRANS(FJD,GAST,SMAT)
      CALLMEANAN(ET,ANM,DNM)
      OBLQ=ANM(6)
      CALLROTATE(1,-OBLQ,RE1)
      CALLTRIM(PRECM,RE1,T1)
      CALLTRIM(DNUTA,T1,T2)
      CALLTRIM(SMAT,T2,A)
      THETA=DARCOS(A(3,3))
      ST=DSIN(THETA)
      PSI=DARCOS(A(3,2)/ST)
      PSI=PI2-PSI
      PHI=DARCOS((A(1,1)*A(3,2)-A(1,2)*A(3,1))/ST)
      DS=-((A(2,1)*A(3,2)-A(2,2)*A(3,1))/ST)
      IF(DS.LT.0.D0)PHI=PI2-PHI
   50 FORMAT(/,(3D25.16))
   60 FORMAT(//,2X,5D20.12)
      RETURN
      END
```

SUBROUTINE : INVSPE

CALL STATEMENT : INVSPE (A, B, M, N, K, L)

SUBROUTINE PURPOSE : Copies a layer of a three dimensional input matrix (A) into a two dimensional matrix (B).

INPUT PARAMETERS :

A. Scalar integers (single precision).
   1. M, N, K are dimensions of the input matrix A (rows, columns, layers).
   2. L is the layer of A (consisting of K layers) which is to be copied into matrix B for output.

B. Matrices (extended precision).
   1. A is the M × N × K input matrix.

OUTPUT PARAMETERS :

A. Matrices (extended precision).
   1. B is the output M × N matrix containing layer L of the input matrix A..

PROGRAM DESCRIPTION : None

SUBROUTINES REQUIRED : None

REFERENCES : None

INVSPE

```
C
      SUBROUTINEINVSPE(A,B,M,N,K,L)
C     MAKING B EQUAL TO LAYER OF A
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONA(M,N,K),B(M,N)
      DO1I=1,M
      DO2J=1,N
    2 B(I,J)=A(I,J,L)
    1 CONTINUE
      RETURN
        END
C
```

SUBROUTINE : LADIS

CALL STATEMENT : LADIS (ET, PE, PM, XE, XM, XCE, X, D, ZD)

SUBROUTINE PURPOSE : The program computes the distance between an earth observatory and a lunar reflector at a given epoch and the zenith distance of the ray at the earth observatory. The program presupposes the observations are made in the simulated environment described in [Papo, 1971] in a refraction free environment assumed in [Fajemirokun, 1971].

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. ET is the epoch of observation in Julian days.

B. Vectors (extended precision, 3 element).
   1. XE is the position vector of the earth observatory in a body fixed system (km).
   2. XM is the position vector of the lunar reflector in a body fixed system (km).
   3. XCE is the geocentric position of the moon in the simulated environment (km).

C. Matrices (extended precision).
   1. PE is a 3×3 matrix used to transform coordinates from the earth fixed system to the mean ecliptic coordinate system.
   2. PM is a 3×3 matrix used to transform coordinates from the moon fixed system to the mean ecliptic system.

OUTPUT PARAMETERS :

A. Vector (extended precision).
   1. X is the topocentric position vector of the lunar reflector (km).

B. Scalars (extended precision).
   1. D is the distance between the earth observing station and the lunar reflector (km). The variable is set to zero if the zenith distance precludes observation.
   2. ZD is the zenith distance of the lunar point in degrees.

PROGRAM DESCRIPTION: The program logic follows the formulation given in [Fajemirokun, 1971] for laser ranging.

A. First, the topocentric position vector of the lunar reflector is obtained.

B. The simulated laser range is found from the topocentric vector.

C. The zenith distance of the observed range is computed.

D. If the observing altitude of the ray is between 30 and 70 degrees, control is returned to the calling program. If the "observation" cannot be made, the distance is set to zero and control is returned to the calling subroutine.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. ONEM

B. O.S.U. Utility Library:
   1. MADD
   2. MSUBT

REFERENCES :

A. Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

B. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

LADIS

```
C
      SUBROUTINE LADIS(ET,PE,PM,XE,XM,XCE,X,D,ZD)
C     THIS PROGRAM SIMULATES RANGES TO THE MOON
C     PROGRAM USES SIMULATED ENVIRONMENT DATA (CREATED BY H. PAPO)
C     ET IS EPOCH OF OBSERVATION IN JULIAN DAY
C     PE MATRIX TRANSFORMS FROM UVW TO MEAN ECLP. SYSTEM
C     PM MATRIX TRANSFORMS XYZ(MOON) TO MEAN ECLP. SYSTEM
C     XE,XM--POS. VECT. OF EARTH PT. AND MOON PT. IN UVW & XYZ SYSTEM
C     XCE-- GEOCENTRIC (ECLP.) POSITION OF SELENOCENTER
C     D IS THE COMPUTED DISTANCE
C     X--THE TOPOCENTRIC POS. OF MOON POINT IN MEAN ECLP. SYSTEM
C     ZD IS THE ZENITH DISTANCE OF MOON POINT
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PE(3,3),PM(3,3),XE(3),XM(3),XCE(3),XEE(3),XME(3),XEM(3),
     *X(3)
      DATA RADDEG/57.29577951308232/
      DS1=DSIN(20.D0/RADDEG)
      DS2=DSIN(60.D0/RADDEG)
      DS3=DCOS(80.D0/RADDEG)
C
      CALLONEM(PE,XE,XEE)
      CALLONEM(PM,XM,XME)
      CALLMADD(XCE,XME,3,1,XEM)
      CALLMSUBT(XEM,XEE,3,1,X)
      D2=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)
      D=DSQRT(D2)
C     THE FOLLOWING CHECKS IF OBSERVING ALTITUDE IS 30<ALT.<70
      AB=0.D0
      EB=0.D0
       A=0.D0
       B=0.D0
       E=0.D0
      DO50I=1,3
      AB=AB+XEE(I)*X(I)
      EB=EB+XME(I)*(-X(I))
      A=A+XEE(I)*XEE(I)
      B=B+X(I)*X(I)
      E=E+XME(I)*XME(I)
   50 CONTINUE
      A=DSQRT(A)
      B=DSQRT(B)
      E=DSQRT(E)
      CST=AB/(A*B)
      CH=EB/(E*B)
      QD=DARCOS(CST)
      ZD=QD*RADDEG
      IF(CST) 60,60,65
   65 CONTINUE
      IF(CST.LE.DS1.OR.CST.GE.DS2)GOTO60
```

```
C      THE FOLLOWING CHECKS IF MOON POINT IS 20 DEG. OFF LUNAR LIMB
       IF(CH.LT.DS3)GOTO60
       GOTO999
    60 CONTINUE
C        IMPOSSIBLE TO OBSERV
       D=0.D0
   999 RETURN
    96 FORMAT(3D25.12)
       END
C
```

SUBROUTINE : LASOLV

CALL STATEMENT : LASOLV (IDP, XP, IDL, XL, B, EMINV, QLA, PX, TEMP, LVEC, MVEC, EN, COREL, NE, NM, NU, NOB, NP, NL, BM, W)

SUBROUTINE PURPOSE : This subroutine is the basic routine for adjusting simulated laser ranging.

INPUT PARAMETERS :

A. Integers (single precision) from main program.
   1. NE is the number of earth station coordinates.
   2. NM is the number of lunar reflector coordinates.
   3. NOBS is the number of laser observations.
   4. NP is the number of earth stations.
   5. NL is the number of lunar reflectors.

B. Integers (single precision), card input within subroutine.
   1. IDP is an NP element vector containing the earth station numbers.
   2. IDL is an NL element vector containing the lunar station numbers.
   3. ID1 is an input earth station number which an observation is made from.
   4. ID2 is an input lunar station number to which an observation is made.

C. Matrices (extended precision) card input within subroutine.
   1. XP is an NP×3 element array containing the geocentric positions of the observing stations in kilometers.
   2. XL is an NL×3 element array containing the selenocentric positions of the laser retroflectors in kilometers.

D. Scalars (extended precision) card input within subroutine.
   1. ET is the epoch in Julian days minus 24370000.0 days of an observation.
   2. X is the topocentric position vector of the lunar reflector at ET (in kilometers).
   3. D is the simulated lunar range in kilometers at ET.
   4. ZD is the zenith distance in degrees of the simulated observation at ET.

E. Scalars (extended precision) input from disk within subprogram.
   1. ETE is the initial epoch for lunar data (see EPHITL).

2. ET1 is an input epoch for earth Eulerian angle data.
3. ET2 is an input epoch for lunar Eulerian angle data.

F. Vectors (extended precision) input from disk within subprogram.
   1. ET and E1 are element vectors described in the PLADIS Program Description.
   2. Y is a 60-element vector described in the FUNPL5 Program Description.

G. Matrices (extended precision) input from disk within subprogram.
   1. FH is a $3 \times 18 \times 15$ array described in the EPHITL Program Description.
   2. SE is a 3×6 array described in the PLADIS Program Description.

OUTPUT PARAMETERS : Many of the extended precision variables in the Call Statement are included for Fortran object time dimensioning. The important output from the routine is threefold:

A. The array QLA dimensioned Nu × NU is printed and represents the variance-covariance matrix of the adjustment. The variances are listed in [Fajemirokun, 1971] Page 200.

B. The array COREL dimensioned NU × NU is the correlation matrix resulting from the adjustment [Fajemirokun, 1971] Pages 202-204.

C. The W vector dimensioned NU is the solution vector obtained from the adjustment

PROGRAM DESCRIPTION · The program carries out the computations outlined theoretically in [Fajemirokun, 1971] Chapter 5.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. EPHITL
   2. PMAT
   3. LADIS
   4. PLADIS
   5. MATPA

B. Fortran Scientific Subroutine Package:
   1. DMINV
   2. DGMPRD

C. O.S.U. Utility Library:
   1. MWRITE
   2. MSCALE
   3. MDUP

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

LASOLV

```
      SUBROUTINE LASOLV(IDP,XP,IDL,XL,B,EMINV,QLA,PX,TEMP,LVEC,MVEC,EN,
     *COREL,NE,NM,NU,NOB,NP,NL,BM,W)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IDP(NP),XP(NP,3),IDL(NL),XL(NL,3),B(NOB,NU),EMINV(NOB),
     *QLA(NU,NU),PX(NU),TEMP(NOB),LVEC(NU),MVEC(NU),EN(NU,NU),
     *COREL(NU,NU),FH(3,16,15),CU(16),XCE(3),XM(3),XE(3),X(3),PE(3,3),
     *PM(3,3),EP(6),EL(6),SE(3,6),SM(3,6),PSM(3,3),B1(21),B2(3),Y(60),
     *Z1(6),Z2(6),E(15)
      DIMENSION BM(NU,NOB),W(NOB)
      COMMON/EPHEM/ETE,FH
    1 CONTINUE
      CON=206.2648062470964D0
      ETC=0.D0
      DO25I=1,NE
      PX(I)=1.D0/(25.D0*25.D0)
   25 CONTINUE
      K1=NE+7
      K2=NE+6+NM
      DO21I=K1,K2
      PX(I)=1.D0/(1000.D0*1000.D0)
   21 CONTINUE
      K3=K2+1
      K4=K2+3
      DO26I=K3,K4
      J1=I-NM-6
      PX(J1)=1.D0/(1.D0*1.D0)
      PX(J1+3)=1.D0/(0.5D0*0.5D0)
      PX(I)=1.D0/(20.D0*20.D0)
      PX(I+3)=1.D0/(10.D0*10.D0)
   26 CONTINUE
      K5=K4+3
      PX(K5+1)=1.D0/(0.5D0*0.5D0)
      PX(K5+2)=1.D0/(2.0D0*2.0D0)
      PX(K5+3)=1.D0/(0.01D0*0.01D0)
      DO29I=1,3
      DO28J=1,3
      PSM(I,J)=0.D0
   28 CONTINUE
      DO29K=1,6
      SE(I,K)=0.D0
      SM(I,K)=0.D0
   29 CONTINUE
      DO30I=1,NP
      READ(5,81)IDP(I),(XP(I,J),J=1,3)
      DO27J=1,3
      IVALUE=IDINT(XP(I,J)*10.D0)
      XP(I,J)=DFLOAT(IVALUE)
      XP(I,J)=XP(I,J)/10.D0
   27 CONTINUE
      WRITE(6,81)IDP(I),(XP(I,J),J=1,3)
```

```
30 CONTINUE
   DO40I=1,NL
   READ(5,81)IDL(I),(XL(I,J),J=1,3)
   DO35J=1,3
   IVALUE=IDINT(XL(I,J))
35 XL(I,J)=DFLOAT(IVALUE)
   WRITE(6,61)IDL(I),(XL(I,J),J=1,3)
40 CONTINUE
   REWIND 4
   REWIND 2
   REWIND 3
   READ(4) ETE,FH
   IJ=10
   I1=16
   NR=0
   L1=NE+6
   L2=NE+6+NM
   L3=L2+6
45 READ(5,92)ET,ID1,ID2,X,D1,ZD
   IF(ET.EQ.0.D0)GOTU70
   IF(ET.EQ.ETC)GOTU56
   CALLEPHITL(ET,4,QU)
   DO50I=1,3
   XCE(I)=QU(I)
50 CONTINUE
57 READ(3)ET1,EP,((SE(I,K),K=1,6),I=1,3)
   IF(ET1.NE.ET)GOTO57
   CONTINUE
   THETAE=EP(1)
     PSIE=EP(2)
     PHIE=EP(3)
   CALLPMAT(PSIE,THETAE,PHIE,PE)
58 READ(2)ET2,EL,Y
   IF(ET2.NE.ET)GOTO58
   CONTINUE
   DO53I=1,3
   DO53J=1,6
   JJ=J*6+I
53 SM(I,J)=Y(JJ)
   DO54I=1,3
   DO54J=1,3
   JJ=J*6+I+36
54 PSM(I,J)=Y(JJ)
   THETAM=EL(1)
   PSIM=EL(2)
   PHIM=EL(3)
   CALLPMAT(PSIM,THETAM,PHIM,PM)
56 CONTINUE
   DO60J=1,3
   XM(J)=XL(ID2,J)
   XE(J)=XP(ID1,J)
60 CONTINUE
```

```
      CALLLADIS( ET,PE,PM,XE,XM,XCE,X,D,ZD)
      WRITE(6,94)ET,ID1,ID2,X,D,ZD
      IF(D.EQ.0.D0)GOTO999
      NR=NR+1
      W(NR)=(D1-D)*1000.D0
      CALLPLADIS(ET,XE,XM,XCE,X,D1,PE,PM,EP,EL,SE,SM,PSM,B1,B2)
      J=(ID1*3)-2
      K=(ID2*3)-2
      B(NR,J)=-B1(1)
      B(NR,J+1)=-B1(2)
      B(NR,J+2)=-B1(3)
      B(NR,K+L1)=-B1(10)
      B(NR,K+L1+1)=-B1(11)
      B(NR,K+L1+2)=-B1(12)
      DO65I=1,6
      B(NR,I+NE)=-B1(I+3)/CON
      B(NR,I+L2)=-B1(I+12)/CON
   65 CONTINUE
      B(NR,L3+1)=-B1(19)/CON
      B(NR,L3+2)=-B1(20)/CON
      B(NR,L3+3)=-B1(21)/CON
      ETC=ET
      GOTO45
   70 CONTINUE
      WRITE(6,95)NR
      CALLMWRITE(W,1,NOB,'OW  ')
      CALLMATPA(B,NR,NU,EMINV,QLA,TEMP)
      DO 75 I=1,NU
      QLA(I,I)=QLA(I,I)+PX(I)
   75 CONTINUE
      FACTOR=0.D0
      DO64I=1,NU
   64 FACTOR=FACTOR+DLOG10(QLA(I,I))
      FACTOR=1.D0/10.D0**(FACTOR/DFLOAT(NU))
      CALLMSCALE(FACTOR,QLA,NU,NU)
      CALLMDUP(QLA,NU,NU,EN)
      CALLDMINV(QLA,NU,DET,LVEC,MVEC)
      CALLDGMPRD(EN,QLA,COREL,NU,NU,NU)
      CALLMSCALE(FACTOR,QLA,NU,NU)
      CALLMWRITE(COREL,NU,NU,'1CHK')
      DO62I=1,NU
      DO62J=1,NU
   62 COREL(I,J)=QLA(I,J)/(DSQRT(QLA(I,I))*DSQRT(QLA(J,J)))
      CALLMWRITE(COREL,NU,NU,'1CCM')
      WRITE(6,90)
      N=0
      DO76I=1,NE,3
      N=N+1
      VA1=QLA(I,I)
      VA2=QLA(I+1,I+1)
      VA3=QLA(I+2,I+2)
```

```
76 WRITE(6,96)N,VA1,VA2,VA3
   WRITE(6,90)
   N=0
   L1=L1+1
   DO77I=L1,L2,3
   N=N+1
   VA1=QLA(I,I)
   VA2=QLA(I+1,I+1)
   VA3=QLA(I+2,I+2)
77 WRITE(6,96)N,VA1,VA2,VA3
   WRITE(6,90)
   L1=L1-1
   DO78I=1,6
   E(I)=QLA(I+NE,I+NL)
   E(I+6)=QLA(I+L2,I+L2)
78 CONTINUE
   E(13)=QLA(L3+1,L3+1)
   E(14)=QLA(L3+2,L3+2)
   E(15)=QLA(L3+3,L3+3)
   WRITE(6,97)E
   WRITE(6,90)
   WRITE(6,120)((COREL(I,J),J=1,9),I=1,9)
   WRITE(6,120)((COREL(I,J),J=10,18),I=10,18)
   WRITE(6,120)((COREL(I,J),J=19,27),I=19,27)
   WRITE(6,90)
   WRITE(6,120)((COREL(I,J),J=1,9),I=10,18)
   WRITE(6,120)((COREL(I,J),J=1,9),I=19,27)
   WRITE(6,120)((COREL(I,J),J=10,18),I=19,27)
   WRITE(6,90)
   WRITE(6,200)((QLA(I,J),J=1,9),I=1,9)
   WRITE(6,200)((QLA(I,J),J=10,18),I=10,18)
   WRITE(6,200)((QLA(I,J),J=19,27),I=19,27)
   WRITE(6,90)
   WRITE(6,200)((QLA(I,J),J=1,9),I=10,18)
   WRITE(6,200)((QLA(I,J),J=1,9),I=19,27)
   WRITE(6,200)((QLA(I,J),J=10,18),I=19,27)
   WRITE(6,90)
   DO73J=1,NU
   DO73I=1,NR
73 BM(J,I)=B(I,J)*EMINV(I)
   CALLDGMPRD(BM,W,EMINV,NU,NR,1)
   CALLDGMPRD(QLA,EMINV,W,NU,NU,1)
   CALLMSCALE(-1.D0,W,NU,1)
   WRITE(6,202)W
   WRITE(6,90)
   WRITE(6,99)DET
   WRITE(6,98)
999 RETURN
81 FORMAT(I5,3D20.9)
62 FORMAT(I2,8X,3F10.4)
90 FORMAT(1H1)
```

```
 92 FORMAT(F7.3,2I2,4F10.7,F5.1)
 94 FORMAT(/,5X,F7.2,2I5,4D19.10,5X,F5.1)
 95 FORMAT(//,I10,//)
 96 FORMAT(15X,I5,5X,3D17.7)
 97 FORMAT(20X,3D17.7)
 98 FORMAT(//,5X,'CONGRATULATIONS, JOB IS OVER ')
 99 FORMAT(//,D25.14)
100 FORMAT(//(5X,12F7.3))
110 FORMAT(//(/,5X,6F7.3))
120 FORMAT(//(5X,9F7.3))
200 FORMAT(//(5X,1P9D10.2))
202 FORMAT(1H1,///(/,5X,6D14.6))
    END
```

SUBROUTINE : LUC A

CALL STATEMENT : LUC A (N, D, DAN)

SUBROUTINE PURPOSE : Creates special skew matrices for differentiation of a rotation matrix as outlined by Lucas (t is an independent variable).

$$\frac{dR_N(\alpha)}{dt} = R_N(\alpha) \quad L_N \quad \frac{d\alpha}{dt} = L_N \, R_N(\alpha) \quad \frac{d\alpha}{dt}$$

INPUT PARAMETERS :

A. Integer N denotes the rotation axis (N = 1, 2, 3).

B. Scalar D is the derivative of the rotation angle $\frac{d\alpha}{dt}$.
If D = 1.0, the DAN matrix will yield $\frac{\partial R_N(\alpha)}{\partial \alpha}$.

OUTPUT PARAMETERS :

A. DAN is a 3×3 matrix of partial derivatives multiplied b
D (i.e. $\frac{\partial R_N(\alpha)}{\partial \alpha}$).

PROGRAM DESCRIPTION :

A. The DAN matrix is zeroed.

B. Indicies N1, N2 are set as (by integer authentic) :

N	N1	N2
1	2	3
2	3	1
3	1	2

C. The following elements of the DAN matrix are set as:

DAN (N1, N2) = D
DAN (N2, N1) = -D

Resulting, for axis 1, in:

Preceding page blank

$$DAN = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & D \\ 0 & -D & 0 \end{bmatrix}$$

For axis 2 :

$$DAN = \begin{bmatrix} 0 & 0 & -D \\ 0 & 0 & 0 \\ D & 0 & 0 \end{bmatrix}$$

For axis 3 :

$$DAN = \begin{bmatrix} 0 & D & 0 \\ -D & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

SUBROUTINES REQUIRED : None

REFERENCES :

Lucas, James (1963). "Differentiation of Orientation Matrix, *Photogrammetric Engineering*, July.

LUCA

```
  SUBROUTINE LUCA (N,D,DAN)
  REAL*8D,DAN(3,3)
  DO1I=1,3
  DO1J=1,3
1 DAN(I,J)=0.D0
  N1=N+1-((N+1)/4)*3
  N2=N+2-((N+2)/4)*3
  DAN(N1,N2)=D
  DAN(N2,N1)=-D
  RETURN
  END
```

SUBROUTINE : MATPA

CALL STATEMENT : MATPA (A, IA, JA, P, ANS, TEMP)

SUBROUTINE PURPOSE : The subroutine forms the product A'PA where A is an IA × JA matrix and P is an IA vector representing a diagonal IA × IA matrix and the product is dimensioned JA × JA.

INPUT PARAMETERS :

- A. Matrix (extended precision).
  1. A is the input matrix which is dimensioned IA × JA.
- B. Vector (extended precision).
  1. P is a vector with IA elements which represent the diagonal elements of an IA × IA matrix.
- C. Integers (single precision).
  1. IA is the number of columns in the A matrix.
  2. JJ is the number of rows in the input A matrix.

OUTPUT PARAMETERS :

- A. Matrix (extended precision).
  1. ANS is an JA × JA matrix representing the product A'PA.
- B. Vector (extended precision).
  1. TEMP is a work vector contained in the I/O parameters for object time dimensioning.

PROGRAM DESCRIPTION : Denoting the ANS matrix by N the subroutine computes:

$$N = A'PA$$

SUBROUTINES REQUIRED : None

REFERENCES : None

```
C
      SUBROUTINE MATPA(A,IA,JA,P,ANS,TEMP)
      DOUBLE PRECISION A(IA,JA),P(IA),ANS(JA,JA),TEMP(IA)
      DO 40 J=1,JA
      DO 25 I=1,IA
      TEMP(I)=0.0D 00
      TEMP(I)=P(I)*A(I,J)
   25 CONTINUE
      DO 40 I=1,JA
      ANS(J,I)=0.0D 00
      DO 40 K=1,IA
      ANS(J,I)=ANS(J,I)+A(K,I)*TEMP(K)
   40 CONTINUE
      RETURN
      END
```

SUBROUTINE : MCROSS

CALL STATEMENT : MCROSS (X, Y, R)

SUBROUTINE PURPOSE : The routine computes the cross product R of two input vectors X and Y.

INPUT PARAMETERS :

A. Vectors (extended precision).
  1. X is a 3-element input vector.
  2. Y is a 3-element input vector.

OUTPUT PARAMETERS :

A. Vector (extended precision).
  1. R is the cross product of the input vectors (3 elements).

PROGRAM DESCRIPTION : The routine computes $R = X \times Y$

SUBROUTINES REQUIRED : None

REFERENCES : None

MCROSS

