TR-DA1509
December 1967

SUBROUTINE DESCRIPTIONS AND LISTINGS FOR THE ORBIT DETERMINATION PROGRAM

VOLUME II

Contract NAS5-9939

Prepared for:

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

PHILCO Ford

PHILCO-FORD CORPORATION
Space & Re-entry Systems Division
Palo Alto, California

TR-DA1509
December 1967

SUBROUTINE DESCRIPTIONS AND LISTINGS FOR THE ORBIT DETERMINATION PROGRAM

Volume II

Contract Number NAS5-9939

Prepared by

SYSTEM DEVELOPMENT AND MISSION ANALYSIS DEPARTMENT
Space and Re-entry Systems Division
Philco-Ford Corporation
Palo Alto, California

for

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

PHILCO-FORD CORPORATION

Space & Re-entry Systems Division

TABLE OF CONTENTS

LIST OF TABLES

Subroutine: GHA

Purpose: To determine the Greenwich hour angle of the first point of Aries for a given date and time.

Calling Sequence: CALL GHA(TSEC,D,GHAN,DA,ØMEGA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TSEC			Seconds	Fractional part of day from D.
I	D			Days	Whole days from 1950 Jan 0.0 UT.
Ø	GHAN			Degrees	Greenwich hour angle $0 \leq GHA < 360$
I	DA			Degrees	Adjustment due to nutation.
Ø	ØMEGA			deg/sec	Rate of rotation of Earth.

See also DEHA.

Common storages used: None

Subroutines required: None

```
$IBFTC MC13GH  NOLIST,NOREF,DECK,M94,NODD,XR3
CMC13GH  SUBROUTINE GHA
      SUBROUTINE GHA(TSEC,D,GHAN,DA,OMEGA)
      DOUBLE PRECISION DD
      OMEGA = .0041780742/(1.+5.21E-13*D)
      DD=D
      DD=DD*(.98564735/360.)
      DF=IDINT(DD)
      DF=DD-DF
      TEM1 = 100.07554+360.*DF    +2.9015E-13*D*D+OMEGA*TSEC
    1 IF (TEM1) 2,3,3
    2 TEM1 = TEM1+360.
      GO TO 1
    3 IF (TEM1-360.) 5,4,4
    4 TEM1 = TEM1-360.
      GO TO 3
    5 GHAN = TEM1+DA*57.2957795
      RETURN
      END
```

Subroutine: GØTØR

Purpose: To solve Kepler's equation for incremental eccentric anomaly on a conic section given the incremental mean anomaly (time).

Calling Sequence: CALL GØTØR (K,VM,C,FF,EO)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	K				Orbit type (1) Elliptic (2) Hyperbolic
I	VM		δM	rad	Incremental mean anomaly
I	C	(2)	C_1, C_2		Kepler's equation coefficients
Ø	FF	(4)	F		Trigonometric functions of incremental eccentric anomaly
I/Ø	EO		α		Incremental eccentric anomaly

Common storages used: None

Subroutines required: None

Method

The incremental eccentric anomaly, α, is implicitly expressed as a function of incremental mean anomaly, δM, and conic coefficients, C_1 and C_2, by the transcendental equations

$$\delta M = (\alpha - \sin\alpha) + C_1 \sin\alpha - C_2(\cos\alpha - 1) \qquad \underline{\text{elliptical}}$$

$$\delta M = (\sinh\alpha - \alpha) + C_1 \sinh\alpha + C_2(\cosh\alpha - 1) \qquad \underline{\text{hyperbolic}}$$

The two equations above may be written as a single equation

$$\delta M = f_1(\alpha) + C_1 f_3(\alpha) + C_2 f_4(\alpha)$$

if the following convention is adopted. For the elliptical case, let

$$f_1(\alpha) = \alpha - \sin\alpha$$

$$f_2(\alpha) = 1 - \cos\alpha$$

$$f_3(\alpha) = \sin\alpha$$

$$f_4(\alpha) = \cos\alpha$$

and for the hyperbolic case, let

$$f_1(\alpha) = \sinh\alpha - \alpha$$

$$f_2(\alpha) = \cosh\alpha - \alpha$$

$$f_3(\alpha) = \sinh\alpha$$

$$f_4(\alpha) = \cosh\alpha$$

When $|\alpha| < 1$, numerical accuracy requires that $f_1(\alpha)$ and $f_2(\alpha)$ be computed by truncated series expansions. Otherwise, the computer library functions SIN, CØS, EXP are used in the computation.

The slope of the function

$$F(\alpha) = f_1(\alpha) + C_1 f_3(\alpha) + C_2 f_4(\alpha)$$

at α is seen to be

$$F'(\alpha) = f_2(\alpha) + C_1 f_4(\alpha) + C_2 f_3(\alpha)$$

$F(\alpha)$ is a monotonically increasing function of α so that any solution of the equation $F(\alpha) = \delta M$ is obviously unique. Newton's method of iteration is used. That is, letting α_n be the n^{th} estimate of α, the (n+1)st estimate is calculated from

$$\alpha_{n+1} = \alpha_n + \frac{\delta M - F(\alpha_n)}{F'(\alpha_n)}$$

The iteration is halted and α is said to be α_n when

$$\left| \frac{\alpha_{n+1} - \alpha_n}{\alpha_{n+1} + \alpha_n} \right| < 3. \times 10^{-8}$$

or when $n = 20$, whichever occurs first. The $f_i(\alpha)$ as well as α are output from GØTØR to avoid their re-computation outside the subroutine.

```
$IBFTC MC13GO  XR3,M94,NODD,LIST
CMC13GO    GOTOR
      SUBROUTINE GOTOR(K,VM,C,FF,E0)
      DIMENSION      C(2),           F(4),           FF(4)
      DIMENSION      A(5),           B(5)
      DATA      A /  .250521083E-7,  .275573192E-5,  .198412698E-3
     1,              .833333333E-2,  .166666667                      /
      DATA      B /  .275573192E-6,  .24801587E-4,   .138888889E-2
     1,              .416666667E-1,  .5                              /
      DATA      NMAX / 20 /
      E1=E0
      N=0
      GO TO (1,6),K
    1 CONTINUE
C     ELLIPTICAL CASE
      SIG=-1.
    2 IF(ABS(E1).GE.1.) GO TO 5
    3 D2=E1*E1
      F=D2*A
      G=D2*B
      DO 4 I=2,5
      F=D2*(A(I)+SIG*F)
    4 G=D2*(B(I)+SIG*G)
      F(1)=E1*F
      F(2)=G
      F(3)=E1+SIG*F
      F(4)=1.+SIG*G
      GO TO 8
    5 F(3)=SIN(E1)
      F(4)=COS(E1)
      F(1)=E1-F(3)
      F(2)=1.-F(4)
      GO TO 8
    6 CONTINUE
C     HYPERBOLIC CASE
      SIG=1.
    7 E3=ABS(E1)
      IF(E3.LT.1.) GO TO 3
      EX=.5*EXP(E3)
      OX=.25/EX
      F(3)=SIGN(EX-OX,E1)
      F(4)=EX+OX
      F(1)=F(3)-E1
      F(2)=F(4)-1.
    8 CM=F(1)+C(1)*F(3)+C(2)*F(2)
      DM=F(2)+C(1)*F(4)+C(2)*F(3)
      DE=(VM-CM)/DM
      AB=ABS(DF)
      IF(AB.GT.1.) DE=DF/AB
      E2=E1+DE
      IF(ABS(DF/(E2+F1)).LF.3.F-8) GO TO 10
      IF(N-NMAX) 9,10,10
    9 N=N+1
      E1=E2
      GO TO (2,7),K
   10 CONTINUF
      E0=E1
      DO 11 I=1,4
   11 FF(I)=F(I)
      RETURN
      END
```

Subroutine: GRAVD

Purpose: To compute the acceleration due to a central body's gravitational attraction. The gravitational potential may include any combination of inverse-square attraction, zonal harmonics, and tesseral harmonics. (See also GRAVDP).

Calling Sequence: CALL GRAVD(A,U,NZ,NT,ZH,TH,RC,**TB2C**,GC,IG)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d	a	km	Central body equatorial radius
I	U	d	u	km^3/sec^2	C.B. gravitational constant
I	NZ,NT		N, N_m		Harmonic summation limits
I	ZH	d(NZ-1)	J_{no}		Zonal harmonic coefficients
I	TH	$d(N_m^2+N_m-2)$	J_{nm}, λ_{nm}		Tesseral harmonic coefficients
I	RC	d(3)	R	km	Position vector, C-frame
I	**TB2C**	d(3,3)			Transformation, body-fixed to C-frame
IØ	GC	d(3)	G	km/sec^2	Acceleration
I	IG				Model and output option

Common storages used: 4N+50 cells

Subroutines required: DDØT,DVNØRM

```
$IBFTC MC133Q  XR3,M94,NODD,LIST
      SUBROUTINE GRAVD (A,U,NZ,NT,ZH,TH,RC,TB2C,GC,IG)
C     COMPUTES GRAVITATIONAL ACCELERATION WITH SPHERICAL HARMONICS
C
      DOUBLE PRECISION  A      , RC(3)      ,TH(1)   ,ZH(1)
     1 .                ,GC(3) ,TB2C(3,3)  ,U
     2                  ,DDOT  ,DVNORM     ,DATAN2 ,DCOS ,DSIN ,DSQRT
C
      COMMON            D ,GI  ,UVEC ,C ,PN
      DOUBLE PRECISION  C(10)  ,GI(3)   ,UVEC(3,3)
     1                  ,D(9)   ,PN(10)
C
      DOUBLE PRECISION  FN(12)
      DATA   FN/ 0.D0, 1.D0, 2.D0, 3.D0, 4.D0, 5.D0
     1          , 6.D0, 7.D0, 8.D0, 9.D0,10.D0,11.D0/
C
C     SET UP UNIT VECTORS AND SPHERICAL COORDINATES
    1 D(1)    = DVNORM(RC,UVEC)
      DO 2 I=1,3
    2 D(I+1) = DDOT  (UVEC,TB2C(1,I))
      D(5)    = DATAN2(D(3),D(2))
      D(3)    = DSIN  (D(5))
      D(2)    = DCOS  (D(5))
      C(1)    = FN(2)
      C(2)    = DSQRT (FN(2)-D(4)*D(4))
      D(8)    =-D(4)*D(2)
      D(9)    =-D(4)*D(3)
      DO 3 I=1,3
      UVEC(I,2) = D(8)*TB2C(I,1)+D(9)*TB2C(I,2)+C(2)*TB2C(I,3)
    3 UVEC(I,3) = D(2)*TB2C(I,2)-D(3)*TB2C(I,1)
      D(2)    = A/D(1)
      D(3)    =-U/D(1)/D(1)
      D(6)    = FN(3)*D(4)
      DO 4 I=1,3
    4 GI(I)   = 0.D0
C
C     SET OPTION SWITCHES
   10 MT = 1
      MZ = 1
      JG = IABS(IG)
      IF (JG.LE.3) GO TO 13
      JG = JG-2
   13 IF (JG-1) 14,16,15
   14 GI(1) = D(3)
      GO TO 50
   15 MT = NT
   16 MZ = NZ
      DO 17 M=1,MT
   17 C(M+2) = C(2)*C(M+1)
C
C     INITIALIZE FOR N-SUMMATION (ZONALS)
   20 D(3)   = D(3)*D(2)
      PN(2) = FN(2)
      JG = 1
      DO 49 N=2,MZ
C
C     GENERATE PN(M)
      D(3)      = D(3)*D(2)
      D(7)      = FN(N+1)+FN(N)
      D(8)      = FN(N+1)*FN(N+2)
      PN(N+1) = D(7)*PN(N)
      PN(N)     = D(4)*PN(N+1)
      M = N
      DO 22 I=2,N
      M = M-1
      D(9)    = D(8)-FN(M)*FN(M+1)
   22 PN(M) = (D(6)*FN(M+1)*PN(M+1)-C(3)*PN(M+2))/D(9)
C
C     COMPUTE AND ADD ZONAL TERMS
   30 D(9)   = ZH(N-1)*D(3)
      GI(1) = GI(1)-D(9)*FN(N+2)*PN(1)
      GI(2) = GI(2)+D(9)*C(2)*PN(2)
      IF (N.GT.MT) GO TO 49
C
C     COMPUTE AND ADD TESSERAL TERMS
```

```
   40 DO 48 M=1,N                                          GRAV0075
      D(7)  = FN(M+1)*(D(5)-TH(JG+1))                      GRAV0076
      D(9)  = TH(JG)*D(3)                                  GRAV0077
      D(8)  = D(9)*DCOS(D(7))                              GRAV0078
      D(9)  = D(9)*DSIN(D(7))                              GRAV0079
      D(7)  = FN(M+1)*C(M)*PN(M+1)                         GRAV0080
      GI(1) = GI(1)-D(8)*FN(N+2)*C(M+1)*PN(M+1)            GRAV0081
      GI(2) = GI(2)+D(8)*(C(M+2)*PN(M+2)-D(4)*D(7))        GRAV0082
      GI(3) = GI(3)-D(9)*D(7)                              GRAV0083
   48 JG = JG+2                                            GRAV0084
   49 CONTINUE                                             GRAV0085
C                                                          GRAV0086
C     COMPUTE ACCELERATION                                 GRAV0087
   50 DO 51 I=1,3                                          GRAV0088
      DO 51 M=1,3                                          GRAV0089
   51 GC(I) = GC(I)+GI(M)*UVEC(I,M)                        GRAV0090
C                                                          GRAV0091
  999 RETURN                                               GRAV0092
      END
```

Subroutine: GRAVDP

Purpose: To compute the acceleration due to a central body's gravitational attraction, and optionally its gradient and partial derivatives with respect to harmonic coefficients. The potential may include inverse-square attraction, zonal harmonics, and/or tesseral harmonics.

Calling Sequence: CALL GRAVDP(A,U,NZ,NT,ZH,TH,RC,TB2C,GC,IG,RG,GP,IZ,IT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d	a	km	Central body equatorial radius
I	U	d	μ	km^3/sec^2	C.B. gravitational constant
I	NZ,NT		N, N_m		Harmonic summation limits
I	ZH	d(NZ-1)	J_{no}		Zonal harmonic coefficients
I	TH	$d(N_m^2+N_m-2)$	J_{nm}, λ_{nm}		Tesseral harmonic coefficients
I	RC	d(3)	R	km	Position vector, C-frame
I	TB2C	d(3,3)			Transformation, body-fixed to C-frame
IØ	GC	d(3)	G	km/sec^2	Acceleration
I	IG				Model and output option
IØ	GR	d(3,3)	ΔG	$1/sec^2$	Gradient of the acceleration
Ø	GP	d(3,n)		km/sec^2	Partial derivatives of the acceleration with respect to the harmonic coefficients
I	IZ	(N)			Output options for zonal partial derivatives
I	IT	$(N_m^2+N_m-2)$			Output options for tesseral partial derivatives

Common storages used: 4N+76 cells

Subroutines required: DDØT,DMVTRN,DVNØRM

Usage:

Model control: The model is controlled by the absolute value of IG. The value of |IG| and the terms included in the model are

|IG|	Spherical Term	Zonal Harmonics	Tesseral Harmonics
1	x		
2	x	x	
3	x	x	x
4		x	
5		x	x

Output control: The gravitational acceleration, G, is summed with the input value for output. If IG < 0, the gradient of the gravitational acceleration is summed with GR and output in GR. The vectors IZ, IT control output of the partial derivatives in GP. For each component of IZ,IT which is ≠0, the partial derivative of G with respect to the corresponding component of ZH,TH respectively, is computed. These partials are upper-loaded into the columns of GP. The components of IZ must be 0 or 1.

Harmonic coefficients: The harmonic coefficients are stored in the arrays ZH,TH in the order:

$$ZH(n-1) = J_{no} \quad , \quad n = 2, 3, \ldots, N$$

$$\begin{aligned}
TH(1) &= J_{21} \\
(2) &= \lambda_{21} \\
(3) &= J_{22} \\
(4) &= \lambda_{22} \\
(5) &= J_{31} \\
(6) &= \lambda_{31} \\
(7) &= J_{32} \\
(8) &= \lambda_{32} \\
(9) &= J_{33} \\
(10) &= \lambda_{33} \\
(11) &= J_{41} \\
&\vdots \\
(k) &= J_{N_m N_m} \\
(k+1) &= \lambda_{N_m N_m}
\end{aligned}$$

The J_{nm} are assumed zero for $m \neq 0$, $n > N_m$.

The internal array FN contains double precision integers from 0 to n+2 (FN(n) = n-1). Its dimension data statement must be at least NZ+3. The common arrays C(i), PN(i) are used to store C^{i-1}, $P_n^{(i-1)}$, respectively, and must be dimensioned at least NZ+2. The dimensions compiled in the listed subroutines are adequate for NZ ≤ 8.

Model

The gravitational field is described by the potential function

$$U = -\frac{\mu}{r} + \frac{\mu}{r} \sum_{n=2}^{N} \sum_{m=0}^{M} J_{nm} \left(\frac{a}{r}\right)^{n} P_n^m(S) \cos m(\lambda - \lambda_{nm}) \qquad (1)$$

where (r, φ, λ) are the spherical coordinates of the field point,

$S = \sin \varphi$

$C = \cos \varphi$

$P_n^m(S)$ = associated Legendre function of the first kind, order n, degree m

$= C^m P_n^{(m)}(S)$

$$P_n^{(m)}(S) = \frac{d^m}{dS^m} P_n(S)$$

The first term is called the spherical term, and the remaining terms are the spherical harmonics. Those harmonics with $m = 0$ are the zonal harmonics, and those with $m \neq 0$ are the tesseral harmonics.

The acceleration, G, is obtained by taking the gradient of U,

$G = - \Delta U.$

Introducing the unit vectors

$\underline{r}$ = unit vector along R

$\underline{k}$ = unit vector along $\varphi = \pi/2$

$\underline{\lambda} = \frac{1}{C} \underline{k} \times \underline{r}$

$\underline{\ell} = \underline{r} \times \underline{\lambda}$

and the abbreviations

$$\sum_{n,m} = \sum_{n=2}^{N} \sum_{m=0}^{n}$$

$$C_{nm} = - \frac{\mu}{r^2} J_{nm} \left(\frac{a}{r}\right)^n \cos m(\lambda - \lambda_{nm}) \tag{2}$$

$$S_{nm} = - \frac{\mu}{r^2} J_{nm} \left(\frac{a}{r}\right)^n \sin m(\lambda - \lambda_{nm})$$

we obtain

$$G = G_r \underline{r} + G_\ell \underline{\ell} + G_\lambda \underline{\lambda}$$

$$\begin{aligned} \nabla G = {} & G_{rr} \underline{rr}^T + G_{r\ell} \underline{r\ell}^T + G_{r\lambda} r\lambda^T \\ & + G_{\ell r} \underline{\ell r}^T + G_{\lambda\lambda} \lambda\lambda^T + G_{\ell\lambda} \underline{\ell\lambda}^T \\ & + G_{\lambda r} \underline{\lambda r}^T + G_{\lambda\ell} \underline{\lambda\ell}^T + G_{\lambda\lambda} \underline{\lambda\lambda}^T \end{aligned} \tag{3}$$

where the coefficients G_r, G_{rr}, etc., are computed from the derivatives of the Legendre polynomials, using

$$G_i = \sum_{n,m} g_{i,nm}$$

$$g_{1,nm} = - C_{nm}(n+1)C^m P_n^{(m)}$$

$$g_{2,nm} = C_{nm}(C^{m+1}P_n^{(m+1)} - mSC^{m-1}P_n^{(m)})$$

$$g_{3,nm} = - S_{nm}\, mC^{m-1}P_n^{(m)}$$

$$g_{4,nm} = - \frac{1}{r}(n+2)g_{1,nm}$$

$$g_{5,nm} = - \frac{1}{r}(n+2)g_{2,nm} \qquad (4)$$

$$g_{6,nm} = - \frac{1}{r}(n+2)g_{3,nm}$$

$$g_{7,nm} = - \frac{1}{r} S_{nm}\, m(C^m P_n^{(m+1)} - (m-1)SC^{m-2}P_n^{(m)})$$

$$g_{8,nm} = \frac{1}{r} C_{nm}(SC^m P_n^{(m+1)} + m(m-1)C^{m-2}P_n^{(m)} + mC^m P_n^{(m)})$$

If the spherical term is omitted,

$$\begin{bmatrix} G_r \\ G_\ell \\ G_\lambda \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$$

$$\begin{bmatrix} G_{rr} & G_{r\ell} & G_{r\lambda} \\ G_{\ell r} & G_{\ell\ell} & G_{\ell\lambda} \\ G_{\lambda r} & G_{\lambda\ell} & G_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} G_4 & G_5 & G_6 \\ G_5 & G_8 - G_4 - \frac{1}{r}G_1 & G_7 \\ G_6 & G_7 & - G_8 + \frac{1}{r}G_1 \end{bmatrix} \qquad (5)$$

and if the spherical term is included, the quantities $(-\mu/r^2, 2\mu/r^3, -\mu/r^3, -\mu/r^3)$ are added to $(G_r, G_{rr}, G_{\ell\ell}, G_{\lambda\lambda})$.

The partial derivatives of G with respect to the J_{nm}, λ_{nm} are

$$\frac{\partial G}{\partial J_{nm}} = \frac{1}{J_{nm}} (g_{1,nm} \underline{r} + g_{2,nm} \underline{\ell} + g_{3,nm} \underline{\lambda})$$

$$\frac{\partial G}{\partial \lambda_{nm}} = - mS_{nm}(n+1)C^m P_n^{(m)} \underline{r} + mS_{nm}(C^{m+1}P_n^{(m+1)} - mSC^{m-1}P_n^{(m)})\underline{\ell} + mC_{nm}(mC^{m-1}P_n^{(m)})\underline{\lambda} \tag{6}$$

Note that the use of $\underline{\ell}$ rather than $\underline{k}$ is computationally superior since the singularities at C = 0 are removed. Indeed, with the equations in this form, we have merely to note that mC^{m-1} and $m(m-1)C^{m-2}$ may be omitted from the summations for m = 0 and m = 0, 1, respectively.

Method

The required Legendre polynomials are generated using the recursion formulas

$$P_n^{(n)} = (2n-1)P_{n-1}^{(n-1)}$$

$$P_n^{(n-1)} = SP_n^{(n)} \tag{7}$$

$$P_n^{(m-1)} = (2mSP_n^{(m)} - C^2P_n^{(m+1)})/(n(n+1)-m(m-1))$$

with the starting value

$$P_1^{(1)} = 1 \qquad (8)$$

For each n, the set $P_n^{(m)}$, m = 0, 1, ... n, is computed and stored in PNM, with

$$PNM(m+1) = P_n^{(m)}$$

For each n, the subroutine first computes the contributions of the zonal harmonics, then the tesseral harmonics, adding the contributions to the G_i. As each $g_{i,nm}$ is computed, the corresponding components of IZ,IT are checked. For those components which are not zero, the corresponding partial derivatives are upper-loaded into the GP-array in the order: (1) all $\partial G/\partial J_{no}$ required; (2) all $\partial G/\partial J_{nm}$, $\partial G/\partial \lambda_{nm}$ in the same order as IT. The components of IZ,IT are discussed under "Usage".

```
$IBFTC MC13GV  XR3,M94,NODD,LIST
      SUBROUTINE GRAVDP (A,U,NZ,NT,ZH,TH,RC,TB2C,GC,IG,GR,GP,IZ,IT)
C     COMPUTES GRAVITATIONAL ACCELERATION WITH SPHERICAL HARMONICS
C     OPTIONALLY COMPUTES GRADIENT OF THE ACCELERATION
C     AND PARTIALS WITH RESPECT TO HARMONIC COEFFICIENTS
C
      DOUBLE PRECISION   A      ,GP(3,1),RC(3)     ,TH(1),ZH(1)
     1                  ,GC(3)  ,GR(3,3),TB2C(3,3),U
     2                  ,DDOT   ,DVNORM ,DATAN2    ,DCOS ,DSIN ,DSQRT
      DIMENSION          IT(1)  ,IZ(1)
C
      COMMON             D ,GI  ,GN  ,UVEC ,C ,PN
      DOUBLE PRECISION   C(10)  ,GI(8)  ,PN(10)  ,UVEC(3,3)
     1                  ,D(9)   ,GN(8)           ,VEC (3,3)
      EQUIVALENCE       (D,VEC)
C
      DOUBLE PRECISION   FN(12)
      DATA   FN/ 0.D0, 1.D0, 2.D0, 3.D0, 4.D0, 5.D0
     1          , 6.D0, 7.D0, 8.D0, 9.D0,10.D0,11.D0/
C
C     SET UP UNIT VECTORS AND SPHERICAL COORDINATES
    1 D(1)    = DVNORM(RC,UVEC)
      DO 2 I=1,3
    2 D(I+1) = DDOT   (UVEC,TB2C(1,I))
      D(5)    = DATAN2(D(3),D(2))
      D(3)    = DSIN   (D(5))
      D(2)    = DCOS   (D(5))
      C(1)    = FN(2)
      C(2)    = DSQRT (FN(2)-D(4)*D(4))
      D(8)    =-D(4)*D(2)
      D(9)    =-D(4)*D(3)
      DO 3 I=1,3
      UVEC(I,2) = D(8)*TB2C(I,1)+D(9)*TB2C(I,2)+C(2)*TB2C(I,3)
    3 UVEC(I,3) = D(2)*TB2C(I,2)-D(3)*TB2C(I,1)
      D(2)    = A/D(1)
      D(3)    =-U/D(1)/D(1)
      D(6)    = FN(3)*D(4)
      DO 4 I=1,8
    4 GI(I)   = 0.D0
C
C     SET OPTION SWITCHES
   10 MT = 1
      MZ = 1
      LZ = 1
      LT = 1
      DO 11 N=2,NZ
   11 LT = LT+IZ(N-1)
      JG = IG
      ASSIGN 35 TO LLA
      ASSIGN 46 TO LLB
      ASSIGN 56 TO LLC
      LN = 3
      IF (JG.GE.0) GO TO 12
      ASSIGN 33 TO LLA
      ASSIGN 44 TO LLB
      ASSIGN 52 TO LLC
      LN = 8
      JG =-JG
   12 IF (JG.LE.3) GO TO 13
      JG = JG-2
   13 IF (JG-1) 14,16,15
   14 GI(1) = D(3)
      GI(4) =-FN(3)*D(3)/D(1)
      GO TO 50
   15 MT = NT
   16 MZ = NZ
      DO 17 M=1,MT
   17 C(M+2) = C(2)*C(M+1)
C
C     INITIALIZE FOR N-SUMMATION (ZONALS)
   20 D(3)    = D(3)*D(2)
      PN(2) = FN(2)
      JG = 1
      DO 49 N=2,MZ
C
```

```
C     GENERATE PN(M)                                                    GRVP0075
      D(3)     = D(3)*D(2)                                              GRVP0076
      D(7)     = FN(N+1)+FN(N)                                          GRVP0077
      D(8)     = FN(N+1)*FN(N+2)                                        GRVP0078
      PN(N+1)  = D(7)*PN(N)                                             GRVP0079
      PN(N)    = D(4)*PN(N+1)                                           GRVP0080
      M = N                                                             GRVP0081
      DO 22 I=2,N                                                       GRVP0082
      M = M-1                                                           GRVP0083
      D(9)   = D(8)-FN(M)*FN(M+1)                                       GRVP0084
   22 PN(M)  = (D(6)*FN(M+1)*PN(M+1)-C(3)*PN(M+2))/D(9)                 GRVP0085
C                                                                       GRVP0086
C     COMPUTE AND ADD ZONAL TERMS                                       GRVP0087
C     ACCELERATION                                                      GRVP0088
   30 GN(1) =-D(3)*FN(N+2)*PN(1)                                        GRVP0089
      GN(2) = D(3)*C(2)*PN(2)                                           GRVP0090
      GN(3) = 0.D0                                                      GRVP0091
      IF (IZ(N-1).EQ.0) GO TO 32                                        GRVP0092
C     PARTIAL WRT JNO                                                   GRVP0093
      DO 31 I=1,3                                                       GRVP0094
   31 GP(I,LZ) = GN(1)*UVEC(I,1)+GN(2)*UVEC(I,2)                        GRVP0095
      LZ = LZ+1                                                         GRVP0096
   32 GN(1) = ZH(N-1)*GN(1)                                             GRVP0097
      GN(2) = ZH(N-1)*GN(2)                                             GRVP0098
      GO TO LLA, (33,35)                                                GRVP0099
C     GRADIENT                                                          GRVP0100
   33 D(7)   =-FN(N+3)/D(1)                                             GRVP0101
      DO 34 I=1,2                                                       GRVP0102
      GN(I+3) = D(7)*GN(I)                                              GRVP0103
   34 GN(I+5) = 0.D0                                                    GRVP0104
      GN(8)    = ZH(N-1)*D(3)*D(4)*PN(2)/D(1)                           GRVP0105
   35 CONTINUE                                                          GRVP0106
      DO 36 I=1,LN                                                      GRVP0107
   36 GI(I) = GI(I)+GN(I)                                               GRVP0108
      IF (N.GT.MT) GO TO 49                                             GRVP0109
C                                                                       GRVP0110
C     COMPUTE AND ADD TESSERAL TERMS                                    GRVP0111
   40 DO 48 M=1,N                                                       GRVP0112
      D(7)   = FN(M+1)*(D(5)-TH(JG+1))                                  GRVP0113
      GN(4) = D(3)*DCOS(D(7))                                           GRVP0114
      GN(5) = D(3)*DSIN(D(7))                                           GRVP0115
      GN(6) =-FN(N+2)*C(M+1)*PN(M+1)                                    GRVP0116
      GN(8) = FN(M+1)*C(M)*PN(M+1)                                      GRVP0117
      GN(7) = C(M+2)*PN(M+2)-D(4)*GN(8)                                 GRVP0118
      GN(1) = GN(4)*GN(6)                                               GRVP0119
      GN(2) = GN(4)*GN(7)                                               GRVP0120
      GN(3) =-GN(5)*GN(8)                                               GRVP0121
      IF (IT(JG).EQ.0) GO TO 41                                         GRVP0122
C     PARTIAL WRT JNM                                                   GRVP0123
      CALL DMVTRN(UVEC,GN,GP(1,LT),1,1)                                 GRVP0124
      LT = LT+1                                                         GRVP0125
   41 DO 42 I=1,5                                                       GRVP0126
   42 GN(I) = TH(JG)*GN(I)                                              GRVP0127
      IF (IT(JG+1).EQ.0) GO TO 43                                       GRVP0128
C     PARTIAL WRT LNM                                                   GRVP0129
      GN(6) = FN(M+1)*GN(5)*GN(6)                                       GRVP0130
      GN(7) = FN(M+1)*GN(5)*GN(7)                                       GRVP0131
      GN(8) = FN(M+1)*GN(4)*GN(8)                                       GRVP0132
      CALL DMVTRN(UVEC,GN(6),GP(1,LT),1,1)                              GRVP0133
      LT = LT+1                                                         GRVP0134
   43 GO TO LLB, (44,46)                                                GRVP0135
C     GRADIENT                                                          GRVP0136
   44 GN(6) = C(M+1)*PN(M+2)                                            GRVP0137
      GN(8) = FN(M)*C(M-1)*PN(M+1)                                      GRVP0138
      GN(7) = GN(5)*FN(M+1)*(D(4)*GN(8)-GN(6))/D(1)                     GRVP0139
      GN(8) = GN(4)*(D(4)*GN(6)+FN(M+1)*(GN(8)+C(M+1)*PN(M+1)))/D(1)    GRVP0140
      GN(6) =-FN(N+3)/D(1)                                              GRVP0141
      DO 45 I=1,3                                                       GRVP0142
   45 GN(I+3) = GN(6)*GN(I)                                             GRVP0143
   46 DO 47 I=1,LN                                                      GRVP0144
   47 GI(I) = GI(I)+GN(I)                                               GRVP0145
   48 JG = JG+2                                                         GRVP0146
   49 CONTINUE                                                          GRVP0147
C                                                                       GRVP0148
C     COMPUTE COMPONENTS OF GR ALONG UVEC                               GRVP0149
```

```
   50 GO TO LLC, (52,56)
   52 GI(12) = GI(1)/D(1)-GI(8)
      GI(11) = GI(7)
      GI(10) = GI(6)
      GI( 9) = GI(7)
      GI( 8) =-GI(4)-GI(12)
      GI( 7) = GI(5)
C
C     COMPUTE GRADIENT
      CALL DMVTRN(UVEC,GI(4),VEC,1,3)
      DO 53 I=1,3
      DO 53 M=1,3
      DO 53 N=1,3
   53 GR(M,N) = GR(M,N)+UVEC(M,I)*VEC(N,I)
C
C     COMPUTE ACCELERATION
   56 DO 57 I=1,3
      DO 57 M=1,3
   57 GC(I) = GC(I)+GI(M)*UVEC(I,M)
C
  999 RETURN
      END
```

Subroutine: GRDAT

Purpose: Computes four Goddard Range-Range Rate System measurements (x, y, range, doppler).