```
      SUBROUTINE MCROSS(X,Y,R)
C     COMPUTES THE CROSS PRODUCT--R-- OF TWO MATRIX VECTORS --X & Y-.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSIONX(3),Y(3),R(3)
      R(1)=(X(2)*Y(3))-(X(3)*Y(2))
      R(2)=(X(3)*Y(1))-(X(1)*Y(3))
      R(3)=(X(1)*Y(2))-(X(2)*Y(1))
      RETURN
      END
```

SUBROUTINE : MEANAN

CALL STATEMENT : MEANAN (ET,. ANM, DNM)

SUBROUTINE PURPOSE : Calculates elements of the mean orbits of the moon and the sun and their time derivatives for a given epoch.

INPUT PARAMETERS :

A. ET is the epoch in Julian days.

OUTPUT PARAMETERS :

A. ANM is a 6-element vector (extended precision). All elements are in radians.
    1. ANM(1) is the longitude of the sun.
    2. ANM(2) is the longitude of the moon measured from the mean equinox along the ecliptic up to the ascending node of the lunar orbit and then along the orbit.
    3. ANM(3) is the longitude of perigee of the sun.
    4. ANM(4) is the longitude of perigee of the moon measured along the ecliptic and the lunar orbit as ANM(2).
    5. ANM(5) is the longitude of the ascending node of the mean lunar orbit.
    6. ANM(6) is the obliquity of the mean equator of the earth with respect to the ecliptic.

B. DNM is a 6-element vector (extended precision). All elements are in radians per day. Each element in DNM is the time derivative of the respective element in ANM.

PROGRAM DESCRIPTION :

A. The number of Julian days since Jan. 0.5, 1900 is computed from:

$$TD = ET - 2415020.0$$

B. The number of Julian centuries since Jan. 0.5, 1900 is computed from:

$$TC = TD/36525.0 = TD/DINC$$

C. The polynomial expressions for T^1, T^2, T^3 are formed in conjunction with 1/57.295· · · = 1/RADD as follows:

$$T(1) = 1./RADD$$
$$T(2) = T/RADD$$
$$T(3) = T^2/RADD$$
$$T(4) = T^3/RADD$$

D. The SSP matrix multiple routines are used to form the mean angle in ANM and their time derivatives in DNM:

$$ {}_6ANM_1 = {}_6CM_4\ {}_4T_1 $$
$$ {}_6DNM_1 = {}_6DC_4 \cdot {}_4T_1 \cdot \frac{1}{36525} $$

E. The ANM vector is examined using the subroutine REDPI to reduce all angles to the interval $(0 - 2\pi)$.

F. The DNM vector is multiplied by the time derivative $\dot{T}$ as below:

$$ DNM = DNM \left(\frac{1}{36252 \times 24 \times 60 \times 0.07436574} \right) $$

SUBROUTINES REQUIRED :

A. Fortran Scientific Subroutine Package:
   1. DGMPRD
B. O.S.U. Project Library:
   1. REDPI

REFERENCES :

Mendez, J. C. and R. J. Stern (1969). "Geographic and Selenodetic Coordinate Transformation Program," TRW Note No. 69-FMT-749, Project Apollo Task, NSC/TRW A-193.

MEANAN

```
      SUBROUTINE MEANAN (ET,ANM,DNM)
C     CALCULATES MEAN ANGLES (RADIANS) AND THEIR TIME DERIVATIVES
C     (RADIANS PER EPHEMERIS DAY)  ;  THE ANGLES ARE MEASURED FROM THE
C     MEAN EQUINOX OF DATE (ET)
C     ET - EPOCH IN JULIAN DAYS
C     THE CONSTANTS OF CM MATRIX GENERATE MEAN ANGLES AS FOLLOWS:
C     SUN LONG, MOON LONG, SUN PERIGEE LONG, MOON PERIGEE LONG,
C     MOON NODE LONG, EARTH EQUATOR OBLIQUITY
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION CM(6,4),DC(6,4),T(4),ANM(6),DNM(6)
      DATA CM   /279.6966778D0,270.4341639D0,281.2208333D0,334.3295556D0
     P,259.183275D0,23.4522944D0,36000.768925D0,481267.8831417D0,1.71917
     P50D0,4069.0340333D0,-1934.1420083D0,-.0130125D0,.3025D-03,-.113333
     P3D-02,.4527778D-03,-.010325D0,.2077778D-02,-.1638889D-05,.0D0,.188
     P889D-05,.333333D-05,-.125D-04,.22222D-05,.5027778D-06/,DC/36000.768
     P925D0,481267.8831417D0,1.719175D0,4069.0340333D0,-1934.1420083D0,-
     P.0130125D0,.605D-03,-.2266667D-02,.9055556D-03,-.02065D0,.4155556D-0
     P2,-.3277778D-05,.0D0,.56667D-05,1.D-05,-.375D-04,.66667D-05,.15083
     P3D-05,6*0.D0/,DINC   /36525.D0                          /,RADD/57.295779
     P5130823209D0/
      TD=ET -2415020.D0
      TC=TD/DINC
      T(1)=1.D0/RADD
      T(2)=TC*T(1)
      T(3)=TC*T(2)
      T(4)=TC*T(3)
      CALLDGMPRD(CM,T,ANM,6,4,1)
      CALLDGMPRD(DC,T,DNM,6,4,1)
      DO1I=1,6
      CALL REDPI (ANM(I))
    1 DNM(I)=DNM(I)/DINC
      RETURN
      END
```

SUBROUTINE : NUTATE

CALL STATEMENT : NUTATE (JD, DOBLQ, DLONG, DNUTA)

SUBROUTINE PURPOSE : Computes the nutation matrix for transformation from the mean to the true celestial Cartesian coordinate system of date.

INPUT PARAMETERS :

A. Scalars (extended precision).
   1. JD is the epoch in Julian days.
   2. DOBLQ is the nutation in obliquity in radians.
   3. DLONG is the nutation in longitude in radians.

OUTPUT PARAMETERS :

A. Matrices (extended precision).
   1. DNUTA is a 3 × 3 vector containing the nutation matrix.

PROGRAM DESCRIPTION : The nutation matrix N is formed from

$$N = R_1(-\epsilon - \Delta\epsilon)\, R_3(-\Delta\psi)\, R_1(\epsilon)$$

where ϵ is the mean obliquity (obtained from the subroutine MEANAN), $\Delta\epsilon$ is the nutation in obliquity and $\Delta\psi$ is the nutation in longitude.

SUBROUTINES REQUIRED :

O.S.U. Project Library:
   1. MEANAN
   2. TRIM

REFERENCES :

Mueller, Ivan I. (1969). "Spherical and Practical Astronomy as Applied to Geodesy," Frederick Ungar Publishing Co., New York.

NUTATE

```
      SUBROUTINE NUTATE(JD,DOBLQ,DLONG,DNUTA)
      IMPLICITREAL*8(A-H,O-Z)
      DOUBLE PRECISION JD
      DIMENSION ANM(6),DNM(6),RTE1(3,3),RE1(3,3),RDPSI(3,3),T1(3,3),
     *DNUTA(3,3)
C THIS ROUTINE COMP. THE NUTATION MATRIX FOR TRANSF. FRM MEAN TO TRUE
C     REQD.INPUT PARAM. --JULIAN DATE, NUTATION IN LONG AND OBLQ.
C     ROUTINES USED ARE MEANAN,TRIM
      CALLMEANAN(JD,ANM,DNM)
      OBLQ=ANM(6)
      TOBLQ=OBLQ+DOBLQ
      CALLROTATE(1,-TOBLQ,RTE1)
      CALLROTATE(1,OBLQ,RE1)
      CALLROTATE(3,-DLONG,RDPSI)
      CALLTRIM(RDPSI,RE1,T1)
      CALLTRIM(RTE1,T1,DNUTA)
      RETURN
   60 FORMAT(/,5X,4D25.16)
      END
```

SUBROUTINE : ONEM

CALL STATEMENT : ONEM (A, B, C)

SUBROUTINE PURPOSE : Multiplies a matrix A by a vector B resulting in a vector C.

INPUT PARAMETERS :

A. MATRIX A (3×3)
B. VECTOR B (3)

OUTPUT PARAMETERS :

A. VECTOR C (3)

PROGRAM DESCRIPTION :

A. ${}_3C_1 = {}_3A_3 \, {}_3B_1$

SUBROUTINES REQUIRED : None

REFERENCES : None

ONEM

```
C
      SUBROUTINE ONEM (A,B,C)
C     PRODUCT OF MATRIX (A) AND VECTOR (B) RESULTS IN VECTOR (C)
      REAL*8A(3,3),B(3),C(3)
      C(1)=A(1,1)*B(1)+A(1,2)*B(2)+A(1,3)*B(3)
      C(2)=A(2,1)*B(1)+A(2,2)*B(2)+A(2,3)*B(3)
      C(3)=A(3,1)*B(1)+A(3,2)*B(2)+A(3,3)*B(3)
      RETURN
      END
C
```

SUBROUTINE : OPTOBS

CALL STATEMENT : OPTOBS (ET, X, JOK, APP, BUNDLE).

SUBROUTINE PURPOSE : The subroutine generates a bundle of rays simulating optical observations from a point in space exterior to the lunar surface (e.g., the earth or a satellite) to an array of 30 points defined in [Papo, 1971] Pages 162-168.

INPUT PARAMETERS :

A. Scalars.
   1. ET is an extended precision variable which contains the Julian date at which the observations are to be simulated.
   2. JOK is a single precision integer indicating the number of the earth observatory from which the optical observation was made (JOK = 1, 2, 3) or JOK = 9 for satellite based observations.
   3. APP is an extended precision variable which contains the sine of one-half of the field angle of the camera used.

B. Vector (extended precision).
   1. X is a 9-element vector which represents:
      a. The geocentric position of the moon or the selenocentric position of the satellite (as applicable).
      b. The Eulerian angles in radians of the senocentric system (elements 4-6).
      c. The Eulerian angles in radians of the mean terrestrial system (elements 7-9). All X quantities are referred to the inertially oriented coordinate systems.

OUTPUT PARAMETERS :

A. Matrix (extended precision).
   1. BUNDLE is a 2 x 30 matrix which contains 30 angular observations in radians of lunar points from a projection center. The first index indicates the angle υ and the second index indicates the angle μ referred to the B_1, B_2, B_3 reference system as described in [Papo, 1971] Pages 164-165 and Figure 4.7.

COMMON AREA PARAMETERS :

A. Area /OPTO/TB, WTER, SSUN, ETER, PCSP, ST1, ST2, ST3, IFLAG.

1. TB is an extended precision 3 × 3 array supplied by the main or calling program and is used to transform coordinates from the simulated inertial system to the B_1, B_2, B_3 system defined in [Papo, 1971].
2. WTER is an extended precision variable evaluated in the subprogram and is the selenodetic longitude of the terminator west of the subsolar point in degrees.
3. SSUN is an extended precisoin variable evaluated in the subprogram and represents the selenodetic longitude of the subsolar point in degrees.
4. ETER is an extended precision variable evaluated in the subprogram and is the selenodetic longitude of the terminator east of the subsolar point in degrees.
5. ST1 is an extended precision variable which is supplied by the main program and is used to check if a particular point is at least 5 degrees from the terminator and on the lighted side of the disk of the moon.
6. ST2 is a variable similar to ST1 and is used to check if the point on the moon as seen from the projection center is at least 20° from the limb of the moon.
7. ST3 is a variable similar to ST1 and ST2 and is used to check if the lunar point is at least 20° above the station horizon (in the case of earth based observations).
8. IFLAG is a single precision integer used as a flag by the subroutine. The flag is set to one if the observing station is in daylight and no observations are made. The flag is set to two if the observing station is in darkness and, in general, observations can be made (subject to other checks).

PROGRAM DESCRIPTION : Data storage is used to input the mean terrestrial coordinates (UVW in kilometers) in the array STAOPT and the selenocentric coordinates of the moon (XYZ in kilometers) in the array PM. Program formulation then represents the theoretical development given in [Papo, 1971]. Appropriate checks for observability are indicated with comment cards.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. ROTATE
   2. TRIM
   3. ONEM
   4. REDPI

B. Fortran Scientific Subroutine Package:
   1. DGMTRA

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science Report No. 156.

OPTOBS

```
      SUBROUTINE OPTOBS (ET,X,JOK,APP,BUNDLE)
C     SUBROUTINE FOR GENERATING OPTICAL OBSERVATION OF TH  MOON
C     X (1 - 3) GEOCENTRIC MOON POSITION OR SELENOCENTRIC POSITION OF SATELLITE
C       (4 - 6) EULERIAN ANGLES OF MOON SYSTEM
C       (7 - 9) EULERIAN ANGLES OF EARTH SYSTEM
C       EULERIAN ANGLES : LONGITUDE NODE INCLINATION
C     X ARE RELATED TO THE ECLIPTIC SYSTEM
C     BUNDLE -      ANGULAR COORDINATES OF OBSERVATION RAYS IN B1-B2-B3 SYSTEM
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSIONEAB(2),OBU(3),T1(3,3),T2(3,3),T3(3,3),STAOPT(3,3),PCSP(3)
      DIMENSION SUN(2),OB(3),TR(3),BV(3),PC(3),TM(3,3),TB(3,3),PM(3,30),     2*P
     PCM(3,30),BUNDLE(2,30),X(9)                                              1
      COMMON /OPTO/ TB, WTER,SSUN,ETER,PCSP,ST1,ST2,ST3,IFLAG
C     TM - TRANSFORMATION MATRIX FROM SELENOGRAPHIC TO ECLIPTIC SYSTEM
C     PM - SELENOGRAPHIC CARTESIAN COORDINATES OF POINTS ON THE MOON
C     PC - SELENOCENTRIC ECLIPTIC CARTESIAN COORDINATES OF PROJECTION CENTER
C     TB - TRANSFORMATION MATRIX FROM ECLIPTIC TO B1-B2-B3 SYSTEM
C     OB - UNIT VECTOR OF EARTH OBSERVATORY (GEOCENTRIC ECLIPTIC SYSTEM)
C     SUN - UNIT VECTOR OF SUN RAYS IN ECLIPTIC SYSTEM (SUN AT INFINITY)
C
      DATA  STAOPT/-1996.0051, -5042.6961,3360.7748,4686.1252, 11.6385,4
     P331.0499,5058.2628, 2698.0251,-2799.8019  /
      DATA PM /284.7144 ,-164.8233 ,1705.6738 ,267.3828 ,742.4968 ,1546.    14*P
     P1653 ,298.9995 ,-1050.4536 ,1350.2634 ,950.9536 ,-484.8089 ,1369.       1
     P1476 ,897.8094 ,293.5282 ,1457.1998 ,297.0151 ,1298.3423 ,1116.         2
     P7341 ,1013.5746 ,-1080.6066 ,894.6916 ,1453.2020 ,-75.2901 ,946.        3
     P1413 ,1076.7970 ,844.2392 ,1069.5859 ,341.1061 ,-1615.9158 ,536.        4
     P6048 ,1506.4731 ,-767.8602 ,390.4892 ,1577.2501 ,544.9516 ,478.         5
     P5874 ,341.1861 ,1617.9158 ,536.6048 ,1157.6880 ,-1287.5193 ,120.        6
     P7443 ,1735.3430 ,1.0 ,29.8342 ,1106.4064 ,1322.5432 ,211.3220 ,         7
     P1726.6207 ,-150.2785 ,-91.4655 ,490.5786 ,-1611.7874 ,-420.9863 ,       8
     P1499.1838 ,733.4201 ,-479.5874 ,294.7239 ,1686.6973 ,-302.3192 ,        9
     P1211.2688 ,-1091.8823 ,-594.9678 ,1511.8965 ,-131.4927 ,-843.1513,     10
     P1004.3727 ,1244.3842 ,-679.6328 ,257.4499 ,-1221.9682 ,-1207.8911,     11
     P1155.7211 ,-614.8371 ,-1140.8012 ,1055.5706 ,741.8690 ,-1163.5211,     12
     P449.7967 ,1120.4735 ,-1250.7900 ,940.5390 ,63.5052 ,-1458.1998 ,       13
     P377.2905 ,-378.7905 ,-1653.5386 ,305.4515 ,368.0022 ,-1671.2764 /      14
C
      IFLAG=2
C     JOK=1 MEANS OBSERVATORY NO 1 IS OBSERVING AND SAME FOR NOS 2 AND 3
C     JOK=9 MEANS THE STATION IS ON A SATELLITE
      DO 12 I=1,3
   12 PC(I)=X(I)
C
      SAN=1.7399359D0+.0172027913D0*(ET+24980.D0)
      SSUN=SAN-X(4)-X(5)-3.1416
      CALL REDPI (SSUN)
      SSUN=SSUN*57.2958
      ETER=SSUN+90.
      WTER=SSUN-90.
```

```
      IF(ETER.GT.360.D0)ETER=ETER-360.D0
      SUN(1)=DCOS(SAN)
      SUN(2)=DSIN(SAN)
C
      IF(JOK-9) 14,13,11
   14 DO15 J=1,3
   15 OBU(J)=STAOPT(J,JOK)
      CALL ROTATE (3,-X  (7),T1)
      CALL ROTATE (1, X (9),T2)
      CALL TRIM (T2,T1,T3)
      CALL ROTATE (3,- X (8),T1)
      CALL TRIM (T1,T3,T2)
      CALL ONEM (T2,OBU,OB)
      DO 10 J=1,3
   10 PC(J)=OB(J)-PC(J)
      GOTO19
C
   13 DO16 J=1,3
   16 OB(J)=-PC(J)
   19 CALL ROTATE (3,- X (4),T1)
      CALL ROTATE (1, X (6),T2)
      CALL TRIM (T2,T1,T3)
      CALL ROTATE (3,- X (5),T1)
      CALL TRIM (T1,T3,TM)
      CALL DGMTRA (TM,T1,3,3)
      CALL ONEM (T1,PC,PCSP)
C
      EAB(1)=DATAN2(-PC(2),-PC(1))
      EAB(2)=DATAN2(-PC(3),DSQRT(PC(1)**2+PC(2)**2))
      CALL ROTATE (3, EAB(1),T3)
      CALL ROTATE (2,-EAB(2),T1)
      CALL TRIM (T1,T3,TB)
C
      SB=DSQRT(OB(1)**2+OB(2)**2+OB(3)**2)
      CH1=(SUN(1)*OB(1)+SUN(2)*OB(2))/SB
      IF(CH1-ST3) 1,1,2
C     CH1 - IS IT NIGHT AT EARTH OBSERVATORY
    2 CALL DGMPRD (TM,PM,CM,3,3,30)
C
      DO 3 I=1,30
      DO 5 J=1,3
    5 TR(J)=CM(J,I)-PC(J)
      SC=DSQRT(CM(1,I)**2+CM(2,I)**2+CM(3,I)**2)
      CH2=(CM(1,I)*SUN(1)+CM(2,I)*SUN(2))/SC
C     CH2 - IS THE MOON POINT ILLUMINATED
      SU=DSQRT(TR(1)**2+TR(2)**2+TR(3)**2)
      CH3=(TR(1)*CM(1,I)+TR(2)*CM(2,I)+TR(3)*CM(3,I))/(SU*SC)
C     CH3 - IS THE MOON POINT AT LEAST 20 DEG FROM THE LIMB
      CH4=(OB(1)*TR(1)+OB(2)*TR(2)+OB(3)*TR(3))/(SU*SB)
C     CH4 - IS THE MOON POINT AT LEAST 20 DEG ABOVE THE HORIZON OF EARTH OBSERV.
      CALL ONEM (TB,TR,BV)
```

OPTOBS (Cont)

```
      R=DSQRT(BV(1)**2+BV(2)**2+BV(3)**2)
      BUNDLE(1,I)=DARCOS(BV(1)/R)
      BUNDLE(2,I)=DATAN2(BV(3),BV(2))
      IF(CH2.GT.ST1.OR.CH3.GT.ST1.OR.CH4.LT.ST3.OR.DABS(BV(2)/R    ).GT.
     PAPP.OR.DABS(BV(3)/R    ).GT.APP)GOTO 4
C     ST1 = -SIN( 5 DEG)  SEE SECTION 4.61
C     ST2 = -SIN (20 DEG)
C     ST3 = SIN (20 DEG)
C
      GOTO 3
    4 BUNDLE(1,I)=0.D0
      BUNDLE(2,I)=0.D0
    3 CONTINUE
C
      RETURN
    1 IFLAG=1
C     IFLAG=1 MEANS THE OBSERVATORY ON EARTH CAN NOT OBSERVE. IT IS DAYTIME
      WRITE(6,76)JOK,ET
      RETURN
   11 WRITE(6,77)ET
      RETURN
   76 FORMAT(1H1 ///,5X,'   EARTH OBSERVATORY NO  ',I3,'  CAN NOT OBSERV
     PE . IT IS DAYTIME  ',F20.8)
   77 FORMAT(5X,'   IMPROPER VALUE GIVEN TO JOK   ',F20.8)
      END
```

SUBROUTINE : OUTADJ

CALL STATEMENT : OUTADJ (ET, Y, YP, IHL, K, PAR)

SUBROUTINE PURPOSE : OUTADJ is the program used to adjust the numerically integrated physical libration angles to Eckhardt's theory [Eckhardt, 1970] as described in [Papo, 1971] Chapter 3.

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. ET is the epoch of interest in Julian days.

B. Vectors (extended precision).
   1. Y and YP are vectors described in the documentation of subroutine FUNPL5.

C. Variables IHK, K, PAR are dummy variables not used in the subroutine.

OUTPUT PARAMETERS : Variables Parameters are output from the subroutine in the common block area MAIOUT in the variables CNORM and CVECT.

COMMON AREA PARAMETERS : Three common areas MAIOUT, MAIF, and ALL are used. The areas MAIF and ALL are described in the FUNPL5 Program Description.

A. Area /MAIOUT/CNORM, CVECT.
   1. CNORM is an extended precision 9 ×9 matrix which contains the normal matrix formed in the subroutine (e.g. each set of observations creates an additional "layer" which increments the normal matrix).
   2. CVECT is an extended precision vector of 9 elements which contains the constant vector corresponding to the normal matrix in CNORM.

PROGRAM DESCRIPTION : Section 3.42 [Papo, 1971] describes the theory used in developing the subroutine. The subroutine is called by the DVDPF series of routine at a desired epoch to evaluate Eckhardt's formulation and form the necessary normal matrix and constant vector for adjustment. Basically the program accomplishes the following computations for each epoch of interest.

A. The differences in the numerically integrated angles and angles computed from Eckhardt's theory are computed.

B. The differences computed above are transformed as described in [Papo, 1971] Equation 3.42.

C. Appropriate portions of the U and Q matrices (Papo's notation) are formed from the input Y vector (elements 7 to 42 and 43 to 60 respectively) as formulated in [Papo, 1971] Equation 3.42.7.

D. The above computations are then used to increment the normal matrix and constant vector (CNORM and CVECT) and control is returned to the calling program.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
    1. EKHARD'
    2. ONEM

B. Fortran Scientific Subroutine Package:
    1. DGMPRD
    2. DGMTRA

REFERENCES :

A. Eckhardt, D. H. (1970). "Lunar Libration Tables," The Moon, Vol. 1, No. 2, February.

B. Papo, Haim B. (1970). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

OUTADJ

```fortran
      SUBRCUTINE OUTADJ (ET,Y,YP,IHL,K,PAR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(42),YP(42),PAR(5),Q(6),PHL(3),DIF(3),DD(3),EM(3,3),
     PU(3,6),UEM(6,3),COW(6,6),CNORM(6,6),DOW(6),CVECT(6),GOG(3),DEN(6)
     P,UUM(3,6),DUM(6)
      COMMON/MA1OUT/CNORM,CVECT,DEN,ROM/ALL/Q
      DATA EM(3,1),EM(1,1),EM(2,1),EM(3,3) /1.D0,3*0.D0/
      IF(NUT.EQ.1)PAR(5)=1.D0
      TEST=(ET-PAR(1))/PAR(3)
      IF(DMOD(TEST,1.D0).GT.1.D-05) GOTO 99
      STE=DSIN(Q(3))
      SFI=DSIN(Q(1))
      CFI=DCOS(Q(1))
      EM(1,2)=-SFI*STE
      EM(1,3)=-CFI
      EM(2,2)=-CFI*STE
      EM(2,3)=SFI
      EM(3,2)=DCOS(Q(3))-1.D0
      DO 1 IFL=1,6
      KAP=(IFL-1)*6
      DO 1 I=1,3
    1 U(I,IFL)=(Y(KAP+I)-Y(36+I))/DEN(IFL)
      CALL EKHARD (ET+2440000.D0,DUM,GOG)
      DO 2 J=1,3
    2 DIF(J)=Y(36+J)-GOG(J)*.484813681109536D-05
      CALL ONEM (EM,DIF,DD)
      CALL DGMPRD (EM,U,UUM,3,3,6)
      CALL DGMTRA (UUM,UEM,3,6)
      CALL DGMPRD (UEM,UUM,COW,6,3,6)
      CALL DGMPRD (UEM,DD,DOW,6,3,1)
      DO 3 I=1,3
    3 ROM=ROM+DD(I)**2
      DO 4 I=1,6
      CVECT(I)=CVECT(I)+DOW(I)
      DO 4 J=1,6
    4 CNORM(I,J)=CNORM(I,J)+COW(I,J)
   99 CONTINUE
      RETURN
      END
```

SUBROUTINE : OUTEPH

CALL STATEMENT : OUTEPH (ET, Y, YP, IHALVE, K, PAR)

SUBROUTINE PURPOSE : Print routine for output of the intermediate results of the simulated ephemeris.

INPUT PARAMETERS :

- A. Integer dummy variables (single precision).
    1. IHALVE
    2. K
- B. Scalars (extended precision).
    1. ET is the desired epoch in Julian days minus 2440000.0.
- C. Vectors (extended precision).
    1. Y is an 18-element vector containing the same variables as described in the Y vector of FUNEPH.
    2. YP is an 18-element vector containing the same variables as the YP vector in FUNEPH.
    3. PAR is a 5-element vector not used in OUTEPH, but described in the MAIN program for simulated ephemeris generation.

OUTPUT PARAMETERS : The subroutine prints out ET and selected elements of the Y vector. The selected elements are the XYZ of the state vector, the Eulerian angles of the moon and the Eulerian angles of the earth.

PROGRAM DESCRIPTION : None

SUBROUTINES REQUIRED : None

REFERENCES : None

Preceding page blank

OUTEPH

```
C
      SUBROUTINE OUTEPH (ET,Y,YP,IHALVE,K,PAR)
C     WATCH OUT NO K VARIABLE PERMISSIBLE IN THIS SUBROUTINE
C     OUTPUT OF SIMULATED EPHEMERIS
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONY(18),YP(18),PAR(5)
      WRITE(6,70)ET,(Y(I),I=1,3),(Y(I),I=7,9),(Y(I),I=13,15)
      WRITE(1) Y
      RETURN
   70 FORMAT (1X,F6.1,3(1X,3D12.5))
      END
```

SUBROUTINE : OUTGAJ

CALL STATEMENT : OUTGAJ (ET, Y, N2, EM3, A1, A1M3, A3, AZ).

SUBROUTINE PURPOSE : The subroutine is used to process simulated earth based optical observation of the moon for solution of lunar coordinates and physical parameters of the moon (initial values of the physical libration parameters and C_{22}, β, C_{20}). The subroutine is called by the DVDPF2 routine at epochs where optical bundles have been observed.

INPUT PARAMETERS :

A. Scalars (extended precision).
  1. ET is the epoch in Julian days at which the optical observation has been taken.

B. Integer (single precision).
  1. N2 is twice the number of optical rays in the bundle.

C. Vector (extended precision).
  1. Y is a 60-element vector input to the subroutine by either the FUNPL5 or FUNPL6 subroutines.

D. Matrices (extended precision).
  1. EM3 through AZ are matrices used within OUTGAJ and are included in the input parameter set to utilize object time dimension facilities of IBM Fortran.

DATA SET INPUT : The subroutine requires binary input from a computer device (either magnetic tape or disk) which in the IBM data device desinator, is named unit 3. The input consists of the following parameters:

A. KOK is the serial number of the bundle of rays to be processed (integer, single precision).

B. ETO is the epoch at which the bundle was "observed" in Julian days (extended precision).

C. IST is the identification number of the bundle.

D. POS is a 3-element extended precision vector containing the position of the projection center in the simulated selenocentric inertially oriented coordinate system (XYZ) in kilometers.

E. EMBT is an extended precision transformation matrix (3 x 3) for coordination transformation from the XYZ to the B_1 B_2 B_3 system [Papo, 1971] Chapter 2.

F. ENU and EKA are extended precision 22-element vectors containin the simulated optical observations.

G. IPO is the identification number of the control point being observed.

DATA SET OUTPUT : As described below, the subroutine essentially forms and increments a normal matrix and constant vector for adjustment of optical observations. The "output" of the routine (e.g. the incremented normal matrix and constant column) are output on a data set device identified as K3 which is provided through the common area MAIOUT. The output is written in binary mode.

COMMON PARAMETERS : Common areas are used extensively to transfer data in and out of the subroutine.

A. Area /MAIOUT/POS, EMBT, ENU, EKA, IPO, TJ, SIG6, SIG7, SGGZ, K3, K2, IST, MM.
   1. The parameters POS through IPO are described in the Data Set Input section.
   2. TJ is an extended precision matrix of (3 × 22) elements, and contains the Cartesian coordinates of the control points in the selenodetic coordinate system (in km).
   3. SIG6 is an extended precision 3 × 3 array containing the a priori covariance matrix of the orientation elements of the optical bundle of rays (in radians squared).
   4. SIG7 is an extended precision scalar which is the a priori variance (in radians squared) of the optical observations relative to the $B_1 B_2 B_3$ coordinate system.
   5. SGGZ is a 3 × 3 extended precision array containing the covariance matrix of the projection center (km squared) in the selenocentric inertially oriented system.
   6. K3 is a single precision integer which identifies the number of the data set device to be used for output.
   7. K2 is a single precision integer identifying a data set device to be used for temporary storage.
   8. IST is a single precision integer which is not used in the program.
   9. MM is a single precision integer containing the number of the first optical bundle to be processed from bundles sequentially stored on unit 3.

B. Area /DVOUT/ETO, KM, ICW, N, ISKIP.
   1. ETO is an extended precision variable containing the epoch in Julian days of an optical bundle read from unit 3.

2. KM is a single precision integer containing the total number of bundles stored on unit 3.
3. ICW is a single precision integer containing the sequential number of the optical bundle being processed.
4. N is a single precision integer containing the number of rays in the next bundle.
5. ISKIP is a single precision integer used as a flag to indicate that some of the sequentially stored optical data are not to be processed ⌈Papo, 1971⌉ Page 204.

C. Area /ALL/. The common area /ALL/ is input from the subroutine FUNPL6 and the parameters are described in FUNPL5 documentation.

PROGRAM DESCRIPTION : The subroutine formulation follows the theoretical development given in ⌈Papo, 1971⌉, particularly Sections 2.52 and 2.6 (Pages 59, 68-70). Basically the logic of the program can be divided into four steps:

A. The calling subroutine DVDPF2 supplies the current physical libration angles and the state transition and parameter sensitivity matrices as input parameters.
B. From a previous step of integration (e.g. the last call of OUTG the optical data and auxillary information has been read from unit 3 and stored in common storage.
C. Calculations according to the theory of ⌈Papo, 1971⌉ Chapter 2 are accomplished and the increment of the normal natrix, etc. are added to the accumulated results and stored on a disk unit (direct access device).
D. The optical and auxillary data of the next, sequentially stored, bundle of rays is read from unit 3 and control is returned to the calling subroutine DVDPF2.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. REDPI
   2. ROTATE
   3. TRIM
   4. LUCA
   5. ONEM
B. Fortran Scientific Subroutine Package:
   1. DGMTRA
   2. DGMPRD
   3. DMINV

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

OUTGAJ

```
      SUBROUTINE OUTGAJ (ET,Y  ,N2,EM3,A1,A1M3,A3,AZ)
      IMPLICITREAL*8 (A-H,O-Z)
      DIMENSION Y(60),SEV(6),Z(3),UQ(3,9),TE1(3,3),TE2(3,3),TE3(3,3)
     P,TEM1(3,3),EMM(3,3),EMB(3,3),EMBT(3,3),EM3(N2,N2),A1(N2,75),U(3)
     P,IPO(22),ENU(22),EKA(22),U5(3),TJ(3,22),TEM4(3,3),POS(3),EMUM(3,3)
     P,EMT(3),B6(2,3),ETT(2,3),TEM2(2,3),B23(2,9),TEM5(2,3),B10(2,3)
     P,SIGZ(3,3),EMZ(2 ,2),B1(2,3),W(2),A3(N2, 3),AZ(N2, 2),LOM(44),MOL
     P(44),SIG6(3,3),A1M3(75,N2),UU(44),TEM3(3),VEN2(44),VEC(75),N(31)
      COMMON/MAIOUT/ POS,EMBT,ENU,EKA,TJ,SIG6,SIG7,SIGZ,K3,K2,IPO,1ST,MM
     P/DVOUT/ ETO,KM ,ICW,N,ISKIP /ALL/ SEV,ROM,NUT
      DATA PI /3.141592653589793D0/
      IF(ICH.LT.MM) GOTO 17
      WRITE(6,73) ET,ETO,ICW,N1,N2
      IF(ICH.GT.KM) GOTO 99
C     BEGIN CALCULATIONS FOR THE PRESENT BUNDLE OF DIRECTIONS
      DO 1 J=1,9
      JJ=J*6
      DO 2 I=1,3
    2 UQ(I,J)=Y(JJ+I)
    1 UQ(1,J)=UQ(1,J)-UQ(2,J)
C     UQ IS THE MATRIX OF PARTIALSFOR PH. L. (TRANSITION AND SENSITIVITY)
      Z(1)=Y(1)-Y(2)+SEV(1)-SEV(2)+PI+(SEV(3)-SEV(4))*(ET-222.5D0)
      CALL REDPI (Z(1))
      Z(2)=Y(2)+SEV(2)+SEV(4)*(ET-222.5D0)
      CALL REDPI (Z(2))
      Z(3)=Y(3)+.026769D0
C     Z ARE THE EULERIAN ANGLES OF THE MOON  1 - PHI  2 - PSI  3 - TETA
      CALL ROTATE (3,-Z(1),TE1)
      CALL ROTATE (1,Z(3),TE2)
      CALL ROTATE (3,-Z(2),TE3)
      CALL TRIM (TE2,TE1,TEM1)
      CALL TRIM (TE3,TEM1,EMM)
C     EMM IS THE TRANSFORMATION MATRIX FROM SELENODETIC TO INERTIAL
      CALL DGMTRA (EMBT,EMB,3,3)
      CALL LUCA (1,-1.D0,TE1)
      CALL TRIM (TE3,TE1,TE2)
      CALL TRIM (TE2,TEM1,TE3)
      CALL LUCA (3,1.D0,TEM1)
      CALL TRIM (EMM,TEM1,TE1)
      CALL TRIM (TEM1,EMM,TE2)
C     TE1,TE2,TE3, ARE NEEDED FOR THE MATRIX OF PARTUALS B23
      DO 12 I=1,66
      DO 12 J=1,N2
   12 A1(J,I)=0.D0
C     INITIALIZING PART OF THE A1 MATRIX
C     THIS COMPLETES MATRICES COMMON TO ALL THE POINTS IN THE BUNDLE
      L=N2/2
      DO 13 I=1,L
      M=IPO(I)
      SN=DSIN(ENU(I))
      CN=DCOS(ENU(I))
```

```
      SK=DSIN(EKA(I))
      CK=DCOS(EKA(I))
      U(1)=CN
      U(2)=SN*CK
      U(3)=SN*SK
C     U ARE THE DIRECTION COSINES OF RAY (I) IN THE B1B2B3 SYSTEM
      CALL ONEM (EMB,U,UB)
      DO 3 II=1,3
      TEM4(II,1)=TE1(II,1)*TJ(1,M)+TE1(II,2)*TJ(2,M)+TE1(II,3)*TJ(3,M)
      TEM4(II,2)=TE2(II,1)*TJ(1,M)+TE2(II,2)*TJ(2,M)+TE2(II,3)*TJ(3,M)
      TEM4(II,3)=TE3(II,1)*TJ(1,M)+TE3(II,2)*TJ(2,M)+TE3(II,3)*TJ(3,M)
C     TEM4 IS THE MATRIX IN BRACKETS FOR B23
C     TJ ARE THE "OBSERVED" COORDINATES OF THE MOON POINTS
      TEM3(II)=-TJ(II,M)
      DO 3 J=1,3
      EMUM(II,J)=UB(II)*UB(J)
      IF(II.EQ.J) EMUM(II,II)=EMUM(II,II)-1.D0
    3 CONTINUE
C     EMUM IS THE EXPRESSION IN BRACKETS FOR B1 ( NEGATIVE)
      CALL ONEM (EMM,TEM3,EMT)
      DO 11 J=1,3
   11 EMT(J)=EMT(J)+POS(J)
C     EMT IS POSITION OF PROJ. CENTR WITH RESP. TO OBSERVED PNT IN INERTIAL COMP
      RO=DSQRT(EMT(1)**2+EMT(2)**2+EMT(3)**2)
      B6(1,1)=-CN*(CK*U(3)-SK*U(2))
      B6(1,2)=-SN*U(3)-CN*SK*U(1)
      B6(1,3)= SN*U(2)+CN*CK*U(1)
      B6(2,1)= SK*U(3)+CK*U(2)
      B6(2,2)=-CK*U(1)
      B6(2,3)=-SK*U(1)
C     B6 IS THE MATRIX OF PARTIALS FOR THE E ORIENTATION PARAMETERS
      ETT(1,1)=-SN/RO
      ETT(1,2)= CN*CK/RO
      ETT(1,3)= CN*SK/RO
      ETT(2,1)= 0.D0
      ETT(2,2)=-SK/RO
      ETT(2,3)= CK/RO
C     ETT IS THE ET/RO MATRIX
      CALL DGMPRD (ETT,EMBT,TEM2,2,3,3)
C     TEM2 IS ET*MBT/RO
      CALL DGMPRD (TEM2,TEM4,TEM5,2,3,3)
      CALL DGMPRD (TEM5,UQ,B23,2,3,9)
      CALL DGMPRD (TEM2,EMUM,B10,2,3,3)
      CALL DGMPRD (B10,SIGZ,TEM5,2,3,3)
      DO 4 K=1,2
      DO 4 J=1,2
    4 EMZ(K,J)=TEM5(K,1)*B10(J,1)+TEM5(K,2)*B10(J,2)+TEM5(K,3)*B10(J,3)
      CALL DGMPRD (B10,EMM,B1,2,3,3)
C     B1 IS THE MATRIX OF PARTIALS FOR THE TRIG POINTS ON THE MOON
      DO 6 K=1,2
      W(K)=0.D0
      DO 5 J=1,3
```

```
      B1(K,J)=B1(K,J)*206264.8062D0
      B10(K,J)=B10(K,J)*206264.8062D0
    5 W(K)=W(K)+TEM2(K,J)*(UB(J)+EMT(J)/RO)
    6 W(K)=W(K)*RO*206264.8062D0
C     W IS THE FREE TERM
C     CALCULATIONS OF B1 , B23 , B6 , B10 ARE COMPLETED FOR POINT M
C     THE SUBMATRICES CREATED ARE STORED IN A1 , A3 , UU AND AZ
      JJ=(M-1)*3
      DO 10 K=1,2
      II=(I-1)*2+K
      UU(II)=W(K)
      DO 7 J=1,3
      A3(II,J)=B6(K,J)
    7 A1(II,JJ+J)=B1(K,J)
      DO 8 J=1,9
    8 A1(II,66+J)=B23(K,J)
      DO 9 J=1,2
    9 AZ(II,J)=EMZ(K,J)
   10 CONTINUE
   13 CONTINUE
      WRITE(6,74) UU
      DO 15 I=1,N2
      DO 15 J=1,N2
      EM3(I,J)=0.D0
      DO 14 K=1,3
   14 EM3(I,J)=EM3(I,J)+A3(I,K)*(A3(J,1)*SIG6(K,1)+A3(J,2)*SIG6(K,2)+
     PA3(J,3)*SIG6(K,3))
      IF(I.EQ.J) EM3(I,J)=EM3(I,J)+SIG7
   15 CONTINUE
      WRITE(6,74)(EM3(I,I),I=1,N2)
      FAT=0.D0
      DO 16 LO=1,N2
   16 FAT=FAT+DLOG10(EM3(LO,LO))
      FAT=1.D0/10.D0**(FAT/DFLOAT(N2))
      DO 22 LO=1,N2
      DO 22 LP=1,N2
   22 EM3(LO,LP)=EM3(LO,LP)*FAT
      CALL DMINV (EM3,N2,DET,LOM,MOL)
      DO 24 LO=1,N2
      DO 24 LP=1,N2
   24 EM3(LO,LP)=EM3(LO,LP)*FAT
      TEST=0.D0
      DO 25 LO=2,N2
      LC=LO-1
      DO 25 LP=1,LC
   25 TEST=TEST+(EM3(LP,LO)-EM3(LO,LP))**2
      WRITE(6,74) DET,FAT,TEST
      WRITE(6,74)(EM3(I,I),I=1,N2)
C     MATRIX M3 IS INVERTED
      DO 19 I=1,75
      DO 19 J=1,N2
```

```
      A1M3(I,J)=0.D0
      DO 18 K=1,N2
   18 A1M3(I,J)=A1M3(I,J)+A1(K,I)*EM3(K,J)
   19 CONTINUE
      REWIND K3
      REWIND K2
      DO 21 I=1,75
      READ(K3) VEC
      DO 20 J=1,75
      DO 20 K=1,N2
   20 VEC(J)=VEC(J)+A1M3(I,K)*A1(K,J)
   21 WRITE(K2) VEC
      READ (K3) VEC
      DO 23 J=1,75
      DO 23 K=1,N2
   23 VEC(J)=VEC(J)+A1M3(J,K)*UU(K)
      WRITE(K2) VEC
      DO 28 IIK=1,75
      READ(K3) VEC
      DO 27 M=1,L
      KA1=(M-1)*2+1
      KA2=(M-1)*2+2
      VEN2(KA1)=A1M3(IIK,KA1)*AZ(KA1,1)+A1M3(IIK,KA2)*AZ(KA2,1)
      VEN2(KA2)=A1M3(IIK,KA1)*AZ(KA1,2)+A1M3(IIK,KA2)*AZ(KA2,2)
   27 CONTINUE
      DO 29 M1=1,75
      DO 29 M=1,N2
   29 VEC(M1)=VEC(M1)+VEN2(M)*A1M3(M1,M)
      WRITE(K2) VEC
   28 CONTINUE
      IAX=K3
      K3=K2
      K2=IAX
      IF(ICW.GE.KM) GOTO 99
      IF(ISKIP.EQ.0) GOTO 17
      DO 707 I=1,ISKIP
      READ(3,END=99) KOK
  707 ICW=ICW+1
   17 N1=N(ICW+1)
      READ(3,END=99) KOK,ETO,IST,POS,EMBT,(IPO(I),ENU(I),EKA(I),I=1,N1)
      WRITE(6,73) ET,ETO,ICW,N1,N2,KOK
      IF((KOK-ICW).NE.1) GOTO 98
      RETURN
   98 WRITE(6,71) ICW,KOK
      NUT=1
      RETURN
   99 NUT=1
      RETURN
   71 FORMAT(//10X,2I5)
   73 FORMAT(/5X,2F12.6,4I5)
   74 FORMAT(/(5X,12D9.2))
      END
```

SUBROUTINE : PLADIS

CALL STATEMENT : PLADIS (ET, EX, XM, XCE, X, D, PE, PM, EP, EL, SE, SM, PSM, B1, B2)

SUBROUTINE PURPOSE : Computation of partial derivatives of simulated laser distances with respect to the considered parameters as outlined in [Fajemirokun, 1971] Chapter 5.

INPUT PARAMETERS :

A. Parameters ET, XE. XM, XCE, X, D, PE and PM are described in the subroutine LADIS Program Description.

B. EP is a 6-element extended precision vector representing the integrated Eulerian earth angles [Fajemirokun, 1971] Page 164.

C. EL is a 6-element extended precision vector representing the integrated Eulerian moon angles [Fajemirokun, 1971] Page 192.

D. SE is a 3x6 extended precision array which contains the state transition matrix of the earth system as described in [Fajemirokun, 1971] Page 64.

E. SM is a 3x6 extended precision array containing the state transition matrix for the moon.

F. PSM is a 3x3 extended precision array containing the state transition matrix of the dynamic parameters of the moon.

OUTPUT PARAMETERS :

A. B1 is a 21 element extended vector containing:
    1. Partials with respect to the earth station position (1-3).
    2. Partials with respect to the earth Eulerian angles (4-9).
    3. Partials with respect to the lunar station position (10-12)
    4. Partials with respect to the lunar Eulerian angles (13-18).
    5. Partials with respect to the dynamic (C_{20}, C_{23}, β) parameters of the moon (19-21).

B. B2 is a 3-element extended precision vector containing the partials with respect to the geocentric coordinates of the center of mass of the moon.

PROGRAM DESCRIPTION : The program computes the partial derivatives formulated in [Fajemirokun, 1971] and in the order given above.

SUBROUTINES REQUIRED :

- A. O.S.U. Project Library:
  - 1. LUCA
  - 2. RATATE
  - 3. ONEM
- B. O.S.U. Utility Library:
  - 1. MMULT
  - 2. MSCALE

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

PLADIS

```
      SUBROUTINE PLADIS(ET,XE,XM,XCE,X,D,PE,PM,EP,EL,SE,SM,PSM,B1,B2)
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION XE(3),XM(3),XCE(3),X(3),PE(3,3),PM(3,3),EP(6),EL(6),
     *B1(21),B2(3),T1(3),SE(3,6),SM(3,6),PSM(3,3),TEM1(3,3),TE1(3,3),
     *TE2(3,3),TE3(3,3),T2(3),TEM2(3,3),B(6)
      DIMENSION A(3),F(3),C1(3,3),C2(3,3)
      DR=1.D0/D
      DO20I=1,3
      DO20J=1,3
      C1(I,J)=0.D0
      C2(I,J)=0.D0
   20 CONTINUE
      C1(1,1)=1.D0
      C1(2,2)=1.D0
      C1(3,3)=1.D0
      C1(3,2)=-1.D0
      C2(1,3)=1.D0
      C2(2,2)=1.D0
      C2(3,1)=1.D0
      C2(3,2)=-1.D0
C     START FORMING PARTIALS W.R.T.THE GEOCENTRIC COORDS. OF EARTH STN.
      CALLMMULT(X,PE,1,3,3,T1)
      CALLMSCALE(-DR,T1,1,3)
      DO30I=1,3
      B1(I)=T1(I)
   30 CONTINUE
C     THIS COMPLETES PARTIALS W.R.T.GEOC.COORDS.OF EARTH STATION
C     START FORMING PARTIALS W.R.T. EARTH EULERIAN ANGLES(INITIAL COND.)
      CALLLUCA(1,1.D0,TEM1)
      CALLROTATE(3,-EP(2),TE2)
      CALLROTATE(1,EP(1),TE1)
      CALLROTATE(3,-EP(3),TE3)
      CALLONEM(TE3,XE,T1)
      CALLONEM(TE1,T1,T2)
      CALLONEM(TEM1,T2,T1)
      CALLONEM(TE2,T1,T2)
      A(1)=0.D0
      DO33I=1,3
      A(1)=A(1)+X(I)*T2(I)
   33 CONTINUE
      A(1)=A(1)/D
      CALLLUCA(3,1.D0,TEM2)
      CALLONEM(PE,XE,T1)
      CALLONEM(TEM2,T1,T2)
      A(2)=0.D0
      DO36I=1,3
      A(2)=A(2)+X(I)*T2(I)
   36 CONTINUE
      A(2)=-A(2)/D
      CALLONEM(TEM2,XE,T1)
      CALLONEM(PE,T1,T2)
```

```
      A(3)=0.D0
      DO39I=1,3
      A(3)=A(3)+X(I)*T2(I)
   39 CONTINUE
      A(3)=-A(3)/D
      CALLMMULT(A,C1,1,3,3,F)
      CALLMMULT(F,SE,1,3,6,B)
      DO40I=1,6
      B1(I+3)=B(I)
   40 CONTINUE
C     THIS COMPLETES PARTIALS W.R.T.INITIAL CONDS. FOR EARTH'S ORIENT.
C     START FORMING PARTIALS FOR SELENOD. COORDS. OF LUNAR POINT
      CALLMMULT(X,PM,1,3,3,T1)
      CALLMSCALE(DR,T1,1,3)
      DO50I=1,3
      B1(I+9)=T1(I)
   50 CONTINUE
C     ENDS FORMATION OF PARTIALS W.R.T.SELENOD. COORDS. OF LUNAR POINT
C     START FORMING PARTIALS W.R.T. MOON'S EULERIAN ANGLS.(INIT. CONDS.)
      CALLROTATE(1,EL(1),TE1)
      CALLROTATE(3,-EL(2),TE2)
      CALLROTATE(3,-EL(3),TE3)
      CALLONEM(TE3,XM,T1)
      CALLONEM(TE1,T1,T2)
      CALLONEM(TEM1,T2,T1)
      CALLONEM(TE2,T1,T2)
      A(1)=0.D0
      DO63I=1,3
      A(1)=A(1)+X(I)*T2(I)
   63 CONTINUE
      A(1)=A(1)/D
      CALLONEM(PM,XM,T1)
      CALLONEM(TEM2,T1,T2)
      A(2)=0.D0
      DO66I=1,3
      A(2)=A(2)+X(I)*T2(I)
   66 CONTINUE
      A(2)=-A(2)/D
      CALLONEM(TEM2,XM,T1)
      CALLONEM(PM,T1,T2)
      A(3)=0.D0
      DO69I=1,3
      A(3)=A(3)+X(I)*T2(I)
   69 CONTINUE
      A(3)=-A(3)/D
      CALLMMULT(A,C2,1,3,3,F)
      CALLMMULT(F,SM,1,3,6,B)
      DO70I=1,6
      B1(I+12)=B(I)
   70 CONTINUE
```