Calling Sequence: CALL GRDAT

Input and Output

Common storages used: /DATCØM/

Subroutines required: SBDAT

COMMON LOCATIONS

COMMON	LOCATION	NAME	DIMENSION	DESCRIPTION
DATCØM	C(1)	BIAS	d(2)	BIAS(1), Doppler bias freq. (CPS).
	C(3)			BIAS(2), Transponder retransmission ratio.
	C(133)	FTR	d	Doppler transmitter frequency (CPS).
	C(135)	ØMEGA	d	Earth rotation rate (rad/sec).
	C(137)	SPDLT	d	Speed of light (km/sec)
	C(139)	STA	d(5)	STA(1-3),Receiving station position in B-frame (km). STA(4-5),Refraction constants.
	C(159)	TAU	d	Doppler count interval (sec).
	C(161)	TB2CØ	d(18)	TB2CØ(1-9), B-frame to C-frame transform at end of doppler count interval at time of signal reception at receiving station. TB2CØ(10-18), Same as above but at beginning of doppler count interval.
	C(197)	TB2CT	d(18)	B-frame to C-frame transforms at end and beginning of doppler count interval at time of signal transmission.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors at station in B-frame.
	C(269)	XV	d(12)	XV(1-6), Spacecraft position and velocity at end of doppler count interval <u>at spacecraft</u> (km, km/sec). XV(7-12), As above but at beginning of doppler count interval.
	C(297)	NALIGN		Should be set to +1 to indicate antenna mount has its principal axis North.
	C(298)	NANG		=1, include angles =2, omit angles
	C(299)	NFRAC		=1, include refraction =2, omit refraction
	C(5)	ØBS	d(64)	Measurements ØBS(1) = x (rad) ØBS(2) = y (rad) ØBS(33) = Range (sec) ØBS(49) = Doppler (counts) See Table 1, in description of CBDAT.

```
$IBFTC MC13DC  M94,NODD,XR3
CMC13DC   GODDARD RANGE AND RANGE-RATE SYSTEM MEASUREMENTS                  GRDT0001
      SUBROUTINE GRDAT                                                      GRDT0002
      COMMON  /DATCOM/  BIAS(2),    OBS(64),    FTR,        OMEGA           GRDT0003
     1,                 SPDLT,      STA(10),    TAU,        TB2CO(18)       GRDT0004
     2,                 TB2CT(18),  TT2BO(9),   TT2BT(9),   XV(12)          GRDT0005
     3,                 MLT,        MODE,       MSTA,       MTIM            GRDT0006
     4,                 NALIGN,     NANG,       NFRAC                       GRDT0007
      DOUBLE PRECISION  BIAS,     OBS,      FTR,      OMEGA,    SPDLT       GRDT0008
     1,                 STA,      TAU,      TB2CO,    TB2CT,    TT2BO       GRDT0009
     2,                 TT2BT,    XV                                        GRDT0010
      MODE=2                                                                GRDT0011
      BIAS(2)=1.                                                            GRDT0012
      CALL SBDAT                                                            GRDT0013
      OBS(33)=OBS(33)/SPDLT                                                 GRDT0014
      OBS(49)=2.*TAU*BIAS(1)-OBS(49)                                        GRDT0015
      RETURN                                                                GRDT0016
      END
```

Subroutine: GRDATP

Purpose: Computes four Goddard Range-Range Rate System measurements and their partials (x, y, range, doppler).

Calling Sequence: CALL GRDATP(R2)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	R2	d	S^2	km^2	Slant range squared

Common storages used: /DATCØM/

Subroutines required: SBDATP

COMMON LOCATIONS

COMMON	LOCATIONS	NAME	DIMENSION	DESCRIPTION
DATCØM	C(1)	BIAS	d(2)	BIAS(1), Doppler bias freq. (CPS).
	C(3)			BIAS(2), Transponder retransmission ratio.
	C(133)	FTR	d	Doppler transmitter frequency (CPS).
	C(135)	ØMEGA	d	Earth rotation rate (rad/sec).
	C(137)	SPDLT	d	Speed of light (km/sec).
	C(139)	STA	d(5)	STA(1-3), Station position in B-frame (km). STA(4-5), Refraction constants.
	C(159)	TAU	d	Doppler count interval (sec).
	C(161)	TB2CØ	d(18)	TB2CØ(1-9), B-frame to C-frame transform at end of doppler count interval at signal reception. TB2CØ(10-18), Same as above, but at beginning of doppler count interval.
	C(197)	TB2CT	d(18)	Same as TB2CØ, but at time of signal transmission.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors at station in B-frame.
	C(269)	XV	d(12)	XV(1-6), Spacecraft position and velocity in C-frame at end of doppler interval at time signal leaves spacecraft. XV(7-12), As above but at beginning of doppler count.
	C(293)	MLT		Speed of light partial option key*.
	C(295)	MSTA		Station location partial option key*.
	C(296)	MTIM		Station clock partial option key*.
	C(297)	NALIGN		Should be set to +1 to indicate antenna mount with principal axis North.
	C(298)	NANG		Angle inclusion option key*.
	C(299)	NFRAC		Refraction correction option key*.
				*If key = +1, option will be included. If key = +2, option will be omitted. If angles are omitted, no angle partials are computed.
	C(5)	ØBS	d(64)	Measurements and partials. See Table 1, description of CBDAT.

```
$IBFTC MC13DG  XR3,M94,NODD,LIST
CMC13DG   GODDARD RANGE AND RANGE-RATE SYSTEM MEASUREMENTS AND PARTIALS
      SUBROUTINE GRDATP(R2OB)
      COMMON  /DATCOM/  BIAS(2),    OBS(64),    FTR,        OMEGA
     1,                 SPDLT,      STA(10),    TAU,        TB2CO(18)
     2,                 TB2CT(18),  TT2BO(9),   TT2BT(9),   XV(12)
     3,                 MLT,        MODE,       MSTA,       MTIM
     4,                 NALIGN,     NANG,       NFRAC
      DOUBLE PRECISION  BIAS,     OBS,      FTR,      OMEGA,    RSPLT
     1,                 SPDLT,    STA,      TAU,      TB2CO,    TB2CT
     2,                 TT2BO,    TT2BT,    XV,       R2OB
      RSPLT=1.D0/SPDLT
      MODE=2
      BIAS(2)=1.
      CALL SBDATP(R2OB)
C     ADJUST RANGING OBSERVABLE AND PARTIALS
C     ADJUST DOPPLER OBSERVABLE AND PARTIALS
      DO 1 I=33,45
      OBS(I+16)=-OBS(I+16)
    1 OBS(I)=OBS(I)*RSPLT
      OBS(40)=1.D0
      OBS(49)=2.*TAU*BIAS(1)+OBS(49)
      OBS(56)=TAU
      RETURN
      END
```

Subroutine: GRTEST

Purpose: To read Goddard Range and Range Rate System raw data tapes, decode the data, convert units and using subroutine PØLYFT, test for outliers and write the edited data in the DCP format on Unit 12.

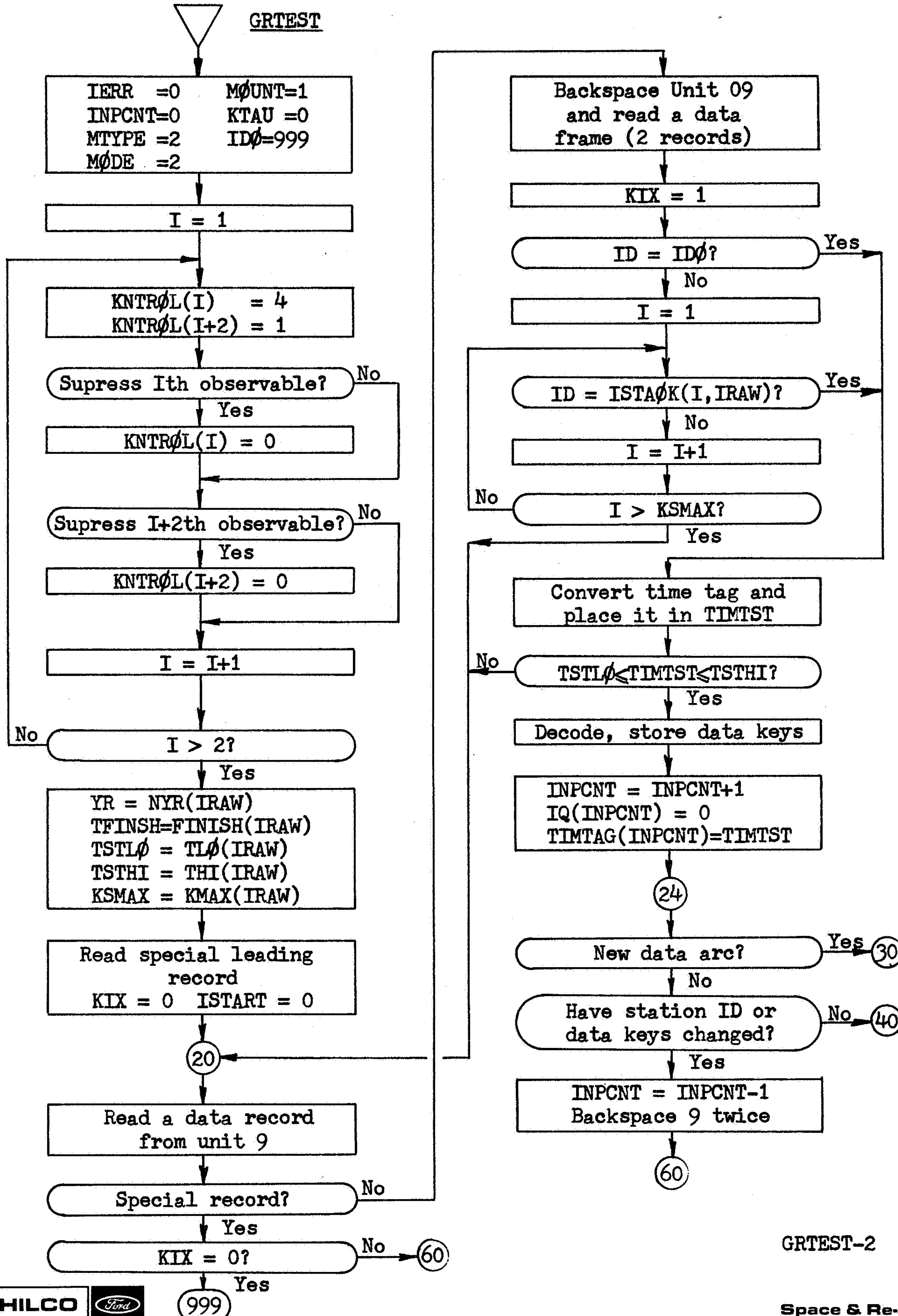
Calling Sequence: CALL GRTEST(IERR)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	IERR				Error flag. Set = 2 by GRTEST if more than 20 stations are accumulated. Set 0 otherwise.

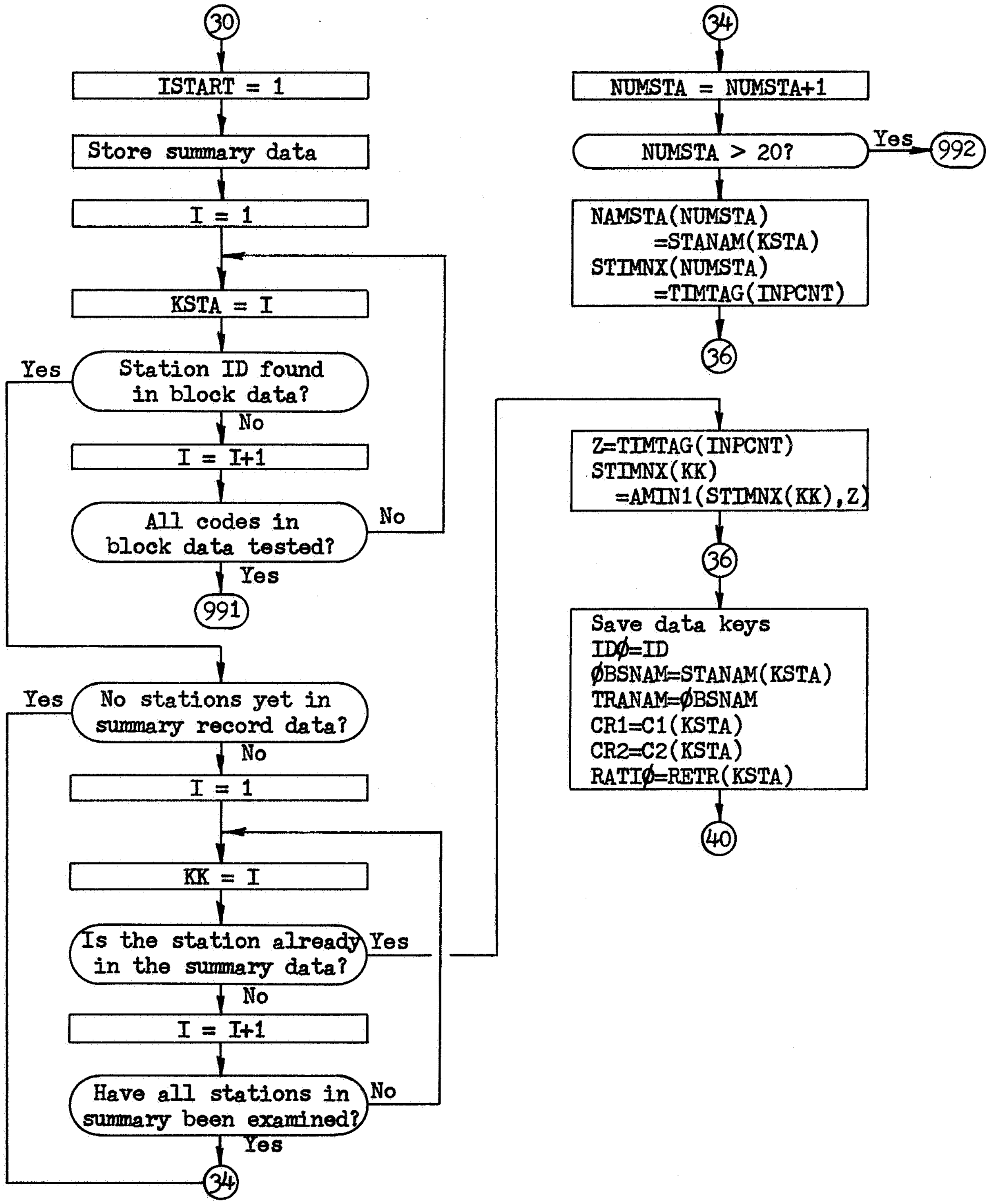
Common storages used: /TRKCØM/,/ØUTCØM/,/TSTCØM/,/DATCØM/,/SUMCØM/

Subroutines required: DATINP, PØLYFT

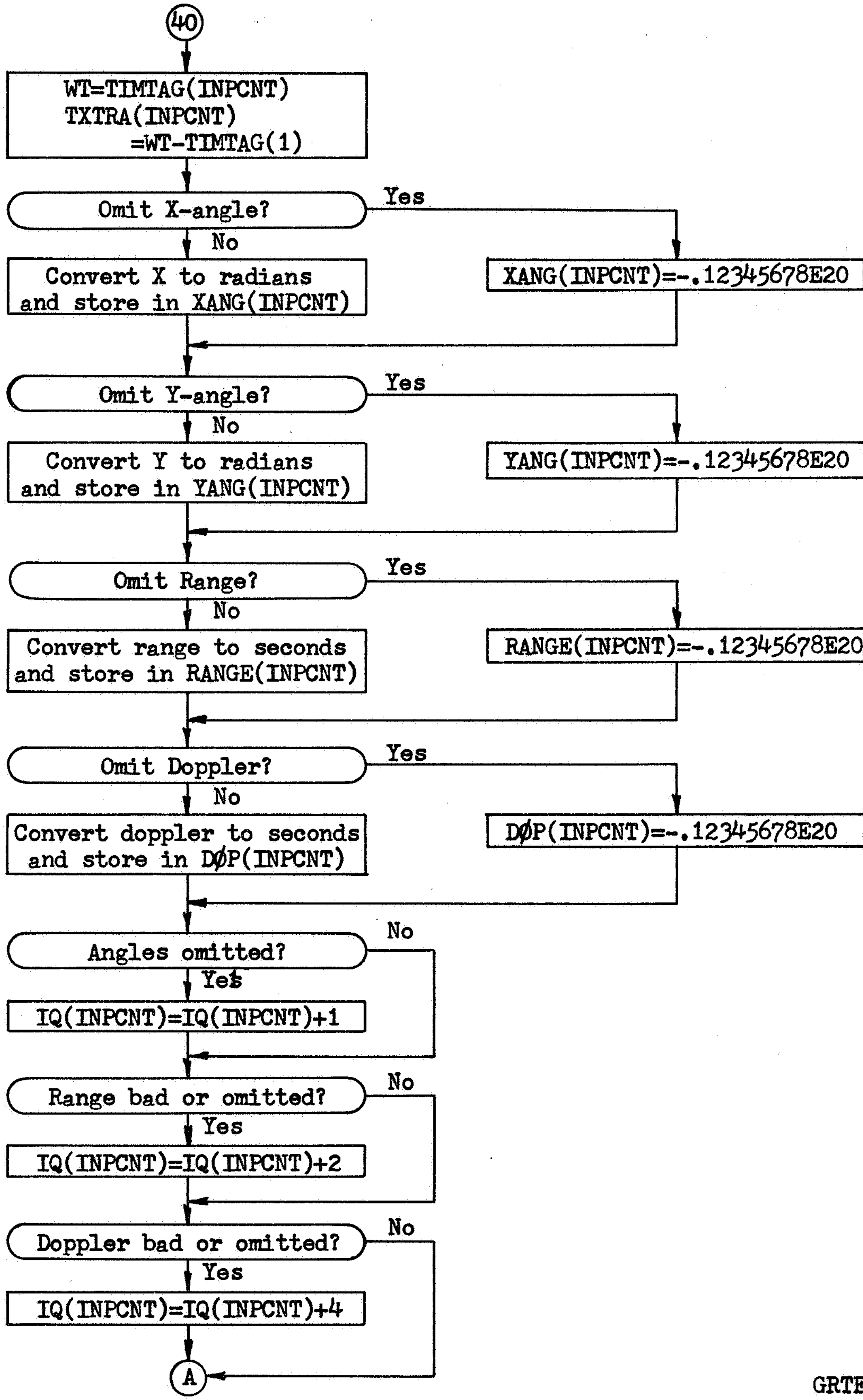


GRTEST
IERR =0
INPCNT=0
MTYPE =2
MØDE =2
MØUNT=1
KTAU =0
IDØ=999
I = 1
KNTRØL(I) = 4
KNTRØL(I+2) = 1
Supress Ith observable?
No
Yes
KNTRØL(I) = 0
Supress I+2th observable?
No
Yes
KNTRØL(I+2) = 0
I = I+1
No
I > 2?
Yes
YR = NYR(IRAW)
TFINSH=FINISH(IRAW)
TSTLØ = TLØ(IRAW)
TSTHI = THI(IRAW)
KSMAX = KMAX(IRAW)
Read special leading record
KIX = 0 ISTART = 0
20
Read a data record from unit 9
Special record?
No
Yes
KIX = 0?
No
60
Yes
999
Backspace Unit 09 and read a data frame (2 records)
KIX = 1
ID = IDØ?
Yes
No
I = 1
ID = ISTAØK(I,IRAW)?
Yes
No
I = I+1
No
I > KSMAX?
Yes
Convert time tag and place it in TIMTST
No
TSTLØ≤TIMTST≤TSTHI?
Yes
Decode, store data keys
INPCNT = INPCNT+1
IQ(INPCNT) = 0
TIMTAG(INPCNT)=TIMTST
24
New data arc?
Yes
30
No
Have station ID or data keys changed?
No
40
Yes
INPCNT = INPCNT-1
Backspace 9 twice
60

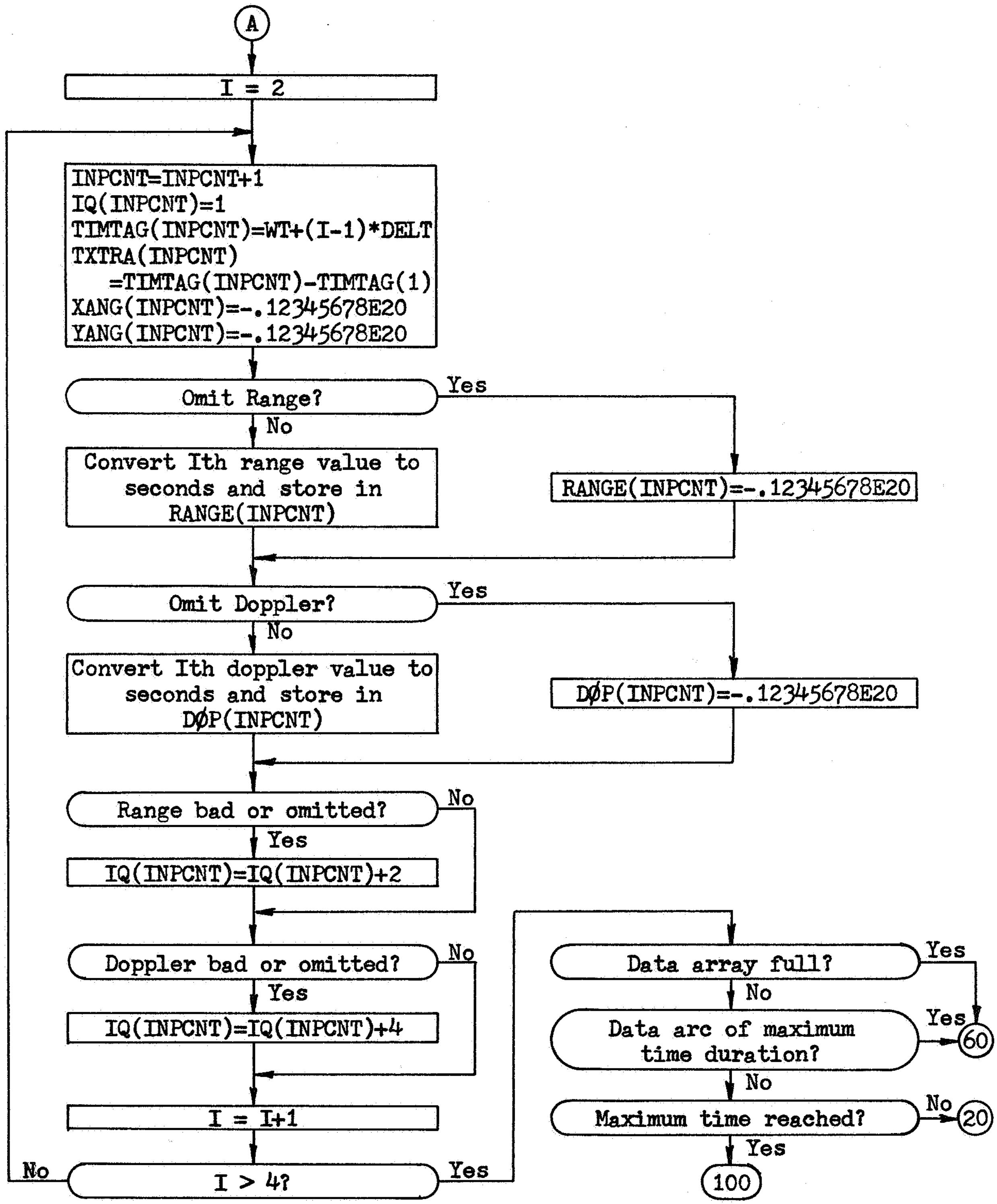
GRTEST-2

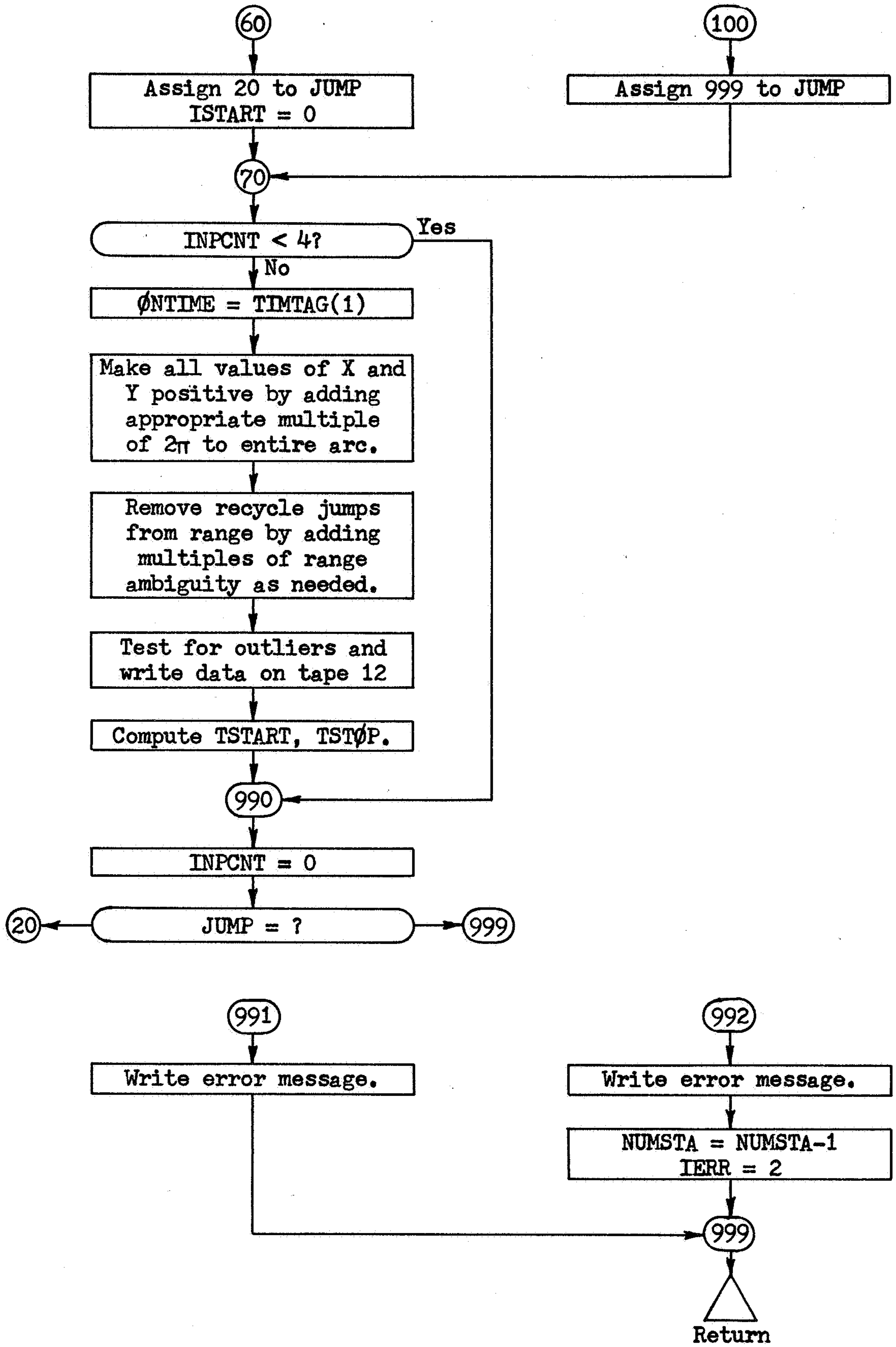

30
ISTART = 1
Store summary data
I = 1
KSTA = I
Station ID found in block data?
Yes
No
I = I+1
All codes in block data tested?
No
Yes
991
No stations yet in summary record data?
Yes
No
I = 1
KK = I
Is the station already in the summary data?
Yes
No
I = I+1
Have all stations in summary been examined?
No
Yes
34
34
NUMSTA = NUMSTA+1
NUMSTA > 20?
Yes
992
NAMSTA(NUMSTA) =STANAM(KSTA)
STIMNX(NUMSTA) =TIMTAG(INPCNT)
36
Z=TIMTAG(INPCNT)
STIMNX(KK) =AMIN1(STIMNX(KK),Z)
36
Save data keys
IDØ=ID
ØBSNAM=STANAM(KSTA)
TRANAM=ØBSNAM
CR1=C1(KSTA)
CR2=C2(KSTA)
RATIØ=RETR(KSTA)
40

GRTEST-3


40
WT=TIMTAG(INPCNT)
TXTRA(INPCNT)
=WT-TIMTAG(1)
Omit X-angle?
Yes
No
Convert X to radians
and store in XANG(INPCNT)
XANG(INPCNT)=-.12345678E20
Omit Y-angle?
Yes
No
Convert Y to radians
and store in YANG(INPCNT)
YANG(INPCNT)=-.12345678E20
Omit Range?
Yes
No
Convert range to seconds
and store in RANGE(INPCNT)
RANGE(INPCNT)=-.12345678E20
Omit Doppler?
Yes
No
Convert doppler to seconds
and store in DØP(INPCNT)
DØP(INPCNT)=-.12345678E20
Angles omitted?
No
Yes
IQ(INPCNT)=IQ(INPCNT)+1
Range bad or omitted?
No
Yes
IQ(INPCNT)=IQ(INPCNT)+2
Doppler bad or omitted?
No
Yes
IQ(INPCNT)=IQ(INPCNT)+4
A

GRTEST-4


A
I = 2
INPCNT=INPCNT+1
IQ(INPCNT)=1
TIMTAG(INPCNT)=WT+(I-1)*DELT
TXTRA(INPCNT)
=TIMTAG(INPCNT)-TIMTAG(1)
XANG(INPCNT)=-.12345678E20
YANG(INPCNT)=-.12345678E20
Omit Range?
Yes
No
Convert Ith range value to seconds and store in RANGE(INPCNT)
RANGE(INPCNT)=-.12345678E20
Omit Doppler?
Yes
No
Convert Ith doppler value to seconds and store in DØP(INPCNT)
DØP(INPCNT)=-.12345678E20
Range bad or omitted?
No
Yes
IQ(INPCNT)=IQ(INPCNT)+2
Doppler bad or omitted?
No
Yes
IQ(INPCNT)=IQ(INPCNT)+4
I = I+1
No
I > 4?
Yes
Data array full?
Yes
No
Data arc of maximum time duration?
Yes
60
No
Maximum time reached?
No
20
Yes
100


60
100
Assign 20 to JUMP
ISTART = 0
Assign 999 to JUMP
70
INPCNT < 4?
Yes
No
ØNTIME = TIMTAG(1)
Make all values of X and Y positive by adding appropriate multiple of 2π to entire arc.
Remove recycle jumps from range by adding multiples of range ambiguity as needed.
Test for outliers and write data on tape 12
Compute TSTART, TSTØP.
990
INPCNT = 0
20
JUMP = ?
999
991
992
Write error message.
Write error message.
NUMSTA = NUMSTA-1
IERR = 2
999
Return

GRTEST-6

```
$IBFTC MC134X  M94,NODD,XR3
CMC134X   GODDARD RAW DATA PROCESSOR                                    GRTS0001
      SUBROUTINE GRTEST(IERR)                                           GRTS0002
      COMMON  /TRKCOM/CTRK(700)                                         GRTS0003
      DOUBLE PRECISION  FTR(50),  C1(50),   C2(50),   BIAS(50), RETR(50)GRTS0004
      DIMENSION   STANAM(50),   KODSTA(50),   NALIGN(50),   PAIR(50)    GRTS0005
      EQUIVALENCE              (CTRK(1),STANAM),     (CTRK(51),KODSTA)  GRTS0006
     1,  (CTRK(101),NALIGN),  (CTRK(151),FTR),      (CTRK(251),C1)      GRTS0007
     2,  (CTRK(351),C2),      (CTRK(451),BIAS),     (CTRK(551),RETR)    GRTS0008
     3,  (CTRK(651),PAIR)                                               GRTS0009
C                                                                       GRTS0010
      COMMON  /OUTCOM/COUT(40)                                          GRTS0011
      DOUBLE PRECISION  ONTIME,   TFIRST,   TLAST,    TAU,      TRF     GRTS0012
     1,        CR1,       CR2,      DR,       BIASF,    RATIO           GRTS0013
      EQUIVALENCE              (COUT(1),NPAIR),     (COUT(2),NEOT)      GRTS0014
     1,  (COUT(3),OBSNAM),    (COUT(4),TRANAM),     (COUT(5),NRCD)      GRTS0015
     2,  (COUT(6),NPTBLK),    (COUT(7),KONT),       (COUT(8),MTYPE)     GRTS0016
     3,  (COUT(9),MOUNT),     (COUT(10),MODE),      (COUT(11),DELT)     GRTS0017
     4,  (COUT(12),KTAU),     (COUT(13),ONTIME),    (COUT(15),TFIRST)   GRTS0018
     5,  (COUT(17),TLAST),    (COUT(19),TAU),       (COUT(21),TRF)      GRTS0019
     6,  (COUT(23),CR1),      (COUT(25),CR2),       (COUT(27),DR)       GRTS0020
     7,  (COUT(29),BIASF),    (COUT(31),RATIO),     (COUT(33),NB1)      GRTS0021
     8,  (COUT(34),NB2),      (COUT(35),NB3),       (COUT(36),NB4)      GRTS0022
C                                                                       GRTS0023
      COMMON  /TSTCOM/CTEST(400)                                        GRTS0024
      DOUBLE PRECISION  FINISH(10),    TLO(10),       THI(10)           GRTS0025
      DIMENSION    NYR(10),       NPTS(10),      NSTEP(10),     NDEG(10)GRTS0026
     1,            CSD(4,10),     KNTROL(4),     IFOMIT(4,10), KMAX(10) GRTS0027
     2,            ISTAOK(20,10)                                        GRTS0028
      EQUIVALENCE              (CTEST(1),NYR),      (CTEST(11),NPTS)    GRTS0029
     1,  (CTEST(21),NSTEP),   (CTEST(31),NDEG),     (CTEST(41),CSD)     GRTS0030
     2,  (CTEST(81),IRAW),    (CTEST(82),KNTROL),   (CTEST(87),FINISH)  GRTS0031
     3,  (CTEST(107),IFOMIT), (CTEST(147),KMAX),    (CTEST(157),ISTAOK) GRTS0032
     4,  (CTEST(357),TLO),    (CTEST(377),THI)                          GRTS0033
C                                                                       GRTS0034
      COMMON  /DATCOM/CDAT(2400)                                        GRTS0035
      DOUBLE PRECISION TIMTAG(300)                                      GRTS0036
      DIMENSION    TXTRA(300),    IQ(300),       XANG(300),     YANG(300)GRTS0037
     1,            RANGE(300),    DOP(300),      RDAT(300,4)            GRTS0038
      EQUIVALENCE              (CDAT(1),TIMTAG),    (CDAT(601),TXTRA)   GRTS0039
     1,  (CDAT(901),IQ),      (CDAT(1201),XANG),    (CDAT(1501),YANG)   GRTS0040
     2,  (CDAT(1801),RANGE),  (CDAT(2101),DOP),     (XANG,RDAT)         GRTS0041
C                                                                       GRTS0042
      COMMON  /SUMCOM/SUMARY(56)                                        GRTS0043
      DOUBLE PRECISION TSTART,TSTOP                                     GRTS0044
      DIMENSION   HEADER(11),    STIMNX(20)                             GRTS0045
      REAL NAMSTA(20)                                                   GRTS0046
      EQUIVALENCE              (SUMARY(1),HEADER),  (SUMARY(12),NUMSTA) GRTS0047
     1,  (SUMARY(13),NAMSTA), (SUMARY(33),TSTART), (SUMARY(35),TSTOP)   GRTS0048
     2,  (SUMARY(37),STIMNX)                                            GRTS0049
C                                                                       GRTS0050
      DOUBLE PRECISION  TFINSH,   TMAX,     TIMTST,   TSTLO,    TSTHI   GRTS0051
      DIMENSION  QR(4),    QRD(4),   R(4),     RD(4),    IDC(4),   DI(3)GRTS0052
      DIMENSION IXXHOL(12),     XXHOL(12),    AR(4),         BRD(4)     GRTS0053
     1,           IXX(4),       INTCOD(5),    LXTCOD(5),     NNN(2)     GRTS0054
      EQUIVALENCE  (IXXHOL,XXHOL)                                       GRTS0055
      DOUBLE PRECISION  XCOUNT(5,2),   XFTR(4),       XBIAS(2),      WT GRTS0056
      DIMENSION         XDEL(3),       XSAMP(5)                         GRTS0057
      LOGICAL HYBRID                                                    GRTS0058
C                                                                       GRTS0059
  602 FORMAT(A2)                                                        GRTS0060
  605 FORMAT(F5.2,A1,F8.2,A1,F7.1,1X,F3.0,3F2.0,A1,F8.2,A1,F7.1/        GRTS0061
     1       F5.2,A1,F8.2,A1,F7.1,1X,I3,I2,2I1,3A1,F8.2,A1,F7.1)        GRTS0062
  901 FORMAT(18H0***  STATION CODE,I3,29H DOES NOT EXIST IN BLOCK DATA/ GRTS0063
     16X,34HPROCESSING OF THIS TAPE ABANDONED./                         GRTS0064
     26X,38HPROGRAM PROCEEDS TO NEXT TAPE, IF ANY.)                     GRTS0065
  902 FORMAT(24H0***  TOO MANY STATIONS./                               GRTS0066
     16X,49HPROGRAM CONTROL PASSES TO TERMINATION OPERATIONS.)          GRTS0067
  903 FORMAT(18H0***  STATION CODE,I3,34H IS NOT IN LIST INTERNAL TO GRTGRTS0068
     1EST/6X,34HPROCESSING OF THIS TAPE ABANDONED./                     GRTS0069
     26X,38HPROGRAM PROCEEDS TO NEXT TAPE, IF ANY.)                     GRTS0070
      DATA XCOUNT/229263.D0,131007.D0,65503.D0,32751.D0,3133956.D0,     GRTS0071
     1 14328.D0,8187.D0,4093.D0,2046.D0,182182.D0/                      GRTS0072
      DATA XFTR/2271.9328D6,2270.9328D6,2270.1328D6,148.26D6/           GRTS0073
      DATA XBIAS/.5D6,.03D6/                                            GRTS0074
```

```
      DATA XDEL/.125,.03125,.00625/                                     GRTS0075
      DATA XSAMP/1.,.5,.25,.125,10./                                    GRTS0076
      DATA EOTIND/2H**/                                                 GRTS0077
C     TO AVOID TRUNCATING GODDARD SYSTEM DATA FRAMES, INPMAX MUST       GRTS0078
C     BE A MULTIPLE OF FOUR                                             GRTS0079
      DATA INPMAX/300/                                                  GRTS0080
      DATA TMAX/.432D5/                                                 GRTS0081
      DATA BLANK/1H /                                                   GRTS0082
      DATA XXHOL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H*,1H /       GRTS0083
      DATA INTCOD/26,22,52,27,28/                                       GRTS0084
      DATA LXTCOD/30,31,32,33,34/                                       GRTS0085
      DATA TEST/-.12345678E20/                                          GRTS0086
      IERR=0                                                            GRTS0087
      INPCNT=0                                                          GRTS0088
      MTYPE=2                                                           GRTS0089
      MODE=2                                                            GRTS0090
      MOUNT=1                                                           GRTS0091
      KTAU=0                                                            GRTS0092
      IDO=999                                                           GRTS0093
      IDSO=999                                                          GRTS0094
      DO 1 I=1,2                                                        GRTS0095
      KNTROL(I)=4                                                       GRTS0096
      KNTROL(I+2)=1                                                     GRTS0097
      IF(IFOMIT(I,IRAW).NE.0) KNTROL(I)=0                               GRTS0098
      IF(IFOMIT(I+2,IRAW).NE.0) KNTROL(I+2)=0                           GRTS0099
    1 CONTINUE                                                          GRTS0100
      YR=NYR(IRAW)                                                      GRTS0101
      TFINSH=FINISH(IRAW)                                               GRTS0102
      TSTLO=TLO(IRAW)                                                   GRTS0103
      TSTHI=THI(IRAW)                                                   GRTS0104
      KSMAX=KMAX(IRAW)                                                  GRTS0105
C     READ FIRST RECORD OF DOUBLE ASTERISKS                             GRTS0106
      READ(9,602) ARCIND                                                GRTS0107
      KIX=0                                                             GRTS0108
      ISTART=0                                                          GRTS0109
C     TEST RECORD FOR DOUBLE ASTERISKS                                  GRTS0110
   20 READ(9,602) ARCIND                                                GRTS0111
      IF(ARCIND.NE.EOTIND) GO TO 21                                     GRTS0112
C     IF DOUBLE ASTERISKS IN TWO SUCCESSIVE RECORDS, END OF TAPE        GRTS0113
C     IF JUST ONCE, END OF DATA ARC                                     GRTS0114
      IF(KIX.EQ.0) GO TO 999                                            GRTS0115
      KIX=0                                                             GRTS0116
      GO TO 60                                                          GRTS0117
   21 BACKSPACE 9                                                       GRTS0118
C     READ A DATA FRAME,   TWO RECORDS                                  GRTS0119
      READ(9,605) X,AR(1),R(1),BRD(1),RD(1),DAY,HR,XMIN,SEC,AR(2),R(2), GRTS0120
     1BRD(2),RD(2),Y,AR(3),R(3),BRD(3),RD(3),ISAT,IDS,IXX,AR(4),R(4),   GRTS0121
     2BRD(4),RD(4)                                                      GRTS0122
      KIX=1                                                             GRTS0123
C     DECODE DATA KEYS AND CONVERT TIME TAG                             GRTS0124
C     FIND IF STATION NUMBER IS UNCHANGED                               GRTS0125
      IF(IDS.EQ.IDSO) GO TO 220                                         GRTS0126
C     TEST IF STATION ID VALID                                          GRTS0127
      DO 22 I=1,KSMAX                                                   GRTS0128
      IF(IDS.EQ.ISTAOK(I,IRAW)) GO TO 220                               GRTS0129
   22 CONTINUE                                                          GRTS0130
      GO TO 20                                                          GRTS0131
C     REASSIGN STATION ID CODE FOR BLOCK DATA                           GRTS0132
  220 DO 221 I=1,5                                                      GRTS0133
      INTLOC=I                                                          GRTS0134
      IF(IDS.EQ.INTCOD(I)) GO TO 222                                    GRTS0135
  221 CONTINUE                                                          GRTS0136
      WRITE(6,903) IDS                                                  GRTS0137
      GO TO 999                                                         GRTS0138
  222 ID=LXTCOD(INTLOC)                                                 GRTS0139
C     SET UP HYBRID SWITCH.  NOTICE THAT NEW ID CODES ARE USED FOR THIS GRTS0140
      HYBRID=ID.GE.33                                                   GRTS0141
   23 CONTINUE                                                          GRTS0142
      DI(1)=YR*100.+1.                                                  GRTS0143
      DI(2)=(DAY*100.+HR)*100.+XMIN                                     GRTS0144
      DI(3)=SEC                                                         GRTS0145
      CALL DATINP(DI,TIMTST)                                            GRTS0146
C     TEST TIME TAG FOR VALID RANGE OF TIMES                            GRTS0147
      IF(TIMTST.LT.TSTLO.OR.TIMTST.GT.THI) GO TO 20                     GRTS0148
      IQAN=0                                                            GRTS0149
```

```
C     TEST IF HYBRID SYSTEM
      IF(HYBRID) GO TO 240
C     HERE IF NOT HYBRID
C     SET UP DATA KEYS
      DO 231 I=1,4
      QRD(I)=BRD(I)
  231 QR(I)=AR(I)
      IDC(1)=IXX(1)
      IDC(2)=IXX(2)
      DO 234 I=3,4
      DO 232 J=1,10
      K=J-1
      IF(IXX(I).EQ.IXXHOL(J)) GO TO 233
  232 CONTINUE
  233 IDC(I)=K
  234 CONTINUE
      IDEL=(IDC(1)-1)/3+1
      IF(IDC(2).EQ.7) TIMTST=TIMTST+.125
      ISAMP=MOD(IDC(2),4)+1
      IDPC=IDC(3)/3+1
      IFTR=IDC(3)+1
      GO TO 299
C     HERE FOR HYBRID SYSTEM
C
  240 IDEL=1
C     THE ABOVE MAY DEPEND ON THE IXX VALUES
C     SET UP DATA KEYS
      ISAMP=MOD(IXX(2),4)+1
      IF(ISAMP.EQ.4) ISAMP=5
      IDPC=IXX(1)/3+1
      IFTR=IXX(1)+1
      IF(ISAMP.NE.5) GO TO 244
C     ADD WHOLE SECONDS TO DOPPLER TIMES
      DO 243 I=1,4
      DO 241 J=1,10
      K=J-1
      IF(BRD(I).EQ. XXHOL(J)) GO TO 242
  241 CONTINUE
      K=0
  242 RD(I)=RD(I)+FLOAT(K)*1.E6
  243 CONTINUE
C     SET UP QUALITY INDICATORS
  244 DO 249 I=1,4
      QR(I)=BLANK
      QRD(I)=BLANK
      IF(AR(I).EQ. XXHOL(12)) GO TO 249
      DO 245 J=1,11
      K=J-1
      IF(AR(I).EQ. XXHOL(J)) GO TO 246
  245 CONTINUE
  246 IF(K.EQ.1) GO TO 247
      IF(K.EQ.3) GO TO 248
      IQAN=1
  247 QRD(I)=XXHOL(11)
  248 QR(I) =XXHOL(11)
  249 CONTINUE
C     SET UP AMBIGUITY COUNT IF POSSIBLE
      DO 252 I=3,4
      NNN(I-2) =0
      DO 250 J=1,10
      K=J-1
      IF(IXX(I).EQ.IXXHOL(J)) GO TO 251
  250 CONTINUE
      GO TO 252
  251 NNN(I-2)=K
  252 CONTINUE
      DO 253 I=1,4
  253 R(I)=R(I)+FLOAT(10*NNN(1)+NNN(2))*XDEL(IDEL)
  299 INPCNT=INPCNT+1
      TIMTAG(INPCNT)=TIMTST
      IQ(INPCNT)=0
      IF(ISTART.EQ.0) GO TO 30
C     IF KEYS CHANGE, A NEW DATA ARC IS FORCED
      IF(IDEL.EQ.IDELO.AND.ISAMP.EQ.ISAMPO.AND.IDPC.EQ.IDPCO.AND.IFTR.EQ
     1.IFTRO.AND.ID.EQ.IDO) GO TO 40
```

```
      INPCNT=INPCNT-1
      BACKSPACE 9
      BACKSPACE 9
      GO TO 60
C     NEW DATA ARC
   30 ISTART=1
C     SET UP CONSTANTS FOR KEY RECORD
      DELT=XSAMP(ISAMP)
      TAU=XCOUNT(ISAMP,IDPC)
      TRF=XFTR(IFTR)
      DR=XDEL(IDEL)
      BIASF=XBIAS(IDPC)
C     FIND STATION ID IN BLOCK DATA
      DO 31 I=1,50
      KSTA=I
      IF(ID.EQ.KODSTA(I)) GO TO 32
   31 CONTINUE
      GO TO 991
   32 IF(NUMSTA.EQ.0) GO TO 34
C     IS STATION NAME ALREADY IN SUMMARY DATA
      DO 33 I=1,NUMSTA
      KK=I
      IF(NAMSTA(I).EQ.STANAM(KSTA)) GO TO 35
   33 CONTINUE
C     NEW STATION NAME
   34 NUMSTA=NUMSTA+1
      IF(NUMSTA.GT.20) GO TO 992
      NAMSTA(NUMSTA)=STANAM(KSTA)
      STIMNX(NUMSTA)=TIMTAG(INPCNT)
      GO TO 36
C     OLD STATION NAME
C     UPDATE STATION FIRST ON TIME
   35 Z=TIMTAG(INPCNT)
      STIMNX(KK)=AMIN1(STIMNX(KK),Z)
   36 IDELO=IDEL
      ISAMPO=ISAMP
      IDPCO=IDPC
      IFTRO=IFTR
      IDO=ID
      IDSO=IDS
      OBSNAM=STANAM(KSTA)
      TRANAM=OBSNAM
      CR1=C1(KSTA)
      CR2=C2(KSTA)
C
C     THIS MIGHT BE MADE TO DEPEND ON ISAT
C     BLOCK DATA VALUE IS FOR S-BAND = 60/80
      RATIO=RETR(KSTA)
C     THIS IS FOR VHF  =12/13
      IF(IFTR.EQ.4) RATIO=.92307692307692308
C
   40 CONTINUE
C     LOAD FOUR ROWS OF DATA ARRAY
      WT=TIMTAG(INPCNT)
      TXTRA(INPCNT)=TIMTAG(INPCNT)-TIMTAG(1)
      XANG(INPCNT)=X*.0174532926
      IF(KNTROL(1).EQ.0) XANG(INPCNT)=TEST
      YANG(INPCNT)=Y*.0174532926
      IF(KNTROL(2).EQ.0) YANG(INPCNT)=TEST
      RANGE(INPCNT)=R(1)*1.E-6
      IF(KNTROL(3).EQ.0.OR.RANGE(INPCNT).EQ.0.) RANGE(INPCNT)=TEST
      DOP(INPCNT)=RD(1)*1.E-6
      IF(KNTROL(4).EQ.0.OR.DOP(INPCNT).EQ.0.) DOP(INPCNT)=TEST
      IF(KNTROL(1).EQ.0.OR.KNTROL(2).EQ.0.OR.IQAN.NE.0)
     1 IQ(INPCNT)=IQ(INPCNT)+1
      IF(QR(1).NE.BLANK.OR.KNTROL(3).EQ.0) IQ(INPCNT)=IQ(INPCNT)+2
      IF(QRD(1).NE.BLANK.OR.KNTROL(4).EQ.0) IQ(INPCNT)=IQ(INPCNT)+4
      DO 41 I=2,4
      INPCNT=INPCNT+1
      IQ(INPCNT)=1
      TIMTAG(INPCNT)=WT+FLOAT(I-1)*DELT
      TXTRA(INPCNT)=TIMTAG(INPCNT)-TIMTAG(1)
      XANG(INPCNT)=TEST
      YANG(INPCNT)=TEST
      RANGE(INPCNT)=R(I)*1.E-6
```

```
      IF(KNTROL(3).EQ.0.OR.RANGE(INPCNT).EQ.0.) RANGE(INPCNT)=TEST
      DOP(INPCNT)=RD(I)*1.E-6
      IF(KNTROL(4).EQ.0.OR.DOP(INPCNT).EQ.0.) DOP(INPCNT)=TEST
      IF(QR(I).NE.BLANK.OR.KNTROL(3).EQ.0) IQ(INPCNT)=IQ(INPCNT)+2
   41 IF(QRD(I).NE.BLANK.OR.KNTROL(4).EQ.0) IQ(INPCNT)=IQ(INPCNT)+4
      IF(INPCNT.EQ.INPMAX) GO TO 60
      IF(TIMTAG(INPCNT)-TIMTAG(1).GE.TMAX) GO TO 60
      IF(TIMTAG(INPCNT).GE.TFINSH) GO TO 100
      GO TO 20
   60 ASSIGN 20 TO JUMP
      ISTART=0
      GO TO 70
  100 ASSIGN 999 TO JUMP
   70 IF(INPCNT.LT.4) GO TO 990
      ONTIME=TIMTAG(1)
C     MAKE ALL ANGLES POSITIVE, BUT RETAIN CONTINUITY
      DO 710 I=1,2
      DO 710 K=1,INPCNT,4
      IF(RDAT(K,I).EQ.TEST) GO TO 710
      IF(RDAT(K,I).GE.0.) GO TO 710
      DO 71 L=1,INPCNT,4
      IF(RDAT(L,I).EQ.TEST) GO TO 71
      RDAT(L,I)=RDAT(L,I)+6.28318531
   71 CONTINUE
  710 CONTINUE
C     REMOVE RANGE RECYCLE DISCONTINUITIES
      DEL=DR
      STEP=.5*DEL
      DO 72 I=1,INPCNT
      KK=I+1
      IF(RANGE(I).NE.TEST) GO TO 73
   72 CONTINUE
      GO TO 79
   73 IF(KK.GT.INPCNT) GO TO 79
      KHOLD=KK-1
      DO 78 I=KK,INPCNT
      IF(RANGE(I).EQ.TEST) GO TO 78
      DIFF=RANGE(I)-RANGE(KHOLD)
      IF(ABS(DIFF).LT.STEP) GO TO 77
      IF(DIFF.LT.0.) GO TO 74
      II=KK
      LL=KHOLD
      GO TO 75
   74 II=I
      LL=INPCNT
   75 DO 76 M=II,LL
      IF(RANGE(M).EQ.TEST) GO TO 76
      RANGE(M)=RANGE(M)+DEL
   76 CONTINUE
   77 KHOLD=I
   78 CONTINUE
   79 CONTINUE
C     DO POLYNOMIAL FITS, TEST FOR OUTLIERS
C     PUT DATA ON TAPE 12
      CALL POLYFT(INPCNT)
      TSTART=DMIN1(TSTART,TIMTAG(1))
      TSTOP=DMAX1(TSTOP,TIMTAG(INPCNT))
  990 INPCNT=0
      GO TO JUMP,(20,999)
  991 WRITE(6,901) ID
      GO TO 999
  992 WRITE(6,902)
      NUMSTA=NUMSTA-1
      IERR=2
  999 RETURN
      END
```

Subroutine: GTR2BD

Purpose: To compute the transformation from body-fixed to inertial coordinates.

Calling Sequence: CALL GTR2BD (ICB,DN,KEY)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ICB				Central body number.
I	DN	d(4)	$\delta\Psi, \delta\epsilon, \delta\alpha$	rad	Nutations in longitude (1) and obliquity (2) are input. The nutation in right ascension (4) is output.
I	KEY				The transformation T_{C2D} is computed only if ICB = 3,11 or KEY > 0.

Common storages used: // 90 cells, /TRJCØM/

Subroutines required: DEHA,DEQTR,DGTRN,DGTSN,DLUNE,DMVTRN

Usage

Ephemeris (ETIMV) and universal (UTIMV) times must be loaded in TRJCØM before entry. The subroutine computes the corresponding transformation T_{B2C}, from body-fixed to Earth's equator, equinox of 1950.0 coordinates.

If KEY > 0 or ICB = 3 or 11, the transformations T_{C2D0}, to Earth's mean equator of date, T_{C2D} to Earth's true equator of date, and T_{B2D} are also computed.

If ICB = 3 or 11, the array CBØ of TRJCØM is computed by GTR2BD. Otherwise, some components must be loaded prior to entry. The components of CBØ are described for the various central bodies in the Table below. Quantities marked *( ) must be loaded prior to entry.

The coordinate systems and transformations are described in the descriptions of the subroutines DEHA,DEQTR,DLUNE and in Appendix A of Reference 1.

CBØ (6)

Central Body Orientation

Component	Central Body			Definition
	Earth	Moon	Other	
1	ω	ω	$^*\omega$	Central body angular velocity.
2	$\delta\Upsilon$		$^*\Upsilon_o$	Nutation in right ascension. Right ascension of prime meridian at 1950 Jan 0.0 ET
3	Υ		Υ	Right ascension of prime meridian.
4	$\bar{\epsilon}$		$^*\epsilon'$	Mean obliquity of date.
5	$\delta\Psi$		$^*L$	Nutation in longitude. Longitude of ascending node of Earth's ecliptic on central body's equator.
6	ϵ		$^*\bar{\epsilon}_{50}$	True obliquity. Earth's mean obliquity of 1950.0

```
$IBFTC MC1321  XR3,M94,NODD,LIST
      SUBROUTINE GTR2BD (ICB,DN,KEY)
C     COMPUTES CENTRAL BODY ORIENTATION
C
      DOUBLE PRECISION   DN(4)
      COMMON            /TRJCOM/CTRJ(246)
      DOUBLE PRECISION   CBO(6) ,ETFMS ,UTIMV
      EQUIVALENCE      (CTRJ(87),CBO   ),(CTRJ( 5),TC2D0),(CTRJ(41),TB2D )
     1                ,(CTRJ( 1),ETFMS),(CTRJ(23),TC2D ),(CTRJ(59),TB2C )
     2                ,(CTRJ( 3),UTIMV)
      COMMON             SAVE(40),T(16),D(9)
      DOUBLE PRECISION   D,T
      DIMENSION          MD(5)
      DATA               MD/1,-3,-1,-3,1/
C
    1 JCB = 1
      IF (ICB.EQ. 3) GO TO 11
      IF (ICB.EQ.11) GO TO 10
      JCB = 3
      IF (KEY) 40,40,11
C     EARTHS EQUATOR EQUINOX OF DATE
   10 JCB = 2
   11 CALL DEQTR (ETFMS,D)
      CALL DGTSN (TC2D0,0,D(2),D(1),D(4))
      T(4) = D(3)
      T(5) = DN(1)
      T(6) = D(3)+DN(2)
      CALL DGTRN  (D,MD,T(4),3)
      CALL DMVTRN (D,TC2D0,TC2D,1,3)
      DN(4) = D(2)
      GO TO (20,30,40) ,JCB
C     EARTH-FIXED
   20 CALL DEHA   (UTIMV,DN(4),T(3),T)
      CALL DGTRN  (TB2D,-3,T(3),1)
      DO 21 I=1,6
   21 CBO(I) = T(I)
   22 CALL DMVTRN (TC2D,TB2D,TB2C,2,3)
      GO TO 999
C     MOON-FIXED
   30 D(1) = T(4)
      CALL DLUNE  (ETFMS,D(1),DN(2),DN(1),TB2D,CBO)
      GO TO 22
C     PLANET-FIXED
   40 CBO(3) = CBO(2)+ETFMS*CBO(1)
      CALL DGTRN  (TB2C,MD(2),CBO(3),4)
      CALL DMVTRN (TC2D,TB2C,TB2D,1,3)
  999 RETURN
      END
```

Subroutine: INITAP

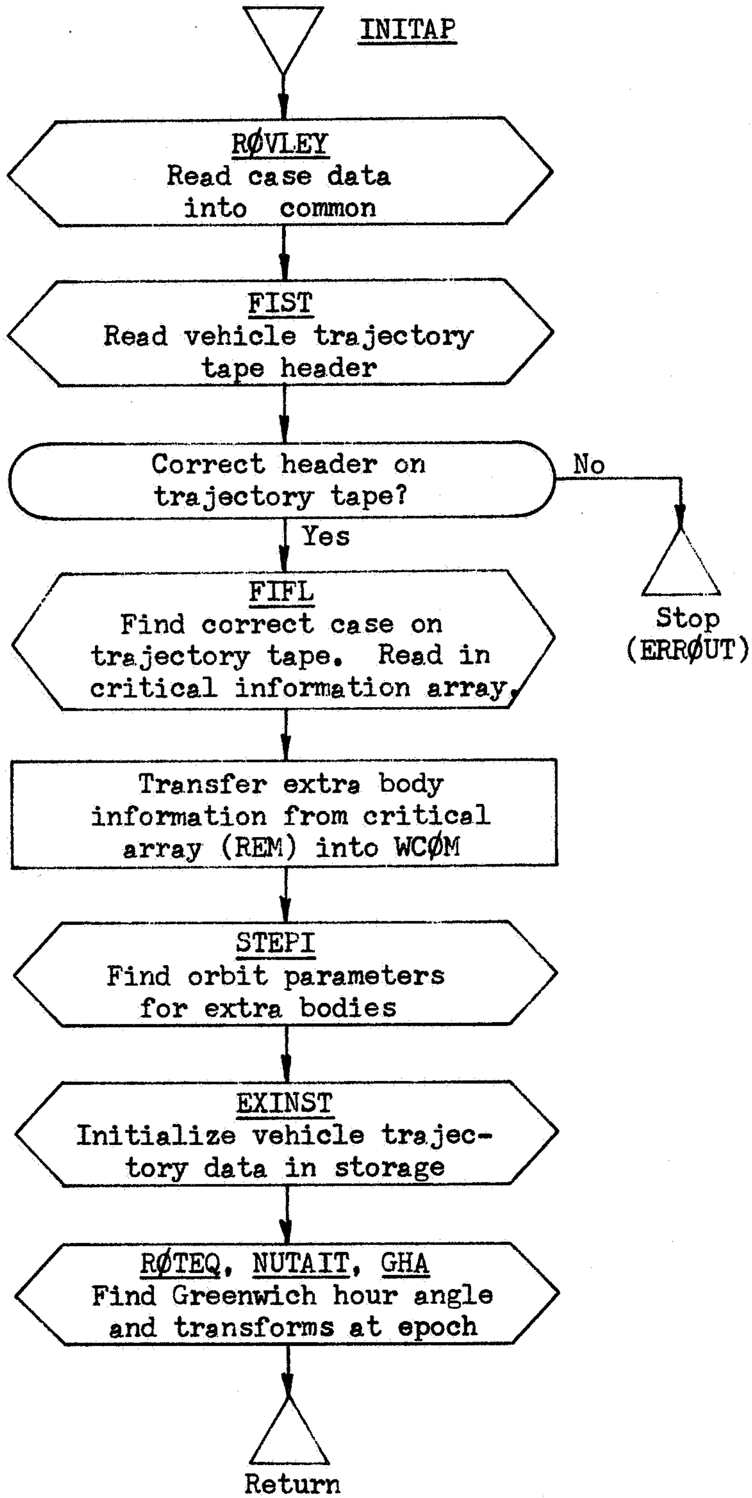
Purpose: To perform initializations for MESERP

Calling Sequence: CALL INITAP

Common storages used: /INPCØM/,/WCØM/,/EXIC/

Subroutines required: DATØUT, EHA, ERRØUT, EXINST, FIFL, FIST, MTRN, NUTAIT, ØRB, RØTEQ, RØVLEY, STEPI, TFRAC, TIMED, TIMES