```
C     THIS ENDS FORMATION OF PARTIALS OF MOON'S ANGLES (INITIAL CONDS.)
C     START FORMING PARTIALS FOR DYNAMIC PARAMETERS OF MOON
      CALLMMULT(F,PSM,1,3,3,6)
      DO80I=1,3
      B1(I+18)=B(I)
   80 CONTINUE
C     ENDS PARTIALS FOR DYNAMIC PARAM. OF MOON--C22,C20,BETA.
C     FORMING PARTIALS FOR COORDS (GEOCENTRIC) OF LUNAR CENTER
      DO90I=1,3
      B2(I)=X(I)/D
   90 CONTINUE
C     ENDS PARTIALS FOR COORDS OF LUNAR CENTER
C     ENDS FORMATION OF ALL PARTIALS
C     VALUES RETURNED IN B1 AND B2 MATRICES WILL HAVE TO BE PUT
C     IN PROPER LOCATIONS IN MAIN PROGRAM/OUTPUT SUBROUTINE.
  999 RETURN
      END
C
```

SUBROUTINE : PMAT

CALL STATEMENT : PMAT (PSI, THETA, PHI, P)

SUBROUTINE PURPOSE : The subroutine computes the transformation matrix necessary to rotate an earth fixed or lunar fixed coordinate system to the celestial system.

INPUT PARAMETERS :

A. Scalars (extended precision).
   1. PSI is the first Eulerian angle ψ in radians.
   2. THETA is the second Eulerian angle θ in radians.
   3. PHI is the third Eulerian angle φ in radians.

OUTPUT PARAMETERS :

A. Vector (extended precision).
   1. P is a 3 × 3 array which contains the transformation matrix to rotate from the fixed body system to the simulated earth-moon system.

PROGRAM DESCRIPTION : The output array (P) is formed by multiplying successive rotation matrices :

$$P = R_3(-\psi)\, R_1(\theta)\, R_3(-\varphi)$$

SUBROUTINES REQUIRED :

O. S. U. Project Library:
   1. ROTATE
   2. TRIM

REFERENCES :

A. Fajemirokun, F. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science. Report No. 157.

B. Papo, Haim B. (1971). "Optimal Selenodetic Control." The Ohio State University, Department of Geodetic Science, Report No. 156.

PMAT

```
      SUBROUTINE PMAT (PSI,THETA,PHI,P)
C     THIS ROUTINE COMPUTES THE P-MATRIX NECCESSARY TO ROTATE FROM AN
C     EARTH-FIXED OR LUNAR FIXED COORD. SYSTEM TO AN "INERTIAL" SYSTEM
C     INPUT QUANTITIES ARE EULER ANGLES PSI,THETA,AND PHI
C     P=R3(-PSI).R1(THETA).R3(-PHI)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION P(3,3),T1(3,3),T2(3,3),T3(3,3),T4(3,3)
      CALLROTATE(3,-PSI,T1)
      CALLROTATE(1,THETA,T2)
      CALLROTATE(3,-PHI,T3)
      CALLTRIM(T1,T2,T4)
      CALLTRIM(T4,T3,P)
      RETURN
      END
```

SUBROUTINE : POLE

CALL STATEMENT : POLE (DT, XPM, YPM)

SUBROUTINE PURPOSE : Computation of the coordinates of the true pole from the pole of the Conventional International Origin using values from the International Polar Motion Service (IPMS) from 1958 to 1970.5.

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. DT is the epoch of observation in Julian days minus 2400000.5 days.

OUTPUT PARAMETERS :

A. Scalars (extended precision).
   1. XM is the polar motion coordinate in the x direction as defined by the IPMS in seconds of arc.
   2. YM is the polar motion coordinate in the y direction as defined by the IPMS in seconds of arc.

PROGRAM DESCRIPTION : The program uses polar motion values stored in data statement storage. The program logic is as follows:

A. The input date is checked to see if the date falls within the table limits (1958-1970.5). If not, an error message is printed and control returned to the calling program.
B. If the input date is valid, a linear interpolation of the tabulated polar motion values is accomplished.

SUBROUTINES REQUIRED : None

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

POLE

```
      SUBROUTINE POLE(DT, XPM,YPM)
      DOUBLE PRECISION DT,XPM,YPM
      DIMENSION PM(251,2),PMT(2)
      DIMENSION PMX1(90),PMY1(90),PMX2(90),PMY2(90),PMX3(15),PMY3(15)
      EQUIVALENCE(PM(1,1),PMX1(1)),(PM(1,2),PMY1(1)),(PM(91,1),PMX2(1)),
     1(PM(91,2),PMY2(1)),(PM(181,1),PMX3(1)),(PM(181,2),PMY3(1))
      DIMENSION PMX4(5),PMY4(5)
      EQUIVALENCE(PM(196,1),PMX4(1)),(PM(196,2),PMY4(1))
      DIMENSION PMX5(51),PMY5(51)
      EQUIVALENCE (PM(201,1),PMX5(1)),(PM(201,2),PMY5(1))
C   POLAR MOTION TABLES FURNISHED BY TOMLINSON (TAKEN FROM IPMS)
      DATA PMX1/
     1-0.173,-0.215,-0.235,-0.237,-0.218,-0.162,-0.097,-0.032, 0.036,       0000
     2 0.111, 0.188, 0.237, 0.348, 0.398, 0.398, 0.368, 0.330, 0.280,       0001
     3 0.218, 0.144, 0.069, 0.000,-0.062,-0.112,-0.140,-0.153,-0.151,       0002
     4-0.126,-0.086,-0.037, 0.026, 0.092, 0.161, 0.223, 0.272, 0.299,       0003
     5 0.308, 0.296, 0.261, 0.202, 0.135, 0.073, 0.046, 0.035, 0.013,       0004
     6-0.026,-0.072,-0.096,-0.107,-0.103,-0.087,-0.039, 0.004, 0.040,       0005
     7 0.070, 0.080, 0.109, 0.117, 0.117, 0.109, 0.092, 0.074, 0.065,       0006
     8 0.064, 0.062, 0.057, 0.046, 0.034, 0.030, 0.032, 0.040, 0.043,       0007
     9 0.042, 0.041, 0.039, 0.028, 0.019,-0.010,-0.027,-0.021,-0.009,       0008
     A 0.008, 0.027, 0.047, 0.071, 0.095, 0.120, 0.144, 0.162, 0.173  /
      DATA PMX2/
     1 0.171, 0.157, 0.128, 0.094, 0.056, 0.017,-0.019,-0.054,-0.086,       0010
     2-0.110,-0.121,-0.119,-0.105,-0.076,-0.038, 0.009, 0.070, 0.134,       0011
     3 0.191, 0.239, 0.274, 0.301, 0.281, 0.237, 0.176, 0.112, 0.048,       0012
     4-0.011,-0.069,-0.122,-0.171,-0.206,-0.194,-0.169,-0.139,-0.101,       0013
     5-0.055, 0.004, 0.074, 0.164, 0.214, 0.240, 0.241, 0.239, 0.255,       0014
     6 0.250, 0.219, 0.161, 0.099, 0.042,-0.012,-0.067,-0.120,-0.160,       0015
     7-0.185,-0.196,-0.194,-0.174,-0.130,-0.072,-0.003, 0.071, 0.127,       0016
     8 0.168, 0.201, 0.221, 0.227, 0.220, 0.194, 0.138, 0.075, 0.033,       0017
     9 0.000,-0.029,-0.058,-0.086,-0.105,-0.116,-0.119,-0.115,-0.104,       0018
     A-0.086,-0.057,-0.010, 0.052, 0.096, 0.117, 0.125, 0.123, 0.115/       0019
      DATA PMY1/
     1 0.022, 0.098, 0.187, 0.265, 0.328, 0.389, 0.443, 0.478, 0.493,       0020
     2 0.476, 0.447, 0.411, 0.365, 0.307, 0.235, 0.165, 0.097, 0.043,       0021
     3-0.007,-0.038,-0.057,-0.064,-0.057,-0.025, 0.032, 0.120, 0.211,       0022
     4 0.285, 0.340, 0.372, 0.393, 0.406, 0.410, 0.401, 0.370, 0.320,       0023
     5 0.260, 0.201, 0.143, 0.090, 0.043, 0.007,-0.012,-0.007, 0.025,       0024
     6 0.059, 0.094, 0.123, 0.153, 0.182, 0.209, 0.238, 0.263, 0.288,       0025
     7 0.300, 0.306, 0.301, 0.288, 0.271, 0.249, 0.220, 0.189, 0.161,       0026
     8 0.150, 0.151, 0.158, 0.161, 0.160, 0.155, 0.153, 0.150, 0.151,       0027
     9 0.154, 0.157, 0.165, 0.174, 0.191, 0.212, 0.242, 0.276, 0.297,       0028
     A 0.309, 0.314, 0.312, 0.304, 0.290, 0.271, 0.246, 0.214, 0.175/       0029
      DATA PMY2/
     1 0.132, 0.092, 0.068, 0.060, 0.067, 0.083, 0.104, 0.128, 0.160,       0030
     2 0.200, 0.248, 0.295, 0.329, 0.356, 0.376, 0.388, 0.387, 0.375,       0031
     3 0.349, 0.307, 0.251, 0.193, 0.139, 0.091, 0.046, 0.008,-0.020,       0032
     4 0.005, 0.041, 0.078, 0.120, 0.168, 0.230, 0.294, 0.353, 0.412,       0033
     5 0.455, 0.467, 0.459, 0.436, 0.394, 0.339, 0.275, 0.219, 0.168,       0034
     6 0.123, 0.085, 0.060, 0.046, 0.043, 0.049, 0.069, 0.103, 0.153,       0035
```

POLE (Cont)

```
 7 0.226, 0.286, 0.334, 0.374, 0.408, 0.434, 0.444, 0.433, 0.399,         0036
 8 0.349, 0.303, 0.259, 0.221, 0.186, 0.156, 0.131, 0.114, 0.103,         0037
 9 0.098, 0.100, 0.108, 0.124, 0.149, 0.181, 0.215, 0.255, 0.296,         0038
 A 0.330, 0.344, 0.345, 0.337, 0.324, 0.308, 0.291, 0.273, 0.253/         0039
  DATA PMX3/
 1 0.099, 0.079, 0.056, 0.031, 0.012,-0.001,-0.006,-0.008,-0.002,
 2 0.012, 0.035, 0.055, 0.046, 0.027, 0.008  /
  DATA PMX4              /-0.010,-0.029,-0.049,-0.063,-0.066/
  DATA PMY3/
 1 0.233, 0.213, 0.194, 0.177, 0.165, 0.157, 0.155, 0.154, 0.152,
 2 0.156, 0.163, 0.172, 0.183, 0.195, 0.208  /
  DATA PMY4              / 0.220, 0.234, 0.249, 0.269, 0.289/
  DATA PMX5/
 1-.056,-.037,-.014,.008,.031,.051,.064,.067,.064,.060,.088,.119,.10
 26 ,.054,.008,-.027,-.056,     -.084,-.109,-.123,-.127,-.120,-.102
 3,-.073,-.033,.010,.052,.091,.125,.154,.174,.185,.164,.168,.127,.07
 47 ,.029,-.021,-.071,-.115,-.157,-.184,-.184,-.166,-.135,-.100,-.06
 53 ,-.025,.017,.083,.154/
  DATA PMY5/
 1.302,.308,.308,.302,.290,.276,.260,.245,.231,.216,.202,.183,.166,
 2.157,.156,.161,.172,     .197,.233,.265,.289,.310,.330,.350,.370,
 3.386,.392,.386,.367,.337,.302,.260,.212,.167,.134,.115,.105,.104,
 4.114,.134,.168,.216,.273,.333,.384,.419,.449,.465,.463,.436,.391/
  A=(DT-0.36203861D5)*0.54758185D-1
  L=A+1.0
  IF(L.LT.2)GOTO901
  IF(L.GT.198)GOTO901
  TL=L
  AN=A+1.0-TL
  B=AN*(AN-1.0)/4.0
  DO10I=1,2
  DELO=PM(L,I)-PM(L-1,I)
  DEL1=PM(L+1,I)-PM(L,I)
  DEL2=PM(L+2,I)-PM(L+1,I)
  PMT(I)=PM(L,I)+AN*DEL1+B*(DEL2-DELO)
10 CONTINUE
  XPM=PMT(1)
  YPM=PMT(2)
  RETURN
901 WRITE(6,9001)
9001 FORMAT(70HOTABLES OF POLAR MOTION COVER ONLY FROM 1958.0 TO 1970.5
 1PLEASE EXTEND.)
  STOP
  END
```

SUBROUTINE : PRECSS

CALL STATEMENT : PRECSS (ET, PRE, DPRE, KK)

SUBROUTINE PURPOSE : Calculates the precession matrix P and its time derivative $\dot{P}$. P is an orthogonal matrix that transforms a vector from the mean equatorial system of 1950.0 into the mean equatorial system of date (ET).

INPUT PARAMETERS :

A. Scalars.
  1. ET (extended precision). The epoch for which P and $\dot{P}$ are required in Julian days minus 2440000.0.
  2. KK (integer). An index to be set to 1 when only P is required or to 2 if P and $\dot{P}$ are required.

OUTPUT PARAMETERS :

A. Matrices (extended precision).
  1. PRE (3 x 3). The precession matrix P.
  2. DPRE (3 x 3). The time derivative of the $\dot{P}$ matrix or P.

PROGRAM DESCRIPTION : Outlined in [Mueller, 1969] Pages 62-65.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
  1. ROTATE
  2. TRIM

B. Fortran Scientific Subroutine Package:
  1. DGMPRD

REFERENCES :

A. Mendez, J. C. and R. J. Stern (1969). "Geographic and Selenodetic Coordinate Transformation Programs," TRW N 69, FMT 749.

B. Mueller, Ivan I. (1969). "Spherical and Practical Astronomy as Applied to Geodesy," Frederick Ungar Publishing Co., New York.

PRECSS

```
      SUBROUTINE PRECSS (ET,PRE,DPRE,KK)
C     CALCULATES PRECESSION MATRIX IN RADIANS AND ITS TIME DERIVATIVE
C     IN RADIANS PER EPHEMERIS DAY  :  THE PRE MATRIX PERFORMS TRANSFORMATION
C     FROM MEAN EQUATURIAL OF 1950.0 (JED=2433282.423) TO MEAN EQUATURIAL
C     OF DATE (ET)
C     KK=1  IF ONLY PRECESSION MATRIX IS DESIRED
C     KK=2  IF PRECESSICN MATRIX AND ITS TIME DERIVATIVE ARE DESIRED
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION CP(3,4),DP(3,4),T(4),PRE(3,3),DPRE(3,3),A1(3),D1(3),A2(3
     P,3),A3(3,3),A4(3,3),A5(3,3),A6(3,3)                                  2
C     THE MATRIX CP CALCULATES THE ANGLES ZETAO , TETA , ZEI
      DATA CP /3*0.D0,2304.95167D0,2004.25826D0,2304.95162D0,.30216SD0,-
     P.42688SD0,1.09519SD0,.018D0,-.0418D0,.01832D0/, DP /2304.95167D0,2   2
     P004.25826D0,2304.95162D0,.60433D0,-.85377D0,2.19039D0,.054D0,-.125   3
     P4D0,.05496D0,3*0.D0/ ,DINTC     /36524.219879D0/                     4
     P,RADSE/206264.806247096355D0/                                        5
      TD=ET-2433282.423D0
      TC=TD/DINTC
      T(1)=1.D0/RADSE
      T(2)=TC*T(1)
      T(3)=TC*T(2)
      T(4)=TC*T(3)
      CALLDGMPRD(CP,T,A1,3,4,1)
      CALLDGMPRD(DP,T,D1,3,4,1)
      DO1I=1,3
    1 D1(I)=D1(I)/DINTC
      CALLROTATE(3,-A1(3),A3)
      CALLROTATE(2, A1(2),A2)
      CALLDGMPRD(A3,A2,A4,3,3,3)
      CALLROTATE(3,-A1(1),A5)
      CALLDGMPRD(A4,A5,PRE,3,3,3)
      IF(KK.NE.2)GOTO707
      CALLLUCA(3,-D1(3),A2)
      CALLDGMPRD(A2,PRE,A3,3,3,3)
      CALLLUCA(2,D1(2),A2)
      CALLDGMPRD(A2,A5,A6,3,3,3)
      CALLDGMPRD(A4,A6,A5,3,3,3)
      CALLLUCA(3,-D1(1),A2)
      CALLDGMPRD(PRE,A2,A4,3,3,3)
      CALLDGMADD(A3,A4,A6,3,3)
      CALLDGMADD(A6,A5,DPRE,3,3)
  707 CONTINUE
      RETURN
      END
```

SUBROUTINE : PREITR

CALL STATEMENT : PREITR (ET, EMU, EKE, N, AM, P)

SUBROUTINE PURPOSE : Prepares elements for STVITR subroutine for the Keplerian motion of a satellite about a primary body.

INPUT PARAMETERS :

A. ET is the initial epoch in Julian days minus 2440000.0.
B. EMU is the product of the gravitational constant and the total mass of the system.
C. EKE is a vector of Keplerian Orbital Elements in extended precision and in radians or radians per day.
    1. EKE(1) is the longitude of the ascending node (Ω).
    2. EKE(2) is the argument of perigee (ω).
    3. EKE(3) is the inclination (i).
    4. EKE(4) is the eccentricity (e).
    5 EKE(5) is the mean anomaly at epoch ET.
    6. EKE(6) is the mean motion (n).

OUTPUT PARAMETERS :

A. P is a vector of output parameters:
    1. P(1) is a precision indicator, defined as:

$$P(1) = 10^{-N}$$

    2. P(2) is the initial epoch (ET) in Julian minus 2440000.
    3. P(3) is the orbital major semi axis computed from:

$$P(3) = \left[\frac{\mu}{n^2}\right]^{\frac{1}{3}}$$

where $\mu = k^2(E + M)$, and k^3 is the Gaussian Gravitational Constant.

    4. P(4 to 6) are the same as EKE(4 to 6).

B. The AM 3 × 3 matrix is formed from:

$$AM = R_3(-\Omega)\, R_1(-i)\, R_3(-\omega)$$

AM is the transformation matrix from $x'y'z'$ (orbital plane perigee) system into the XYZ ecliptic or equatorial celestial system.

PROGRAM DESCRIPTION : For a given Keplerian orbit of a satellite (ET, EMU, EKE) the routine sets the AM, P output parameters which are then transferred by the calling program to the STVITR routine every time the state vector of the satellite is required.

SUBROUTINES REQUIRED :

O. S. U. Project Library:

1. ROTATE
2. TRIM

REFERENCES :

Mueller, Ivan I. (1964). "Introduction to Satellite Geodesy," Frederick Ungar Publishing Co., New York.

PREITR

```
C
      SUBROUTINE PREITR (ET,EMU,EKE,N,AM,P)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AM(3,3),B(3,3),C(3,3),EKE(6),P(6)
      P(1)=1.D0/10.D0**N
      P(2)=ET
      P(3)=(EMU/(EKE(6)*EKE(6)))**.3333333333333333D0
      DO 1 I=4,6
      P(I)=EKE(I)
      CALL ROTATE (3,-EKE(2),AM)
      CALL ROTATE (1,-EKE(3),B)
      CALL TRIM (B,AM,C)
      CALL ROTATE (3,-EKE(1),B)
      CALL TRIM (B,C,AM)
      RETURN
      END
```

SUBROUTINE : PRESTV

CALL STATEMENT : PRESTV (ET, EM, EK)

SUBROUTINE PURPOSE : Generation of series of constants and a transformation matrix for the calculation of the state vector of a satellite in a Keplerian orbit about the primary body. (The state vector is created by use of a companion subroutine STVKEP).

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. ET is the initial epoch in Julian days minus 2440000. 0.
   2. EM is the product of the gravitational constant and the total mass of the two bodies.

B. Vector (extended precision).
   1. EK(6) is a 6-element vector containing the Keplerian orbital elements in the following order and in radians or radians per day:
      a. Longitude of ascending node (Ω).
      b. Argument of Perigee (ω).
      c. Inclination (i).
      d. Eccentricity (e).
      e. Mean anomaly (M) at epoch ET.
      f. Mean motion (n).

OUTPUT PARAMETERS : (All in common storage sector /EPLER/).

A. Scalars (extended precision).
   1. E is the eccentricity.
   2. E2 is the eccentricity squared.
   3. RZ is 1 + $e^{2/2}$.
   4. ELO is the mean anomaly (M) at the standard epoch set equal to Input Parameter EK(5).
   5. ETO is the mean motion (set equal to Input Parameter E
   6. EN is the mean motion set equal to Input Parameters EK(
   7. AO is the orbit's semi major axis.

B. Vectors (extended precision).
   1. R(1) through R(7) are coefficients of a series expansion for the radius vector.

2. F(1) through F(7) are coefficients of a series expansion for the true anomaly (see [Brouwer and Clemence, 1961]).

C. Matrix (extended precision).

1. T1 (3 × 3) is an orthogonal transformation matrix to rotate a vector from the $x'y'z'$ (satellite orbital plane) system to the xyz (fixed ecliptic) system.

?ROGRAM DESCRIPTION :

A. The radius vector semi major axis ratio is given in the form [Brouwer and Clemence, 1961] Equation 73, Page 76:

$$\frac{r}{a} = R_1 + R_2 \cos \ell + R_3 \cos 2\ell \cdots$$

where R_i are functions of the eccentricity and ℓ is the mean anomaly, for example:

$$R_1 = 1 + \frac{1}{2}$$

$$R_2 = -e + \frac{3}{8} e^3 - \frac{5}{192} e^5 + \frac{7}{9216} e^7$$

B. The true anomaly (f) is expressed by [Brouwer and Clemence, 1961] Equation 75, Page 77.

$$f = \ell + F_1 \sin \ell + F_2 \sin 2\ell \cdots$$

where F_i are functions of the eccentricity, for example

$$F_i = 2e - \frac{1}{4} e^3 + \frac{5}{96} e^5 + \frac{107}{4608} e^7.$$

C. In both the R and F coefficients one must note that the eccentricity terms are truncated beyond the e^7 term. Therefore, for orbits with large eccentricites the effect of higher order terms should be closely investigated.

D. A rotation matrix to rotate coordinates in the orbital plane system to the mean ecliptic system is formed in T1 as:

$$T1\ (3,\ 3) = R_3(-\Omega)\ R_1(-i)\ R_3(-\omega).$$

SUBROUTINES REQUIRED :

O.S.U. Project Library
1. ROTATE
2. TRIM

REFERENCES :

Brouwer, D. and G. Clemence (1961). "Methods of Celestial Mechanics," Academic Press, New York, pp. 76-77.

PRESTV