INITAP


RØVLEY
Read case data into common
FIST
Read vehicle trajectory tape header
Correct header on trajectory tape?
No
Stop (ERRØUT)
Yes
FIFL
Find correct case on trajectory tape. Read in critical information array.
Transfer extra body information from critical array (REM) into WCØM
STEPI
Find orbit parameters for extra bodies
EXINST
Initialize vehicle trajectory data in storage
RØTEQ, NUTAIT, GHA
Find Greenwich hour angle and transforms at epoch
Return

```
$IBFTC MC13IT   NOREF,M94,NODD,XR3
CMC13IT    INITAP PERFORMS INITIALIZATIONS FOR MESERP
      SUBROUTINE INITAP
      DIMENSION     KRX(9,2)
      DIMENSION              AN(3,3),        BCD(11)
     1,    BODC(10,8),     DUM(3,3),        EN(3,3)
     2,                                     HEAD(24),       ICAS(3)
     3,    IREM(255)
     4,        REM(255),    TI(3)
     5,   XIN(6),          XSTOR(9,10),    MSTAR(9,10)
     6,    NOX(3),          RX(3,3),        VX(3,3),          UM(10)
     7,    SHFC(18),        XSTAR(9,10)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)/EXIC/W(36),CRX(9,2)
      EQUIVALENCE         (C(6),SPMSD),  (C(7),RSPMSD), (C(11),BODC)
     2,                                                (C(675),HEAD)
     3,             (C(700),IOCAS), (C(21),UM),   (C(673),IKAS)
     4,       (C(5),DTR)
      EQUIVALENCE                 (IW(377),ICAS)
     1,                           (IW(382),NOR),       (IW(4),NOX)
     2,             (IW(489),ITRIG),   (IW(490),KOUNT)
      EQUIVALENCE  (IW(7),IXT),   (IW(383),ITARG)
     1,    (CW(1134),XIN),        (CW(1140),TSEC)
     2,    (CW(1),RX),            (CW(10),VX),          (CW(1160),TI)
     3,    (CW(1163),AN),         (CW(1172),TSECO),     (CW(1179),FLTIM)
     4,                                                 (CW(1173),GHAR)
     5,    (CW(1372),SHFC),       (CW(1427),TFO),       (CW(1426),TWO)
      EQUIVALENCE     (KRX,CRX)
      EQUIVALENCE  (IREM,REM),     (REM(36),XSTAR,MSTAR)
     1,                           (REM(162),BCD)
      DOUBLE PRECISION TWR
  701 FORMAT(1H1,11X,28HSTARTING CONDITIONS FOR CASE,I3,1X,12A6)
  702 FORMAT(1H0,1A6,1X,18HIS THE TARGET BODY/13H FLIGHT TIME ,F12.4,
     1 5HDH.MS/10H REF. TIME,2X,F12.4,5HDH MS)
 703  FORMAT(1H ,1A6,1X,8HCENTERED,91X,12HEQUATOR 1950/
     15H     X,E15.8,5H     Y,E15.8,5H     Z,E15.8,5H    DX,E15.8,
     25H    DY,E15.8,5H    DZ,E15.8)
  710 FORMAT(1H0,11X,21HFROM TAPE STORED CASE,I3,1X,12A6)
  711 FORMAT(39H0*WARNING* START TIME LT TAPE ZERO TIME)
      CALL ROVLEY(C,IW,HEAD)
      IER=0
      CALL FIST(HEAD,IER,10)
      IF(IER.EQ.2) CALL ERROUT(1,27H(22HOUNABLE TO FIND HEADER))
      ICAS(2)=0
      ICAS(3)=1
      CALL FIFL(ICAS,J,REM,IER,10)
      IF(IER.LT.0) CALL ERROUT(1,31H(23HOBIN TAPE READ ERR TYPE I3),IER)
      IKAS=IKAS+1
      ITRIG=0
      KOUNT=0
      WRITE (6,701) IKAS,(HEAD(I),I=13,24)
      WRITE(  6 ,710) ICAS(1),BCD
      NNN=IREM(35)
      TI(1)=REM(173)
      TI(2)=REM(174)
      DO 111 I=1,9
      DO 111 J=1,10
  111 XSTOR(I,J)=XSTAR(I,J)
C TRANSFER EXTRA BODY INFO FROM REM ARRAY INTO WCOM
      DO 115 I=1,7
  115 IW(I)=IREM(I+3)
      DO 120 I=1,24
  120 CW(I)=REM(I+10)
      IF(IXT.EQ.0) GO TO 123
      DO 122 J=1,IXT
      KK=6*J-5
      L=NOX(J)
      CALL STEPI(RX(1,J),VX(1,J),UM(L),SHFC(KK))
  122 CONTINUE
 123  CONTINUE
      IOCAS=IOCAS+1
      FLTIM=(XSTAR(8,NNN)+XSTAR(9,NNN))*SPMSD
      CALL TIMES(FLTIM,C(101),DUM)
      CALL TIMED(C(99),TSECO)
      ITARG=MSTAR(7,NNN)
      WRITE(  6 ,702) BODC(ITARG,1),C(101),C(99)
```

```
      TZERO=(XSTAR(8, 1 )+XSTAR(9, 1 ))*SPMSD
C FIND RELATION OF NEW REFERENCE TIME TO STORED ORBIT
      I=2
      IF(TSECO-TZERO) 125,140,130
 125  WRITE(6,711)
      TSECO=TZERO
      GO TO 140
130     CONTINUE
      T=(XSTAR(8,NNN)+XSTAR(9,NNN))*SPMSD
      IF(TSECO-T) 140,135,135
 135  CONTINUE
      CALL ERROUT(1,36H(31HOSTART TIME EXCEEDS FLIGHT TIME))
 140  CONTINUE
      NNI=3
      NOR=MSTAR(7,2)
C SAVE INITIAL WHOLE AND FRACTIONAL DAYS
      TWO=TI(1)
      TFO=TI(2)
C UPDATE TW,TF TO CORRESPOND TO NEW ZERO TIME
      CALL TFRAC(TI(1),TI(2)+(TSECO-TZERO)*RSPMSD,TI(1),TI(2))
      M=0
      N=18
      TSEC=TSECO
      TWR=TSEC
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
 148  CONTINUE
      M=1
      N=3
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
      IF(N.GE.0) GO TO 1485
      NOR=KRX(7,2)
      NNI=NNI+2
      GO TO 148
1485    CONTINUE
      CALL DATOUT (TI,TI(2),DUM,DUM(2,1),0)
      WRITE(6,703) BODC(NOR,1),(XIN(I),I=1,6)
      CALL ORB ( XIN,XIN(4),UM(NOR),6)
      TSEX=TI(2)*SPMSD
      TIME=TI(1)+TI(2)
      CALL ROTEQ(TIME,DUM)
      CALL NUTAIT(TIME,WM,CR,DA,EN,EPSIL)
      CALL MTRN(EN,DUM,AN)
      CALL GHA(TSEX,TI(1),GHAR,EN(2,1),WET)
      GHAR=GHAR*DTR
      RETURN
       END
```

Subroutine: LAYØ

Purpose: To input the station, beacon, and on-board data cards and convert the information to units used internally.

Calling Sequence: CALL LAYØ(S,N,ITEMP)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	S	(23)			Card input is read into S(J) where J is the index on the input cards.
I	N				N = 1, read station data N = 2, read beacon data N = 3, read on-board data N = 4, read equation of motion data (this option not used by the TDS).
I	ITEMP	(23)			If N ≠ 2, ITEMP(2) is not used. If N = 2, and ITEMP(2) = 0, beacon measurement data. ITEMP(2) = 1, individual beacon data.

Common storages used: None

Subroutines required: TIMED

```
$IBFTC MC13LO  NOREF,M94,NODD,XR3
CMC13LO    LAYO    SAME AS LAYOVR EXCEPT ERRORS NOT SQUARED              LAYO0001
      SUBROUTINE LAYO  (S,N,ITEMP)                                        LAYO0002
      DIMENSION S(23),ITEMP(23),IND(4),BUF(4)                             LAYO0003
      DATA DTR,OT,STR/ .0174532926,.001, .48481368E-5/                    LAYO0004
      KSTOP=0                                                             LAYO0005
      IT3=ITEMP(3)                                                        LAYO0006
    1 CONTINUE                                                            LAYO0007
      READ(5,100) (IND(I),BUF(I),I=1,4)                                   LAYO0008
  100 FORMAT(4(I3,E12.8))                                                 LAYO0009
      IF(IND)3,70,3                                                       LAYO0010
    3 DO 60 I=1,4                                                         LAYO0011
      J=IND(I)                                                            LAYO0012
      IF(J)4,60,5                                                         LAYO0013
    4 J=-J                                                                LAYO0014
      KSTOP=J                                                             LAYO0015
 5    CONTINUE                                                            LAYO0016
      B = BUF(I)                                                          LAYO0017
      GO TO (10,20,30,59),N                                               LAYO0018
 10   GO TO (60,51,52,52,53,52,52,53,59,53,53,53,59,53,53,53,53,53,59,57LAYO0019
     1,59,57,59),J                                                        LAYO0020
 20   IF(ITEMP(2))60,21 ,22                                               LAYO0021
21    IF(J.EQ.11) GO TO 59                                                LAYO0022
      IF(J-2) 51,51,53                                                    LAYO0023
 22   GO TO (60,52,52,53,52,53,53,53),J                                   LAYO0024
 30   GO TO (53,53,53,53,59,59,51,56,59,56,59,56,59,51,52,52,52,52),J   LAYO0025
 51   BVD = B                                                             LAYO0026
      CALL TIMED(BVD,B)                                                   LAYO0027
      GO TO 59                                                            LAYO0028
 52   B = DTR*B                                                           LAYO0029
C     DEGREES TO RADIANS                                                  LAYO0030
      GO TO 59                                                            LAYO0031
   56 B=STR*B                                                             LAYO0032
C     ARCSECONDS TO RADIANS                                               LAYO0033
      GO TO 59                                                            LAYO0034
   57 B=B*1.E6                                                            LAYO0035
C     MEGACYCLES PER SECOND TO CYCLES PER SECOND                          LAYO0036
      GO TO 59                                                            LAYO0037
   53 B=OT*B                                                              LAYO0038
C     METERS TO KILOMETERS OR MILLIRADIANS TO RADIANS                     LAYO0039
   59 S(J)=B                                                              LAYO0040
      IF(KSTOP.EQ.J) GO TO 70                                             LAYO0041
   60 CONTINUE                                                            LAYO0042
      GO TO 1                                                             LAYO0043
   70 CONTINUE                                                            LAYO0044
      RETURN                                                              LAYO0045
      END                                                                 LAYO0046
```

Subroutine: LØCAT

Purpose: To add N, E, D bias errors to station and beacon locations.

Calling Sequence: CALL LØCAT

Common storages used: /INPCØM/,/WCØM/

Subroutines required: FNØRM, STAT

```
$IBFTC MC13LC  NOREF,M94,NODD,XR3
CMC13LC  LOCAT  ADDS N,E,D,BIAS ERRORS TO STATION,BEACON LOCATIONS      LCAT0001
      SUBROUTINE LOCAT                                                  LCAT0002
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)                        LCAT0003
      DIMENSION     ISC(12),  IS(11,12),S(23,12), BRDE(2),  BIASL(3,22) LCAT0004
     1,             IBC(10),  IB(3,12), B(8,10),  BRDB(2)               LCAT0005
     2,             BODC(10,8),         CON(3),   TM(3,3),  RT(3)       LCAT0006
     3,             SC(5)                                               LCAT0007
      EQUIVALENCE   (C(11),BODC),       (C(200),S),         (C(476),B)  LCAT0008
     1,             (IW(8),ISC),        (IW(20),IBC),       (IW(180),IS)LCAT0009
     2,             (IW(312),IB),       (IW(347),NBEAC)                 LCAT0010
     3,            (CW(156),BIASL)                                      LCAT0011
      DATA CON(3)/-1./                                                  LCAT0012
      BRDE(1)=BODC(1,3)                                                 LCAT0013
      BRDE(2)=BODC(1,4)                                                 LCAT0014
      DO 19 I=1,12                                                      LCAT0015
      IF(ISC(I)) 20,20,10                                               LCAT0016
10    NSTAT=1                                                           LCAT0017
      K=ISC(I)                                                          LCAT0018
      DO 19 M=6,8                                                       LCAT0019
      IF(IS(M,K)) 11,19,11                                              LCAT0020
11    GO TO (12,15),NSTAT                                               LCAT0021
12    CALL STAT(S(3,K),0.,TM,RT,SC,BRDE)                                LCAT0022
      NSTAT=2                                                           LCAT0023
      CON(1)=FNORM(RT)                                                  LCAT0024
      CON(2)=CON(1)*SC(2)                                               LCAT0025
   15 L=M-3                                                             LCAT0026
      J=M+10                                                            LCAT0027
      N=M-5                                                             LCAT0028
      BIASL(N,K)=S(J,K)/CON(N)                                          LCAT0029
      S(L,K)=S(L,K)+BIASL(N,K)                                          LCAT0030
19    CONTINUE                                                          LCAT0031
20    IF(IBC.EQ.0) GO TO 50                                             LCAT0032
      BRDB(1)=BODC(NBEAC,3)                                             LCAT0033
      BRDB(2)=BODC(NBEAC,4)                                             LCAT0034
      DO 29 I=1,10                                                      LCAT0035
      K=IBC(I)                                                          LCAT0036
      IF(K) 50,50,21                                                    LCAT0037
21    NSTAT=1                                                           LCAT0038
      DO 29 M=1,3                                                       LCAT0039
      IF(IB(M,K)) 22,29,22                                              LCAT0040
22    GO TO (23,25),NSTAT                                               LCAT0041
23    CALL STAT(B(2,K),0.,TM,RT,SC,BRDB)                                LCAT0042
      NSTAT=2                                                           LCAT0043
      CON(1)=FNORM(RT)                                                  LCAT0044
      CON(2)=CON(1)*SC(2)                                               LCAT0045
25    L=M+1                                                             LCAT0046
      J=M+5                                                             LCAT0047
      BIASL(M,K+12)=B(J,K)/CON(M)                                       LCAT0048
      B(L,K)=B(L,K)+BIASL(M,K+12)                                       LCAT0049
29    CONTINUE                                                          LCAT0050
50    CONTINUE                                                          LCAT0051
CCCCC   ANY RELATED WORK FOR ONBOARD                                    LCAT0052
      RETURN                                                            LCAT0053
      END                                                               LCAT0054
```

Subroutine: MEASUR

Purpose: Computes and stores an arc of appropriate type measurements.

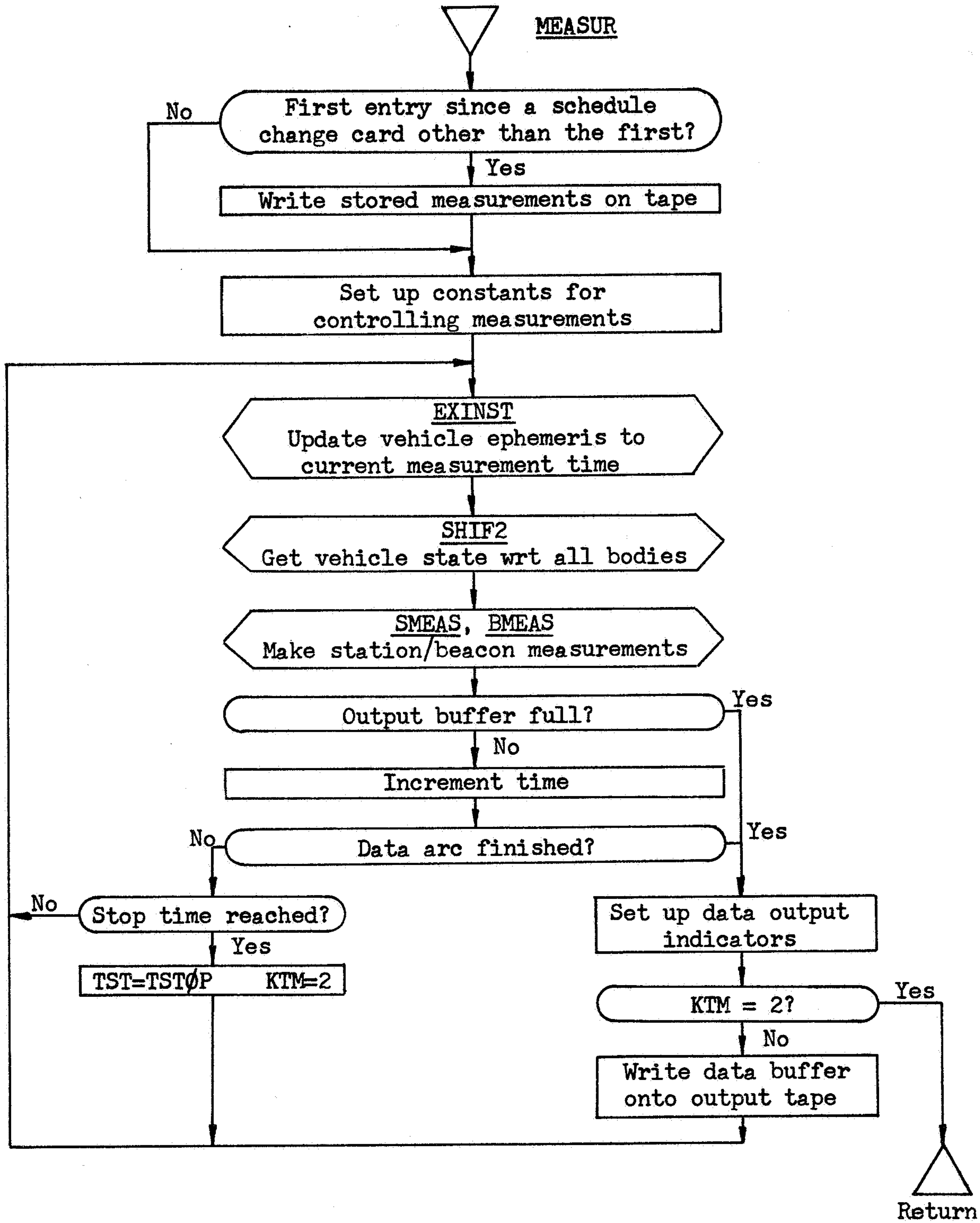
Calling Sequence: CALL MEASUR (TSTP,TSTØP,KSTØP,DEL,L,BRAD, KMAXS,KMAXB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I/Ø	TSTP			Seconds	Current time from epoch.
I	TSTØP			Seconds	Time to stop making this arc of measurements.
I	KSTØP				Key to indicate measurement type (i.e, station, beacon, etc.).
I	DEL			Seconds	Time step between sequential measurements.
I	L				Station or beacon number.
I	BRAD	(3)		km,km rad/sec	Earth semi-major axis, semi-minor axis, rotation rate.
I	KAMXS				Total number of on-off time pairs in SSTART,SSTØP arrays.
I	KMAXB				Total number of on-off time pairs in BSTART,BSTØP arrays.

Common storages used: /INPCØM/,/WCØM/,/EXIC/,/BUFCØM/

Subroutines required: BMEAS,EXINST,ERROUT,SHIF2,SMEAS,TFRAC


MEASUR
First entry since a schedule change card other than the first?
No
Yes
Write stored measurements on tape
Set up constants for controlling measurements
EXINST
Update vehicle ephemeris to current measurement time
SHIF2
Get vehicle state wrt all bodies
SMEAS, BMEAS
Make station/beacon measurements
Output buffer full?
Yes
No
Increment time
Data arc finished?
No
Yes
Stop time reached?
No
Yes
TST=TSTØP KTM=2
Set up data output indicators
KTM = 2?
Yes
No
Write data buffer onto output tape
Return

```
$IBFTC MC13UR  NOREF,M94,NODD,XR3
CMC13UR  MEASUR  COMPUTES AND STORES MEASUREMENTS EVERY DEL
      SUBROUTINE MEASUR(TSTP,TSTOP,KSTOP,DEL,L,BRAD,KMAXS,KMAXB)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)/EXIC/W(36),CRX(9,2)
      COMMON  /BUFCOM/BUFF1(40) ,KEY
      DOUBLE PRECISION  DBUFF(10)
      DOUBLE PRECISION  ONTM,      TFST,      TLST,      TAU,      FTR
     1,                 C1,        C2,        RAMB,      BIAS,     OMG4
      EQUIVALENCE  (BUFF1(1),IKY),    (BUFF1(2),KEE),    (BUFF1(3),STANAM)
     1,            (BUFF1(4),STNAM1),(BUFF1(5),NRCD),    (BUFF1(6),NKOUNT)
     2,            (BUFF1(7),KONTIN),(BUFF1(8),MTYPE),  (BUFF1(9),NALIGN)
     3,            (BUFF1(10),MODE), (BUFF1(11),FMS),   (BUFF1(12),KTAU)
     4,            (BUFF1(13),DBUFF,ONTM)
      EQUIVALENCE  (DBUFF(2),TFST),   (DBUFF(3),TLST),   (DBUFF(4),TAU)
     1,            (DBUFF(5),FTR),    (DBUFF(6),C1),     (DBUFF(7),C2)
     2,            (DBUFF(8),RAMB),   (DBUFF(9),BIAS),   (DBUFF(10),OMG4)
      DIMENSION    ABC(2,4), BRDB(3),  BODC(10,8),      TI(2)
     1,            XIN(6),    XOUT(6,10),               S(23,12)
     2,      BRAD(3),     IS(11,12),XMES(5, 85), B(91)
     3,    IB(36)
      DIMENSION   KRX(9,2)
      DIMENSION   SSTART(75),    BSTART(75)
      EQUIVALENCE  (C(11),BODC),  (C(7),RSPMSD),  (C(200),S)
     1,            (C(476),B),     (C(6),SPMSD),  (C(115),XTROUT)
     2,            (CW(1160),TI), (CW(1140),TSEC),     (CW(1134),XIN)
     3,            (IW(382),NOR), (IW(380),NMAX),      (IW(312),IB)
     4,            (IB(36),NB),    (IB(35),IBTBIS),    (CW(1426),TWO)
     5,            (IW(180),IS),   (CW(522),XMES),     (CW(1427),TFO)
      EQUIVALENCE  (C(116),CFRAC1),  (C(117),CFRAC2)
      EQUIVALENCE   (CRX,KRX)
      EQUIVALENCE  (IW(384),IK1),    (IW(386),IK2),     (IW(489),ITRIG)
     1,            (IW(490),KOUNT),  (CW(222),SSTART), (CW(372),BSTART)
      DIMENSION   TYPE(4)
      DATA TYPE/6HC-BAND,6HGODDRD,6HS-BAND,6H DSIF /
      DATA QYES,QNO/3HYES,2HNO/
C  SUBROUTINES REQUIRED    SMEAS,    BMEAS,    TFRAC,    SHIF2,    EXINST
C                    ERROUT
C     AND EVENTURLLY
C                         OBOMES,  OBRMES
      DOUBLE PRECISION TWR
      DATA (ABC(1,J),J=1,4)/
     112HSTATION  NO.,
     212HBEACON   NO.,
     212HONBRDOPT NO.,
     412HONBRDRAD NO./
      DATA IQUAL/0/
      IF(IABS(KEY).EQ.2) GO TO 1
      ASSIGN 1 TO JUMP
      GO TO 72
    1 CONTINUE
      NRCD=0
      MODE=2
      FMS=DEL
      KTAU=1
      RAMB=1.D20
      NMS=0
      KTM=1
      KEY=1
      IF(KSTOP-2) 39,40,43
   39 CONTINUE
      STANAM=S(1,L)
      STNAM1=STANAM
      MTYPE=IS(1,L)
      NALIGN=IS(10,L)
      IF(MTYPE.EQ.2) NALIGN=1
      TAU=S(23,L)
      FTR=S(22,L)
      C1=CFRAC1
      C2=CFRAC2
      BIAS=S(20,L)
      OMG4=S(21,L)
      GO TO 43
40    BRDB(1)=BODC(NB,3)
      BRDB(2)=BODC(NB,4)
      BRDB(3)=BODC(NB,7)
```

```
43      CONTINUE
        KSORCE=KSTOP
        NSORCE=L
   45 CALL TFRAC(TI(1),TI(2)+(TSTP-TSEC)*RSPMSD,TI(1),TI(2))
      IF(IK1.GE.KMAXS) GO TO 450
      IF(TSTP.GE.SSTART(IK1+1)) ITRIG=1
  450 CONTINUE
      IF(IK2.GE.KMAXB) GO TO 451
      IF(TSTP.GE.BSTART(IK2+1)) ITRIG=1
  451 CONTINUE
      TWR=TSTP
      M=1
      N=3
105   CONTINUE
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
      IF(M.LE.0) CALL ERROUT(1,10H(4H0M1=I2),M)
      IF(N) 115,110,120
110   CONTINUE
      CALL ERROUT(1,28H(23H0ERROR ON BIN TAPE READ))
115   CONTINUE
      N=-N
      NOR=KRX(7,N)
      GO TO 105
120   CONTINUE
      TSEC=TSTP
      CALL SHIF2(2,NOR,TI,XIN,1,XOUT)
      NMS=NMS+1
      GO TO (47,48,49,50),KSTOP
47    CONTINUE
      CALL SMEAS(S(1,L),BRAD,XOUT,XMES(1,NMS),IS(1,L))
      GO TO 60
48    CALL BMEAS(L      ,BRDB,XOUT(1,NB),XMES(1,NMS))
      GO TO 60
49    CONTINUE
C     CALL OBOMES FOR ONBOARD OPTICAL MEASUREMENTS
      GO TO 60
50    CONTINUE
C     CALL OBRMES FOR ONBOARD RADAR   MEASUREMENTS
60    CONTINUE
      IF(NMS.EQ.NMAX) GO TO 70
      GO TO (61,70),KTM
61    TSTP=TSTP+DEL
      IF(TSTP-TSTOP) 45,62,63
   62 KTM=2
      GO TO 45
   63 KTM=2
70    CONTINUE
      IKY=IKY+1
      KEE=0
      IF(IKY.EQ.1) KEE=-1
      NRCD=NRCD+1
      KONTIN=MIN0(NRCD-1,1)
      IF(NRCD.EQ.1) ARCST=XMES(1,1)-TSECO
      ONTM=ARCST+(TWO+TFO)*SPMSD
      TFST=XMES(1,1)+ONTM-ARCST
      TLST=XMES(1,NMS)+ONTM-ARCST
      DO 71 I=1,85
   71 XMES(1,I)=XMES(1,I)-ARCST
      NKOUNT=NMS
      IF(KTM.EQ.2) GO TO 65
      ASSIGN 640 TO JUMP
   72 WRITE(12) BUFF1
      WRITE(12) (IQUAL,I=1,85),((XMES(I,J),J=1,85),I=1,5)
C***  CHECK BINARY OUTPUT
      QOUT=QNO
      IF(KONTIN.EQ.1) QOUT=QYES
      WRITE(6,601) IKY,STANAM,STNAM1,TYPE(MTYPE),QOUT,TFST
  601 FORMAT(//24H ***  RECORD PAIR NUMBER,I5//75H RECEIVING STATION   T
     1RANSMITTING STATION   MEASUREMENT TYPE   CONTINUATION/7X,A6,15X,A6
     2,15X,A6,12X,A3//12H DATA BEGINS,D24.16,20H SECONDS FROM 1950.0)
      TEMP=(TFST -.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,602) TLST
  602 FORMAT(12H0 DATA ENDS ,D24.16,20H SECONDS FROM 1950.0)
```

```
      TEMP=(TLST -.5364576D9)*RSPMSD                                    MSUR0150
      CALL TFRAC(6209.,TEMP,X01,X02)                                    MSUR0151
      CALL DATOUT(X01,X02,Y01,Y02,0)                                    MSUR0152
      WRITE(6,606) NKOUNT                                               MSUR0153
  606 FORMAT(12H0TIME POINTS/I7)                                        MSUR0154
      IF(XTROUT.EQ.0.) GO TO 64                                         MSUR0155
      WRITE(6,607) KEE,NRCD,NALIGN,MODE,FMS,KTAU,ONTM, (DBUFF(I),I=4,10)MSUR0156
     1,(IQUAL,(XMES(I,J),I=1,5),J=1,NKOUNT)                             MSUR0157
  607 FORMAT(5H0NEOT,3X,4HNRCD,3X,6HNALIGN,3X,4HMODE,7X,8HMEAS INT,6X   MSUR0158
     1,4HKTAU,11X,6HONTIME,22X,3HTAU/I4,I7,2I8,E19.8,I6,D27.16,D26.16// MSUR0159
     213X,3HFTR,23X,2HC1,24X,2HC2,24X,2HDR/4D26.16//12X,4HBIAS,21X,5HRATMSUR0160
     3IO/2D26.16//3H IQ,11X,4HTIME,15X,5HANG 1,15X,5HANG 2,15X,5HRANGE, MSUR0161
     414X,7HDOPPLER/(I3,5E20.8))                                        MSUR0162
   64 GO TO JUMP,(1,640)                                                MSUR0163
  640 NMS=0                                                             MSUR0164
      GO TO 61                                                          MSUR0165
65    RETURN                                                            MSUR0166
      END                                                               MSUR0167
```

Subroutine: MEAS2X

Purpose: Converts vehicle state from measurement space to Cartesian space, earth-centered equator and equinox of 1950.0. Also computes the gradient of the Cartesian state vector with respect to the measurement space state vector.

Calling Sequence: CALL MEAS2X(XC,PX2M,XM,NRØT,TL2C,RC2L,ØMEGAC,ØMEGA).

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	XC	d(6)	X_c	km, km/sec	Spacecraft position and velocity in Cartesian space.
Ø	PX2M	d(6,6)	$\frac{\partial X_c}{\partial X_M}$		Gradient of Cartesian state vector with respect to measurement space state vector.
I	XM	d(6)	X_M	km, km/sec,	Spacecraft state in measurement space. (See discussion in subsequent pages.)
I	NRØT	(3)			Defines the axes of rotation of the measurement angles. (See discussion following.)
I	TL2C	d(3,3)			Transformation from coordinates in which measurements are made to C-frame. (See discussion).
I	RC2L	d(3)		km	Origin of measurement coordinate system with respect to C-frame origin in C-frame coordinates.
I	ØMEGAC	d(3)	$\hat{\Omega}$		Unit vector along earth spin axis expressed in C-frame coordinates.
I	OMEGA	d	ω	rad/sec	Earth spin rate, inertial. Set = 0. by calling program (MLESTT) for DSIF(JPL) measurements.

Common storages used: //96 cells.

Subroutines required: DCRØSS,DGTRN,DMVTRN

1. Coordinate Systems

Three coordinate systems are used by the Data Start link of the Orbit Determination Program System. All three are right-handed Cartesian (dextral orthogonal) systems. Two are considered to be inertial systems by the Data Start link; the third is earth-fixed and therefore rotates in inertial space.

The fundamental inertial system used by the ODP system is an earth-centered, dextral orthogonal, equator and equinox of 1950.0 system. This system is always referred to as the C-frame.

The secondary "inertial" system is an earth-centered, dextral orthogonal, equator and equinox of date system. This system is always referred to as the D-frame. Obviously the C-frame and D-frame cannot both be truly inertial systems, and the D-frame is not generally so considered by the ODP system. However, within the Data Start link, the relative motion of the two systems is ignored. The third system is a tracker-centered, dextral orthogonal, earth-fixed, local tangent plane system. The first basis vector lies along the local North vector; the second, along the local East vector; the third, along the local Down vector. This system is always referred to as the L-frame.

Tracking station measurements are always made in the L-frame with one exception: DSIF(JPL) angles are always made in a special coordinate system that is identical to the D-frame except that it is translated (without rotation) so that its center is at the tracker. In this subroutine description only, any reference to the L-frame will mean the true L-frame when any except DSIF measurements are involved and will mean the above described translated D-frame when DSIF measurements are involved; i.e., any reference to the L-frame in this subroutine will mean the coordinate frame in which the measurements are made.

2. State Spaces

The spacecraft state (position and velocity) is expressed and computed in two distinct ways in the Data Start link. The first is a normal Euclidian 3-space whose basis vectors are identical to those of the C-frame. Thus, the state elements in this system are the familiar X, Y, Z, $\dot{X}$, $\dot{Y}$, and $\dot{Z}$ in the C-frame. This system is used by the Data Start link primarily for output and communication with the Differential Correction Program.

The second system, in which most of the actual Data Start link calculations are performed, may be referred to as a measurement space. In this system the spacecraft state is defined by slant range, two angles (e.g., azimuth and elevation), and the time derivatives of these three elements. Since the units are obviously incompatible (kilometers mixed with radians), scaling is employed to avoid numerical difficulties with certain matrix operations. The scaling is of no concern to this subroutine since the input in measurement space is unscaled prior to entry into the subroutine.

3. Extended Description of Input and Output

The previous two sections provide sufficient material so that an extended discussion of input and output is now possible.

3.1 Input

XM, or X_M, is the spacecraft state expressed in measurement space. The actual components depend on the measurement system type as follows:

X_M	MEASUREMENT SYSTEM: C-Band	Goddard	USBS	DSIF	Symbol
1	Range, km	Range, km	Range, km	Range, km	ρ
2	Azimuth, rad	X-angle, rad	X-angle, rad	Right Ascension, rad	α_1
3	Elevation, rad	Y-angle, rad	Y-angle, rad	Declination, rad	α_2
4	Range rate, km/sec	Range rate, km/sec	Range rate km/sec	Range rate, km/sec	$\dot{\rho}$
5	Azimuth rate, rad/sec	X-angle rate, rad/sec	X-angle rate, rad/sec	Right Ascension rate, rad/sec	$\dot{\alpha}_1$
6	Elevation rate, rad/sec	Y-angle rate, rad/sec	Y-angle rate, rad/sec	Declination rate, rad/sec	$\dot{\alpha}_2$

It will be noted that the elements of X_M are not exactly the observables truly measured by the tracker. Conversion from doppler counts to km/sec, for example, is performed at an early stage in the Data Start link, as is conversion from Hour Angle to Right Asecnsion. Angle rates, which are not truly measured, are obtained by the polynomial fitting operation that precedes the maximum likelihood estimation.

TL2C is the transformation from the L-frame (See Section 2 above) to the C-frame. Its columns are the basis vectors of the L-frame expressed in the C-frame.

NRØT contains three integers that define the axes of rotation of the measurement angles. The values depend on the measurement system type as follows:

NRØT	MEASUREMENT SYSTEM				
	C-Band	Goddard	USBS, 30 ft.	USBS, 85 ft.	DSIF
1	1	-3	-3	-3	1
2	2	-2	-2	1	-2
3	3	1	1	2	-3

NRØT is used to obtain the slant-range vector expressed in the C-frame given the range and the two angles along with the L-frame to C-frame transform TL2C. The process, in general, is: Take the NRØT(1)th column vector of TL2C, where a minus sign on NRØT(1) means take the negative of the vector; rotate it about the NRØT(2)th vector of TL2C by XM(3) radians in a right-handed sense if NRØT(2) is positive, left-handed if NRØT(2) is negative; rotate the resulting vector about the NRØT(3)th vector of TL2C, with the same sign convention, by XM(2) radians; and finally multiply the result by the slant range XM(1). The result is the position of the spacecraft with respect to the tracker in the C-frame. For C-Band, for example, one takes the unit North vector (in C-frame components); rotates it about the East vector by the elevation angle; rotates again about the Down vector by the azimuth angle; and multiplies by range. (Note that it is the vector and not the coordinate system that is being rotated.)

RC2L is the position of the tracker with respect to the C-frame origin expressed in the C-frame. Adding this to the above described slant-range vector gives the spacecraft position in the C-frame.

ØMEGAC and ØMEGA define the rotation rate of the L-frame with respect to the C-frame. This vector will be used in deriving the output gradient PX2M. Note that when DSIF measurements are involved, the calling program (MLESTT) sets ØMEGA = 0.

3.2 Output

XC or X_C is the spacecraft position and velocity in the C-frame. The position, X_C(1-3), which we shall denote as R_C is computed as described above in the description of NRØT. After the introduction of some notation, R_C will be derived in mathematical notation, as will the velocity, X_C(4-6), to be denoted as V_C.

Notation:

R_X = Spacecraft/position in X-frame (X = C or X = L or X = M)

V_X = Spacecraft velocity in X-frame

$$X_X = \begin{pmatrix} R_X \\ \\ V_X \end{pmatrix}$$

Ω = L-frame rotation rate vector expressed in C-frame. (Null vector for DSIF.)

βX = the skew-symmetric matrix

$$\begin{pmatrix} 0 & -\beta_3 & \beta_2 \\ \beta_3 & 0 & -\beta_1 \\ -\beta_2 & \beta_1 & 0 \end{pmatrix}$$

which has the effect of crossing the vector β into the vector on which βX operates.

$T(\beta, \alpha)$ is a rotation matrix that rotates the vector on which it operates about the vector β in a right-handed sense through angle α.

Note that

$$\frac{\partial}{\partial\alpha} T(\beta, \alpha) = \beta \times T(\beta, \alpha) = T\beta X$$

Definitions:

e_0 = |NRØT(1)|-th column of TL2I times the sign of NRØT(1)

β_1 = |NRØT(2)|-th column of TL2I times the sign of NRØT(2)

β_2 = |NRØT(3)|-th column of TL2I times the sign of NRØT(3)

Then

$$R_L = \rho \cdot (T_2 \cdot T_1 \cdot e_0) \qquad (1)$$

where

$$T_1 = T(\beta_1, \alpha_2)$$

$$T_2 = T(\beta_2, \alpha_1)$$

and

$$V_L = \frac{\partial R_L}{\partial t} = \frac{\partial R_L}{\partial R_M}\frac{\partial R_M}{\partial t} + \frac{\partial R_L}{\partial V_M}\frac{\partial V_M}{\partial t}$$

$$= \frac{\partial R_L}{\partial R_M} V_M \qquad (2)$$

(The value $\frac{\partial R_L}{\partial R_M}$ will be derived below.)

We may now derive $X_C = \begin{pmatrix} RC \\ VC \end{pmatrix}$ as follows

$$R_C = R_L + RC2L$$

$$V_C = V_L + \Omega X R_C$$

<u>PX2M</u> is the gradient of X_C with respect to X_M and is partitioned as

$$\begin{array}{c|c} & 3 \quad\quad 3 \\ \begin{array}{c} 3 \\ 3 \end{array} & \left(\begin{array}{c|c} \dfrac{\partial R_C}{\partial R_M} & \dfrac{\partial R_C}{\partial V_M} \\ \hline \dfrac{\partial V_C}{\partial R_M} & \dfrac{\partial V_C}{\partial V_M} \end{array}\right) \end{array}$$

The four 3x3 submatrices are derived as follows:

$$R_C = R_L + RC2L$$

whence

$$\frac{\partial R_C}{\partial R_M} = \frac{\partial R_L}{\partial R_M}$$

$$\frac{\partial R_C}{\partial V_M} = 0 \text{ , since } R_L \text{ is independent of } V_M$$

$$\frac{\partial V_C}{\partial V_M} = \frac{\partial V_L}{\partial V_M} + \Omega X \frac{\partial R_C}{\partial V_M} = \frac{\partial R_L}{\partial R_M}$$

$$\frac{\partial V_C}{\partial R_M} = \frac{\partial V_L}{\partial R_M} + \Omega X \frac{\partial R_C}{\partial R_M} = \frac{\partial V_L}{\partial R_M} + \Omega X \frac{\partial R_L}{\partial R_M}$$

All that remains is to derive $\frac{\partial R_L}{\partial R_M}$ and $\frac{\partial V_L}{\partial R_M}$

From Equation (1)

$$\frac{\partial R_L}{\partial \rho} = T_2 T_1 e_0 = \frac{R_L}{\rho}$$

$$\frac{\partial R_L}{\partial \alpha_1} = \frac{\partial R_L}{\partial T_2} \frac{\partial T_2}{\partial \alpha_1} = \rho(\beta_2 \times T_2 T_1 e_0) = \beta_2 \times R_L$$

$$\frac{\partial R_L}{\partial \alpha_2} = \frac{\partial R_L}{\partial T_1} \frac{\partial T_1}{\partial \alpha_2} = \rho(T_2(\beta_1 \times T_1 e_0))$$

$$= \hat{\beta}_1 \times (\rho T_2 T_1 e_0) = \hat{\beta}_1 \times R_L$$

where

$$\hat{\beta}_1 = T_2 \beta_1$$

whence

$$\left(\frac{\partial R_L}{\partial R_M} = \frac{R_L}{\rho}, \beta_2 \times R_L, \hat{\beta}_1 \times R_L\right)$$

From Equation (2)

$$\frac{\partial V_L}{\partial R_M} = \frac{\partial^2 R_L}{\partial R_M^2} V_M$$

More explicitly

$$\frac{\partial V_L}{\partial \rho} = \dot{\alpha}_1 \beta_2 \times \frac{\partial R_L}{\partial \rho} + \dot{\alpha}_2 \beta_1 \times \frac{\partial R_L}{\partial \rho}$$

$$= \left(\frac{\dot{\rho}}{\rho} I + \dot{\alpha}_2 \ (\hat{\beta}_1) \times + \dot{\alpha}_1 \ (\beta_2) \ \times \right) \frac{\partial R_L}{\partial \rho} - \frac{\dot{\rho}}{\rho^2} R_L$$

$$= A \frac{\partial R_L}{\partial \rho}$$

where

$$A = \frac{\dot{\rho}}{\rho} I + (\dot{\alpha}_2 \hat{\beta}_1 + \dot{\alpha}_1 \beta_2) \times$$

$$\frac{\partial V_L}{\partial \alpha_1} = A \frac{\partial R_L}{\partial \alpha_1} + \dot{\alpha}_1 \frac{\partial \hat{\beta}_1}{\partial \alpha_1} \times R_L$$

$$= A \frac{\partial R_L}{\partial \alpha_1} + \dot{\alpha}_1 (\beta_2 \times \hat{\beta}_1) \times R_L$$

$$\frac{\partial V_L}{\partial \alpha_2} = A \frac{\partial R_L}{\partial \alpha_2}$$

whence

$$\frac{\partial V_L}{\partial R_M} = A \frac{\partial R_L}{\partial R_M} + \left(- \frac{\rho}{\rho^2} R_L, \ \dot{\alpha}_1 (\beta_2 \times \hat{\beta}_1) \times R_L \ , \ 0 \right)$$

```
$IBFTC MC13X5  XR3,M94,LIST,NODD
      SUBROUTINE MEAS2X (XC,PX2M,XM,NROT,TL2C,RC2L,OMEGAC,OMEGA)
C   O  XC     = CARTESIAN STATE, C-FRAME
C   O  PX2M   = PARTIAL OF X W/R XM
C   I  XM     = STATE IN MEASUREMENT COORDINATES
C               XM(1) = RANGE              XM(4) = RANGE RATE
C               XM(2) = SECOND ROTATION XM(5) = SECOND ROTATION RATE
C               XM(3) = FIRST ROTATION  XM(6) = FIRST ROTATION RATE
C   I  NROT   = SLANT RANGE VECTOR IS TL2C(1,NROT(1)) ROTATED ABOUT
C   I  TL2C           TL2C(1,NROT(2)) BY XM(3) RADIANS AND THEN ABOUT
C                     TL2C(1,NROT(3)) BY XM(2) RADIANS.
C               NEGATIVE VALUES INDICATE NEGATIVE ROTATION
C   I  RC2L   = L-FRAME ORIGIN IN C-FRAME COORDINATES
C   I  OMEGAC= ROTATION VECTOR OF L-FRAME W/R C-FRAME IN C-FRAME COORD
      DOUBLE PRECISION   XC(6) ,PX2M(6,6) ,RC2L(3) ,OMEGAC(3)
     1                   ,XM(6) ,TL2C(3,3)          ,OMEGA
      DIMENSION          NROT(3)
      COMMON             SAVE(66) ,R(3,4) ,W(3,4) ,V(3,4)
      DOUBLE PRECISION   W,R,V
      DOUBLE PRECISION   TP
      DIMENSION          NANG(2)
      DATA  NANG/3,2/
      DO  8 K=1,3
      N = NROT(K)
      IF( N.LT.0 ) GO TO 4
      DO 2 I=1,3
    2 W(I,K) = TL2C(I,N)
      GO TO 8
    4 N =-N
      DO 6 I=1,3
    6 W(I,K) = -TL2C(I,N)
    8 CONTINUE
      DO 10 K=1,2
      NA1 = NANG(K)
      CALL DGTRN( V,0,XM(NA1),W(1,K+1) )
      DO 10 J=1,K
      CALL DMVTRN( V,W(1,J),R,2,1 )
      DO 10 I=1,3
      W(I,J) = R(I,1)
   10 CONTINUE
      DO 12 I=1,3
   12 R(I,4) = XM(1)*W(I,1)
      DO 14 J=1,2
      NA1 = NANG(J)
   14 CALL DCROSS( W(1,J+1),R(1,4),R(1,NA1) )
      DO 16 I=1,3
   16 R(I,1) = W(I,1)
      CALL DCROSS( W(1,3),W(1,2),W(1,1) )
      DO 18 I=1,3
   18 W(I,2) = XM(6)*W(I,2)+XM(5)*W(I,3)
      TP = XM(4)/XM(1)
      DO 20 K=1,3
   20 CALL DCROSS( W(1,2),R(1,K),V(1,K) )
      DO 22 I=4,9
   22 V(I,1) = V(I,1)+TP*R(I,1)
      CALL DCROSS( W,R(1,4),W(1,2) )
      DO 26 I=1,3
      V(I,2) = V(I,2)+XM(6)*W(I,2)
      V(I,4) = OMEGA*OMEGAC(I)
   26 R(I,4) = R(I,4)+RC2L(I)
      DO 30 J=1,4
   30 CALL DCROSS( V(1,4),R(1,J),W(1,J) )
      DO 32 J=1,3
      DO 32 I=1,3
      PX2M(I,J) = R(I,J)
      PX2M(I,J+3) = 0.D0
      PX2M(I+3,J) = V(I,J)+W(I,J)
      PX2M(I+3,J+3) = R(I,J)
   32 CONTINUE
      CALL DMVTRN( R(1,1),XM(4),V,1,1 )
      DO 34 I=1,3
      XC(I) = R(I,4)
   34 XC(I+3) = V(I,1)+W(I,4)
      RETURN
      END
```

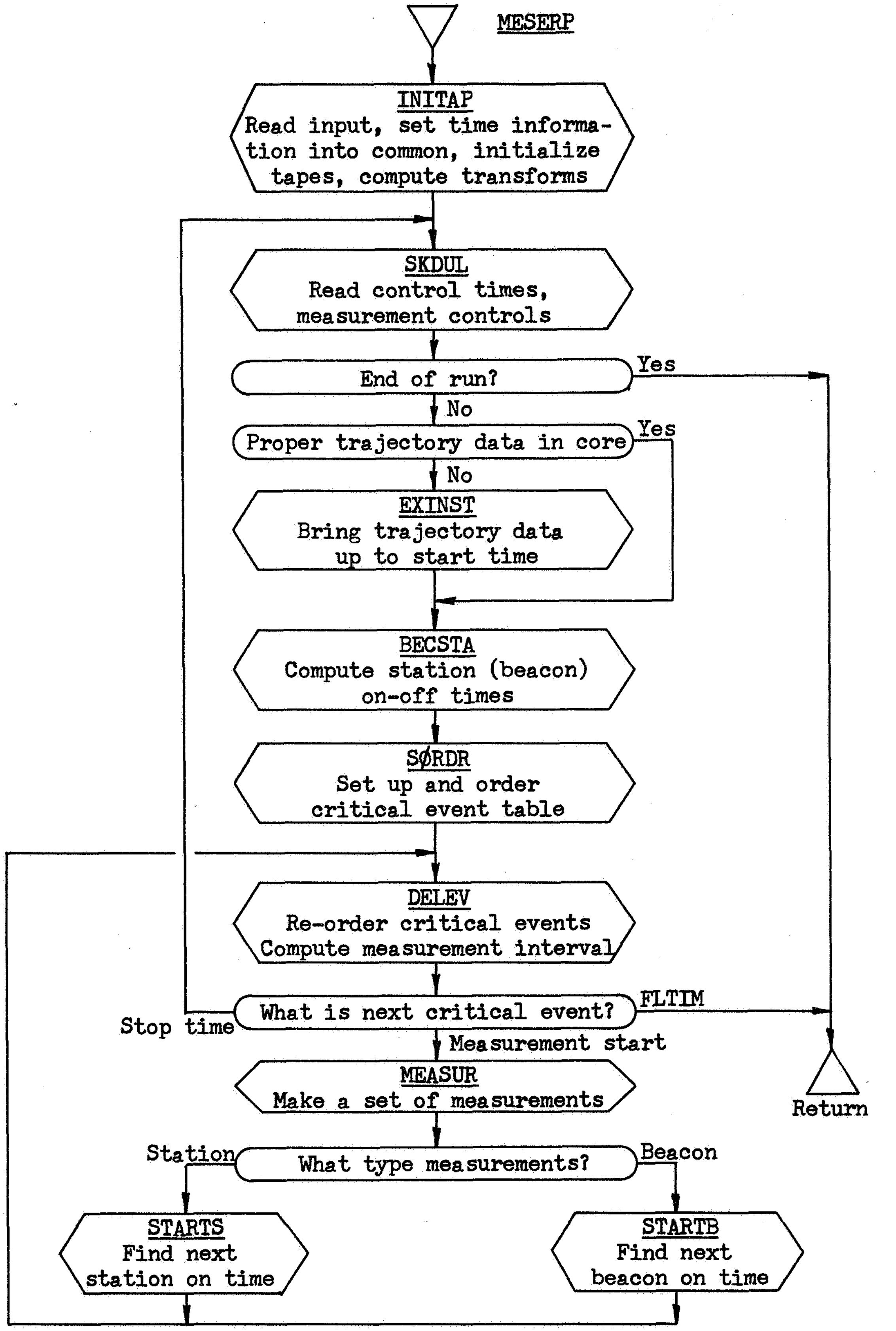
Subroutine: MESERP

Purpose: To control writing of a simulated measurement data tape. MESERP calls subroutines that read the control times and other data, manipulate the trajectory tape, compute station (beacon, and onboard) measurements, and write the measurement data tape.

Calling Sequence: CALL MESERP

Common storages used: /BUFCØM/,/INPCØM/,/WCØM/,/EXIC/

Subroutines required: BECSTA, DELEV, ERRØUT, EXINST, INITAP, MEASUR, SKDUL, SØRDR, STARTB, STARTS, TFRAC


MESERP
INITAP
Read input, set time information into common, initialize tapes, compute transforms
SKDUL
Read control times, measurement controls
End of run?
Yes
No
Proper trajectory data in core
Yes
No
EXINST
Bring trajectory data up to start time
BECSTA
Compute station (beacon) on-off times
SØRDR
Set up and order critical event table
DELEV
Re-order critical events
Compute measurement interval
What is next critical event?
Stop time
FLTIM
Measurement start
MEASUR
Make a set of measurements
Return
Station
What type measurements?
Beacon
STARTS
Find next station on time
STARTB
Find next beacon on time

MESERP-2

```
$IBFTC MC13ES  NOREF,M94,NODD,XR3
CMC13ES  MESERP    READS SCHEDULE,COMPUTES + WRITES MEASUREMENTS ACCORD.MSRP0001
      SUBROUTINE MESERP                                                 MSRP0002
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)/EXIC/W(36),CRX(9,2)    MSRP0003
      COMMON  /BUFCOM/BUFF1(40) ,KEY                                    MSRP0004
C  DIMENSION AND EQUIVALENCE INCLUDE ALL COMMON VARIABLES USED IN ALL   MSRP0005
C     PARTS OF THE MESERP PROGRAM                                       MSRP0006
      DIMENSION      BODC(10,8),    UM(10),    S(23,12),      B(91)     MSRP0007
     1,              OB(18),        EMP(24),   HEAD(24)                 MSRP0008
     2,              NSTX(3),       NOX(3),    ISC(12),       IBC(10)   MSRP0009
     3,              IEMP(24),      IOBR(4),   IOBA(18),      IS(11,12) MSRP0010
     4,              IB(3,12),      ISEE(12),  IBEE(10),      ICAS(3)   MSRP0011
     5,              ISTIM(50),     IBTIM(50)                           MSRP0012
     6,              RX(3,3),       VX(3,3),   TX(2,3),       BIASL(3,22) MSRP0013
     7,              SSTART(75),    SSTOP(75),BSTART(75),     BSTOP(75) MSRP0014
     8,              XMES(5, 85),   XIN(6),    REFS(3),       TI(3)     MSRP0015
      DIMENSION      AN(3,3),       EVNT(6),   TWT(2),        STIME(50) MSRP0016
     1,              BTIME(50),     SECR(12),  SHFC(18),      BECR(10)  MSRP0017
     2,              KRX(9,2)                                           MSRP0018
      DIMENSION      BRAD(3),       KSTA(75),  KBEA(75),      KEV(6)    MSRP0019
      EQUIVALENCE  (C(4),RTD),     (C(5),DTR),     (C(6),SPMSD)         MSRP0020
     1,            (C(7),RSPMSD), (C(9),SPDLT),   (C(11),BODC)          MSRP0021
     2,            (C(21),UM),    (C(112),XMIN), (C(113),XMAX)          MSRP0022
     3,            (C(114),RDELAY), (C(200),S),   (C(476),B)            MSRP0023
     4,            (C(556),DELBEC), (C(567),OB), (C(573),DELRAD)        MSRP0024
     5,            (C(580),DELOPT), (C(585),EMP),(C(608),EMP24)         MSRP0025
     6,            (C(673),IKAS),   (C(675),HEAD),(C(699),ITAPE)        MSRP0026
     7,            (C(700),IOCAS),(C(115),XTROUT)                       MSRP0027
     8,            (C(116),CFRAC1),  (C(117),CFRAC2)                    MSRP0028
      EQUIVALENCE  (IW(1),NSTX),   (IW(4),NOX),    (IW(7),IXT)          MSRP0029
     1,            (IW(8),ISC),    (IW(20),IBC),   (IW(30),IEMP)        MSRP0030
     2,            (IW(53),IEMP24), (IW(54),IOBR), (IW(58),IOBA)        MSRP0031
     3,            (IW(180),IS),     (IW(312),IB),  (IW(347),NBEAC)     MSRP0032
     4,            (IW(355),ISEE),   (IW(367),IBEE),(IW(377),ICAS)      MSRP0033
     5,            (IW(380),NMAX),   (IW(382),NOR), (IW(383),ITARG)     MSRP0034
     6,            (IW(384),IK1),    (IW(385),KSMAX),  (IW(386),IK2)    MSRP0035
     7,            (IW(387),KBMAX), (IW(389),ISTIM),  (IW(439),IBTIM)   MSRP0036
     8,            (IW(489),ITRIG),  (IW(490),KOUNT)                    MSRP0037
      EQUIVALENCE  (CW(1),RX),       (CW(10),VX),      (CW(19),TX)      MSRP0038
     1,            (CW(156),BIASL), (CW(222),SSTART), (CW(297),SSTOP)   MSRP0039
     2,            (CW(372),BSTART),(CW(447),BSTOP),  (CW(522),XMES)    MSRP0040
     3,            (CW(1134),XIN),   (CW(1140),TSEC),  (CW(1157),REFS)  MSRP0041
     4,            (CW(1160),TI),    (CW(1163),AN),    (CW(1172),TSECO) MSRP0042
     5,            (CW(1173),GHAR), (CW(1175),OBTIME),(CW(1178),TSTART) MSRP0043
     6,            (CW(1179),FLTIM),(CW(1180),EVNT),  (CW(1188),TWT)    MSRP0044
     7,            (CW(1200),STIME),(CW(1250),BTIME), (CW(1360),SECR)   MSRP0045
     8,            (CW(1372),SHFC), (CW(1426),TWO),   (CW(1427),TFO)    MSRP0046
     9,            (CW(1441),BECR), (CRX,KRX)                           MSRP0047
      DOUBLE PRECISION  DBUFF(10)                                       MSRP0048
      DOUBLE PRECISION  ONTM,      TFST,      TLST,      TAU,      FTR  MSRP0049
     1,                 C1,        C2,        RAMB,      BIAS,     OMG4 MSRP0050
      EQUIVALENCE  (BUFF1(1),IKY),    (BUFF1(2),KEE),    (BUFF1(3),STANAM)MSRP0051
     1,            (BUFF1(4),STNAM1),(BUFF1(5),NRCD),  (BUFF1(6),NKOUNT)MSRP0052
     2,            (BUFF1(7),KONTIN),(BUFF1(8),MTYPE), (BUFF1(9),NALIGN)MSRP0053
     3,            (BUFF1(10),MODE), (BUFF1(11),FMS),  (BUFF1(12),KTAU) MSRP0054
     4,            (BUFF1(13),DBUFF,ONTM)                               MSRP0055
      EQUIVALENCE  (DBUFF(2),TFST),   (DBUFF(3),TLST),  (DBUFF(4),TAU)  MSRP0056
     1,            (DBUFF(5),FTR),    (DBUFF(6),C1),    (DBUFF(7),C2)   MSRP0057
     2,            (DBUFF(8),RAMB),   (DBUFF(9),BIAS),  (DBUFF(10),OMG4) MSRP0058
      DOUBLE PRECISION TWR                                              MSRP0059
      DIMENSION   TYPE(4)                                               MSRP0060
      DATA TYPE/6HC-BAND,6HGODDRD,6HS-BAND,6H DSIF /                    MSRP0061
      DATA QYES,QNO/3HYES,2HNO/                                         MSRP0062
      DATA IQUAL/0/                                                     MSRP0063
C     ASSUMES TRAJECTORY ON BINARY TAPE 10                              MSRP0064
C SUBROUTINES CALLED HERIN                                              MSRP0065
C MC13      NAME                                                        MSRP0066
C  BA      BECSTA                                                       MSRP0067
C  DL      DELEV                                                        MSRP0068
C  ER      ERROUT                                                       MSRP0069
C  JY      EXINST                                                       MSRP0070
C  IT      INITAP                                                       MSRP0071
C  UR      MEASUR                                                       MSRP0072
C  UL      SKDUL                                                        MSRP0073
C  SO      SORDR                                                        MSRP0074
```

```
C  AB      STARTB                                                       MSRP0075
C  AS      STARTS                                                       MSRP0076
C  TM      TFRAC                                                        MSRP0077
C OTHER SUBROUTINES REQUIRED BY THE MESERP PROGRAM                       MSRP0078
C MC13      NAME                                                        MSRP0079
C  FM      ANTR1                                                        MSRP0080
C  AT      ARKTNS                                                       MSRP0081
C  J3      BACK                                                         MSRP0082
C  B4      BARN                                                         MSRP0083
C  I6      BIBCD                                                        MSRP0084
C  IP      /INPCOM/B.D.                                                 MSRP0085
C  MB      BMEAS                                                        MSRP0086
C  AR      CARDIN                                                       MSRP0087
C  CA      CRITA                                                        MSRP0088
C  CC      CRITIC                                                       MSRP0089
C  CR      CRITO                                                        MSRP0090
C  MC      CROSS                                                        MSRP0091
C  MF      DOT,FNORM,VNORM                                              MSRP0092
C  98      FIEF                                                         MSRP0093
C  97      FIFL                                                         MSRP0094
C  96      FIST                                                         MSRP0095
C  GH      GHA                                                          MSRP0096
C  GO      GOTOR                                                        MSRP0097
C  LO      LAYO                                                         MSRP0098
C  LC      LOCAT                                                        MSRP0099
C  MA      MNA                                                          MSRP0100
C  MT      MTRN,VTRN,VTRT                                               MSRP0101
C  NU      NUTAIT                                                       MSRP0102
C  OR      ORB                                                          MSRP0103
C  OY      OYAL                                                         MSRP0104
C  PB      PARAB                                                        MSRP0105
C  QC      QUARTC                                                       MSRP0106
C  50      ROTEQ                                                        MSRP0107
C  9Q      ROVLEY                                                       MSRP0108
C  JW      SBEV2                                                        MSRP0109
C  S2      SHIF2                                                        MSRP0110
C  KD      SKDOUT                                                       MSRP0111
C  SM      SMEAS                                                        MSRP0112
C  O2      SORDR2                                                       MSRP0113
C  SA      STAT                                                         MSRP0114
C  SI      STEPI                                                        MSRP0115
C  SG      STEPT                                                        MSRP0116
C  TC      TCONIC                                                       MSRP0117
C  TM      TIME PACKAGE   DATOUT,TFRAC,TIMED,TIMES                      MSRP0118
      NMAX=85                                                           MSRP0119
      BRAD(1)= BODC(1,3)                                                MSRP0120
      BRAD(2)= BODC(1,4)                                                MSRP0121
      BRAD(3)= BODC(1,7)                                                MSRP0122
      IKY=0                                                             MSRP0123
      INITS=-1                                                          MSRP0124
      CALL INITAP                                                       MSRP0125
 20   CONTINUE                                                          MSRP0126
      CALL SKDUL(INITS,NCH)                                             MSRP0127
      IF(NCH.EQ.111) GO TO 250                                          MSRP0128
      IF(TSEC.EQ.TSTART) GO TO 30                                       MSRP0129
      TWR=TSTART                                                        MSRP0130
      CALL TFRAC(TI(1),TI(2)+(TSTART-TSEC)*RSPMSD,TI(1),TI(2))          MSRP0131
      M=1                                                               MSRP0132
      N=3                                                               MSRP0133
 105  CONTINUE                                                          MSRP0134
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)              MSRP0135
      IF(M.LE.0) CALL ERROUT(1,10H(4HOM1=I2),M)                         MSRP0136
      IF(N) 115,110,120                                                 MSRP0137
 110  CONTINUE                                                          MSRP0138
      CALL ERROUT(1,28H(23HOERROR ON BIN TAPE READ))                    MSRP0139
 115  CONTINUE                                                          MSRP0140
      N=-N                                                              MSRP0141
      NOR=KRX(7,N)                                                      MSRP0142
      GO TO 105                                                         MSRP0143
 120  CONTINUE                                                          MSRP0144
      TSEC=TSTART                                                       MSRP0145
 30   CONTINUE                                                          MSRP0146
      KMAXS=0                                                           MSRP0147
      KMAXB=0                                                           MSRP0148
      CALL BECSTA(KSTA,KBEA,KMAXS,KMAXB)                                MSRP0149
```

```
      EVNT(1) = SSTART
      EVNT(2) = BSTART
      EVNT(3) = FLTIM + 9999.
      EVNT(4) = EVNT(3)
      EVNT(5) = TSTART
      EVNT(6) = FLTIM
C EVNT(3)  ONBOARD OPTICAL  TEMPORARILY OUT
C EVNT(4)  ONBOARD RADAR    TEMPORARILY OUT
      DO 1 I=1,6
1     KEV(I)=I
      CALL SORDR(EVNT,KEV,6)
43    CONTINUE
      CALL DELEV(EVNT,KEV,DEL,KSTOP,KSTA(IK1))
      IF(KSTOP-5) 50,20,250
50    CONTINUE
      TSTP=EVNT(1)
      GO TO (60,70,80,90),KSTOP
60    CONTINUE
      TSTOP=SSTOP(IK1)
      N=KSTA(IK1)
      L=ISC(N)
      GO TO 100
70    CONTINUE
      TSTOP=BSTOP(IK2)
      N=KBEA(IK2)
      L=IBC(N)
      GO TO 100
80    CONTINUE
C SET TSTOP,L, FOR MEASUR CALL FOR  ONBOARD OPTICAL MEASUREMENTS
      GO TO 100
90    CONTINUE
C SET TSTOP,L  FOR MEASUR CALL FOR  ONBOARD RADAR   MEASUREMENTS
100   CONTINUE
      CALL MEASUR(TSTP,TSTOP,KSTOP,DEL,L,BRAD,KMAXS,KMAXB)
      IF(KOUNT.LE.0) GO TO 150
      DO 101 I=1,KOUNT
  101 BACKSPACE 10
      M=0
      N=18
      TWR=TSEC
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
  150 ICAS(3)=KSTOP
      ITRIG=0
      KOUNT=0
      GO TO (160,170,180,190),KSTOP
160   CONTINUE
      CALL STARTS(KMAXS,KSTA)
      GO TO 43
170   CONTINUE
      CALL STARTB(KMAXB,KBEA)
      GO TO 43
180   CONTINUE
C  SOMETHING FOR ONBOARD OPTICAL IF NECESSARY
      GO TO 43
190   CONTINUE
C SOMETHING FOR ONBOARD RADAR  IS NECESSARY
      GO TO 43
250   CONTINUE
      WRITE(12) BUFF1
      WRITE(12)(IQUAL,I=1,85),((XMES(I,J),J=1,85),I=1,5)
C***  CHECK BINARY OUTPUT
      QOUT=QNO
      IF(KONTIN.EQ.1) QOUT=QYES
      WRITE(6,601) IKY,STANAM,STNAM1,TYPE(MTYPE),QOUT,TFST
  601 FORMAT(//24H ***  RECORD PAIR NUMBER,I5//75H RECEIVING STATION   T
     1RANSMITTING STATION   MEASUREMENT TYPE   CONTINUATION/7X,A6,15X,A6
     2,15X,A6,12X,A3//12H DATA BEGINS,D24.16,20H SECONDS FROM 1950.0)
      TEMP=(TFST -.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,602) TLST
  602 FORMAT(12H0 DATA ENDS ,D24.16,20H SECONDS FROM 1950.0)
      TEMP=(TLST -.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
```

```
      WRITE(6,606) NKOUNT                                               MSRP0225
  606 FORMAT(12H0TIME POINTS/I7)                                        MSRP0226
      IF(XTROUT.EQ.0.) GO TO 61                                         MSRP0227
      WRITE(6,607) KEE,NRCD,NALIGN,MODE,FMS,KTAU,ONTM, (DBUFF(I),I=4,10)MSRP0228
     1,(IQUAL,(XMES(I,J),I=1,5),J=1,NKOUNT)                             MSRP0229
  607 FORMAT(5H0NEOT,3X,4HNRCD,3X,6HNALIGN,3X,4HMODE,7X,8HMEAS INT,6X   MSRP0230
     1,4HKTAU,11X,6HONTIME,22X,3HTAU/I4,I7,2I8,E19.8,I6,D27.16,D26.16// MSRP0231
     213X,3HFTR,23X,2HC1,24X,2HC2,24X,2HDR/4D26.16//12X,4HBIAS,21X,5HRATMSRP0232
     3IO/2D26.16//3H IQ,11X,4HTIME,15X,5HANG 1,15X,5HANG 2,15X,5HRANGE, MSRP0233
     414X,7HDOPPLER/(I3,5E20.8))                                        MSRP0234
C***  END OF BINARY OUTPUT                                              MSRP0235
   61 KEE=1                                                             MSRP0236
      IKY=IKY+1                                                         MSRP0237
      NRCD=NRCD+1                                                       MSRP0238
      WRITE(12) BUFF1                                                   MSRP0239
C***  CHECK BINARY OUTPUT                                                MSRP0240
      WRITE(6,703)                                                      MSRP0241
  703 FORMAT(//42X,35H *** END OF SIMULATED DATA TAPE ***)              MSRP0242
C*** END BINARY OUTPUT                                                  MSRP0243
      ENDFILE 12                                                        MSRP0244
      ICAS(3)=6                                                         MSRP0245
      RETURN                                                            MSRP0246
      END                                                               MSRP0247
```