```
C
      SUBROUTINE PRESTV (ET,EM,EK)
C     PREPARING ELEMENTS FOR STVKEP SUBROUTINE IN COMMON /EPLER/
C  ET EPOCH IN (JD-2440000.)
C  EM PRODUCT OF GRAVITATIONAL CONSTANT AND TOTAL MASS OF THE TWO BODIES
C  EK KEPLERIAN ORBIT ELEMENTS IN THE FOLLOWING ORDER :
C     LONGITUDE OF NODE , ARGUMENT OF PERIGEE , INCLINATION
C     ECCENTRICITY , MEAN ANOMALY , MEAN MOTION
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION T1(3,3),T2(3,3),T3(3,3),R(7),F(7),EK(6)
      COMMON /EPLER/ E,E2,RZ,R,F,ELO,ETO,EN,AO,T1
      DATA ONE,TWO,TRI,FOR,FIF,SEV/1.D0,2.D0,3.D0,4.D0,5.D0,7.D0/
C     SERIES FOR RADIUS VECTOR
      E=EK(4)
      E2=E*E
      E3=E2*E
      E4=E2*E2
      E5=E3*E2
      E6=E3*E3
      E7=E4*E3
      RZ=ONE+E2/TWO
      R(1)=SEV*E7/9216.-FIF*E5/192.D0+TRI*E3/8.D0-E
      R(2)=-E2/TWO+E4/TRI-E6/16.
      R(3)=-TRI*E3/8.D0+45.D0*E5/128.D0-567.*E7/5120.
      R(4)=-E4/TRI+TWO*E6/FIF
      R(5)=-125.D0*E5/384.D0+4375.*E7/9216.
      R(6)=-27.*E6/80.
      R(7)=-16807.*E7/46080.
C     SERIES FOR TRUE ANOMALY
      F(1)=TWO*E-E3/FOR+FIF*E5/96.D0+107.*E7/4608.
      F(2)=FIF*E2/FOR-11.D0*E4/24.D0+17.*E6/192.
      F(3)=13.D0*E3/12.D0-43.D0*E5/64.D0+95.*E7/512.
      F(4)=103.D0*E4/96.D0-451.*E6/480.
      F(5)=1097.D0*E5/960.D0-5957.*E7/4608.
      F(6)=1223.*E6/960.
      F(7)=47273.*E7/32256.
      CALL ROTATE (3,-EK(2),T1)
      CALL ROTATE (1,-EK(3),T2)
      CALL TRIM (T2,T1,T3)
      CALL ROTATE (3,-EK(1),T2)
      CALL TRIM (T2,T3,T1)
      EN=EK(6)
      ETO=ET
      ELO=EK(5)
      AO=(EM/(EN*EN))**.33333333333333300
      RETURN
      END
```

SUBROUTINE : REDPI

CALL STATEMENT : REDPI (YTH)

SUBROUTINE PURPOSE : To reduce a given angle θ to the interval

$0 < \theta < 2\pi$

INPUT PARAMETERS :

A. Scalar (extended precision).
   1. YTH is the input angle in radians.

OUTPUT PARAMETERS :

A. Scalar (extended precision).
   1. YTH is returned from the subroutine in the desired range (in radians).

PROGRAM DESCRIPTION : None

SUBROUTINES REQUIRED : None

REFERENCES : None

REDPI

```
C
      SUBROUTINE REDPI (YTH)
C     REDUCTION OF ANGLE YTH TO THE INTERVAL ZERO AND + TWO PI
      REAL*8 YTH,PI2
      PI2=6.283185307179586D0
      YTH=YTH-PI2*DFLOAT(IDINT(YTH/PI2))
   10 IF(YTH   .LE.0.D0)GOTO13
   11 IF(YTH   .LT.PI2)GOTO12
      YTH   =YTH   -PI2
      GOTO11
   13 YTH   =YTH   +PI2
      GOTO10
   12 CONTINUE
      RETURN
      END
C
```

SUBROUTINE : ROTATE

CALL STATEMENT : ROTATE (N, ANG, RNA)

SUBROUTINE PURPOSE : Forms a 3×3 rotation matrix, RNA, representing a rotation of angle ANG (in radians) about an axis xyz, designated 1, 2 and 3 respectively.

INPUT PARAMETERS :

A. Axis about which rotation is to occur N.

B. Rotation angle in radians ANG which may be positive or negative in sign.

OUTPUT PARAMETERS :

A. Rotation matrix RNA (3×3).

PROGRAM DESCRIPTION :

A. $R_i(\theta)$ is formed where $i = N$, θ = ANG and $R_i(\theta)$ = RNA.

B. The integer variables N1 and N2 are set to the following value dependent on N (by use of integer arithmetic)

N	N1	N2
1	2	3
2	3	1
3	1	2

Then after RNA is zeroed :

RNA (N, N) = 1
RNA (N1, N1) = cos (ANG)
RNA (N2, N2) = cos (ANG)
RNA (N1, N2) = sin (ANG)
RNA (N2, N1) = -sin (ANG)

SUBPROGRAMS REQUIRED : None

REFERENCES :

Mueller, I. I. (1969). "Spherical and Practical Astronomy as Applied to Geodesy," Frederick Ungar Publishing Co., New York, p. 43.

ROTATE

```
      SUBROUTINEROTATE(N,ANG,RNA)
C     ROTATION MATRIX "RNA" ABOUT AXIS "N" BY ANGLE "ANG"
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONRNA(3,3)
      DO304I1=1,3
      DO303I2=1,3
  303 RNA(I1,I2)=0.D0
  304 CONTINUE
      SANG=DSIN(ANG)
      CANG=DCOS(ANG)
      RNA(N,N)=1.D0
      N1=N+1-((N+1)/4)*3
      N2=N+2-((N+2)/4)*3
      RNA(N1,N1)=CANG
      RNA(N2,N2)=CANG
      RNA(N1,N2)=SANG
      RNA(N2,N1)=-SANG
      RETURN
      END
C
```

SUBROUTINE : SEFODI

CALL STATEMENT : SEFODI (M, N, K, L, K1, K2, FF, FD)

SUBROUTINE PURPOSE : Generation of second and fourth modified differences for Everett interpolation of L sets of tabulated quantities.

INPUT PARAMETERS :

A. Integers (single precision).
  1. M is the number of a tape unit which contains L sets of K tabulated ephemeris quantities written in binary m
  2. N is the number of an output storage unit (disk) on whic the function table with second and fourth differences are to be written in binary mode.
  3. K is the number of variables in a set. For example, if the tabulated quantites are components of a state vector, K should be set to 6.
  4. K1 and K2 must be set to 9 and 3 respectively.

B. Matrices (extended precision).
  1. FF (9 × K) contains 9 rows of K tabulated quantities input from the unit designed by M. The second and fourth modified differences are computed for the middle row (i.e., row 5).

OUTPUT PARAMETERS :

A. Matrices (extended precision).
  1. FD (3 × K) contains the following information in rows:
    a. The K tabulated quantites from unit M (identical to row 5 of FF).
    b. The K modified second differences.
    c. The K modified fourth differences.

PROGRAM DESCRIPTION : The second and fourth modified differences are computed as outlined in [O'Handley, 1969]. The method is intended to facilitate the use of Everett's fifth order interpolation formula (see EVERAL).

SUBROUTINES REQUIRED :

Fortran Scientific Subroutine Package:
  1. DGMPRD

REFERENCES :

O'Handley, Douglas A. et al., (1969). "JPL Development Ephemeris Number 69," JPL Technical Report 32-1465.

SEFODI

```
C
      SUBROUTINE SEFODI (M,N,K,L,K1,K2,FF,FD)
C     GENERATING SECOND AND FOURTH MODIFIED DIFFERENCES
C     M - UNIT CONTAINING TABULATED QUANTITIES IN L ROWS OF K EACH IN BINARY
C     N - UNIT TO WRITE IN BINARY MODE THE TABLE OF DIFFERENCES
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION FF(K1,K),FD(K2,K),CC(3,9)
      DATACC(1,1),CC(1,2),CC(1,3),CC(1,4),CC(1,6),CC(1,7),CC(1,8),CC(1,9
     P),CC(1,5),CC(2,1),CC(2,9),CC(2,2),CC(2,8),CC(2,3),CC(2,7),CC(2,4),          1
     PCC(2,6),CC(2,5),CC(3,1),CC(3,9),CC(3,2),CC(3,8),CC(3,3),CC(3,7),CC          2
     P(3,4),CC(3,6),CC(3,5)/8*0.D0,1.D0,2*.4299D-2,2*-.047512D0,2*.19909          3
     P2D0,2*.562456D0,-1.43667D0,2*.066489D0,2*-.826161D0,2*4.587306D0,2          4
     P*-12.009419D0,16.35961D0/                                                   5
      REWIND M
      REWIND N
      DO 1 I=1,9
    1 READ(M) (FF(I,KO),KO=1,K)
      NK=L-9
      DO 2 I=1,NK
      CALLOGMPRD(CC,FF,FD,3,9,K)
      DO 3 KO=1,K
      DO 3 J=1,8
    3 FF(J,KO)=FF(J+1,KO)
      READ(M,END=9)(FF(9,KO),KO=1,K)
    2 WRITE(N)    ((FD(J.KO),J=1,3),KO=1,K)
    9 REWIND N
      RETURN
      END
```

SUBROUTINE : SKEPTR

CALL STATEMENT : SKEPTR (S, M, K)

SUBROUTINE PURPOSE : Transformation of a satellite state vector into instantaneous Keplerian elements.

INPUT PARAMETERS :

- A. Scalar (extended precision).
  - 1. M is the product of the universal gravitational constant (k^2) and the combined masses $(m_1 + m_2)$ of the primary body and the satellite.
- B. Vector.
  - 1. S is a six element vector containing the state vector in linear units and linear units per day. The units in the state vector must be compatible with those in the scaler M, above.

OUTPUT PARAMETERS :

- A. Vector (extended precision).
  - 1. K is a six element vector containing the elements of the instantaneous Keplerian orbit in radians/radians per day, in the following order:
    - a. The longitude of node.
    - b. The argument of perigee.
    - c. The inclination.
    - d. the eccentricity.
    - e. The mean anomaly.
    - f. The mean motion.
  - 2. The following factors are used in the programs (within the extent of IBM 360 extended precision):
    - a. If the inclination is zero, then the longitude of node is set to zero.
    - b. If the eccentricity is zero, then the argument of perigee is set to zero.

PROGRAM DESCRIPTION : See references.

SUBROUTINES REQUIRED : None.

REFERENCES :

A. Escobal, P. R. (1965). "Methods of Orbit Determination," John Wiley and Sons, New York.

B. Mueller, I. I. (1964). "Introduction to Satellite Geodesy," Frederick Ungar Publishing Co., New York.

SKEPTR

```
      SUBROUTINE SKEPTR (S,M,K)
C     TRANSFORMATION FROM STATE VECTOR INTO INSTANTANEOUS KEPLERIAN ELEMENTS
C   S STATE VECTOR
C   M PRODUCT OF GRAVITATIONAL CONSTANT AND TOTAL MASS OF THE TWO BODIES
C   K KEPLERIAN ORBIT ELEMENTS IN THE FOLLOWING ORDER :
C     LONGITUDE OF NODE , ARGUMENT OF PERIGEE , INCLINATION
C     ECCENTRICITY , MEAN ANOMALY , MEAN MOTION
C     IF INCLINATION IS ZERO - LONGITUDE OF NODE IS SET TO ZERO
C     IF ECCENTRICITY IS ZERO - ARGUMENT OF PERIGEE IS SET TO ZERO
C     MEAN ANOMALY IS MEASURED FROM PERIGEE OR FROM NODE OR FROM AXIS (1)
C     ARGUMENT OF PERIGEE IS MEASURED FROM NODE OR FROM AXIS (1)
      IMPLICITREAL*8(A-H,O-Z)
      REAL*8 M,K
      DIMENSION S(6),K(6),U(3),V(3)
      R=0.D0
      VE=0.D0
      RD=0.D0
      DO 1 I=1,3
      R=R+S(I)**2
      VE=VE+S(I+3)**2
    1 RD=RD+S(I)*S(I+3)
      R=DSQRT(R)
      RT=RD/R
      A=1.D0/(2.D0/R-VE/M)
      K(6)=DSQRT(M/(A**3))
      T=RD/DSQRT(A*M)
      Q=1.D0-R/A
      K(4)=DSQRT(T**2+Q**2)
      IF(K(4)) 8,9,8
    8 K(5)=DATAN2(T/K(4),Q/K(4))-T
    9 P=A*(1.D0-K(4)**2)
      DO 2 I=1,3
      U(I)=S(I)/R
    2 V(I)=(S(I+3)*R-S(I)*RT)/DSQRT(P*M)
      CI=DSQRT((U(1)+V(2))**2+(U(2)-V(1))**2)-1.D0
      SL=(U(2)-V(1))/(1.D0+CI)
      CL=(U(1)+V(2))/(1.D0+CI)
      EL=DATAN2(SL,CL)
      SI=DSQRT(U(3)**2+V(3)**2)
      IF(SI) 3,4,3
    3 K(3)=DATAN2(SI,CI)
      SU=U(3)/SI
      CU=V(3)/SI
      UU=DATAN2(SU,CU)
      GOTO 10
    4 UU=EL
   10 IF(K(4)) 5,6,5
    5 SV=RT*DSQRT(P/M)/K(4)
      CV=(P/R-1.D0)/K(4)
      TA=DATAN2(SV,CV)
```

```
  GOTO 7
6 TA=UU
  K(5)=UU
7 K(1)=EL-UU
  K(2)=UU-TA
  CALL REDPI (K(1))
  CALL REDPI (K(2))
  CALL REDPI (K(5))
  RETURN
  END
```

SUBROUTINE : SPEQU

CALL STATEMENT : SPEQU (A, B, M, N, K, L)

SUBROUTINE PURPOSE : Copies a two dimensional matrix (B) into layer L of a three dimensional matrix A.

INPUT PARAMETERS :

- A. Scalar Integers (single precision).
  - 1. M, N, K. Dimensions of the three dimensional output matrix A (Rows, Columns, Layers).
  - 2. L. The number of the layer of A (consisting of K layer into which the input matrix B is to be copied.
- B. Matrices (extended precision).
  - 1. B. The M × N matrix to be copied.

OUTPUT PARAMETERS :

- A. Matrices (extended precision).
  - 1. A. The output M × N × K three dimensional matrix.

PROGRAM DESCRIPTION : None

SUBROUTINES REQUIRED : None

REFERENCES : None

SPEQU

```
      SUBROUTINESPEQU(A,B,M,N,K,L)
C     MAKING LAYER L OF A EQUAL TO B
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSIONA(M,N,K),B(M,N)
      DO1I=1,M
      DO2J=1,N
    2 A(I,J,L)=B(I,J)
    1 CONTINUE
      RETURN
      END
```

SUBROUTINE : STRANS

CALL STATEMENT : STRANS (JD, GAST, SMAT)

SUBROUTINE PURPOSE : The subroutine computes the rotation matrix S to rotate coordinates from the true celestial system to the average terrestrial system.

INPUT PARAMETERS :

A. Scalars (extended precision).
   1. JD is the epoch in Julian days.
   2. GAST is the Greenwich apparent siderial time in radians.

OUTPUT PARAMETERS :

A. Matrix (extended precision).
   1. SMAT is a 3 ×3 array containing the rotation matrix from the true celestial system to the average terrestrial system.

PROGRAM DESCRIPTION : The SMAT matrix (in Mueller's notation the S matrix) is computed from the polar motion components (x, y) and the Greenwich apparent siderial time (θ) from

$$S = R_1(-x)\, R_1(-y)\, R_3(\theta)$$

SUBROUTINES REQUIRED :

O.S.U. Project Library:
1. POLE
2. ROTATE
3. TRIM

REFERENCES :

Mueller, I. I. (1969). "Spherical and Practical Astronomy as Applied to Geodesy," Frederick Ungar Publishing Co., New York.

STRANS

```
      SUBROUTINE STRANS(JD,GAST,SMAT)
      IMPLICITREAL*8(A-H,O-Z)
      DOUBLE PRECISION MJD,JD
      DIMENSION R3TETA(3,3),R1MY(3,3),R2MX(3,3),T1(3,3),SMAT(3,3)
      DATARADSE/206264.8062470964D0/
C     COMPUTES THE SMATRIX TO ROTATE FRM TRUE CEL. TO AVERAGE TERR. SYST
C     INPUT PARAM. --JD,GAST.
C     EXTERNAL ROUTINES--POLE(FOR POLAR MOTION),AND TRIM
      MJD=JD-2400000.5D0
      CALLPOLE(MJD,XPP,YPP)
      XPP=XPP/RADSE
      YPP=YPP/RADSE
      CALLROTATE(3,GAST,R3TETA)
      CALLROTATE(1,-YPP,R1MY)
      CALLROTATE(2,-XPP,R2MX)
      CALLTRIM(R2MX,R1MY,T1)
      CALLTRIM(T1,R3TETA,SMAT)
      RETURN
   60 FORMAT(//,20X,2F20.9)
      END
```

SUBROUTINE : STVITR

CALL STATEMENT : STVITR (ET, STV, AM, P)

SUBROUTINE PURPOSE : To calculate a state vector of a satellite in a Keplerian orbit for a given epoch and in components of a fixed Cartesian coordinate system.

INPUT PARAMETERS : (Extended precision.)

A. ET is the epoch in Julian days (t) for which the state vector is required minus 24400000.

B. AM and P are the 3 X 3 matrix and the six element vector described in PREITR.

OUTPUT PARAMETERS : (Extended precision.)

A. STV is a 6-element state vector of epoch in the XYZ system of a satellite moving in a Keplerian orbit about a primary body.

PROGRAM DESCRIPTION :

A. The mean anomaly (ℓ) for the epoch of observation is obtained from

$$\ell = \ell_0 + n(t - t_0)$$

$$\text{where} \quad 0 \leq \ell < 2\pi$$

B. The eccentric anomaly E is then found from the solution of Kepler's equation by:

1. Estimating the initial value as

$$E_0 = \ell$$

2. Then using a Newton iterative procedure to find the E_p of epoch as follows:

    a. Compute ΔE_i from:

$$\Delta E_i = E_i - e \sin E_i - \ell$$

b. Compare ΔE_i with the precision estimator $P(1)$ which is denoted P

$$|\Delta E_i| < P$$

c. Then:
If $|\Delta E_i| < P$, terminate iteration and continue program.

If $|\Delta E_i| \geq P$, continue iteration by setting

$$E_{i+1} = E_i - \Delta E_i.$$

If convergence is not achieved after 50 iterations, print error message and stop.

C. Compute the auxillaries (notation from Escobal).

$S_e = \sin E_p$	Computer Variables	SE
$C_e = \cos E_p$		CE
$a\dot{E} = \dfrac{na}{1 - {}_e C_e}$		CON

D. Compute the state vector in the x', y', z' system:

$$\begin{aligned} x' &= a(C_e - e) \\ y' &= a(1 - e^2)^{\frac{1}{2}} S_e \\ z' &= 0 \end{aligned}$$ XOR Vector

$$\begin{aligned} \overset{\circ}{x}' &= -a\dot{E} \cdot S_e \\ \overset{\circ}{y}' &= a\overset{\circ}{E}(1 - e^2)^{\frac{1}{2}} C_e \\ \overset{\circ}{z}' &= 0 \end{aligned}$$ VOR Vector

E. Rotate the above state vector into the XYZ system by (see PREITR).

$$STV = \begin{bmatrix} AM & 0 \\ 0 & AM \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{bmatrix} \equiv \begin{bmatrix} AM & 0 \\ 0 & AM \end{bmatrix} \begin{bmatrix} XOR \\ VOR \end{bmatrix}$$

SUBROUTINES REQUIRED :

O. S. U. Project Library:

1. ONEM

REFERENCES :

A. Escobal, P. R. (1965). "Methods of Orbit Determination,' John Wiley and Sons, New York.

B. Mueller, Ivan I. (1964). "Introduction to Satellite Geodesy Frederick Ungar Publishing Co., New York.

STVITR

```
      SUBROUTINE STVITR (ET,STV,AM,P)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AM(3,3),STV(6),XOR(3),VOR(3),V1(3),V2(3),P(6)
      DATA ONE,XOR(3),VOR(3)/1.D0,2*0.D0/
      EM=DMOD((P(5)+P(6)*(ET-P(2))),6.283185307179586D0)
      E=EM
      DO 1 I=1,50
      DE=(E-P(4)*DSIN(E)-EM)/(ONE-P(4)*DCOS(E))
      IF(DABS(DE).LT.P(1)) GOTO 2
    1 E=E-DE
      WRITE(6,70)DE,E
    2 SE=DSIN(E)
      CE=DCOS(E)
      CON= P(6)*P(3) /(ONE-P(4)*CE)
      XOR(1)=P(3)*(CE-P(4))
      XOR(2)=P(3)*DSQRT(ONE-P(4)*P(4))*SE
      VOR(1)=-CON*SE
      VOR(2)=DSQRT(ONE-P(4)*P(4))*CE*CON
      CALL ONEM (AM,XOR,V1)
      CALL ONEM (AM,VOR,V2)
      DO 3 I=1,3
      STV(I)=V1(I)
    3 STV(I+3)=V2(I)
      RETURN
   70 FORMAT(10X,'NO CONVERGENCE AFTER 50 ITERATIONS',2D24.16)
      END
```

SUBROUTINE : STVKEP

CALL STATEMENT : STVKEP (ET, STV)

SUBROUTINE PURPOSE : To calculate a state vector in a Keplerian orbit. (This subroutine must be used in conjunction with subroutine PRESTV which generates a series of coefficients in common storage used in the state vector computation.)

INPUT PARAMETERS : (Extended precision.)

A. Scalar.

1. ET is the epoch for which the state vector is required in Julian days minus 2440000.0.

OUTPUT PARAMETERS :

A. Vector (extended precision).

1. STV (6) is the output state vector.

COMMON AREA PARAMETERS :

A. Area /EPLER/ E, E2, RZ, R, F, ELO, ETO, EN, AO, T3. The common area parameters have been described in PRESTV.

PROGRAM DESCRIPTION :

A. The mean anomaly (ℓ) at epoch t (variable_ET) is computed from the mean anomaly at epoch t_o, (which is variable ETO in common storage), and n which is variable EN in common storage:

$$EL = \ell = \ell_o + n(t - t_o)$$

$$\text{where} \quad 0 < |\ell| < 2\pi .$$

B. The S vector is then computed by:

$$S_1 = \sin \ell$$

$$S_2 = 2 \sin^2 \ell = \sin 2\ell$$

$$S_3 = (3 - 4 \sin^2 \ell) \sin \ell = \sin 3\ell$$

$$S_4 = (2 - 4 \sin^2 \ell) 2 \sin \ell \cos \ell = \sin 4\ell$$

$$\vdots$$

$$S_7 = \sin 7\ell .$$

C. The C vector is then computed from:

$$\begin{aligned} C_1 &= \cos \ell \\ C_2 &= (2 - 4\sin^2 \ell)\tfrac{1}{2} = 1 - 2\sin^2 \ell = \cos 2\ell \\ &\vdots \\ C_7 &= \cos 7\ell \, . \end{aligned}$$

D. The radius vector $(\frac{r}{a})$ is computed from [Brouwer and Clemence, 1961] Equation 73, Page 76:

$$\frac{r}{a} = \sum_{i=1}^{7} R_i \, C_i \; .$$

E. The true anomaly (f) is computed from [Brouwer and Clemence, 1961] Equation 75, Page 77:

$$f = \ell + \sum_{i=1}^{7} F_i \, S_i \; .$$

F. Note in the above expressions that the R and F coefficients exclude terms greater than e^7, thus the effect of the exclusion should be investigated before using this subroutine.

G. The radius (r) is then computed from:

$$r = \frac{r}{a} \cdot a \; .$$

H. The Cartesian coordinates in the orbital plane are then computed from:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{matrix} = \\ = \\ = \end{matrix} \begin{bmatrix} r\cos f \\ r\sin f \\ 0 \end{bmatrix} = \vec{x}$$

I. The time derivatives are then formed from:

$$\begin{bmatrix} \dot{x} \\ \\ \dot{y} \\ \\ \dot{z} \end{bmatrix} \begin{matrix} = \\ \\ = \\ \\ = \end{matrix} \begin{bmatrix} -na \dfrac{y}{r\sqrt{1-e^2}} \\ \\ na \dfrac{x + ea\sqrt{1-e^2}}{r} \\ \\ 0 \end{bmatrix} = \vec{v}$$

J. The coordinates and time derivations are then rotated into the XYZ system through the matrix T3 (from PRESTV) which is effectively:

$$T3 = R_3(-\Omega)\, R_1(-i)\, R_3(-\omega)$$

and $$S = \begin{bmatrix} T3 & 0 \\ 0 & T3 \end{bmatrix} \cdot \begin{bmatrix} x \\ v \end{bmatrix}$$

where S is the output vector STV.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. REDPI
2. ONEM

REFERENCES :

Brouwer, D. and G. Clemence (1961). "Methods of Celestial Mechanics," Academic Press, New York, pp. 76-77.

STVKEP