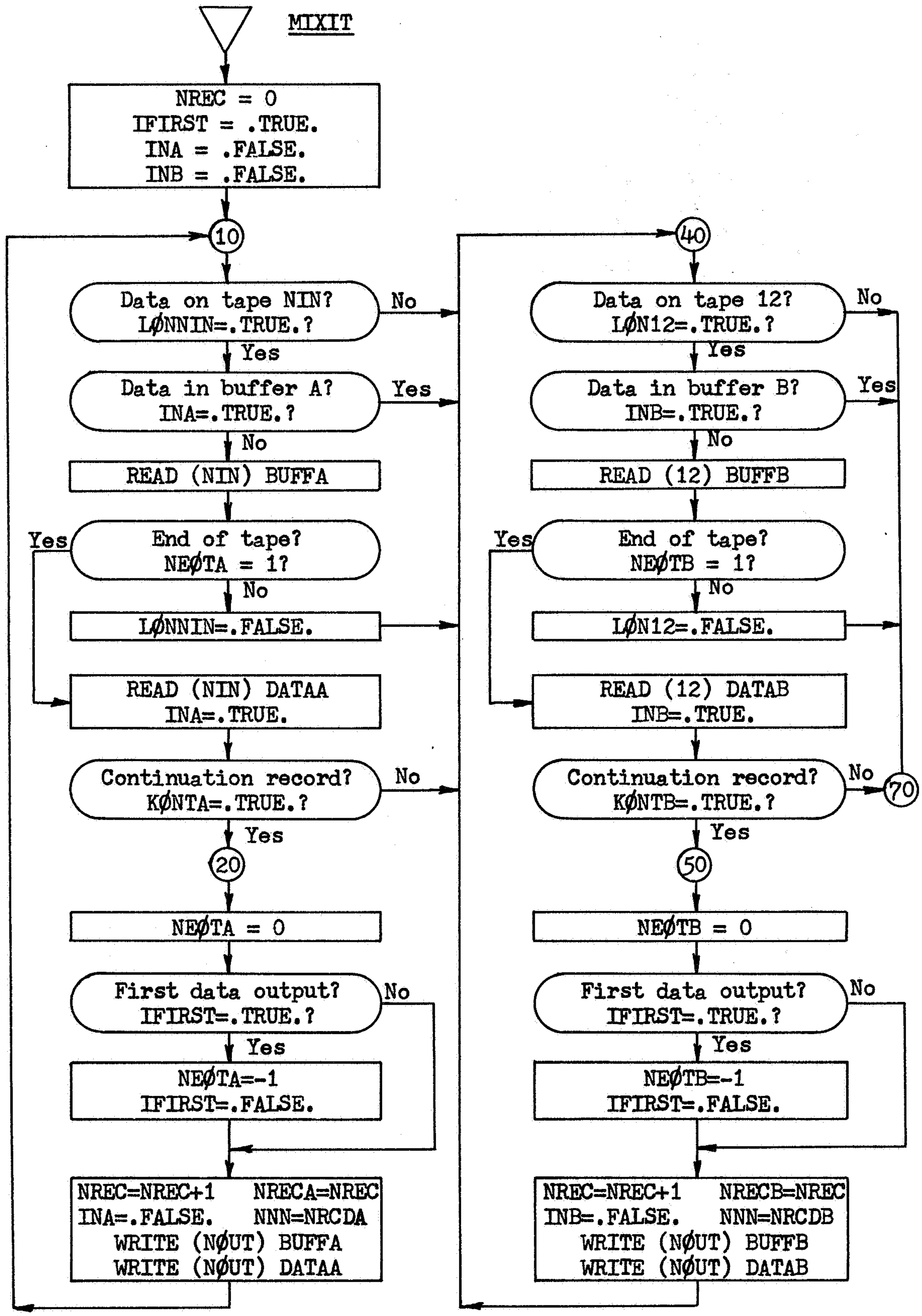
Subroutine: MIXIT

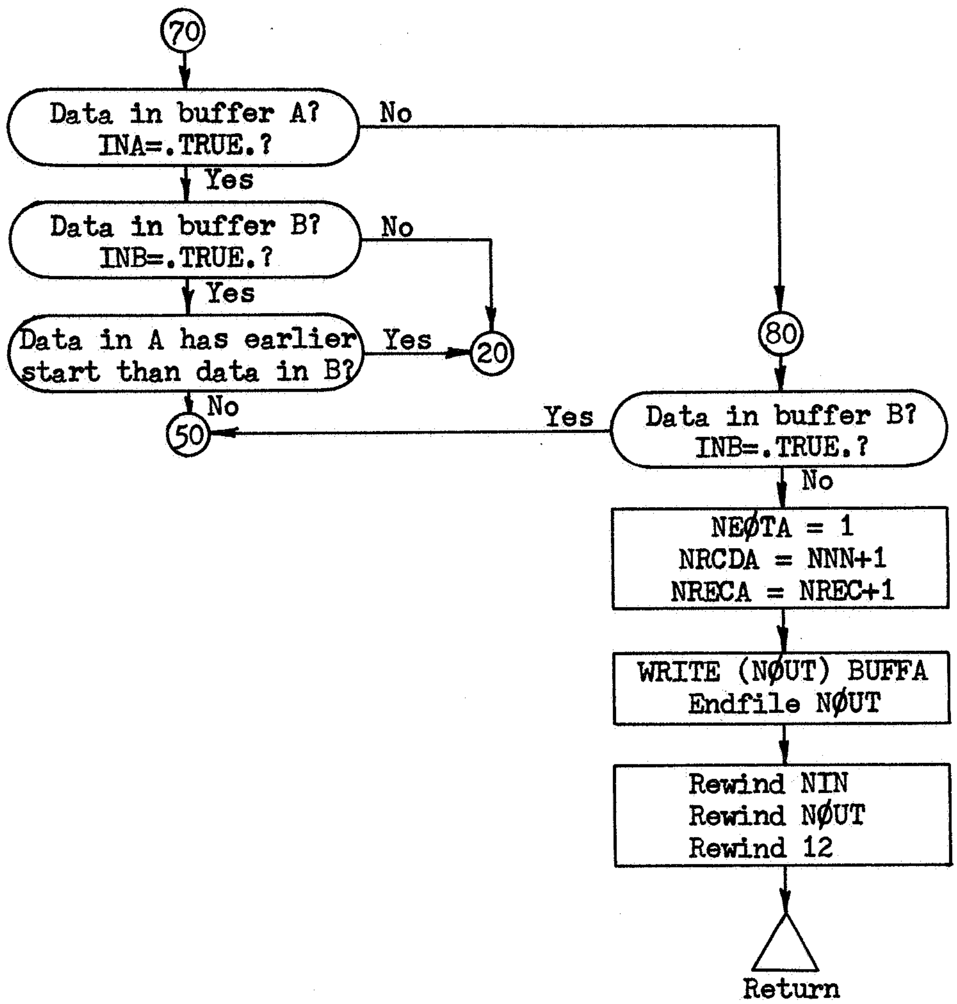
Purpose: Reads edited data from tape 12 and previously edited data from tape NIN, arranges data in order of data arc start times, and writes merged data on tape NØUT.

Calling Sequence: CALL MIXIT

Common storages used: /MIXCØM/,/ØUTCØM/,/MESCØM/

Subroutines requires: None.


MIXIT
NREC = 0
IFIRST = .TRUE.
INA = .FALSE.
INB = .FALSE.
10
Data on tape NIN?
LØNNIN=.TRUE.?
No
Yes
Data in buffer A?
INA=.TRUE.?
Yes
No
READ (NIN) BUFFA
End of tape?
NEØTA = 1?
Yes
No
LØNNIN=.FALSE.
READ (NIN) DATAA
INA=.TRUE.
Continuation record?
KØNTA=.TRUE.?
No
Yes
20
NEØTA = 0
First data output?
IFIRST=.TRUE.?
No
Yes
NEØTA=-1
IFIRST=.FALSE.
NREC=NREC+1 NRECA=NREC
INA=.FALSE. NNN=NRCDA
WRITE (NØUT) BUFFA
WRITE (NØUT) DATAA
40
Data on tape 12?
LØN12=.TRUE.?
No
Yes
Data in buffer B?
INB=.TRUE.?
Yes
No
READ (12) BUFFB
End of tape?
NEØTB = 1?
Yes
No
LØN12=.FALSE.
READ (12) DATAB
INB=.TRUE.
Continuation record?
KØNTB=.TRUE.?
No
70
Yes
50
NEØTB = 0
First data output?
IFIRST=.TRUE.?
No
Yes
NEØTB=-1
IFIRST=.FALSE.
NREC=NREC+1 NRECB=NREC
INB=.FALSE. NNN=NRCDB
WRITE (NØUT) BUFFB
WRITE (NØUT) DATAB


70
Data in buffer A?
INA=.TRUE.?
No
Yes
Data in buffer B?
INB=.TRUE.?
No
Yes
Data in A has earlier start than data in B?
Yes
20
No
50
80
Yes
Data in buffer B?
INB=.TRUE.?
No
NEØTA = 1
NRCDA = NNN+1
NRECA = NREC+1
WRITE (NØUT) BUFFA
Endfile NØUT
Rewind NIN
Rewind NØUT
Rewind 12
Return

```
$IBFTC MC134T  M94,NODD,XR3
CMC134T   MERGES EDITED DATA AND PUTS IT ON OUTPUT TAPE NOUT
      SUBROUTINE MIXIT
      COMMON  /MIXCOM/IMIX(4)
      LOGICAL LONNIN,   LON12
      EQUIVALENCE             (IMIX(1),LONNIN),    (IMIX(2),LON12)
     1,  (IMIX(3),NIN),       (IMIX(4),NOUT)
C
      COMMON  /OUTCOM/BUFFA(40)
      DOUBLE PRECISION  ASTA
      LOGICAL KONTA
      EQUIVALENCE             (BUFFA(1),NRECA),    (BUFFA(2),NEOTA)
     1,  (BUFFA(5),NRCDA),    (BUFFA(7),KONTA),    (BUFFA(13),ASTA)
C
      COMMON  /MESCOM/DATAA(510)
C
      DIMENSION   BUFFB(40),    DATAB(510)
      DOUBLE PRECISION  ASTB
      LOGICAL KONTB
      EQUIVALENCE             (BUFFB(1),NRECB),    (BUFFB(2),NEOTB)
     1,  (BUFFB(5),NRCDB),    (BUFFB(7),KONTB),    (BUFFB(13),ASTB)
C
      LOGICAL INA,      INB,      IFIRST
C
C     INITIALIZE
      NREC=0
      IFIRST=.TRUE.
      INA=.FALSE.
      INB=.FALSE.
   10 IF(.NOT.LONNIN.OR.INA) GO TO 40
C     READ A KEY RECORD FROM TAPE NIN
      READ(NIN) BUFFA
C     TEST FOR END OF TAPE RECORD
      IF(NEOTA.EQ.1) GO TO 30
C     READ A DATA RECORD FROM TAPE NIN
      READ(NIN) DATAA
      INA=.TRUE.
      IF(.NOT.KONTA) GO TO 40
C     IF CONTINUATION RECORD PAIR, WRITE IT ON NOUT IMMEDIATELY
   20 NEOTA=0
      IF(.NOT.IFIRST) GO TO 25
C     IF FIRST OUTPUT RECORD, SET NEOTA=-1
      NEOTA=-1
      IFIRST=.FALSE.
C     INCREMENT RECORD PAIR OUTPUT COUNT
   25 NREC=NREC+1
      NRECA=NREC
      NNN=NRCDA
C     WRITE BOTH KEY AND DATA RECORDS ON NOUT
      WRITE(NOUT) BUFFA
      WRITE(NOUT) DATAA
      INA=.FALSE.
      GO TO 10
   30 LONNIN=.FALSE.
   40 IF(.NOT.LON12.OR.INB) GO TO 70
C     READ A KEY RECORD FROM TAPE 12
      READ(12) BUFFB
C     TEST FOR END OF TAPE RECORD
      IF(NEOTB.EQ.1) GO TO 60
C     READ A DATA RECORD FROM TAPE 12
      READ(12) DATAB
      INB=.TRUE.
      IF(.NOT.KONTB) GO TO 70
C     IF CONTINUATION RECORD PAIR, WRITE IT ON NOUT IMMEDIATELY
   50 NEOTB=0
      IF(.NOT.IFIRST) GO TO 55
C     IF FIRST OUTPUT RECORD, SET NEOTB=-1
      NEOTB=-1
      IFIRST=.FALSE.
C     INCREMENT RECORD PAIR OUTPUT COUNT
   55 NREC=NREC+1
      NRECB=NREC
      NNN=NRCDB
C     WRITE BOTH KEY AND DATA RECORDS ON NOUT
      WRITE(NOUT) BUFFB
```

```
      WRITE(NOUT) DATAB
      INB=.FALSE.
      GO TO 40
   60 LON12=.FALSE.
   70 IF(.NOT.INA) GO TO 80
      IF(.NOT.INB) GO TO 20
C     HERE WE TEST ARC START TIMES TO SEE WHICH RECORD PAIR GOES ON
C     NOUT FIRST
      IF(ASTA-ASTB) 20,20,50
   80 IF(INB) GO TO 50
C     SET NEOTA =+1 FOR END OF TAPE RECORD
      NEOTA=1
C     INCREMENT RECORD PAIR COUNT AND ARC RECORD COUNT
      NRCDA=NNN+1
      NRECA=NREC+1
C     WRITE END OF TAPE KEY RECORD
      WRITE(NOUT) BUFFA
      ENDFILE NOUT
      REWIND NIN
      REWIND NOUT
      REWIND 42
      RETURN
      END
```

Subroutine: MLESTT

Purpose: Control subroutine for the data start link of the DCP. Computes an estimate of the vehicle state by maximum likelihood estimation without a priori knowledge of the state.

Calling Sequence: CALL MLESTT

Common storages used: //253 cells,/DCPCØM/,/MLECØM/

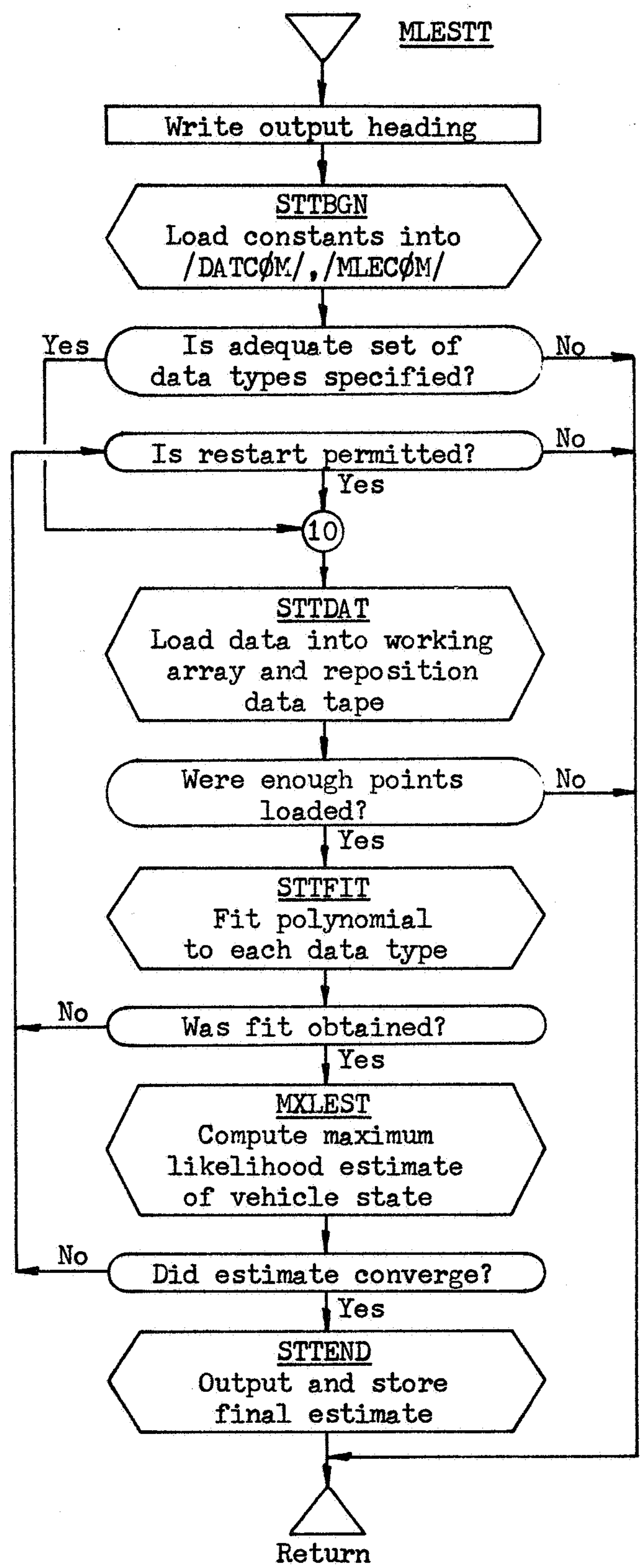
Subroutines required: MXLEST, STTBGN, STTDAT, STTEND, STTFIT

Discussion

MLESTT controls the computation of a maximum likelihood estimate of the vehicle state from a small number of data points, to be used as an *a priori* estimate for differential correction. The various steps required are controlled by the following subroutines:

STTBGN	Reads overlay data into DCPCØM. Loads constants into DATCØM and MLECØM, and initializes the data start.
STTDAT	Selects appropriate data from the edited data tape, and loads the working array DATA with the selected data.
STTFIT	Fits each type of data with a time polynomial in the least squares sense. Outputs the zeroth and first order coefficients as smoothed values of the measurement and its derivative.
MXLEST	Computes the maximum likelihood estimate of vehicle state. The subroutine uses the smoothed measurements as the starting point of an iterative estimation.
STTEND	Stores the estimate computed by MXLEST in ESTCØM. Also writes the estimate on the estimate tape.

The subroutine requires measurement of two angles and either range or doppler. The controls for the selection of data, smoothing, and convergence control are described in the individual subroutine descriptions.


MLESTT
Write output heading
STTBGN
Load constants into
/DATCØM/,/MLECØM/
Is adequate set of
data types specified?
Yes
No
Is restart permitted?
No
Yes
10
STTDAT
Load data into working
array and reposition
data tape
Were enough points
loaded?
No
Yes
STTFIT
Fit polynomial
to each data type
Was fit obtained?
No
Yes
MXLEST
Compute maximum
likelihood estimate
of vehicle state
Did estimate converge?
No
Yes
STTEND
Output and store
final estimate
Return

```
$IBFTC MC13A5  XR3,M94,NODD,LIST
.      SUBROUTINE MLESTT                                                MLES0001
C      CONTROL ROUTINE FOR MAXIMUM LIKELIHOOD DATA START                MLES0002
C                                                                       MLES0003
       COMMON              /DCPCOM/CDCP(900)                            MLES0004
       EQUIVALENCE         (CDCP(111),IERR   )                          MLES0005
       COMMON              /MLECOM/CMLE(1070)                           MLES0006
       DIMENSION            DATA(5,100)                                 MLES0007
       LOGICAL              LAUTO,LMLE                                  MLES0008
       EQUIVALENCE         (CMLE(570),DATA   ) ,(CMLE(432),NPTSTT)      MLES0009
      1                    ,(CMLE(405),LAUTO ) ,(CMLE(   1),TSTART)     MLES0010
      2                    ,(CMLE(406),LMLE  )                          MLES0011
C                                                                       MLES0012
  601 FORMAT(//48X,23H*** DATA START LINK ***/)                         MLES0013
  602 FORMAT(//46X,28H*** EXIT DATA START LINK ***)                     MLES0014
  603 FORMAT(//52H **** DID NOT SUCCEED IN OBTAINING STARTING ESTIMATE/)MLES0015
  604 FORMAT(//30H  *** INSUFFICIENT DATA LOADED)                       MLES0016
  605 FORMAT(//48H   ** POLYNOMIAL FIT FAILED FOR THIS SET OF DATA)     MLES0017
  606 FORMAT(//25H   ** TRY NEW SET OF DATA)                            MLES0018
  607 FORMAT(//74H   ** MAXIMUM-LIKELIHOOD ESTIMATOR FAILED TO CONVERGE MLES0019
     1FOR THIS SET OF DATA)                                             MLFS0020
  608 FORMAT(//41H  *** TRIAL OF NEW SET OF DATA PROHIBITED)            MLES0021
  609 FORMAT(//45H  *** INSUFFICIENT DATA TYPES TO DEFINE STATE)        MLES0022
C                                                                       MLES0023
C      INITIALIZE                                                       MLES0024
    1 CONTINUE                                                          MLES0025
      WRITE (6,601)                                                     MLES0026
      CALL STTBGN                                                       MLES0027
      IF (IERR.NE.0)      GO TO 80                                      MLES0028
C                                                                       MLES0029
C     LOAD DATA                                                         MLES0030
   10 CONTINUE                                                          MLES0031
      CALL STTDAT                                                       MLES0032
      IF (IERR.NE.0)      GO TO 81                                      MLES0033
C                                                                       MLES0034
C     FIT POLYNOMIALS TO DATA                                           MLES0035
   20 CONTINUE                                                          MLES0036
      CALL STTFIT                                                       MLES0037
      IF (IERR.NE.0)      GO TO 82                                      MLES0038
C                                                                       MLES0039
C     COMPUTE MAXIMUM LIKELIHOOD ESTIMATE                               MLES0040
   30 CONTINUE                                                          MLES0041
      CALL MXLEST                                                       MLES0042
      IF (IERR.NE.0)      GO TO 999                                     MLES0043
      IF (.NOT.LMLE)      GO TO 83                                      MLES0044
C                                                                       MLES0045
C     OUTPUT AND STORE FINAL ESTIMATE                                   MLES0046
   40 CONTINUE                                                          MLES0047
      CALL STTEND                                                       MLES0048
      GO TO 999                                                         MLES0049
C                                                                       MLES0050
C     WRITE ERROR MESSAGES                                              MLES0051
   80 WRITE (6,609)                                                     MLES0052
      GO TO 88                                                          MLES0053
   81 WRITE (6,604)                                                     MLES0054
      GO TO 88                                                          MLES0055
   82 WRITE (6,605)                                                     MLES0056
      GO TO 84                                                          MLES0057
   83 WRITE (6,607)                                                     MLES0058
   84 IF (.NOT.LAUTO)     GO TO 86                                      MLES0059
      WRITE (6,606)                                                     MLES0060
      TSTART = DATA(1,NPTSTT)*1.000001                                  MLES0061
      GO TO 10                                                          MLES0062
   86 WRITE (6,608)                                                     MLES0063
   88 IERR = 15                                                         MLES0064
      WRITE (6,603)                                                     MLES0065
C                                                                       MLES0066
  999 RETURN                                                            MLES0067
      END
```

Subroutine: MNA

Purpose: To provide the rotation matrix EMN which transforms Earth's true equator, equinox coordinates to moon's time equator, prime meridan coordinates.

Calling Sequence: CALL MNA(TIME,ØM,CR,DT,EPSIL,RØ,G,GP, WW,EMN)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TIME		d	days	Days since 1950 Jan 0.0
I	OM		Ω	rad	Argument of the descending node.
I	CR		Γ	rad	Mean longitude of the moon.
I	DT		$\Delta\Psi+\delta\Psi$	rad	Nutation in longitude.
I	EPSIL		$\bar{\epsilon}+\Delta\epsilon+\delta\epsilon$	rad	True obliquity.
Ø	RØ		ρ	rad	Libration in inclination.
Ø	G		g	rad	Mean anomaly of the moon.
Ø	GP		g'	rad	Mean anomaly of the sun.
Ø	WW		ω	rad	Argument of perigee of the moon.
Ø	EMN	(3,3)			Rotation matrix.

Common storages used: None.

Subroutines required: ARKTNS

Transformation From Earth's True Equator to Moon's True Equator

The two rectangular systems are related through Λ, Ω', and i by the rotation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{MOON} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix}_{EARTH}$$

where

$$b_{22} = - \sin \Lambda \sin \Omega' + \cos \Lambda \cos \Omega' \cos i$$

$$b_{23} = \cos \Lambda \sin i$$

$$b_{31} = \sin \Omega' \sin i$$

$$b_{32} = -\cos \Omega' \sin i$$

$$b_{33} = \cos i$$

$$b_{11} = \cos \Lambda \cos \Omega' - \sin \Lambda \sin \Omega' \cos i$$

$$b_{12} = \cos \Lambda \sin \Omega' + \sin \Lambda \cos \Omega' \cos i$$

$$b_{13} = \cos \Lambda \sin i$$

$$b_{21} = -\sin \Lambda \cos \Omega' - \cos \Lambda \sin \Omega' \cos i$$

i is the inclination of the moon's true equator to the earth's equator.

Ω' is the right ascension of the ascending node of the moon's true equator

Λ is the anomaly from the node to the X axis.

$$\Lambda = \Delta + (\Gamma + \tau) - (\Omega + \sigma)$$

Δ is the anomaly from the node to the ascending node of the moon's true equator on the ecliptic.

Ω is the mean longitude of the descending node of the moon's mean equator on the ecliptic.

Γ is the mean longitude of the moon.

σ is the libration in the node.

τ is the libration in the mean longitude.

ρ is the libration in the inclination.

$\delta\Psi$, ε, Ω, and Γ are input quantities obtained from NUTAIT. The remainder are computed from the following equations.

I = inclination of moon's equator to ecliptic.

$I = 1.535^\circ$

g = mean anomaly of moon.

$g = 215.54013 + 13.064992\ d$

g' = mean anomaly of sun.

$g' = 358.009067 + .9856005\ d$

ω = argument of perigee of moon.

$\omega = 196.745632 + .1643586\ d$

where d = days from 1950.

$$\sigma \sin I = -.0303777 \sin g + .0102777 \sin (g + 2\omega) - .00305555 \sin (2g + 2\omega)$$

$$\tau = -.003333 \sin g + .0163888 \sin g' + .005 \sin 2\omega$$

$$\rho = -.0297222 \cos g + .0102777 \cos (g + 2\omega) - .00305555 \cos (2g + 2\omega)$$

$$\cos i = \cos (\Omega + \sigma + \delta\Psi) \sin \varepsilon \sin (I + \rho) + \cos \varepsilon \cos (I + \rho) \qquad 0 < i < 90^\circ$$

$$\sin \Omega' = -\sin (\Omega + \sigma + \delta\Psi) \sin (I + \rho) \csc i \qquad -90^\circ < \Omega' < 90^\circ$$

$$\sin \Delta = -\sin (\Omega + \sigma + \delta\Psi) \sin \varepsilon \csc i$$

$$\cos \Delta = -\sin (\Omega + \sigma + \delta\Psi) \sin \Omega' \cos \varepsilon - \cos (\Omega + \sigma + \delta\Psi) \cos \Omega' \qquad 0^\circ \leq \Delta < 360^\circ$$

Reference: JPL Technical Report No. 32-223

```
$IBFTC MC13MA   NOREF,M94,NODD,XR3
      SUBROUTINE MNA(TIME,OM,CR,DT,EPSIL,RO,G,GP,WW,EM)
CMC13MA   MNA (IV)
      DIMENSION EM(3,3),DF(3)
      DOUBLE PRECISION DD
      DATA DTR/.017453293/
      D = TIME
      T = D/36525.
      T2 = T*T
      T3 = T2*T
      A=13.064992
      DO 6 I=1,3
      DD=D
      DD=DD*(A/360.)
      DF(I)=IDINT(DD)
      DF(I)=DD-DF(I)
      GO TO (4,5,6),I
    4 A=.9856005
      GO TO 6
    5 A=.1643586
    6 CONTINUE
      G=215.54013+360.*DF(1)
      GP=358.009067+360.*DF(2)
      WW=196.745632+360.*DF(3)
      G=G*DTR
      GP=GP*DTR
      WW=WW*DTR
      YN=1.535*DTR
      RO = -.0297222*COS (G) + .01020777*COS (G+2.*WW)
     1 -.00305555*COS (2.*G+2.*WW)
      TA = -.003333*SIN (G) + .0163888*SIN (GP)
     1 +.005*SIN (2.*WW)
      SG = -.0302777*SIN (G) + .0102777*SIN (G+2.*WW)
     1 -.00305555*SIN (2.*G+2.*WW)
      SG=(SG*DTR)/SIN(YN)
      RO=RO*DTR
      TA=TA*DTR
      YN = YN + RO
      RO = OM + SG + DT
      CI = COS (RO)*SIN (EPSIL)*SIN (YN)
     1 +COS (EPSIL)*COS (YN)
      SI=1.-CI*CI
      SI = SQRT (SI)
      SO = -SIN (RO)*SIN (YN)/SI
      CO=1.-SO*SO
      CO=SQRT(CO)
      SD = -SIN (RO)*SIN (EPSIL)/SI
      CD = -SIN (RO)*SO*COS (EPSIL) - COS (RO)*CO
      DL=ARKTNS(360,CD,SD)
      CA = DL + (CR + TA) - (OM + SG)
      SA=SIN(CA)
      CA=COS(CA)
      RO = COS (RO)*SIN (EPSIL)/(SI*CD)
      EM(1,1) = CA*CO - SA*SO*CI
      EM(1,2) = CA*SO + SA*CO*CI
      EM(1,3) = SA*SI
      EM(2,1) = -SA*CO - CA*SO*CI
      EM(2,2) = -SA*SO + CA*CO*CI
      EM(2,3) = CA*SI
      EM(3,1) = SO*SI
      EM(3,2) = -CO*SI
      EM(3,3) = CI
      RETURN
      END
```

Subroutine: MTRN

Purpose: Computes the matrix product of two 3X3 matrices.

Calling Sequence: CALL MTRN(A,B,C)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A,B	(3,3)			Input matrices.
Ø	C	(3,3)			AB

Common storages used: None.

Subroutines required: None.

```
$IBFTC MTRN     XR3,M94,NODD
      SUBROUTINE MTRN(A,B,C)                                           MTRN0001
      DIMENSION  A(3,3) ,B(3,3) ,C(3,3)                                MTRN0002
      DO 1 I=1,3                                                       MTRN0003
      DO 1 J=1,3                                                       MTRN0004
      C(I,J) = 0.                                                      MTRN0005
      DO 1 K=1,3                                                       MTRN0006
    1 C(I,J) = C(I,J)+A(I,K)*B(K,J)                                    MTRN0007
      RETURN                                                           MTRN0008
      END                                                              MTRN0009
```

Subroutine: MXLEST

Purpose: Computes a maximum likelihood estimate of vehicle state using the data from a single tracking station. The computations are described in detail in Appendix B of Reference 1.

Calling Sequence: CALL MXLEST

Common storages used: //162 cells,/DATCØM/,/DCPCØM/,/DFMCØM/,/EDTCØM/, /ESTCØM/,/MLECØM/

Subroutines required: DEHA, DEQTR, DGTRN, DGTSN, DMPLY, DMVTRN, DNØRM, DPFMRS, MEAS2X, PSEUDØ, STEPDI, STTIME

```
$IBFTC MC13M5  XR3,M94,NODD,LIST
      SUBROUTINE MXLEST                                                 MXLE0001
C     COMPUTES MAXIMUM LIKELIHOOD ESTIMATE OF VEHICLE STATE             MXLE0002
C                                                                       MXLE0003
      DOUBLE PRECISION   DATAN2 ,DCOS ,DNORM ,DSIN                      MXLE0004
C                                                                       MXLE0005
      COMMON             /DCPCOM/CDCP(900)                              MXLE0006
      DOUBLE PRECISION   ALOWR ,BDOWN ,CNST(7) ,EPSBAR ,CONTRL(20)      MXLE0007
     1                  ,AUPPR ,BUP   ,RCNST   ,EPSDEL ,RTD             MXLE0008
     2                         ,BO    ,VCNST           ,SCALE  (6)      MXLE0009
      DIMENSION          CBODY(8,11) ,NSCTRL(14)                        MXLE0010
      EQUIVALENCE               (CDCP(112),ETAPE ) ,(CDCP(683),NSTTM )  MXLE0011
     1    ,(CDCP(758),INDSTA) ,(CDCP(777),ICB    ) ,(CDCP(684),NSTTR )  MXLE0012
     2    ,(CDCP( 17),CBODY ) ,(CDCP(111),IERR   ) ,(CDCP(   7),RTD   )  MXLE0013
     3    ,(CDCP(825),CONTRL) ,(CDCP(787),NSCTRL) ,(CDCP(813),SCALE )   MXLE0014
      EQUIVALENCE               (CONTRL(18),BDOWN ),(CONTRL( 2),EPSBAR) MXLE0015
     1    ,(CONTRL(20),ALOWR ),(CONTRL(17),BUP    ),(CONTRL( 3),EPSDEL) MXLE0016
     2    ,(CONTRL(19),AUPPR ),(CONTRL( 8),CNST   ),(CONTRL( 4),EPSEUD) MXLE0017
     3    ,(CONTRL(16),BO     ),(CONTRL(15),CNTL   ),(CONTRL( 6),RCNST ) MXLE0018
     4                                             ,(CONTRL( 7),VCNST ) MXLE0019
      EQUIVALENCE               (NSCTRL(11),MAXAMB),(NSCTRL( 8),KSCALE) MXLE0020
     1                         ,(NSCTRL( 7),MTRY   )                    MXLE0021
C                                                                       MXLE0022
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)                MXLE0023
      DOUBLE PRECISION   DRNGE ,OMG3 ,ONTIME                            MXLE0024
      EQUIVALENCE        (INDDAT(27),DRNGE ) ,(INDDAT(29),OMG3   )      MXLE0025
     1                  ,(INDDAT(12),KTAU  ) ,(INDDAT(13),ONTIME)       MXLE0026
     2                  ,(INDDAT( 8),MTYPE )                            MXLE0027
C                                                                       MXLE0028
      COMMCN             /ESTCOM/CEST(804)                              MXLE0029
      DOUBLE PRECISION   DELDAN(2) ,SE(14,20)                           MXLE0030
      EQUIVALENCE        (CEST( 97),DELDAN) ,(CEST( 61),EFEDAN)         MXLE0031
     1                  ,(CEST(245),SE     )                            MXLE0032
      COMMCN             /DATCOM/CDAT(299)                              MXLE0033
      DOUBLE PRECISION   BIAS(2) ,FTR ,SPDLT ,STA(10) ,TAU              MXLE0034
      EQUIVALENCE               (CDAT(133),FTR    ) ,(CDAT(139),STA    ) MXLE0035
     1                         ,(CDAT(135),OMEGA ) ,(CDAT(159),TAU    ) MXLE0036
     2    ,(CDAT(   1),BIAS   ) ,(CDAT(137),SPDLT ) ,(CDAT(233),TT280 ) MXLE0037
C                                                                       MXLE0038
      COMMON             /DFMCOM/IFM(14),RFM(6,12)                      MXLE0039
     1                         ,DFM (4),BFM (577),SNT(2,102)            MXLE0040
      DOUBLE PRECISION   BFM,DFM,RFM                                    MXLE0041
C                                                                       MXLE0042
      COMMON             /MLECOM/CMLE(1070)                             MXLE0043
      DOUBLE PRECISION   AMU       ,EDEL   ,ETIMR1    ,STBIAS (4),UTIMR MXLE0044
     1                  ,COEF(2,4),EDHAT  ,ETIMV1    ,STIMR      ,UTIMR1 MXLE0045
     2                  ,CTAU      ,EDHATX,HY     (6),STIMR1     ,X1 (6) MXLE0046
     3                  ,DATAT (4),EHAT   ,OBIAS  (4),TB12C(3,3),XA1(6) MXLE0047
     4                  ,DRA       ,EHATL ,PI    (6,6),TD2C (3,3),XM (6) MXLE0048
     5                  ,E         ,EHATX ,PX12M(6,6),TEST   (20)       MXLE0049
     6                  ,EBAR      ,ELAST ,QI     (4),TMEAN             MXLE0050
      LOGICAL            LDATA(4) ,LMEAS(4) ,LMLE ,LRESDL               MXLE0051
      DIMENSION          DATA(5,100) ,LBM(12) ,NROT(4)                  MXLE0052
     1                              ,LBS(12) ,NYR (4)                   MXLE0053
      EQUIVALENCE               (CMLE( 55),QI     ) ,(CMLE(135),PX12M ) MXLE0054
     1                         ,(CMLE(531),LBM    ) ,(CMLE( 27),STBIAS) MXLE0055
     2    ,(CMLE(363),AMU    ) ,(CMLE(543),LBS    ) ,(CMLE( 13),STIMR ) MXLE0056
     3    ,(CMLE(347),COEF   ) ,(CMLE(401),LDATA ) ,(CMLE(   5),STIMR1) MXLE0057
     4    ,(CMLE(365),CTAU   ) ,(CMLE(397),LMEAS ) ,(CMLE( 81),TB12C ) MXLE0058
     5    ,(CMLE(570),DATA   ) ,(CMLE(406),LMLE  ) ,(CMLE( 63),TD2C  ) MXLE0059
     6    ,(CMLE( 39),DATAT ) ,(CMLE(407),LRESDL) ,(CMLE(307),TEST  ) MXLE0060
     7    ,(CMLE(367),DRA    ) ,(CMLE(432),NPTSTT) ,(CMLE(   3),TMEAN ) MXLE0061
     8    ,(CMLE(   9),ETIMR1) ,(CMLE(423),NROT  ) ,(CMLE( 15),UTIMR ) MXLE0062
     9    ,(CMLE( 11),ETIMV1) ,(CMLE(431),NSTATE) ,(CMLE(   7),UTIMR1) MXLE0063
     1    ,(CMLE(295),HY     ) ,(CMLE(427),NYR   ) ,(CMLE( 99),X1    ) MXLE0064
     2    ,(CMLE(433),IQUAL ) ,(CMLE( 31),OBIAS ) ,(CMLE(111),XA1   ) MXLE0065
     3    ,(CMLE(434),ITIME ) ,(CMLE(223),PI     ) ,(CMLE(123),XM    ) MXLE0066
      EQUIVALENCE              (TEST( 6),EBAR   ) ,(TEST( 8),EHAT  )    MXLE0067
     1                        ,(TEST( 7),EDEL   ) ,(TEST(14),EHATL )    MXLE0068
     2                        ,(TEST( 9),EDHAT ) ,(TEST(17),EHATX )     MXLE0069
     3    ,(TEST( 5),E       ) ,(TEST(18),EDHATX) ,(TEST(11),ELAST )    MXLE0070
C                                                                       MXLE0071
      COMMON             SAVE(120),D(21)                                MXLE0072
      DOUBLE PRECISION   D                                              MXLE0073
C                                                                       MXLE0074
```

```
      DOUBLE PRECISION   ALPHA(2) ,DXM(6) ,GHAI    ,XMHAT (6) ,YR(4)    MXLE0075
     1                  ,BETA     ,EPSBT   ,OMEGAX ,XMLAST(6)           MXLE0076
     2                  ,DRGE     ,G  (6) ,P(6,6) ,XXHAT (6)            MXLE0077
      DIMENSION          LBEST(6) ,LBG  (7) ,LBSC(7) ,LBSXM(2)          MXLE0078
     1                  ,LBETA(5) ,LBORB(9) ,LBSS(3) ,MD     (3)        MXLE0079
C                                                                       MXLE0080
      DATA  LBEST/6HALL CO,6HORDINA,6HTES   ,6HRANGE ,6HONLY  ,6H      /MXLE0081
      DATA  LBETA/6HFIXED ,6HAT     ,6HVARIAB,6HLE    ,6HDOWN  /        MXLE0082
      DATA  LBG  /6HEXTERN,6HALLY   ,6HEACH T,6HRIAL  ,6HFIRST ,6HTRIAL MXLE0083
     1          ,6HONLY  /                                              MXLE0084
      DATA  LBORB/6HSMA   ,6HECC    ,6HINC    ,6HLAN    ,6HAPF    ,6HTHE    MXLE0085
     1          ,6HSLR   ,6HHEV    ,6HRCA    /                          MXLE0086
      DATA  LBSC /6HEXTERN,6HALLY   ,6HEQUAL ,6HTO 1.0,6HTO NOR,6HMALIZE MXLE0087
     1          ,6H XM   /                                              MXLE0088
      DATA  LBSS /6HKM    ,6HRAD    ,6HRAD    /                         MXLE0089
      DATA  LBSXM/6HOF SCA,6HLED XM/                                    MXLE0090
      DATA  MD    / 1,3,-1 /                                            MXLE0091
C                                                                       MXLE0092
  600 FORMAT(45H1 T  OUTPUT FROM MAXIMUM LIKELIHOOD ESTIMATOR/3H  R/3H  MXLE0093
     1I/3H  A/3H  L)                                                    MXLE0094
  601 FORMAT(///1X,I2,16H  ESTIMATION OF ,3A6//5X,12HANCHOR TIMES,10X,14MXLE0095
     1HSTATION TIME =,D24.16,8X,14HVEHICLE TIME =,D24.16)               MXLE0096
  602 FORMAT(18H0     VEHICLE STATE//8X,19HM-FRAME COORDINATES,48X,5HRATEMXLE0097
     1S,3(/18X,2A6,D28.16,1X,A6,7X,D23.16,1X,A3,4H/SEC))                MXLE0098
  603 FORMAT(/8X,31HCARTESIAN COORDINATES (C-FRAME)/20X,18HEARTH CENTEREMXLE0099
     1D (X),7X,A6,14H CENTERED (XA),5X,A6,16H TO EARTH VECTOR)          MXLE0100
  604 FORMAT(15X,3D25.16)                                               MXLE0101
  605 FORMAT(/8X,52HORBITAL ELEMENTS IN KILOMETERS AND DEGREES (C-FRAME)MXLE0102
     1,3(/12X,3(6X,A3,D25.16)))                                         MXLE0103
  606 FORMAT(1H0,4X,20HCONVERGENCE CONTROLS/8X,33HMAXIMUM NUMBER OF TRIAMXLE0104
     1LS (MTRY) =,I4/8X,55HCONVERGENCE LEVEL FOR MEAN WGT SQ OF RESIDUALMXLE0105
     2S (EPSBAR),17X,1H=,E16.8/8X,73HFRACTIONAL LEVEL FOR FRACTIONAL CHAMXLE0106
     3NGE IN WGT SQ OF RESIDUALS (EPSDEL)  =,E16.8)                     MXLE0107
  607 FORMAT(24H0     CONSTRAINT CONTROLS/8X,25HSCALING FACTOR (BETA) IS MXLE0108
     1,A6,A2,A6,4HONLY)                                                 MXLE0109
  608 FORMAT(11X,25HINITIAL VALUE   (B0)      =,E16.8)                  MXLE0110
  609 FORMAT(11X,25HDECREASE VALUE (BDOWN)   =,E16.8,6X,44HMINIMUM FRACTIMXLE0111
     1ONAL CORRECTION      (ALOWR)  =,E16.8)                            MXLE0112
  610 FORMAT(11X,25HINCREASE VALUE (BUP)     =,E16.8,6X,44HMAXIMUM ALPHA MXLE0113
     1FOR INCREASING BETA (AUPPR)  =,E16.8)                             MXLE0114
  611 FORMAT(1H0,7X,38HUNSCALED CONSTRAINT VECTOR (G) IS SET ,3A6)      MXLE0115
  612 FORMAT(20X,3D26.16)                                               MXLE0116
  613 FORMAT(11X,22HPOSITION CONSTRAINT IS,E16.8,19H  OF DISTANCE FROM ,MXLE0117
     1A6/11X,22HPOSITION CONSTRAINT IS,E16.8,13H  OF VELOCITY)          MXLE0118
  615 FORMAT(43H0     M-FRAME SCALING VECTOR (SCALE) IS SET ,3A6)       MXLE0119
  616 FORMAT(/5X,20H  CONVERGED ON TRIAL,I3)                            MXLE0120
  617 FORMAT(/5X,18HCOVARIANCE MATRIX ,2A6)                             MXLE0121
  618 FORMAT(10X,3(14X,2A6))                                            MXLE0122
  619 FORMAT(8X,2A6,3(D24.16,2X))                                       MXLE0123
  620 FORMAT(/5X,23HSUMMARY OF RESIDUAL FIT/24X,11HMEAN SQUARE,15X,13HNUMXLE0124
     1MBER OF PTS)                                                      MXLE0125
  621 FORMAT(8X,2A6,D24.16,I9)                                          MXLE0126
  622 FORMAT(62X,5HVALUE,18X,17HCONVERGENCE LEVEL/8X,27HTOTAL MEAN WGT SMXLE0127
     1QUARE(EBAR),23X,1H=,D24.16,E18.8/8X,51HFRACTIONAL IMPROVEMENT IN TMXLE0128
     2OTAL MEAN SQUARE(EDEL) =,D24.16,E18.8)                            MXLE0129
  623 FORMAT(1H1,9X,15HRESIDUAL OUTPUT//13X,4HTIME,13X,4(2A6,5X))       MXLE0130
  631 FORMAT(1H0,I2,26H  DIFFERENTIAL CORRECTIONS)                      MXLE0131
  632 FORMAT(/20X,18HSCALED EST (XXHAT),6X,7H-SCALE-,18X,18HUNSCALED CONMXLE0132
     1ST (G))                                                           MXLE0133
  633 FORMAT(5X,2A6,D23.16,2D25.16)                                     MXLE0134
  634 FORMAT(/20X,16HESTIMATE (XMHAT),8X,21HCONST VECTOR (BETA*G),4X,21HMXLE0135
     1DIFF CORRECTION (DXM))                                            MXLE0136
  635 FORMAT(/5X,31HINFORMATION MATRIX OF SCALED XM)                    MXLE0137
  636 FORMAT(1H0,I2,17H  TEST PARAMETERS,18X,10HLAST TRIAL,6X,7HCURRENT,MXLE0138
     19X,9HPREDICTED,7X,9HPREDICTED,7X,13HUNCONSTRAINED/70X,11HON LAST TMXLE0139
     2RY,5X,12HFOR NEXT TRY,4X,9HPREDICTED/5X,29HSUM OF WGT SQ OF RESD(EMXLE0140
     33)     =,5E16.8/5X,29HMEAN WGT SQ OF RESD(EBAR)    =,2E16.8/5X,29HFRAMXLE0141
     4CTIONAL IMPRV IN E(EDEL) =,5E16.8)                                MXLE0142
  637 FORMAT(1H0,I2,21X,4H-HY-,22X,19HUNCONSTRAINED HYHAT,7X,17HCONSTRAIMXLE0143
     1NED HYHAT)                                                        MXLE0144
  638 FORMAT(1H0,I2,19H  CONVERGENCE TESTS,24X,10HTEST LEVEL/5X,11HEBAR MXLE0145
     1      =,D24.16,E18.8/5X,11HEDEL       =,D24.16,E18.8/5X,11HALPHA   MXLE0146
     2 =,D24.16)                                                        MXLE0147
  639 FORMAT(5X,4HBETA,6X,1H=,D24.16)                                   MXLE0148
  642 FORMAT(/5X,4HRANK,6X,1H=,I24)                                     MXLE0149
```

```
  690 FORMAT(/38H    * MAXIMUM NUMBER OF TRIALS REACHED)
  691 FORMAT(/24H    * FAILED TO CONVERGE)
  692 FORMAT(1H0,I2,37H *PROCESS DIVERGING, EDEL IS POSITIVE)
  693 FORMAT(1H0,I2,18H  INCREASE BETA TO,E16.8,20H, PROCESS CONVERGING)
  694 FORMAT(1H0,I2,37H *MAXIMUM FRACTIONAL DIFF CORRECTION(,E15.8,11H).
     1LT.ALCWR(,E15.8,1H))
  695 FORMAT(5X,14HREDUCE BETA TO,E16.8,47H AND MAKE NEW TRIAL FROM LAST
     1 ACCEPTED ESTIMATE)
C
C**   CONVERT POLYNOMIAL COEFFICIENTS TO MEASUREMENT COORDINATES
C
C     SET UP ANCHOR POINT TIME
    1 CONTINUE
      LMLE = .FALSE.
      IFM(ICB) = 2
      STIMR1 = TMEAN+ONTIME
      UTIMR1 = STIMR1+STBIAS(1)+STBIAS(2)*STIMR1
      ETIMR1 = UTIMR1+DELDAN(1)+DELDAN(2)*UTIMR1
C
C     EARTH ORIENTATION (SEE GTR2BD)
      CALL DPFMRS (ETIMR1,EFEDAN,ICB,IERR,ETAPE)
      IF (IERR.NE.0)      GO TO 998
      CALL DEQTR  (ETIMR1,D)
      CALL DGTSN  (D(13),0,D(2),D(1),D(4))
      D(1) = D(3)+DFM(2)
      D(2) = DFM(1)
      CALL DGTRN  (D(4),MD,D,3)
      CALL DMVTRN (D(13),D(4),TD2C,2,3)
      DRA = D(7)
      CALL DEHA   (UTIMR1,DRA,GHA1,OMEGA)
      OMEGAX = OMEGA
      D(1) = DCOS(GHA1)
      D(2) = DSIN(GHA1)
      DO 2 I=1,3
      TB12C(I,1) = D(1)*TD2C(I,1)+D(2)*TD2C(I,2)
      TB12C(I,2) = D(1)*TD2C(I,2)-D(2)*TD2C(I,1)
      TB12C(I,3) = TD2C(I,3)
    2 CONTINUE
C
C     TRANSFER COEFFICIENTS TO MEASUREMENT COORDINATES
   10 CONTINUE
      XM(1) = COEF(1,3)
      XM(2) = COEF(1,1)
      XM(3) = COEF(1,2)
      XM(4) = COEF(2,3)
      XM(5) = COEF(2,1)
      XM(6) = COEF(2,2)
      NSTATE = 6
      IF (MTYPE.EQ.1)     GO TO 22
C
C     C-BAND IS SET UP
C     CONTINUE FOR OTHERS
      IF (IQUAL.EQ.2)     GO TO 15
C
C     RANGE DATA IS AVAILABLE
      D(1) = 0.5D0
      IF (MTYPE.EQ.2)     D(1) = D(1)*SPDLT
      DRGE = D(1)*DRNGE
      XM(1) = D(1)*XM(1)
      XM(4) = D(1)*XM(4)
      LRESDL = .TRUE.
      NSTATE = 0
C
C     MINIMIZE RESIDUALS VS RANGE AMBIGUITY
      ITP = NSTTR
      NSTTR = 0
      ELAST = 1.D+20
      IAM = 1
      GO TO 35
   11 IF (E.GT.ELAST)     GO TO 12
      ELAST = E
      IAM = IAM+1
      XM(1) = XM(1)+DRGE
      IF (IAM.LE.MAXAMB) GO TO 35
   12 CONTINUE
```

```
      XM(1) = XM(1)-DRGE
      NSTATE = 6
      NSTTR = ITP
      GO TO 22
C
C     RANGE DATA NOT AVAILABLE
C     ESTIMATE RANGE FIRST
   15 CONTINUE
      XM(1) = 25.D+3
      IF (ICB.NE.3)       XM(1) = DNORM(REM(1,3))
      GO TO (999,18,17,16) ,MTYPE
C     DSIF
   16 XM(2) = DATAN2 (STA(2),STA(1))+GHA1-XM(2)
      XM(5) =-XM(5)
      OMEGAX = 0.DO
C     USBS
   17 D(2) = BIAS(2)
      GO TO 19
C     STADAN/VHF
   18 D(2) =-1.DO
   19 CONTINUE
      D(1) = COEF(1,4)/TAU
      IF (KTAU.EQ.0)      D(1) = TAU/COEF(1,4)
      D(1) =-(BIAS(1)-D(1))/(2.DO*FTR*D(2))
      XM(4) = SPDLT*D(1)/(1.DO+D(1))
      NSTATE = 1
      GO TO 22
C
C**   OUTPUT HEADINGS AND CONTROL DATA
C
C     RESIDUAL OUTPUT ONLY
   20 CONTINUE
      LRESDL = .TRUE.
      WRITE (6,623)      (LBM(I),LBM(I+4),I=1,4)
      GO TO 35
C
C     MAXIMUM LIKELIHOOD ESTIMATION
   21 NSTATE = 6
   22 CONTINUE
      LRESDL = .FALSE.
      IF (NSTTM.EQ.0)     GO TO 30
      WRITE (6,600)
      IF (NSTTM.LT.4)     GO TO 30
      WRITE (6,606)       MTRY,EPSBAR,EPSDEL
      I = 1
      IF (CNTL.NE.0.)     I=3
      J = I+1
      IF (CNTL.LT.0.)     J=5
      WRITE (6,607)      (LBETA(K),K=I,J)
      WRITE (6,608)       BO
      IF (CNTL.EQ.0.)     GO TO 23
      WRITE (6,609)       BDOWN,ALOWR
      IF (CNTL.LE.0.)     GO TO 23
      WRITE (6,610)       BUP,AUPPR
   23 CONTINUE
      I = 1
      IF (CNST.NE.0.)     I=3
      J = I+1
      IF (CNST.GE.0.)     GO TO 24
      I = 5
      J = 7
   24 WRITE (6,611)      (LBG(K),K=I,J)
      IF (CNST.NE.0.)     GO TO 25
      WRITE (6,612)      (CNST(I),I=2,7)
      GO TO 26
   25 WRITE (6,613)       RCNST,CBODY(1,ICB),VCNST
   26 CONTINUE
      I = 3
      IF (KSCALE.EQ.0)    GO TO 27
      I = 1
      IF (KSCALE.LT.0)    GO TO 27
      I = 5
      J = 7
      GO TO 28
   27 J = I+1
```

```
   28 WRITE (6,615)      (LBSC(K),K=I,J)
      IF (KSCALE.GE.0)   GO TO 30
      WRITE (6,612)      SCALE
C
C**   INITIALIZE
C
C     HERE FOR M.L.E.
   30 CONTINUE
      DO 31 I=1,6
      DXM(I) = 0.DO
   31 TEST(I+10) = 9.9999999999999D+21
      KSS = 2
      IF (NSTATE.EQ.1)   KSS=0
      NTRY = 0
      DO 32 I=1,36
   32 PI(I,1) = 0.DO
      EPSBT = 0.
      DO 33 I=1,4
   33 IF (LMEAS(I))      EPSBT=EPSBT+EPSBAR
      BETA = BO
C
C     HERE FOR RESIDUAL COMPUTATION
   35 CONTINUE

C
C**   M.L.E. ITERATION LOOP STARTS HERE
C
C     ANCHOR POINT
  100 NTRY = NTRY+1
      ETIMV1 = ETIMR1-XM(1)/SPDLT
      IF (MTYPE.EQ.4)    GO TO 101
      CALL DMVTRN (TB12C,TT2BO,TD2C,1,3)
  101 CALL DMVTRN (TB12C,STA,D(13),1,1)
      CALL MEAS2X (X1,PX12M,XM,NROT,TD2C,D(13),TB12C(1,3),OMEGAX)
      CALL DPEMRS (ETIMV1,EFEDAN,ICB,IERR,ETAPE)
      IF (IERR.NE.0)     GO TO 998
C
C     CLEAR SUMMATION STORAGE
      DO 106 I=1,4
  106 NYR(I) = 0
      DO 102 I=1,6
      HY(I) = 0.DO
      TEST(I) = 0.DO
  102 XA1(I) = X1(I)+RFM(I,3)
      CALL STEPDI (ETIMV1,XA1,AMU)
C
C     OUTPUT ANCHOR POINT
      IF (LRESDL)        GO TO 110
      IF (NSTTM.LT.2)    GO TO 110
      IF (NSTTM.EQ.2.AND.NTRY.NE.1) GO TO 110
  103 I = 1
      IF (NSTATE.NE.6)   I = 4
      J = I+2
      WRITE (6,601)      NTRY,(LBEST(K),K=I,J),STIMR1,ETIMV1
      WRITE (6,602)     (LBS(I),LBS(I+6),XM(I),LBSS(I),
     1                   XM(I+3),LBSS(I),I=1,3)
      IF (NSTTM.LT.3)    GO TO 105
      CALL X2ORBD (XA1,D,AMU)
      DO 104 I=3,6
  104 D(I) = D(I)*RTD
      WRITE (6,605)     (LBORB(I),D(I),I=1,9)
  105 WRITE (6,603)      CBODY(1,ICB),CBODY(1,ICB)
      WRITE (6,604)     (X1(I),XA1(I),RFM(I,3),I=1,6)
      IF (LMLE)     GO TO 242
C
C     SET UP SCALING
  110 IF (LRESDL)   GO TO 120
      IF (KSCALE)   115,111,113
  111 DO 112 I=1,6
  112 SCALE(I) = 1.DO
      GO TO 115
  113 DO 114 J=1,6
      SCALE(J) = DABS(XM(J))
      DO 114 I=1,6
  114 PX12M(I,J) = SCALE(J)*PX12M(I,J)
```

```
  115 CONTINUE
      IF (CNST.LT.0.)     GO TO 117
      IF (NTRY.NE.1)      GO TO 120
      IF (CNST.NE.0.)     GO TO 117
      DO 116 I=1,6
  116 G(I) = CNST(I+1)
      GO TO 120
  117 CONTINUE
      YR(1) = RCNST
      YR(2) = VCNST
      IF (ABS(CNST).EQ.2.) GO TO 118
      YR(1) = YR(1)*DNORM(XA1)
      YR(2) = YR(2)*DNORM(XA1(4))
  118 CONTINUE
      G(1) = 1.D0
      G(3) = 1.D0/XM(1)
      D(1) = DABS(DCOS(XM(3)))
      IF (D(1).LT.1.D-20) D(1) = 1.D-20
      G(2) = G(3)/D(1)
      DO 119 I=1,3
      G(I+3) = G(I)*YR(2)
  119 G(I)   = G(I)*YR(1)
C
C**   COMPUTE RESIDUALS AND INFORMATION MATRIX
C
  120 CONTINUE
      DO 123 ITIME=1,NPTSTT
      DO 121 I=1,4
      LDATA(I) = LMEAS(I)
      DATAT(I) = DATA(I+1,ITIME)
  121 IF (DATAT(I).LT.0.) LDATA(I) = .FALSE.
      STIMR = CNTIME+DATA(1,ITIME)
      UTIMR = STIMR+STBIAS(1)+STBIAS(2)*STIMR
      IF (KTAU.NE.0)      GO TO 122
      TAU = DATAT(4)
      DATAT(4) = CTAU
      BIAS(1)  = OMG3+SE(10,INDSTA)/TAU
  122 CONTINUE
      CALL STTIME
  123 IF (IERR.NE.0)      GO TO 999
C
C**   COMPUTE DIFFERENTIAL CORRECTION DXM AND ESTIMATE XMHAT
C
  140 IF (LRESDL)         GO TO 160
      DO 141 I=1,36
  141 P(I,1) = PI(I,1)
      NRANK = NSTATE
      CALL PSEUDO (P,NRANK,6,EPSEUD)
      CALL DMPLY  (P,HY,XXHAT,NSTATE,NSTATE,1,6,6,6,0)
      ALPHA = .999D+20
      DO 142 I=1,NSTATE
      D(I+1) = G(I)*BETA
      XMHAT(I) = XXHAT(I)*SCALE(I)
      D(1) = DABS(XMHAT(I))
      IF (D.EQ.0.D0)      GO TO 142
      D(1) = D(I+1)/D(1)
      IF (D.LT.ALPHA)     ALPHA = D
  142 CONTINUE
      ALPHA(2) = ALPHA
      IF (ALPHA(2).GT.1.D0) ALPHA(2) = 1.D0
      DO 143 I=1,NSTATE
  143 DXM(I) = XMHAT(I)*ALPHA(2)
C
C**   COMPUTE TEST PARAMETERS
C
  160 CONTINUE
      DO 161 I=1,4
      IF (.NOT.LMEAS(I)) GO TO 161
      D(1) = TEST(I)*QI(I)
      E = E+D(1)
      D(2) = NYR(I)
      TEST(I) = TEST(I)/D(2)
      EBAR = EBAR+D(1)/D(2)
  161 CONTINUE
      IF (.NOT.LRESDL)    GO TO 162
```

```
      IF (NSTTR.NE.0)     GO TO 244
      IF (NSTATE.EQ.0)    GO TO 11
      GO TO 999
  162 CONTINUE
      EDEL = (E-ELAST)/E
      CALL DMPLY  (PI,XXHAT,D,NSTATE,NSTATE,1,6,6,6,0)
      CALL DMPLY  (XXHAT,D,D(7),1,NSTATE,1,6,6,6,1)
      CALL DMPLY  (XXHAT,HY,D(8),1,NSTATE,1,6,6,6,1)
      D( 9) = D(7)-2.D0*D(8)
      D(10) = (D(7)*ALPHA(2)-2.D0*D(8))*ALPHA(2)
      EHATX  = E+D( 9)
      EHAT   = E+D(10)
      EDHATX = D( 9)/EHATX
      EDHAT  = D(10)/EHAT
C
C**   OUTPUT TRIAL SUMMARY
C
  180 IF (NSTTM.LT.6)     GO TO 184
      WRITE (6,631)       NTRY
      IF (NSTTM.LT.8)     GO TO 181
      WRITE (6,632)
      WRITE (6,633)      (LBS(I),LBS(I+6),XXHAT(I),SCALE(I),G(I)
     1                   ,I=1,NSTATE)
  181 WRITE (6,634)
      DO 182 I=1,NSTATE
      D(7) = BETA*G(I)
  182 WRITE (6,633)       LBS(I),LBS(I+6),XMHAT(I),D(7),DXM(I)
      WRITE (6,642)       NRANK
      WRITE (6,639)       BETA
  184 CONTINUE
      WRITE (6,638)       NTRY,EBAR,EPSBT,EDEL,EPSDEL,ALPHA(1)
      IF (NSTTM.LT.5)     GO TO 200
      WRITE (6,636)       NTRY,ELAST,E,EHATL,EHAT,EHATX
     1                        ,TEST(12),EBAR
     2                        ,TEST(13),EDEL,TEST(15),EDHAT,EDHATX
      IF (NSTTM.LT.7)     GO TO 200
      WRITE (6,637)       NTRY
      DO 185 I=1,NSTATE
      D(7) = D(I)*ALPHA(2)
  185 WRITE (6,619)       LBS(I),LBS(I+6),HY(I),D(I),D(7)
      IF (NSTTM.LT.9)     GO TO 200
      WRITE (6,635)
      ASSIGN 252 TO N1
      ASSIGN 186 TO N2
      GO TO 250
  186 IF (NSTTM.LT.10)    GO TO 200
      WRITE (6,617)       LBSXM
      ASSIGN 251 TO N1
      ASSIGN 200 TO N2
      GO TO 250
C
C**   TEST FOR CONVERGENCE
C
  200 CONTINUE
      IF (EDEL.GT.0.D0)   GO TO 280
      IF (EBAR.LT.EPSBT)  GO TO 220
      IF (EPSDEL+EDEL.GT.0.D0.AND.ALPHA.GT.0.99D0) GO TO 220
C
C     DID NOT CONVERGE ON THIS TRY
      IF (NTRY.GE.MTRY)   GO TO 900
      IF (CNTL.LE.0.)     GO TO 210
      IF (ALPHA.GT.AUPPR)GO TO 210
  201 BETA = BETA*BUP
      IF (NSTTM.LT.2)     GO TO 210
      WRITE (6,693)       NTRY,BETA
C
C     SET UP ANOTHER ITERATION
  210 CONTINUE
      DO 211 I=1,36
  211 PI(I,1) = 0.D0
      DO 212 I=1,6
      XMLAST(I) = XM(I)
  212 XM(I) = XM(I)+DXM(I)
      DO 213 I=1,5
  213 TEST(I+10) = TEST(I+4)
```

```
      GO TO 100
C
C     CONVERGED
  220 LMLE = .TRUE.
      DO 221 J=1,NSTATE
      DO 221 I=1,NSTATE
  221 P(I,J) = P(I,J)*SCALE(I)*SCALE(J)
C
C**   OUTPUT SUMMARY OF M.L.E.
C
  240 CONTINUE
      IF (NSTTM.EQ.0)     GO TO 260
      WRITE (6,616)       NTRY
      IF (NSTTM.EQ.1)     GO TO 103
  242 CONTINUE
      IF (NSTTM.GE.3)     WRITE (6,622)  EBAR,EPSBT,EDEL,EPSDEL
      IF (NSTTM.LT.5)     GO TO 246
  244 WRITE (6,620)
      DO 245 I=1,4
      IF (.NOT.LMEAS(I))  GO TO 245
      WRITE (6,621)       LBM(I),LBM(I+4),TEST(I),NYR(I)
  245 CONTINUE
      IF (LRESDL)         GO TO 999
  246 CONTINUE
      IF (NSTTM.LT.6)     GO TO 260
      WRITE (6,617)
      ASSIGN 260 TO N2
      ASSIGN 251 TO N1
C
C     OUTPUT INFORMATION MATRIX
  250 CONTINUE
      DO 253 I=1,NSTATE,3
      II = I+KSS
      WRITE (6,618)      (LBS(J),LBS(J+6),J=I,II)
      DO 253 J=I,NSTATE
      GO TO N1, (251,252)
  251 WRITE (6,619)       LBS(J),LBS(J+6),(P(J,K),K=I,II)
      GO TO 253
  252 WRITE (6,619)       LBS(J),LBS(J+6),(PI(J,K),K=I,II)
  253 CONTINUE
      GO TO N2, (186,200,260)
C
C     REPEAT FOR RESIDUALS
  260 IF (NSTATE.EQ.1)    GO TO 21
      IF (NSTTR.GE.1)     GO TO 20
      GO TO 999
C
C     PROCESS DIVERGED ON LAST TRY
  280 CONTINUE
      IF (NSTTM.GE.2)     WRITE (6,692)  NTRY
      IF (CNTL.EQ.0.)     GO TO 910
      S = 0.
      DO 281 I=1,NSTATE
  281 S = S+ABS(DXM(I)/XM(I))
      IF (S.GE.ALOWR)     GO TO 282
      IF (NSTTM.GE.2)     WRITE (6,694)  NTRY,S,ALOWR
      GO TO 910
  282 BETA = BETA*BDOWN
      IF (NSTTM.GE.2)     WRITE (6,695)  BETA
      DO 283 I=1,36
  283 PI(I,1) = 0.D0
      DO 284 I=1,6
  284 XM(I) = XMLAST(I)
      GO TO 100
C
C**   PROCESS FAILED TO CONVERGE
C
  900 IF (NSTTM.GE.1)     WRITE (6,690)
  910 IF (NSTTM.GE.1)     WRITE (6,691)
      IF (NSTTR.GE.2)     GO TO 20
      GO TO 999
C
  998 IERR = IERR+15
  999 RETURN
      END
```