```
      SUBROUTINE STVKEP (ET,STV)
C     CALCULATION OF STATE VECTOR IN A KEPLERIAN ORBIT
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION R(7),F(7),S(7),C(7),T3(3,3),XOR(3),VOR(3),POS(3),T2(3),S
     PTV(6)
      COMMON /EPLER/ E,E2,RZ,R,F,ELO,ETO,EN,AO,T3
      DATA ONE,TWO,TRI,FOR,FIF,SEV/1.D0,2.D0,3.D0,4.D0,5.D0,7.D0/
C     MULTIPLE SINES AND COSINES OF MEAN ANOMALY
      EL=ELO+EN*(ET-ETO)
      CALL REDPI (EL)
      SL=DSIN(EL)
      CL=DCOS(EL)
      SLQ=FOR*SL*SL
      CLQ=FOR-SLQ
      SCL=TWO*SL*CL
      S(1)=SL
      S(2)=SCL
      S(3)=SL*(TRI-SLQ)
      S(4)=SCL*(TWO-SLQ)
      S(5)=SL*(FIF-SLQ*(FIF-SLQ))
      S(6)=SCL*(TRI-CLQ*(FOR-CLQ))
      S(7)=SL*(SEV-SLC*(14.D0-SLO*(SEV-SLQ)))
      C(1)=CL
      C(2)=.5D0*(TWO-SLQ)
      C(3)=CL*(CLQ-TRI)
      C(4)=ONE-CLQ*.5D0*(FOR-CLQ)
      C(5)=CL*(FIF-CLQ*(FIF-CLQ))
      C(6)=CLC*.5D0*(SEV+TWO-CLQ*(FIF+ONE-CLQ))-ONE
      C(7)=CL*(CLQ*(14.D0-CLQ*(SEV-CLQ))-SEV)
C     CALCULATION OF RADIUS VECTOR AND TRUE ANOMALY
      RV=RZ
      TA=EL
      DO 1 I=1,7
      RV=RV+R(I)*C(I)
    1 TA=TA+F(I)*S(I)
      RV=RV*AO
      XOR(1)=DCOS(TA)*RV
      XOR(2)=DSIN(TA)*RV
      XOR(3)=0.D0
      VOR(1)=-EN*AO*XOR(2)/(RV*DSQRT(ONE-E2))
      VOR(2)=EN*AO*(XOR(1)+E*AO)*DSQRT(ONE-E2)/RV
      VOR(3)=0.D0
      CALL ONEM (T3,XOR,POS)
      CALL ONEM (T3,VOR,T2)
      DO 2 I=1,3
      J=I+3
      STV(I)=POS(I)
    2 STV(J)=T2(I)
      RETURN
      END
C
```

SUBROUTINE : TRIK

CALL STATEMENT : TRIK (Y, YP, TETA, FETA)

SUBROUTINE PURPOSE : The subroutine is used with either FUNPL5 or FUNPL6 and provides the second time derivatives of the physical libration parameters and the Θ, Φ partial derivative matrices (see [Papo, 1971] Pages 99-106).

INPUT PARAMETERS :

- A. Vector (extended precision).
  - 1. Y is a 6-element vector containing the Eulerian angle of the Moon and their time derivatives as input throug FUNPL5 or FUNPL6.

OUTPUT PARAMETERS :

- A. Vector (extended precision).
  - 1. YP is a 3-element vector containing the second time derivatives of the physical libration angles (λ, δ, ρ).
- B. Matrices (extended precision).
  - 1. TETA is a 6×6 matrix which is the Θ partial derivative array evaluated at ET [Papo, 1971].
  - 2. FETA is a 6×3 matrix which is the Φ partial derivativ array evaluated at ET [Papo, 1971].

COMMON AREA PARAMETERS : Common areas are used extensively to transfer data in and out of this subroutine, the calling subroutine (FUNPL5 or FUNPL6) and the main program.

- A. Area /MAIF/ALF, BET, GAM, TEQ, CCE, CCS, P, G.
  - 1. ALF, BET, GAM, TEQ are described in subroutine documentation for FUNPL5.
  - 2. CCE is an extended precision scalar which contains the gravitational constant of the earth.
  - 3. CCS is an extended precision scalar which contains the gravitational constant of the sun.
  - 4. P is an extended precision 3-element vector containing th rotational velocities e_x, e_y, e_z respectively of the ecliptic coordinate system defined in [Papo, 1971] Page 83.

5. G is an extended precision 3×3 matrix containing partial derivatives of τ, σ, ρ, with respect to C_{22}, β, C_{20} respectively as defined in [Papo, 1971] Equation 3.32.12, Page 101.

B. Area /EXE/ET, B, SUN.

1. The above parameters correspond to the parameters TO, B, SUN described in the FUNPL5 subroutine documentation.

C. Area /ALL/SEV, ESH, NUT, KLU.

1. The above parameters correspond to the parameters SEV, A, NUT, KLU described in the FUNPL5 subroutine documentation.

PROGRAM MATHEMATICS : The expressions used to evaluate the second partial derivatives are given in [Papo, 1971] Pages 99-101. Comments are included in the statement listings to refer the next segment of statements to specific equations given in [Papo, 1971].

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. ROTATE
2. TRIM
3. ONEM
4. LUCA

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

TRIK

```
      SUBROUTINE TRIK (Y,YP,TETA,FETA)
C     SUBROUTINE TRIK FOR CALCULATING EQUATIONS OF MOTION OF THE PHYSICAL
C     LIBRATION ANGLES AND THE PARTIALS FOR GENERATING STATE TRANSITION
C     AND PARAMETER SENSITIVITY MATRICES
C     PROGRAMMING SECTION 3.8 IN THE DISSERTATION (PAPO)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Y(6),YP(3)      ,SEV(6),P(3),B(3),SUN(3),EM1IN(3,3),D(3),
     PPOP(3),OME(3),PRE(3),T1(3,3),T2(3,3),T3(3,3),T4(3;3),TEM1(3),
     PTEM2(3),TEM3(3),T5(3,3),T6(3,3),T7(3,3),T8(3,3),SUND(3),DDD(3,3)
     P,DDD(3,3),TETA(6,6),A(3),E(3,3),AVI(3,3),RAC(3,3)
     P,G(3,3),BO(3,3),FETA(6,3)
      COMMON/MAIF/ALF,BET,GAM,TEQ,CCE,CCS,P,G/EXE/ET,B,SUN/ALL/  SEV,ESH
     P,NUT,KLU
      DATA EM1IN(1,3),EM1IN(2,3),EM1IN(3,3),PRE(3),ONE/1.D0,3*0.D0,1.D0/
C     Y - EULERIAN ANGLES OF MOON
C     SEV - (1) LONG (2) OMEGA (3) LONG DOT (4) OMEGA DOT (5) LONG DDOT (6) OMDD
C     G - MATRIX OF PARTIALS ALF,BET,GAM / C22,BET,C20
C     TEQ - MEAN INCLINATION OF MOON EQUATOR
C     P - ECLIPTIC ROTATION RATE VECTOR
C     B,SUN - EARTH AND SUN POSITION IN ECLIPTIC SYSTEM
      CALLROTATE(3,Y(1),T1)
      CALLROTATE(1,-Y(3),T2)
      CALLROTATE(3,Y(2),T3)
      CALL TRIM (T1,T2,T4)
      CALL TRIM (T4,T3,T5)
C     T5 - TRANSFORMATION MATRIX M
      STEIN=ONE/T2(3,2)
C     STEIN = 1 / SIN (TETA)
      EM1IN(1,1)=T1(1,2)*T2(3,3)*STEIN
      EM1IN(1,2)=T1(1,1)*T2(3,3)*STEIN
      EM1IN(2,1)=-T1(1,2)*STEIN
      EM1IN(2,2)=-T1(1,1)*STEIN
      EM1IN(3,1)=-T1(1,1)
      EM1IN(3,2)=T1(1,2)
C     CREATING W(-1)
C     D : NEGATIVE COORDINATES OF EARTH IN TRUE SELENOGRAPHIC
      CALL ONEM (T5,B,D)
C     SUND : COORDINATES OF SUN IN TRUE SELENOGRAPHIC
      CALL ONEM (T5,SUN,SUND)
      ZE=0.D0
      ZS=0.D0
      DO 1 I=1,3
      DO 6 J=1,6
    6 TETA(I,J)=0.D0
      DO 7 J=1,3
    7 FETA(I,J)=0.D0
      ZS=ZS+SUND(I)*SUND(I)
    1 ZE=ZE+D(I)*D(I)
      TETA(1,4)=1.D0
      TETA(2,5)=1.D0
      TETA(3,6)=1.D0
```

```
C     FORMATION OF TRIVIAL PARTS IN CAPITAL FI (FETA) AND TETA
      ZE=CCE/ZE**2.5D0
      ZS=CCS/ZS**2.5D0
C     CALCULATING THE EFFECT OF THE MOTION OF THE ECLIPTIC
      R=Y(2)
      DR=Y(5)
      CR=DCOS(R)
      SR=DSIN(R)
      POP(1)=-Y(6)+CR*P(1)+SR*P(2)
      POP(2)=      -SR*P(1)+CR*P(2)
      POP(3)=Y(5)+P(3)
C     POP - IS P(0)
      CALL ONEM   (T4,POP,OME)
      OME(3)=OME(3)+Y(4)
C     OME - RATES OF ROTATION ABOUT SELENOGRAPHIC AXES
      PRE(1)=DR*(-SR*P(1)+CR*P(2))
      PRE(2)=DR*(-CR*P(1)-SR*P(2))
C     PRE - PSIDOT * DR3(PSI) / DPSI * P
      CALL ONEM   (T4,PRE,TEM1)
C     TEM1 IS T(C)
      CALL LUCA (3,Y(4),T6)
      CALL TRIM (T6,T4,T7)
      CALL LUCA (1,-Y(6),T6)
C     T6 IS T(B)
      CALL TRIM (T4,T6,T8)
      CALL DGMADD (T7,T8,T6,3,3)
      CALL ONEM (T6,POP,TEM2)
C     TEM2 IS T(B) * P(0)
      A(1)=  ZE*D(2)*D(3)+ZS*SUND(2)*SUND(3)-OME(2)*OME(3)
      A(2)=-(ZE*D(1)*D(3)+ZS*SUND(1)*SUND(3)-OME(1)*OME(3))
      A(3)=  ZE*D(1)*D(2)+ZS*SUND(1)*SUND(2)-OME(1)*OME(2)
C     A IS T(A)
      TEM3(1)=ALF*A(1)-TEM2(1)-TEM1(1)
      TEM3(2)=BET*A(2)-TEM2(2)-TEM1(2)
      TEM3(3)=GAM*A(3)-TEM2(3)-TEM1(3)
C     TEM3 IS T(D)
      IF(KLU.GT.1) GOTO 5
C     IF NO GENERATION OF PARTIALS IS NEEDED KLU IS GT. THAN 1
C
      G11=OME(2)
      G21=-OME(1)
      G12=TEM1(1)/DR
      G22=TEM1(2)/DR
      G32=TEM1(3)/DR
      G13=T4(1,3)*POP(2)-T4(1,2)*POP(3)
      G23=T4(2,3)*POP(2)-T4(2,2)*POP(3)
      G33=T4(3,3)*POP(2)-T4(3,2)*POP(3)
C     G123 ARE G(I) ROW VECTORS
C
      DDD(1,1)=-OME(2)*(ONE+ALF)
      DDD(2,1)= OME(1)*(ONE+BET)
      DDD(3,1)=0.D0
```

```
      DDD(1,2)=-ALF*(T4(2,3)*OME(3)+T4(3,3)*OME(2))-G12-T6(1,3)
      DDD(2,2)= BET*(T4(1,3)*OME(3)+T4(3,3)*OME(1))-G22-T6(2,3)
      DDD(3,2)=-GAM*(T4(1,3)*OME(2)+T4(2,3)*OME(1))-G32-T6(3,3)
      DDD(1,3)= ALF*(T4(2,1)*OME(3)+T4(3,1)*OME(2))-G13+T6(1,1)
      DDD(2,3)=-BET*(T4(1,1)*OME(3)+T4(3,1)*OME(1))-G23+T6(2,1)
      DDD(3,3)= GAM*(T4(1,1)*OME(2)+T4(2,1)*OME(1))-G33+T6(3,1)
C     DDD IS DT(D) / D EULERIAN DOTED
C
      CALL TRIM (EM11N,DDD,AVI)
C     AVI ARE PARTIALS OF EULERIAN DOUBLE DOTS VS. EULERIAN DOTS
C
      C11=ZE*(T5(2,1)*B  (1)+T5(2,2)*B  (2)+T5(2,3)*B  (3))
      C21=ZE*(-T5(1,1)*B  (1)-T5(1,2)*B  (2)-T5(1,3)*B  (3))
      C12=ZE*(-T5(1,2)*B  (1)+T5(1,1)*B  (2))
      C22=ZE*(-T5(2,2)*B  (1)+T5(2,1)*B  (2))
      C32=ZE*(-T5(3,2)*B  (1)+T5(3,1)*B  (2))
      C13=ZE*(T4(1,3)*T3(2,1)*B(1)+T4(1,3)*T3(2,2)*B(2)-T4(1,2)*B(3))
      C23=ZE*(T4(2,3)*T3(2,1)*B(1)+T4(2,3)*T3(2,2)*B(2)-T4(2,2)*B(3))
      C33=ZE*(T4(3,3)*T3(2,1)*B(1)+T4(3,3)*T3(2,2)*B(2)-T4(3,2)*B(3))
C     C123 ARE C(I) ROWS OF EARTH
      H11=ZS*(T5(2,1)*SUN(1)+T5(2,2)*SUN(2)+T5(2,3)*SUN(3))
      H21=ZS*(-T5(1,1)*SUN(1)-T5(1,2)*SUN(2)-T5(1,3)*SUN(3))
      H12=ZS*(-T5(1,2)*SUN(1)+T5(1,1)*SUN(2))
      H22=ZS*(-T5(2,2)*SUN(1)+T5(2,1)*SUN(2))
      H32=ZS*(-T5(3,2)*SUN(1)+T5(3,1)*SUN(2))
      H13=ZS*(T4(1,3)*T3(2,1)*SUN(1)+T4(1,3)*T3(2,2)*SUN(2)-T4(1,2)*SUN
     P(3))
      H23=ZS*(T4(2,3)*T3(2,1)*SUN(1)+T4(2,3)*T3(2,2)*SUN(2)-T4(2,2)*SUN
     P(3))
      H33=ZS*(T4(3,3)*T3(2,1)*SUN(1)+T4(3,3)*T3(2,2)*SUN(2)-T4(3,2)*SUN
     P(3))
C     H123 ARE C(I) ROWS OF SUN
C
      DOD(1,1)= ALF*(C21*D(3)+H21*SUND(3)-G21*OME(3))-TEM2(2)-TEM1(2)
      DOD(2,1)=-BET*(C11*D(3)+H11*SUND(3)-G11*OME(3))+TEM2(1)+TEM1(1)
      DOD(3,1)= GAM*(C11*D(2)+C21*D(1)+H11*SUND(2)+H21*SUND(1)-G11*OME(2
     P)-G21*OME(1))
      DOD(1,2)= ALF*(C22*D(3)+C32*D(2)+H22*SUND(3)+H32*SUND(2)-G22*OME(
     P3 )-G32*OME(2))+T4(1,2)*PRE(1)-T4(1,1)*PRE(2)-(T6(1,1)*PRE(1)
     P+T6(1,2)*PRE(2))/DR
      DOD(2,2)=-BET*(C12*D(3)+C32*D(1)+H12*SUND(3)+H32*SUND(1)-G12*OME
     P(3)-G32*OME(1))+T4(2,2)*PRE(1)-T4(2,1)*PRE(2)-(T6(2,1)*PRE(1)
     P+T6(2,2)*PRE(2))/DR
      DOD(3,2)= GAM*(C12*D(2)+C22*D(1)+H12*SUND(2)+H22*SUND(1)-G12*OME
     P(2)-G22*OME(1))+T4(3,2)*PRE(1)-T4(3,1)*PRE(2)-(T6(3,1)*PRE(1)
     P+T6(3,2)*PRE(2))/DR
      DOD(1,3)= ALF*(C23*D(3)+C33*D(2)+H23*SUND(3)+H33*SUND(2)-G23*OME(3
     P)-G33*OME(2))-T4(1,3)*PRE(2)+T4(1,2)*PRE(3)+T6(1,2)*POP(3)-T6(1,3)
     P*POP(2)
      DOD(2,3)=-BET*(C13*D(3)+C33*D(1)+H13*SUND(3)+H33*SUND(1)-G13*OME(3
```

```
     P)-G33*OME(1))-T4(2,3)*PRE(2)+T4(2,2)*PRE(3)+T6(2,2)*POP(3)-T6(2,3)
     P*POP(2)
      DOD(3,3)= GAM*(C13*D(2)+C23*D(1)+H13*SUND(2)+H23*SUND(1)-G13*OME(2
     P)-G23*OME(1))-T4(3,3)*PRE(2)+T4(3,2)*PRE(3)+T6(3,2)*POP(3)-T6(3,3)
     P*POP(2)
C     DOD ARE PARTIALS OF T(D) VS. EULERIAN ANGLES
C
      CALL TRIM (EM1IN,DOD,T7)
C
      RAC(1,1)=(T1(1,1)*T2(3,3)*TEM3(1)-T1(1,2)*T2(3,3)*TEM3(2))*STEIN
     P+T7(1,1)
      RAC(2,1)=(-T1(1,1)*TEM3(1)+T1(1,2)*TEM3(2))*STEIN+T7(2,1)
      RAC(3,1)=T1(1,2)*TEM3(1) +T1(1,1)*TEM3(2)+T7(3,1)
      RAC(1,2)=T7(1,2)
      RAC(2,2)=T7(2,2)
      RAC(3,2)=T7(3,2)
      RAC(1,3)=(-T1(1,2)*TEM3(1)-T1(1,1)*TEM3(2))*STEIN*STEIN+T7(1,3)
      RAC(2,3)=(T1(1,2)*T2(3,3)*TEM3(1)+T1(1,1)*T2(3,3)*TEM3(2))
     P*STEIN*STEIN+T7(2,3)
      RAC(3,3)=T7(3,3)
C     RAC ARE PARTIALS OF EULERIAN DOUBLE DOTS VS. EULERIAN ANGLES
C
      TETA(4,1)=RAC(1,1)+RAC(2,1)
      TETA(4,2)=RAC(1,2)+RAC(2,2)-TETA(4,1)
      TETA(4,3)=RAC(1,3)+RAC(2,3)
      TETA(4,4)=AVI(1,1)+AVI(2,1)
      TETA(4,5)=AVI(1,2)+AVI(2,2)-TETA(4,4)
      TETA(4,6)=AVI(1,3)+AVI(2,3)
      TETA(5,1)=RAC(2,1)
      TETA(5,2)=RAC(2,2)-RAC(2,1)
      TETA(5,3)=RAC(2,3)
      TETA(5,4)=AVI(2,1)
      TETA(5,5)=AVI(2,2)-AVI(2,1)
      TETA(5,6)=AVI(2,3)
      TETA(6,1)=RAC(3,1)
      TETA(6,2)=RAC(3,2)-RAC(3,1)
      TETA(6,3)=RAC(3,3)
      TETA(6,4)=AVI(3,1)
      TETA(6,5)=AVI(3,2)-AVI(3,1)
      TETA(6,6)=AVI(3,3)
C     FILLING IN THE 4,5,6 ROWS OF CAPITAL TETA
C
      DO 3 I=1,3
      DO 2 J=1,3
    2 E(J,I)=EM1IN(J,I)*A(I)
    3 E(1,I)=E(1,I)+E(2,I)
      CALL TRIM (E,G,BO)
      DO 4 I=1,3
      DO 4 J=1,3
```

```
    4 FETA(I+3,J)=SO(I,J)
C       FILLING IN THE 4,5,6 ROWS OF FETA (CAPITAL FI)
    5 CALL ONEM (EM1IN,TEM3,TEM1)
      YP(1)=TEM1(1)-SEV(5)+TEM1(2)
      YP(2)=TEM1(2)-SEV(6)
      YP(3)=TEM1(3)
C     YP - ENCKE EQUATIONS OF MOTION OF PHYSICAL LIBRATION ANGLES
      RETURN
      END
```

SUBROUTINE : TRIM

CALL STATEMENT : TRIM (A, B, C)

SUBROUTINE PURPOSE : Multiplies two 3x3 matrices, A and B, to form a 3×3 product C.

INPUT PARAMETERS :

A. Matrix A (3, 3)
B. Matrix B (3, 3)

OUTPUT PARAMETERS :

A. Matrix C (3, 3)

PROGRAM DESCRIPTION :

A. ${}_3C_3 = {}_3A_3\,{}_3B_3$

SUBROUTINES REQUIRED : None

REFERENCES : None

TRIM

```
  SUBROUTINE TRIM (A,B,C)
  REAL*8A(3,3),B(3,3),C(3,3)
  DO1I=1,3
  DO1J=1,3
1 C(I,J)=A(I,1)*B(1,J)+A(I,2)*B(2,J)+A(I,3)*B(3,J)
  RETURN
  END
```

DESCRIPTION OF MAIN PROGRAMS

MAIN PROGRAM INDEX

Preceding page blank

MAIN PROGRAM : A

PROGRAM PURPOSE : Generation of the simulated ephemeris of the earth-moon system described in [Papo, 1971].

INPUT PARAMETERS : An example of card input parameters is given in Appendix A/2.

- A. Vectors (extended precision).
  - 1. PAR is a 5-element vector containing:
    - a. The initial epoch in Julian days minus 2440000.0.
    - b. The final epoch is Julian days minus 2440000.0.
    - c. The initial step size for the integration in days.
    - d. Elements 4 and 5 are not used.
- B. Scalars (single precision).
  - 1. HMIN is the minimum step size to be allowed in the integration routine.
  - 2. HMAX is the maximum step size to be allowed in the integration routine.
  - 3. RAT1 is the relative accuracy required (number of correct significant figures) for the geocentric state vector ot the moon.
  - 4. RAT2 is the relative accuracy required for the numerical integration of the earth and moon Eulerian angles.

OUTPUT PARAMETERS :

- A. The main output from the routine is the state vector of the earth-moon system, the Eulerian angles of the system and time derivatives of the vectors and angles. The units and sequence of the quantities are described in the subroutine FUNEPH description (for the Y and YP vectors) and the subroutine OUTEPH (for the FD matrix layered into the FH array). In addition the FH array of the simulated ephemeris is written on a direct access disk device identified in the program as unit 4.

PROGRAM DESCRIPTION : A flow chart of the program logic is given in Appendix A/1.

Preceding page blank

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. FUNEPH
2. EKHARD
3. OUTEPH
4. PREITR
5. DVDPFI
6. MEANAN
7. REDPI
8. SEFODI

REFERENCES :

Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

Program A

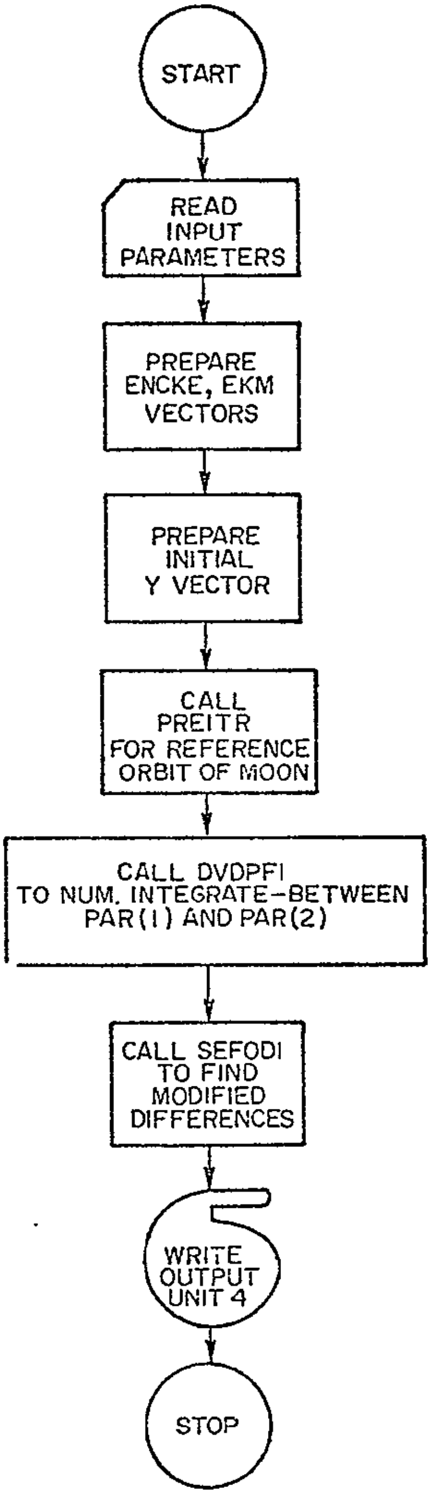
```
C     GENERATION OF SIMULATED EPHEMERIS OF THE MOON+EARTH SYSTEM
      IMPLICITREAL*8(A-H,O-Z)
      REAL*4 EP,HMIN,HMAX,RAT1,RAT2
      DIMENSION Y(18),YP(18),KQ(18),DT(20,18),ENCKE(7),EKM(6),AMM(3,3)
     P,PM(6),EP(18),YNN(18),KD(18),PAR(5),PLA(3),DNM(6),ANM(6)
      DIMENSION FF(9,18),FD(3,18),FH(3,18,15)
      COMMON/MAIFUN/ENCKE,AMM,PM
      EXTERNAL FUNEPH,OUTEPH
      DATAZERO,PI,EKM(3),EKM(4)/0.D0,3.141592653589793D0,.08980D0,.0549D
     P0/
      READ(5,61) PAR
      READ(5,61) HMIN,HMAX,DELT,RAT1,RAT2
      D0=PAR(1)
      EJD=D0+2440000.D0
      CET=(D0+24980.D0)/36525.D0
      CALL MEANAN (EJD,ANM,DNM)
      ENCKE(1)=D0
      ENCKE(2)=PI+ANM(2)-ANM(5)
      CALL REDPI (ENCKE(2))
      ENCKE(3)=ANM(5)
      ENCKE(4)=.23D0
      ENCKE(5)=(DMOD(D0,1.D0)*2.D0+(23925.836D0+8640184.542D0*CET+.0929D
     P0*CET*CET)/43200.D0)*PI
      CALL REDPI (ENCKE(5))
      ENCKE(6)=.40916D0
      ENCKE(7)=6.300388098D0
      EKM(1)=ANM(5)
      CALL REDPI (EKM(1))
      EKM(2)=ANM(4)-ANM(5)
      CALL REDPI (EKM(2))
      EKM(5)=ANM(2)-ANM(4)
      CALL REDPI (EKM(5))
      EKM(6)=DNM(2)
      CALL EKHARD (EJD,DNM,PLA)
      DO 1 I=1,6
      Y(I)=ZERO
      Y(I+6)=DNM(I)
    1 Y(I+12)=ZERO
      Y(7)=Y(7)-ENCKE(2)
      Y(8)=Y(8)-ENCKE(3)
      Y(10)=Y(10)-ENCKE(4)
      WRITE(6,76)Y,YP     ,PAR,ENCKE,EKM
      EMUM=3012159753997540.D0
      CALLPREITR (D0,EMUM,EKM,12,AMM,PM)
      REWIND 1
      CALL DVDPF1 (18,Y,YP,KD,EP,RAT1,RAT2,HMIN,HMAX,DELT,KST,IHL,PAR
     P,FUNEPH,OUTEPH,KQ,YNN,DT,0)
      WRITE(6,76) ENCKE,AMM,PM
```

Program A (Cont)

```
   CALL SEFODI (1,2,18,753,9,3,FF,FD)
   REWIND 4
   ET=215.5D0
   DO 33 I=1,53
   DO 32 J=1,15
   READ (2) FD
   DO 31 M=1,18
   DO 31 N=1,3
31 FH(N,M,J)=FD(N,M)
32 CONTINUE
   ET=ET+7.D0
   WRITE(4) ET,FH
   BACKSPACE 2
33 CONTINUE
   REWIND 4
   READ(4) ET,FH
   WRITE(6,81) ET,FH
   DO 34 I=1,50
34 READ(4) ET
   READ(4) ET,FH
   WRITE(6,81) ET,FH
   STOP
61 FORMAT(6F13.6,2X)
76 FORMAT(5X,6D19.11)
81 FORMAT(//10X,F10.3/,(2X,6D19.11))
   END
```

APPENDIX A/1

Flowchart of the Simulated Ephemeris Generator



APPENDIX A/2

Sample Input

```
220.5          596.5               .5       .000001
.005           .5                  .5       .000001       .00000001
```

MAIN PROGRAM : B

PROGRAM PURPOSE : To generate optical observations of lunar points.

INPUT PARAMETERS : Examples of card input are given in Appendix B/2.

A. Simulated lunar ephemeris.
   1. The simulated ephemeris is assumed to be stored on a direct access disk device designated unit 4 (see the main program description for ephemeris generation).

B. Scalars.
   1. ST1, ST2 and ST3 extended precision variables are described in the subroutine description OPTOBS. Basically, they define the conditions under which a point can be observed.
   2. JOK is a single precision integer variable denoting the type of observation (see OPTOBS).
   3. APP is an extended precision variable which contains one-half of the field angle in degrees.
   4. ET is the epoch of observation in Julian days minus 2440000.

C. Vector (extended precision).
   1. X is a 9-element vector input only if the observing station is on a satellite which was not the case in the example given. For examples of satellite borne observa see [Papo, 1971].

OUTPUT PARAMETERS :

A. The subroutine computes the simulated optical observations of the moon for a bundle of nominally 30 observations of lunar points. The angles created are printed (unit 6) and punched (unit 7).

PROGRAM DESCRIPTION : A flow chart of the program logic is given in Appendix B/1.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. EPHITL
   2. OPTOBS

REFERENCES :

A. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

B. Sprague, M. (1961). "An Investigation to Improve Selenodetic Control on the Lunar Far Side Using Apollo Mission Trans-Earth Photography," Department of Geodetic Science, Report No. 155, June.

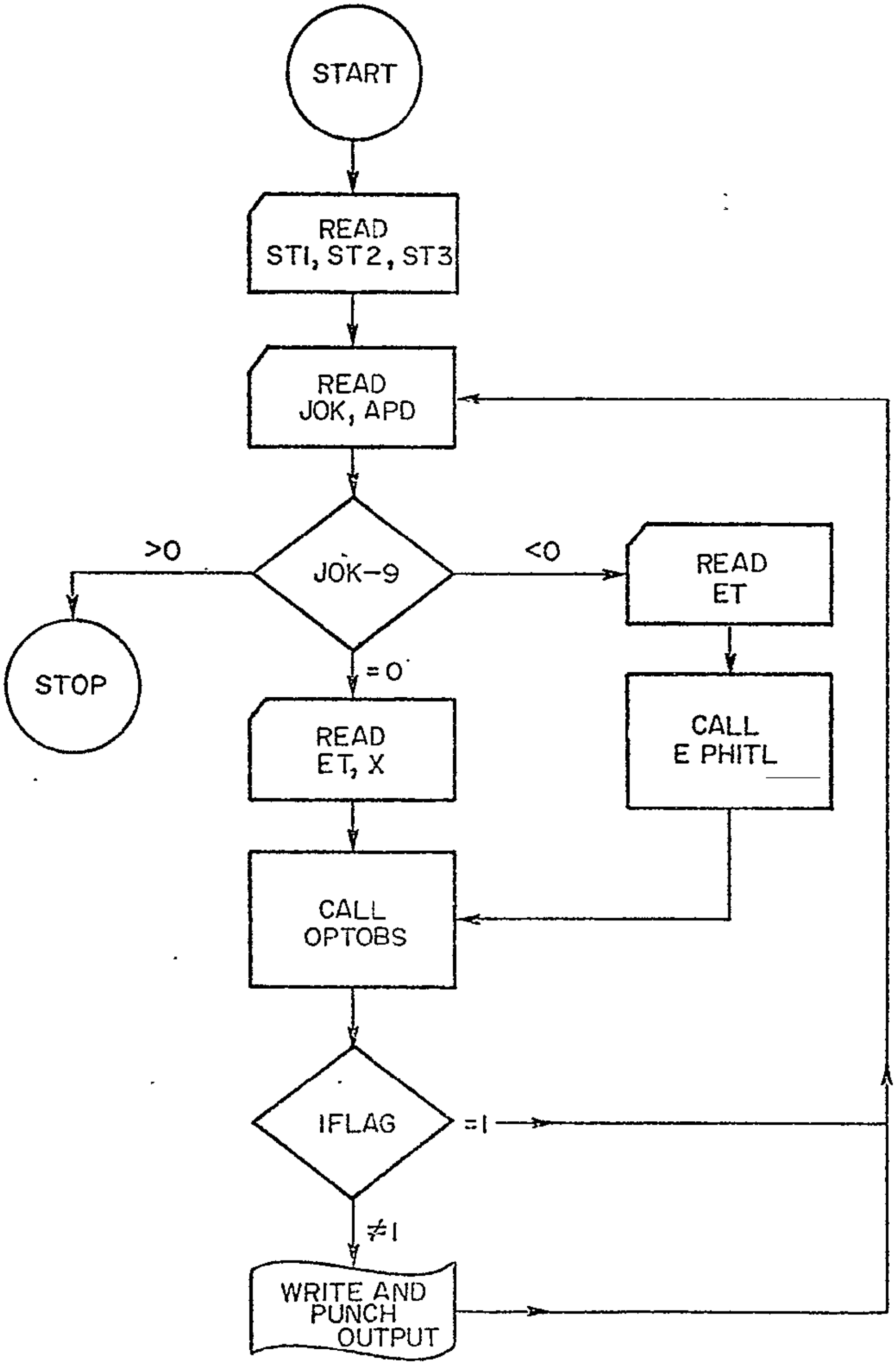
Program B

```
C     MAIN PROGRAM FOR GENERATING OPTICAL OBSERVATIONS OF THE MOON
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION X(9),BUNDLE(2,30),PCSP(3),Y(18),TB(3,3)
      DIMENSION JOKK(2),ISATT(2),IPOINT(2),UNU(2),AKA(2)
      DIMENSION FH(3,16,15)
      COMMON /OPTO/ TB, WTER,SSUN,ETER,PCSP,ST1,ST2,ST3,IFLAG
      COMMON /EPHEM/ ETE,FH
      REWIND 4
      READ (4) ETE,FH
C
      ISET=1
      READ(5,60)ST1,ST2,ST3
    1 READ(5,61,END=99) JOK,APP
C     APP - SIN OF FIELD ANGLE / 2
      IF(JOK-9) 4,5,99
    4 READ(5,60,END=99)ET
      CALL EPHITL (ET,4,Y)
      DO 3 M=1,3
      X(M)=Y(M)
      X(M+3)=Y(M+6)
    3 X(M+6)=Y(M+12)
      GOTO6
    5 READ(5,60,END=99)ET,X
C
    6 CALL OPTOBS (ET,X,JOK,APP,BUNDLE)
      IF(IFLAG-1) 2,1,2
    2 WRITE(6,70)ISET,ET
      WRITE(6,75)WTER,SSUN,ETER
      ISET=ISET+1
      IF(JOK.EQ.9) GOTO 7
      R=DSQRT (PCSP(1)**2+PCSP(2)**2)
      ALLON=DATAN2 (PCSP(2),PCSP(1))*57.295779500
      ALLAT=DATAN2 (PCSP(3),R)*57.2957795D0
      WRITE(6,72)JOK
      WRITE(6,77)ALLON,ALLAT
      GOTO 8
    7 WRITE(6,73)
    8 WRITE(6,76)PCSP
      WRITE(6,78)TB
      WRITE(6,71)(I,(BUNDLE(J,I),J=1,2),I=1,30)
      ISAT=JOK/9
      KOUNT=1
      DO 9 I=1,30
      IF(BUNDLE(1,I).EQ.0.D0) GOTO 9
      KOUNT=KOUNT+1
      IM=MOD(KOUNT,2)+1
      IPOINT(IM)=I
      UNU(IM)=BUNDLE(1,I)
      AKA(IM)=BUNDLE(2,I)
      IF(IM.NE.2) GOTO 9
      WRITE(7,79) ET,JOK,ISAT,IPOINT(1),UNU(1),AKA(1),ET,JOK,ISAT,IPOINT
```

```
     P(2),UNU(2),AKA(2)
    9 CONTINUE
      IF(IM.EQ.2) GOTO 10
      WRITE(7,79) ET,JOK,ISAT,IPOINT(1),UNU(1),AKA(1)
   10 CONTINUE
      WRITE(7,80)(ET,JOK,ISAT,(TB(I,J),I=1,3),ISET,J=1,3)
      WRITE(7,80)ET,JOK,ISAT,PCSP,ISET
      WRITE(6,81) X
      GOTO 1
C
   99 STOP
   60 FORMAT(5F15.10)
   61 FCRMAT( I10,F10.5)
   70 FORMAT(1H1    /10X,'SET NUMBER',I4,10X,'EPOCH OF OBSERVATION IN (JD
     P - 2440000.) IS',F17.9/ )
   71 FORMAT( 10X,I5,2D25.16)
   72 FORMAT(10X,'STATICN OBSERVING IS OBSERVATORY NO',I3/ )
   73 FORMAT(10X,'STATICN OBSERVING IS ON A SATELLITE'/ )
   74 FORMAT(2(10X,5D20.10/))
   75 FORMAT(10X,'SELENOGRAPHIC LONGITUDE OF TERMINATORS AND SUBSOLAR PO
     PINT'/ 10X,3F1C.3/ )
   76 FORMAT(10X,'SELENOGRAPHIC COORDINATES OF PROJECTION CENTER'/ 10X,3
     PF15.3/ )
   77 FORMAT(10X,'TOTAL LIBRATION : IN LONGITUDE',F8.3,'  - IN LATITUDE'
     P,F8.3/)
   78 FORMAT(10X,'ORTHOGONAL TRANSFORMATION MATRIX FROM THE B1B2B3 TO TH
     PE ECLIPTIC SYSTEM'/3(12X,3D25.16/)/10X,'OPTICAL OBSERVATIONS NU AN
     PD KAPPA'/)
   79 FORMAT(2(F6.2,I1,2I2,D13.6,D16.9))
   80 FORMAT(F6.2,I1,I2,3D23.16,I2)
   81 FORMAT(/10X,'SIMULATED LUNAR EPHEMERIS AND EULERIAN ANGLES , EARTH
     P EULERIAN ANGLFS'/3(12X,3D25.16/))
      END
C
```

APPENDIX B/1

Flowchart for Generating Optical Observations of Lunar Points



APPENDIX B/2

Sample Input

```
-0.08          -0.34          .34
         3.007                               3.007
  224.93                              428.87
         2.007                               3.007
  225.01                              430.92
         3.007                               2.007
  254.92                              431.
         2.007                               3.007
  255.                                459.93
         3.007                               2.007
  283.9                               460.01
         2.007                               3.007
  283.98                              461.98
         3.007                               2.007
  312.88                              462.06
         2.007                               3.007
  312.96                              490.96
         3.007                               2.007
  313.93                              491.04
         2.007                               3.007
  314.01                              519.94
         3.007                               2.007
  342.92                              520.02
         2.007                               3.007
  343.0                               548.91
         3.007                               2.007
  371.91                              548.99
         2.007                               3.007
  371.99                              578.9
         3.007                               2.007
  400.9                               578.98
```

MAIN PROGRAM : C

PROGRAM PURPOSE : To adjust the numerically integrated physical librations in the simulated environment of [Papo, 1971] to the forced physical librations given in [Echardt, 1970].

INPUT PARAMETERS : An example of card input to the program is given in Appendix C/2.

A. AU through M are input parameters required by the JPL subroutines for reading the Development Ephemeris (DE - 69) on tape as described in [O'Handley, 1969].
B. PARAM is an extended precision vector with 5 elements which are described below.
   1. The first element is the initial epoch of integration in Julian days minus 2440000.0 days.
   2. The second element is the final epoch of integration in Julian days minus 2440000.0 days.
   3. The third parameter is the step size (in days) used to initialize the numerical integration routine.
   4. The last two parameters are not used as input parameters.
C. OSH is a 6-parameter vector which is used to input the initial values of the physical libration parameters and their time derivatives $(\tau_0, \sigma_0, \rho_0, \dot{\tau}_0, \dot{\sigma}_0, \dot{\rho}_0)$.
D. BE is a 6-element vector which is used to input the weight matrix of the physical libration parameters.
E. ALF, BET, GAM, TEQ are inputs of the dimensionless ratio of inertia of the moon (α, β, γ) and the inclination of the lunar equator to the ecliptic in radians, respectively.
F. The parameters CCE, CCS and P are described in the subroutine FUNPL5.

OUTPUT PARAMETERS :

A. The output of the program is the comparison between the simulated solution described in [Papo, 1971] and the actual case described in [Eckhardt, 1970]. Examples of the output are given in [Papo, 1971].

PROGRAM DESCRIPTION : A flowchart of the program logic is given in AppendixC/1

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. DVDPF
   2. FUNPL5
   3. OUTADJ

B. Fortran Scientific Subroutine Package:
   1. DMINV
   2. DGMPRD

REFERENCES :

A. Eckhardt, D. H. (1970). "Lunar Libration Tables," The Moon, Vol. 1, No. 2, February.

B. O'Handley, Douglas A. et al. (1969). "JPL Development Ephemeris Number 69," Technical Report 32-1465.

C. Papo, Haim B. (1971). "Optimal Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

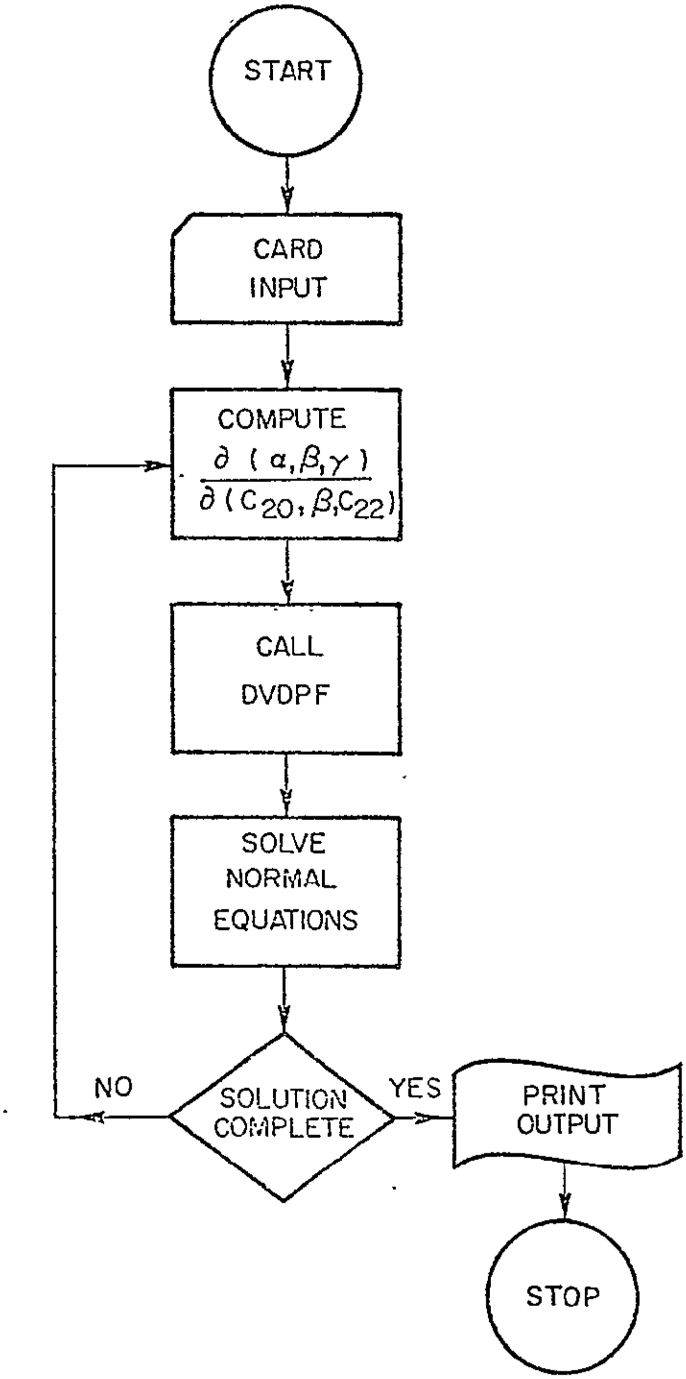
Program C

```
C
C     MAIN FOR ADJUSTING INTEGRATED NUMERICALLY PHYSICAL LIBRATIONS TO ECKHARDTS
      IMPLICITREAL*8(A-H,O-Z)
      REAL*4 EP(60),HMIN,HMAX,EMX,RAT1,RAT2
      DIMENSION KD(60),KQ(60),YN(60),DT(20,60)
      DIMENSION P(3),BEB(3),SUN(3),CNORM(9 ,9 ),CVECT(9 ),G(3,3),
     PSEV(6),H(13),H(6,12),F(4),PARAM(5),CSH(6),BE(6)          ,Q(3),
     PY(60),YP(60),LVEC(9 ),MVEC(9 ),TE2(9 ),COREL(9 ,9 ),WE(9 )
      DIMENSIONCNOR8(8,8),CVEC8(8),LVE8(8),MVE8(8),TE8(8),CORE8(8,8)
      COMMON/MAIF/ALF,BET,GAM,TEQ,CCE,CCS,P,G /EXE/TOO,BEB,SUN,NUMB
     P/MAIOUT/CNORM,CVECT   /ALL/   SEV,ROM,NUT,KLU
     P/CETBL1/AU,REA,TPD,EMR/CETBL2/ICW,ICE,M/CETBL4/H,F
      DATA ONE,TWO,FOR/1.D0,2.D0,4.D0/
      EXTERNAL FUNPLS,OUTADJ
      READ(5,60)AU,REA,TPD,EMR
      READ(5,65)ICE,M
      READ(5,60)PARAM
      READ(5,65)NQ
      READ(5,60)WE
      READ(5,60)HMIN,HMAX,DELT,RAT1,RAT2
      REWIND 1
      ICW=1
      NUT=2
      KLU=1
    2 CONTINUE
      READ(5,60)OSH
      READ(5,60)Q,TEQ
      READ(5,62)CCE,CCS
      READ(5,62)P
      ICHE=1
  101 ICHE=ICHE+1
    1 A1=ONE/(Q(3)-(TWO-FOR*Q(2))*Q(1))
      B1=ONE/((ONE+Q(2))*Q(3)-(TWO-TWO*Q(2))*Q(1))
      ALF=Q(2)*(Q(3)+TWO*Q(1))*A1
      BET=Q(2)
      GAM=-FOR*Q(2)*Q(1)*B1
      WRITE(6,79)         PARAM,WE ,Q,ALF,BET,GAM,TEQ,CCE,CCS,P
      ROM=0.D0
      NUMB=0
   16 DO 3 I=1,6
    3 Y(I)=OSH(I)
      IF(KLU.GT.1) GOTO 8
   17 A1=A1*A1
      B1=B1*B1
      G(1,1)= FOR*Q(2)*(ONE-Q(2))*Q(3)*A1
      G(1,2)=(Q(3)**2-FOR*Q(1)**2)*A1
      G(1,3)=-FOR*Q(2)*(ONE-Q(2))*Q(1)*A1
      G(2,1)=0.D0
      G(2,2)=1.D0
      G(2,3)=0.D0
      G(3,1)=-FOR*Q(2)*(ONE+Q(2))*Q(3)*B1
```

```
      G(3,2)=-FOR*Q(1)*(Q(3)-TWO*Q(1))*B1
      G(3,3)= FOR*Q(2)*(ONE+Q(2))*Q(1)*B1
C     PARTIALS OF (ALF,BET,GAM) VS. (C22,BET,C20)
      DO 9 K=1,9
      CVECT(K)=0.D0
      DO 9 I=1,9
    9 CNORM(K,I)=0.D0
   18 DO 5 I=7,NQ
    5 Y(I)=0.D0
      DO 6 I=7,42,7
    6 Y(I)=1.D0
C     INITIALIZATION OF TRANSITION AND SENSITIVITY MATRICES
    8 TOO=PARAM(1)+1.0D0
      CALL DVDPF (NQ,Y   ,YP,KD,EP,RAT1,RAT2,HMIN,HMAX,DELT,KST,
     1PIHL,PARAM,FUNPL5,OUTADJ,KQ,YN,DT)
      WRITE(6,79) ROM
      IF(ICHE.EQ.3) GOTO 998
      DO 14 I=1,8
      CVEC8(I)=CVECT(I)
      DO 14 J=1,8
   14 CNOR8(I,J)=CNORM(I,J)
      CALL DMINV (CNOR8,8,DET,LVE8,MVE8)
      CALL DGMPRD (CNOR8,CVEC8,TE8,8,8,1)
      DO 10 I=1,6
   10 OSH(I)=OSH(I)-TE8(I)
      WRITE(6,78) TE8,CNOR8
      Q(1)=Q(1)-TE8(7)
      Q(2)=Q(2)-TE8(8)
      DO 11 I=1,8
      DO 11 J=1,8
   11 CORE8(I,J)=CNOR8(I,J)/ DSQRT(CNOR8(I,I)*CNOR8(J,J))
      WRITE(6,78) CORE8
      IF(ICHE.EQ.1) GOTO 101
      NQ=6
      DELT=10.D0
      KLU=2
      GOTO 101
   19 NQ=60
      KLU=1
      GOTO 101
  998 STOP
   60 FORMAT(6F13.5)
   62 FORMAT(4D20.13)
   65 FORMAT(16I5)
   78 FORMAT(/(5X,8D13.5))
   79 FORMAT(5X,6D18.10)
   80 FORMAT(5X,9 D11.4)
   81 FORMAT(5X,8D11.4)
   90 FORMAT(5D16.9)
      END
```

APPENDIX C/1

Flowchart



APPENDIX C/2

Sample Input

```
149597893.   6378.1492    1.           81.301
    3                                                  1    1
203.         570.         1.           0.00001
-.00001461873.023799910601.000018879248.000008645041-.00021278188-.00014866572
1000.        10.          1000.        1000.        100.         1000.
.0003968     .0006268     .00023       .026769
.8926626        D 16
```

MAIN PROGRAM : D

PROGRAM PURPOSE : To process earth-based optical observations of elected lunar points in a weighted least squares procedure.

INPUT PARAMETERS : An example of the card input required for the program is given in Appendix D/2 (except for estimates of the TQ).

A. PARAM is an extended precision vector of 5 elements.
   1. The first element is the initial epoch in Julian days minus 2440000.0 days.
   2. The second element is the final epoch in Julian days minus 2440000.0 days.
   3. The third element is the initial step size in days.
   4. The remaining elements are not used for input.

B. NQ is a single precision integer used to input the number of integrands.

C. OSH is an extended precision 6 element vector used to input the initial values of the physical librations (τ, σ, ρ) in radians an their time rates ($\dot{\tau}$, $\dot{\sigma}$, $\dot{\rho}$) in radians per day.

D. The vector SEV is described in the subroutine FUNPL5 description.

E. The variables HMIN, HMAX, DELT, RATI, RAT2 are described in the subroutine DVDPF2 description.

F. The variables Q, TEQ, CCE, CCS, P and G are described in the subroutine FUNPL5 description.

G. TJ is an extended precision array of 3 X 22 elements which contain approximate (initial) selenocentric positions of the lunar points in kilometers.

H. JOBN is a single precision integer vector of 10 elements which is used to input the identification number of batches of optical observations to be processed.

I. Simulated ephemeris data is assumed to be available on a direct access disk device (designated unit 4). The parameters read from the disk are ETE and the array FH (see EPHITL subroutine description).

OUTPUT PARAMETERS :

A. Basically output is created in the subroutine OUTADJ which in turn is called by DVDPF2 Samples of the normal equation are given in [Papo, 1971].

PROGRAM DESCRIPTION : The program was formulated according to the theoretical development given in [Papo, 1971] Chapter 2. A basic flowchart is given in Appendix D/1.

SUBROUTINES REQUIRED :

A. O.S.U. Project Library:
   1. DVDPF2
   2. FUNPL6
   3. OUTGAJ

REFERENCES :

A. Papo, Haim B. (1971). "Optical Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 156.

Program D

```
C     MAIN FOR ADJUSTING SIMULATED EARTH-BOUND OPTICAL OBSERVATIONS
      IMPLICITREAL*8(A-H,O-Z)
      REAL*4 EP(60),HMIN,HMAX,EMX,RAT1,RAT2
      DIMENSION KD(60),KQ(60),YN(60),DT(20,60),FH(3,18,15),VEC(75)
      DIMENSIONPOS(3),EMBT(3,3),ENU(22),EKA(22),TJ(3,22),SIG6(3,3)
     P,SIGZ(3,3),IPO(22),JOBN(10)
      DIMENSIONP(3),BEB(3),SUN(3),G(3,3),Y(60),YP(60),
     PSEV(6),M(13),H(6,12),F(4),PARAM(5),OSH(6),Q(3)
      COMMON/MAIF/ALF,BET,GAM,TEQ,CCE,CCS,P,G /EXE/TCO,BEB,SUN,NUMB
     P/MAIOUT/POS,EMBT,ENU,EKA,TJ,SIG6,SIG7,SIGZ,K3,K2,IPO,IST,IICW
     P   /ALL/SEV,ROM,NUT,KLU,N2,KM,ISKIP          /EPHEM/ETE,FH
      DATA ONE,TWO,FOR/1.D0,2.D0,4.D0/
      DATA PI /3.14159265358979D0/
      EXTERNAL FUNPL6,OUTGAJ
      READ(5,60)PARAM
      READ(5,65)NQ
      READ(5,60)OSH
      READ(5,60)SEV
      READ(5,60)HMIN,HMAX,DELT,RAT1,RAT2
      READ(5,60)Q,TEQ
      READ(5,62)CCE,CCS
      READ(5,62)P
      READ(5,61) TJ
      READ(5,65) JOBN
      KLU=1
      NUMB=1
      KM=30
      ISKIP=4
      IICW=1
      III=1
      DO 4 I=1,3
      DO 2 J=1,3
      SIG6(I,J)=0.D0
    2 SIGZ(I,J)=0.D0
      SIG6(I,I)=.02D0
    4 SIGZ(I,I)=.04D0
      SIGZ(1,1)=.004D0
      SIG7=.01D0
      REWIND 8
  709 JOB=JOBN(III)
      IF(JOB.EQ.99) GOTO 998
      REWIND 3
      REWIND 4
      READ (4) ETE,FH
      NUT=0
      N2=44
      K3=1
      K2=2
      REWIND K3
      DO 28 I=1,151
      DO 21 J=1,75
```

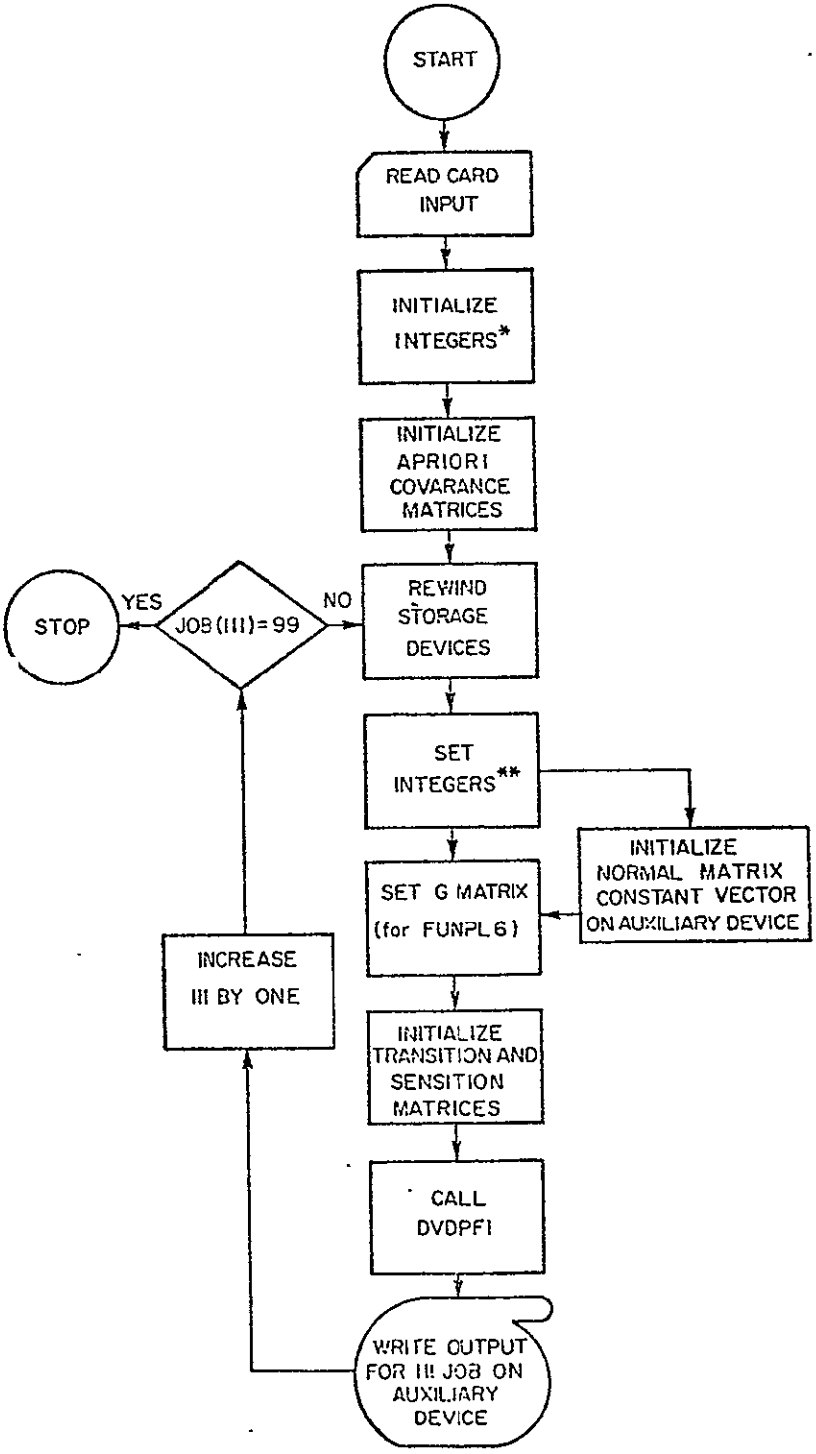
Program D (Cont)

```
   21 VEC(J)=0.D0
      WRITE(K3) VEC
   28 CONTINUE
    1 A1=ONE/(Q(3)-(TWO-FOR*Q(2))*Q(1))
      B1=ONE/((ONE+Q(2))*Q(3)-(TWO-TWO*Q(2))*Q(1))
      ALF=Q(2)*(Q(3)+TWO*Q(1))*A1
      BET=Q(2)
      GAM=-FOR*Q(2)*Q(1)*B1
      WRITE(6,79) SEV,  PARAM    ,Q,ALF,BET,GAM,TEQ,CCE,CCS,P
   16 DO 3 I=1,6
    3 Y(I)=OSH(I)
   17 A1=A1*A1
      B1=B1*B1
      G(1,1)= FOR*Q(2)*(ONE-Q(2))*Q(3)*A1
      G(1,2)=(Q(3)**2-FOR*Q(1)**2)*A1
      G(1,3)=-FOR*Q(2)*(ONE-Q(2))*Q(1)*A1
      G(2,1)=0.D0
      G(2,2)=1.D0
      G(2,3)=0.D0
      G(3,1)=-FOR*Q(2)*(ONE+Q(2))*Q(3)*B1
      G(3,2)=-FOR*Q(1)*(Q(3)-TWO*Q(1))*B1
      G(3,3)= FOR*Q(2)*(ONE+Q(2))*Q(1)*B1
C     PARTIALS OF (ALF,BET,GAM) VS. (C22,BET,C20)
   18 DO 5 I=7,NQ
    5 Y(I)=0.D0
      DO 6 I=7,42,7
    6 Y(I)=1.D0
   15 CONTINUE
C     INITIALIZATION OF TRANSITION AND SENSITIVITY MATRICES
    8 T00=PARAM(1)+1.0D0
      DELT=1.D0
      CALL DVDPF2(NQ,Y   ,YP,KQ,EP,RAT1,RAT2,HMIN,HMAX,DELT,KST,
     PIHL,PARAM,FUNPL6,OUTGAJ,KQ,YN,DT,1)
      REWIND K3
      DO 151 I=1,151
      READ(K3) VEC
  151 WRITE(8) JOB,VEC
      III=III+1
      DO 801 I=1,3
  801 SIG6(I,I)=.2D0
      SIG7=.1D0
      GOTO 709
  998 STOP
   60 FORMAT(6F13.5)
   61 FORMAT(2(10X,3F10.4))
   62 FORMAT(4D20.13)
   65 FORMAT(16I5)
   79 FORMAT(/(5X,6D18.10))
      END
```

APPENDIX D/1

Flowchart of the Optical Data Processor

(Earth-based Observations Assumed)



INITIALIZED INTEGERS*

KLU - Indicates if partial derviative are integrate or read from disk

NUMB - Not used

KM - Number of optical bundles to be processe

ISKIP - OUTGAJ

IICN - Bundle sequence number

III - Job serial number

SET INTERGERS**

NUT - If equal to 2, termina

NZ - Two times the number lunar points

K2 - K3 - Unit numbers of auxiliary device

APPENDIX D/2

Sample Input

```
222.5        579.         .5           .000001
   60
-.00006071823-.00881857445.000606182246.000027448591.00530418844 .000062451555
1.196721511  0.08051275   .2299715022  -.00092421392
.005         .5           1.           .000001      .00001
.0000207     .000629      -.000207     .026769
.892668         D 16

   31   32   99
```

MAIN PROGRAM : E

PROGRAM PURPOSE : To create simulated laser ranges.

INPUT PARAMETERS :

A. Integers (single precision) card input.
   1. IDP is an 8-element vector used to input earth station identification numbers.
   2. IDL is an 8-element vector used to input lunar station identification numbers.

B. Matrices (extended precision) card input.
   1. XP is an 8 × 3 array containing the U, V, W geocentric coordinates of the earth stations in kilometers.
   2. XL is an 8 × 3 array containing the X, Y, Z selenocentric coordinates of the lunar reflectors in kilometers.

C. Scalar (extended precision) disk input.
   1. ETE is the Julian day epoch for the lunar ephemeris stored on disk. The parameter is described in the EPHITL subroutine description.

D. Matrix (extended precision) disk input.
   1. FH is a 3 × 18 × 15 array containing tabulated ephemeris quantities as described in the EPHITL subroutine description.

OUTPUT PARAMETERS : The program prints and punches output for the simulated laser ranges. The parameters are:

A. Integers (single precision).
   1. ID1 is the station identification number of the observing station.
   2. ID2 is the reflector identification number observed.

B. Scalars (extended precision).
   1. JED is the Julian date of observation minus 240000.0 days.
   2. D is the simulated observed distance in kilometers.
   3. ZD is the zenith distance of the observation in degrees.

C. Vector (extended precision).
   1. X is the topocentric Cartesian coordinates of the reflector as described in the LADIS subroutine documentation.

PROGRAM DESCRIPTION : A flowchart is given in Appendix E/1 to outline the programming of the theory developed in [Fajemirokun, 1971]. Sample input is illustrated in Appendix E/2.

SUBROUTINES REQUIRED :

O.S.U. Project Library:

1. EPHITL
2. PMAT
3. LADIS

REFERENCES :

Fajemirokun, F. A. (1971). "Application of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

Program E

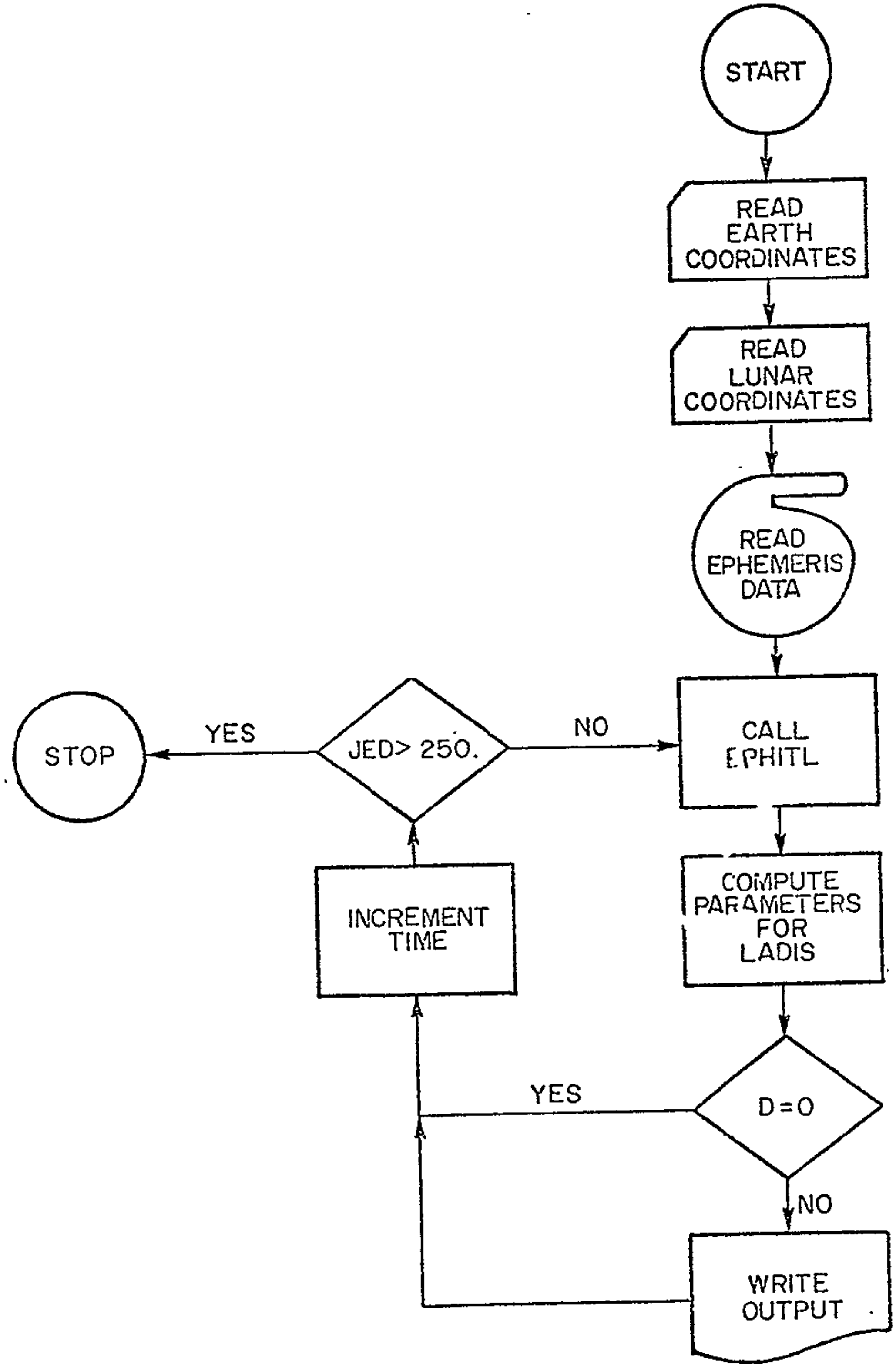
```
      IMPLICIT REAL*8(A-H,O-Z)
C     MAIN PROGRAM FOR COMPUTING SIMULATED LASER DISTANCES
      DOUBLE PRECISION JED
      DIMENSIONXP(8,3),XL(8,3),XE(3),XM(3),XCE(3),PE(3,3),PM(3,3),X(3),
     *FH(3,18,15),QU(16),IDP(8),IDL(8)
      COMMON/EPHEM/ETE,FH
      I=1
      READ(5,81)IDP(I),(XP(I,J),J=1,3)
      WRITE(6,81)IDP(I),(XP(I,J),J=1,3)
   30 CONTINUE
      DO40I=1,3
      READ(5,81)IDL(I),(XL(I,J),J=1,3)
      WRITE(6,81)IDL(I),(XL(I,J),J=1,3)
   40 CONTINUE
      REWIND 4
      READ(4) ETE,FH
      JED=222.5D0
   45 CALLEPHITL(JED,4,QU)
      DO50I=1,3
      XCE(I)=QU(I)
   50 CONTINUE
      PHIM=QU(7)
      PSIM=QU(8)
      THETAM=QU(9)
      CALLPMAT(PSIM,THETAM,PHIM,PM)
      PHIE=QU(13)
      PSIE=QU(14)
      THETAE=QU(15)
      CALLPMAT(PSIE,THETAE,PHIE,PE)
      DO60I=1,3
      ID2=IDL(I)
      XM(1)=XL(I,1)
      XM(2)=XL(I,2)
      XM(3)=XL(I,3)
      J=1
      ID1=IDP(J)
      XE(1)=XP(J,1)
      XE(2)=XP(J,2)
      XE(3)=XP(J,3)
      CALLLADIS(JED,PE,PM,XE,XM,XCE,X,D,ZD)
      IF(D.EQ.0.0D0)GOTO60
      WRITE(6,91)JED,ID1,ID2,X,D,ZD
      WRITE(7,92)JED,ID1,ID2,X,D,ZD
   60 CONTINUE
      JED=JED+0.5D0
      IF(JED.GE.250.0D0)GOTO70
      GOTO45
   70 WRITE(6,93)
      STOP
```

Program E (Cont)

```
81 FORMAT(I5,3D20.9)
82 FORMAT(I2,6X,3F10.4)
91 FORMAT(/,2X,F7.2,2I5,5D19.10)
92 FORMAT(F7.3,2I2,4F10.7,F5.1)
93 FORMAT('1',10X,'JOB REQUIRED IS DONE')
96 FORMAT(3D25.12)
   END
```

APPENDIX E/1

Flowchart for Laser Range Generation



APPENDIX E/2

Sample Input

```
-1   -0.135080000D+04   -0.547478000D+04   0.323560000D+04MCDONALD
 1    0.159180000D+04    0.691878000D+04   0.193560000D+02  APOLLO 11
 2    0.165200000D+04   -0.520000000D+03  -0.110000000D+03  APOLLO 14
 3    0.155400000D+04    0.992000000D+02   0.762000000D+03  APOLLO 15
```

MAIN PROGRAM : F

PROGRAM PURPOSE : To adjust simulated laser distances to the moon.

INPUT PARAMETERS :

A. Integers (single precision). See Appendix F/2.
   1. NE is the number of earth station coordinates.
   2. NM is the number of lunar reflector coordinates.
   3. NV is the number of unknowns in the solution.
   4. NOB is the number of observations in the solution.

OUTPUT PARAMETERS : Output for the program is accomplished in the subroutine LASOLV.

PROGRAM DESCRIPTION : A flow chart of the program is given in Appendix F/1.

SUBROUTINES REQUIRED :

O.S.U. Project Library:
1. LASOLV

REFERENCES :

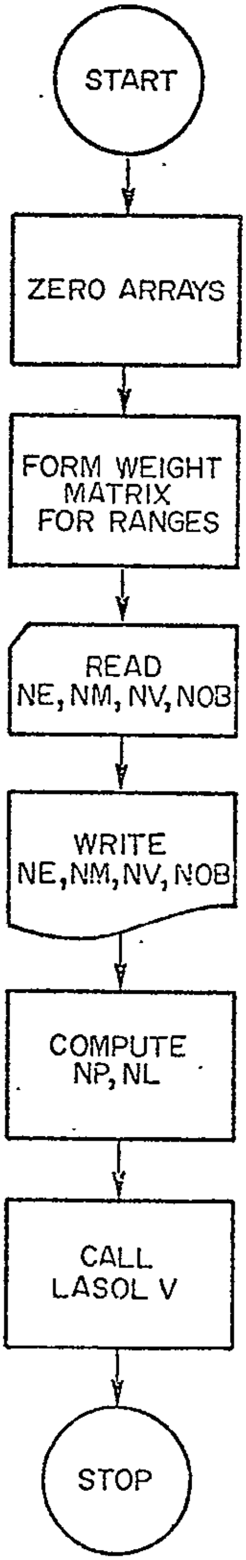
Fajemirokun, F. A. (1971). "Applications of New Observational Systems for Selenodetic Control," The Ohio State University, Department of Geodetic Science, Report No. 157.

Program F

```
C
C     MAIN (1ST) PROGRAM FOR ADJUSTING SIMULATED LASER DISTANCES
      IMPLICITREAL*8(A-H,O-Z)
      DIMENSION IDP(8),XP(8,3),IDL(8),XL(8,3),B(210,63),EMINV(210),
     *QLA(63,63),PX(63),TEMP(210),LVEC(63),MVEC(63),EN(63,63),
     *COREL(63,63)
      DIMENSION BM(63,210),W(210)
    1 CONTINUE
      DO15I=1,210
      DO15J=1,63
      B(I,J)=0.0D0
      BM(J,I)=0.0D0
   15 CONTINUE
      DO20I=1,210
      EMINV(I)=1.D0/(0.15D0*0.15D0)
      W(I)=0.0D0
   20 CONTINUE
      READ(5,10)NE,NM,NU,NOB
      WRITE(6,10)NE,NM,NU,NOB
      NP=NE/3
      NL=NM/3
      CALL       LASOLV(IDP,XP,IDL,XL,B,EMINV,QLA,PX,TEMP,LVEC,MVEC,EN,
     *COREL,NE,NM,NU,NOB,NP,NL,BM,W)
      STOP
   10 FORMAT(4I5)
      END
```

APPENDIX F/1

Flowchart for Laser Range Adjustment



APPENDIX F/2

Sample Input

```
        9        9      33      50
```