Subroutine: NUDATA

Purpose: To allow a convenient means whereby data may be entered into the various station arrays used by the Tracking Data Editor.

Calling Sequence: CALL NUDATA

Common storages used: /TRKCOM/

Subroutines required: None

Overlay data for the Tracking Data Editing Program consists of one or more sets of overlay cards. Each set consists of an ARRAY NAME card followed by one or more data cards. The ARRAY NAME card contains the code name of the array into which the data on the following data cards are to be put. The code name is six or fewer characters and begins in column 1. The allowable code names with a description of the related arrays are shown in Table 1.

The data cards for an array each contain up to three pairs of numbers. Each pair consists of an integer k and a value v. The integer k is a subscript indicating the relative location in the array C into which the value v is to be placed; thus C(k) = v. Three types of value data are permitted: BCD (Hollerith), integer, and double precision. The type used must be appropriate to the array code name as indicated in Table 1. The formats for the three types are:

```
BCD
    3(I3,A6,15X)
Integer
    3(I3,I2,19X)
Double Precision
    3(I3,D21.16)
```

Thus the integers k always occupy the same locations on a data card regardless of the type of data; i.e., the integers k are in columns 1-3, 25-27, and 49-51.

If k = 0 and $v \neq 0$ (or blank), v will be stored in the next successive location of the array, C, following the previous entry into that array; if there were no previous entry, storage will start in C(1). If k and v are both zero (v is blank for BCD), reading of the data for this array is terminated if they are the first pair in a card; otherwise the next card is processed. If k is negative, k is set positive, v is properly stored, and reading of data for this array is terminated. Thus reading of data for an array may be terminated by a card on which k_1 and v_1 are both zero

TABLE 1

STATION DATA ARRAYS

CODE NAME	DATA TYPE	DESCRIPTION
STANAM	BCD	6-letter station name.
PAIR	BCD	6-letter name of associated transmitting station, if three way doppler.
KODSTA	Integer	Two digit decimal station ID.
NALIGN	Integer	S-Band dish size 01 = 30 ft dish 02 = 85 ft dish
FTR	Double Precision	Transmitter frequency, HZ (cps), used for doppler.
CFRAC1	Double Precision	Atmospheric Refraction constants
CFRAC2	Double Precision	Atmospheric Refraction constants
BIAS	Double Precision	Doppler fixed offset bias, HZ (cps).
RATIO	Double Precision	Spacecraft transponder retransmission ratio, for doppler (transmit/receive).

(v_1 = blank for BCD), or by a negative value of k in any of the three integer fields.

When reading of data for an array is terminated, another ARRAY NAME card is expected. The overlay process may be stopped by an ARRAY NAME card on which the array code name field has either been left blank or contains the word STOP.

All data read by NUDATA are written on the system output tape with zero values of k replaced by the relative locations actually used.

If an ARRAY NAME card with an invalid code name is encountered, an error message is printed and the program is stopped.

```
$IBFTC MC134N  M94,NODD,LIST,XR3
CMC134N   BLOCK DATA OVERLAY FOR TRACKING DATA EDITOR                   NUDA0001
      SUBROUTINE NUDATA                                                  NUDA0002
      COMMON  /TRKCOM/CTRK(700)                                          NUDA0003
      DOUBLE PRECISION  FTR(50),  C1(50),   C2(50),   BIAS(50), RETR(50)NUDA0004
      DIMENSION   STANAM(50),   KODSTA(50),   NALIGN(50),    PAIR(50)    NUDA0005
      EQUIVALENCE               (CTRK(1),STANAM),    (CTRK(51),KODSTA)   NUDA0006
     1,  (CTRK(101),NALIGN),   (CTRK(151),FTR),     (CTRK(251),C1)      NUDA0007
     2,  (CTRK(351),C2),       (CTRK(451),BIAS),    (CTRK(551),RETR)    NUDA0008
     3,  (CTRK(651),PAIR)                                                NUDA0009
C                                                                        NUDA0010
      DOUBLE PRECISION  DTEMP(3)                                         NUDA0011
      DIMENSION   IND(3),        HTEMP(6),      ITEMP(3),      ARRAY(9)  NUDA0012
      EQUIVALENCE               (DTEMP,HTEMP),       (HTEMP(4),ITEMP)    NUDA0013
      REAL NAME                                                          NUDA0014
C                                                                        NUDA0015
  501 FORMAT(A6)                                                         NUDA0016
  502 FORMAT(I3,A6,2(15X,I3,A6))                                         NUDA0017
  503 FORMAT(I3,I2,2(19X,I3,I2))                                         NUDA0018
  504 FORMAT(3(I3,D21.16))                                               NUDA0019
  600 FORMAT(1H1,52X,13HOVERLAY INPUT)                                   NUDA0020
  601 FORMAT(16H0THE ARRAY NAME ,A6,15H DOES NOT EXIST)                  NUDA0021
  602 FORMAT(7H0ARRAY ,A6)                                               NUDA0022
  603 FORMAT(6X,3(I3,3X,A6,22X))                                         NUDA0023
  604 FORMAT(6X,3(I3,3X,I2,26X))                                         NUDA0024
  605 FORMAT(6X,3(I3,D25.16,6X))                                         NUDA0025
C                                                                        NUDA0026
      DATA ARRAY/6HSTANAM,6HPAIR  ,6HKODSTA,6HNALIGN,6HFTR   ,6HCFRAC1, NUDA0027
     16HCFRAC2,6HBIAS  ,6HRATIO /                                        NUDA0028
      DATA BLANKS/6H      /                                              NUDA0029
      DATA STOP/6HSTOP  /                                                NUDA0030
C                                                                        NUDA0031
    1 WRITE(6,600)                                                       NUDA0032
   10 K=1                                                                NUDA0033
      LOC=0                                                              NUDA0034
      READ(5,501) NAME                                                   NUDA0035
      IF(NAME.EQ.BLANKS.OR.NAME.EQ.STOP) GO TO 999                       NUDA0036
      DO 11 I=1,9                                                        NUDA0037
      KK=I                                                               NUDA0038
      IF(NAME.EQ.ARRAY(I)) GO TO 12                                      NUDA0039
   11 CONTINUE                                                           NUDA0040
      WRITE(6,601) NAME                                                  NUDA0041
      STOP                                                               NUDA0042
   12 CONTINUE                                                           NUDA0043
      WRITE(6,602) NAME                                                  NUDA0044
      GO TO (20,20,30,30,40,40,40,40,40),KK                              NUDA0045
   20 CONTINUE                                                           NUDA0046
C                                                                        NUDA0047
C     INPUT BCD DATA                                                     NUDA0048
  201 KOUNT=0                                                            NUDA0049
      READ(5,502) (IND(I),HTEMP(I),I=1,3)                                NUDA0050
      IF(IND(1).EQ.0.AND.HTEMP(1).EQ.BLANKS) GO TO 10                    NUDA0051
      DO 24 I=1,3                                                        NUDA0052
      IF(IND(I).NE.0) GO TO 21                                           NUDA0053
      IF(HTEMP(I).EQ.BLANKS) GO TO 26                                    NUDA0054
      IND(I)=LOC+1                                                       NUDA0055
   21 LOC=IABS(IND(I))                                                   NUDA0056
      KOUNT=KOUNT+1                                                      NUDA0057
      IF(KK.EQ.2) GO TO 23                                               NUDA0058
      STANAM(LOC)=HTEMP(I)                                               NUDA0059
   22 IF(IND(I).LT.0) GO TO 25                                           NUDA0060
      GO TO 24                                                           NUDA0061
   23 PAIR(LOC)=HTEMP(I)                                                 NUDA0062
      GO TO 22                                                           NUDA0063
   24 CONTINUE                                                           NUDA0064
      GO TO 26                                                           NUDA0065
   25 K=0                                                                NUDA0066
   26 WRITE(6,603) (IND(I),HTEMP(I),I=1,KOUNT)                           NUDA0067
      IF(K.EQ.0) GO TO 10                                                NUDA0068
      GO TO 201                                                          NUDA0069
   30 CONTINUE                                                           NUDA0070
C                                                                        NUDA0071
C     INPUT INTEGER DATA                                                 NUDA0072
  301 KOUNT=0                                                            NUDA0073
      READ(5,503) (IND(I),ITEMP(I),I=1,3)                                NUDA0074
```

```
      IF(IND(1).EQ.0.AND.ITEMP(1).EQ.0) GO TO 10                        NUDA0075
      DO 34 I=1,3                                                       NUDA0076
      IF(IND(I).NE.0) GO TO 31                                          NUDA0077
      IF(ITEMP(I).EQ.0) GO TO 36                                        NUDA0078
      IND(I)=LOC+1                                                      NUDA0079
   31 LOC=IABS(IND(I))                                                  NUDA0080
      KOUNT=KOUNT+1                                                     NUDA0081
      IF(KK.EQ.4) GO TO 33                                              NUDA0082
      KODSTA(LOC)=ITEMP(I)                                              NUDA0083
   32 IF(IND(I).LT.0) GO TO 35                                          NUDA0084
      GO TO 34                                                          NUDA0085
   33 NALIGN(LOC)=ITEMP(I)                                              NUDA0086
      GO TO 32                                                          NUDA0087
   34 CONTINUE                                                          NUDA0088
      GO TO 36                                                          NUDA0089
   35 K=0                                                               NUDA0090
   36 WRITE(6,604) (IND(I),ITEMP(I),I=1,KOUNT)                          NUDA0091
      IF(K.EQ.0) GO TO 10                                               NUDA0092
      GO TO 301                                                         NUDA0093
   40 CONTINUE                                                          NUDA0094
      KK=KK-4                                                           NUDA0095
C                                                                       NUDA0096
C     INPUT DOUBLE PRECISION DATA                                       NUDA0097
  401 KOUNT=0                                                           NUDA0098
      READ(5,504) (IND(I),DTEMP(I),I=1,3)                               NUDA0099
      IF(IND(1).EQ.0.AND.DTEMP(1).EQ.0.) GO TO 10                       NUDA0100
      DO 48 I=1,3                                                       NUDA0101
      IF(IND(I).NE.0) GO TO 41                                          NUDA0102
      IF(DTEMP(I).EQ.0.) GO TO 50                                       NUDA0103
      IND(I)=LOC+1                                                      NUDA0104
   41 LOC=IABS(IND(I))                                                  NUDA0105
      KOUNT=KOUNT+1                                                     NUDA0106
      GO TO (42,44,45,46,47),KK                                         NUDA0107
   42 FTR(LOC)=DTEMP(I)                                                 NUDA0108
   43 IF(IND(I).LT.0) GO TO 49                                          NUDA0109
      GO TO 48                                                          NUDA0110
   44 C1(LOC)=DTEMP(I)                                                  NUDA0111
      GO TO 43                                                          NUDA0112
   45 C2(LOC)=DTEMP(I)                                                  NUDA0113
      GO TO 43                                                          NUDA0114
   46 BIAS(LOC)=DTEMP(I)                                                NUDA0115
      GO TO 43                                                          NUDA0116
   47 RETR(LOC)=DTEMP(I)                                                NUDA0117
      GO TO 43                                                          NUDA0118
   48 CONTINUE                                                          NUDA0119
      GO TO 50                                                          NUDA0120
   49 K=0                                                               NUDA0121
   50 WRITE(6,605) (IND(I),DTEMP(I),I=1,KOUNT)                          NUDA0122
      IF(K.EQ.0) GO TO 10                                               NUDA0123
      GO TO 401                                                         NUDA0124
  999 RETURN                                                            NUDA0125
      END                                                               NUDA0126
```

Subroutine: NUTAIT

Purpose: To compute EN, the transformation from mean equator, equinox to true equator, equinox; and to compute ØM,CR,DT and EPSIL required by MNA.

Calling Sequence: CALL NUTAIT(TIME,ØM,CR,DT,EN,EPSIL)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TIME			days	Total number of days from 1950 Jan 0.0
Ø	ØM			rad	Argument of moon's descending node.
Ø	CR			rad	Mean longitude of moon.
Ø	DT			rad	Nutation in longitude of equinox.
Ø	EN				Nutation matrix.
Ø	EPSIL			rad	True obliquity.

Common storages used: None

Subroutines required: None

```
$IBFTC MC13NU  XR3,LIST
      SUBROUTINE NUTAIT (TIME,OM,CR,DT,EN,EPSIL)                        NVTA0000
CMC13NU  SUBROUTINE NUTAIT                                              NVTA001
      DIMENSION EN(3,3)                                                 NVTA0020
      DATA DTR/.0174532925/                                             NVTA003
      D = TIME                                                          NVTA0040
      T = D/36525.                                                      NVTA0050
      T2 = T*T                                                          NVTA0060
      T3 = T2*T                                                         NVTA0070
      OM = 12.112790-.052953922*D+.0020795*T+.002081*T2+.000002*T3      NVTA0080
      CR = 64.375452+13.176397*D-.001131575*T-.00113015*T2+.0000019*T3  NVTA0090
      GP = 208.84399+.11140408*D-.010334*T-.010343*T2-.000012*T3        NVTA0100
      VL = 280.08121+.98564734*D+.000303*(T+T2)                         NVTA0110
      G = 282.08053+.0000470684*D+.00045525*T+.0004575*T2+.000003*T3    NVTA0120
      OM = OM*DTR                                                       NVTA013
      CR = CR*DTR                                                       NVTA014
      GP = GP*DTR                                                       NVTA015
      VL = VL*DTR                                                       NVTA016
      G = G*DTR                                                         NVTA017
      DE = 25.5844*COS (OM)-.2511*COS (2.*OM)+1.5336*COS (2.*VL)        NVTA0180
     1 +.0666*COS (3.*VL-G)-.0258*COS (VL+G)-.0183*COS (2.*VL-OM)       NVTA0190
     2 -.0067*COS (2.*GP-OM)                                            NVTA0200
      DD = .2456*COS (2.*CR)+.0508*COS (2.*CR-OM)+.0369*COS (3.*CR-GP)  NVTA0210
     1 -.0139*COS (CR+GP)-.0086*COS (CR-GP+OM)+.0083*COS (CR-GP-OM)     NVTA0220
     2 +.0061*COS (3.*CR+GP-2.*VL)+.0064*COS (3.*CR-GP-OM)              NVTA0230
      DT = -(47.8927+.0482*T)*SIN (OM)+.58*SIN (2.*OM)                  NVTA0240
     1 -3.5361*SIN (2.*VL)-.1378*SIN (3.*VL-G)+.0594*SIN (VL+G)         NVTA0250
     2 +.0344*SIN (2.*VL-OM)+.0125*SIN (2.*GP-OM)+.35*SIN (VL-G)        NVTA0260
     3 +.0125*SIN (2.*VL-2.*GP)                                         NVTA0270
      DS = -.5658*SIN (2.*CR)-.095*SIN (2.*CR-OM)-.0725*SIN (3.*CR-GP)  NVTA0280
     1 +.0317*SIN (CR+GP)+.0161*SIN (CR-GP+OM)+.0158*SIN (CR-GP-OM)     NVTA0290
     2 -.0144*SIN (3.*CR+GP-2.*VL)-.0122*SIN (3.*CR-GP-OM)              NVTA0300
     3 +.1875*SIN (CR-GP)+.0078*SIN (2.*CR-2.*GP)                       NVTA0310
     4 +.0414*SIN (CR+GP-2.*VL)+.0167*SIN (2.*CR-2.*VL)                 NVTA0320
     5 -.0089*SIN (4.*CR-2.*VL)                                         NVTA0330
      DE = DTR*.0001*(DE+DD)                                            NVTA034
      DT=DTR*.0001*(DT+DS)                                              NVTA035
      EB = 23.4457587-.01309404*T-.00000088*T2+.0000005*T3              NVTA0360
      EB = EB*DTR                                                       NVTA037
      EPSIL = EB+DE                                                     NVTA0380
      EN(1,1) = 1.                                                      NVTA0390
      EN(1,2) = -DT*COS (EB)                                            NVTA0400
      EN(1,3) = -DT*SIN (EB)                                            NVTA0410
      EN(2,1) = -EN(1,2)                                                NVTA0420
      EN(2,2) = 1.                                                      NVTA0430
      EN(2,3) = -DE                                                     NVTA0440
      EN(3,1) = -EN(1,3)                                                NVTA0450
      EN(3,2) = DE                                                      NVTA0460
      EN(3,3) = 1.                                                      NVTA0470
      RETURN                                                            NVTA0480
      END
```

Subroutine: .ØPTW.

Purpose: To change the option words ØPTWD1 and ØPTWD2 to allow the optional return from FXEM when an illegal character is read on a raw data tape or when a bad record causes a permanent read redundancy.

Calling Sequence: None.

Common storages used: /.ØPTW./

Subroutines required: None

Discussion

This is a MAP subroutine by absolute necessity for use of the IBM 7094 with the IBSYS Operating System, Version 13. For other systems, an equivalent subroutine must be supplied by the User.

When an attempt is made to read illegal characters from a raw data tape, the usual response is a program stop. In the IBSYS Operating System, the object program error procedure is controlled by subroutine FXEM, which normally terminates execution and passes control to subroutine .LXCON when the above stated error condition (or any of a number of other error conditions) exists. However, some error conditions allow execution to be resumed by using optional exits from FXEM. The use of these optional exits is controlled by bits in words ØPTWD1, ØPTWD2, and ØPTWD3. The optional exit for illegal characters is to replace the illegal characters with zeros and proceed. This subroutine alters bits 32 and 33 of ØPTWD1 from the delivered version to allow this optional response.

On some raw data tapes an occasional bad record causing a permanent read redundancy has been found. This also normally causes a program stop. The optional return from FXEM for this condition is to accept the bad record as read the hundredth time and proceed. This subroutine alters bit 35 of ØPTWD1 and bit 7 of ØPTWD2 to allow this optional response.

For futher details see IBM Form C28-6389-2, IBM 7090/7094 IBSYS Operating System, Version 13, IBJOB Processor, page 107, "Fortran IV Utility Library", and page 158, "Subroutine Library Error Messages, FORTRAN IV Subroutine Messages".

```
$IBMAP MC138F
       ENTRY   .OPTW.                                                  OPTW0001
       REM     WHEN AN ATTEMPT IS MADE TO READ ILLEGAL CHARACTERS      OPTW0002
       REM     FROM RAW DATA TAPES, THE USUAL RESPONSE IS A PROGRAM    OPTW0003
       REM     STOP.  THIS SUBROUTINE PERMITS THE OPTIONAL RESPONSE,   OPTW0004
       REM     WHICH IS TO REPLACE THE ILLEGAL CHARACTER WITH A ZERO.  OPTW0005
       REM     THIS IS DONE BY ADDING ONES TO BITS 32 AND 33 OF THE    OPTW0006
       REM     OPTION WORD OPTWD1 USED BY FXEM.                        OPTW0007
       REM     A BAD RECORD MAY ALSO CAUSE A PERMANENT READ REDUNDANCY. OPTW0008
       REM     THE OPTIONAL RETURN IS TO ACCEPT THE RECORD AS READ THE  OPTW0009
       REM     HUNDREDTH TIME.  THIS IS DONE BY ADDING ONES TO BIT 35  OPTW0010
       REM     OF OPTWD1 AND BIT 7 OF OPTWD2.                          OPTW0011
       REM     REF.  IBM FORM C28-6389-2, IBM 7090/7094 IBSYS OPERATING OPTW0012
       REM     SYSTEM, VERSION 13, IBJOB PROCESSOR, PAGES 107,         OPTW0013
       REM     FORTRAN IV UTILITY LIBRARY, AND 158, FORTRAN IV         OPTW0014
       REM     SUBROUTINE MESSAGES.                                    OPTW0015
.OPTW. OCT     377777777755  ADD ONES TO BITS 32, 33, AND 35 OF OPTWD1 OPTW0016
       OCT     002000000000  ADD ONE TO BIT 7 OF               OPTWD2  OPTW0017
       OCT     376000000000  LEAVE AS DELIVERED                OPTWD3  OPTW0018
       END
```

Subroutine: ØRB

Purpose: To compute and print out orbital elements.

Calling Sequence: CALL ØRB (X,DX,U)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	(3)		km	Position
I	DX	(3)		km/sec	Velocity
I	U			km^3/sec^2	Gravitational constant

Common storages used: 22 cells

Subroutines required: CRØSS, FNØRM, DØT, ARKTNS, TCØNIC

```
$IBFTC MC13OR  XR3,M94,NODD,LIST
      SUBROUTINE ORB (X,DX,U)                                           ORB 0001
C     COMPUTES AND WRITES ORBITAL ELEMENTS                              ORB 0002
      COMMON     C(12),D(10)                                            ORB 0003
      DIMENSION  X(3) ,DX(3) ,B(3)                                      ORB 0004
      EQUIVALENCE(D( 1),A   ) ,(C( 7),C3  ) ,(C( 9),PRV) ,(C(12),TPR)   ORB 0005
     1          ,(D( 4),B   ) ,(C( 2),ECC) ,(D( 9),R   ) ,(D(10),UM )   ORB 0006
     2          ,(D( 3),B1  ) ,(C( 3),OIN) ,(C( 6),RCA) ,(C( 7),V2 )    ORB 0007
     3          ,(D( 2),B2  ) ,(C( 4),OMG) ,(C( 1),SMA) ,(C(11),VIM)    ORB 0008
     4          ,(C( 5),BEP) ,(C(10),P   ) ,(C( 8),THE)                 ORB 0009
      DATA   R2D/57.2957795/                                            ORB 0010
  601 FORMAT(5H  SMAE15.8,5H  ECCE15.8,5H  INCE15.8,5H  LANE15.8,5H  APFORB 0011
     1E15.8,5H  RCAE15.8/5H   C3E15.8,5H THETE15.8,5H PERVE15.8,5H  SLREORB 0012
     215.8,5H IMPVE15.8,5H TPERE15.8)                                   ORB 0013
    1 UM = U                                                            ORB 0014
      CALL CROSS (X,DX,B)                                               ORB 0015
      R  = FNORM (X)                                                    ORB 0016
      V2 = DOT(DX,DX)                                                   ORB 0017
      A  = DOT(X,DX)/UM                                                 ORB 0018
      B1 = B(1)*B(1)+B(2)*B(2)                                          ORB 0019
      B2 = B1+B(3)*B(3)                                                 ORB 0020
      P  = B2/UM                                                        ORB 0021
      C3 = V2-2.*UM/R                                                   ORB 0022
      SMA=-UM/C3                                                        ORB 0023
      ECC= SQRT(1.+C3*P/UM)                                             ORB 0024
      B1 = SQRT(B1)                                                     ORB 0025
      B2 = SQRT(B2)                                                     ORB 0026
      A  = A*B2                                                         ORB 0027
      RCA= P/(1.+ECC)                                                   ORB 0028
      PRV=SQRT(C3+2.*UM/RCA)                                            ORB 0029
      VIM= PRV-SQRT(UM/RCA)                                             ORB 0030
      C(20) = X(2)*B(1)-X(1)*B(2)                                       ORB 0031
      C(19) = -B(2)                                                     ORB 0032
      C(17) = X(3)*B2                                                   ORB 0033
      DO 2 I=1,3                                                        ORB 0034
    2 C(I+2) = R2D*ARKTNS(360,C(I+17),C(I+14))                          ORB 0035
      THE = ARKTNS(180,P-R,A)                                           ORB 0036
      CALL TCONIC(UM,ECC,SMA,P,THE,TPR,A)                               ORB 0037
      THE = THE*R2D                                                     ORB 0038
      TPR = TPR/86400.                                                  ORB 0039
      BEP = BEP-THE                                                     ORB 0040
      IF(BEP.LT.0.)BEP=BEP+360.                                         ORB 0041
      IF(BEP.GE.360.)BEP=BEP-360.                                       ORB 0042
      WRITE (6,601) C                                                   ORB 0043
      RETURN                                                            ORB 0044
      END
```

Subroutine: ØUTTRJ

Purpose: To control output on the system output tape and the accumulation of interpolation coefficients during trajectory integration for the DCP estimation/propagation link.

Calling Sequence: CALL ØUTTRJ

Common storages used: //40 cells,/DCPCØM/,/DFMCØM/,/DQDCØM/,/ESTCØM/,/SBFCØM/,/TRJCØM/

Subroutines required: DATØUP, DEHA, DMVTRN, DNØRM, DPFMRS, GTR2BD, ØUTXPD, STEPDP, STEPDT

Discussion

ØUTTRJ is called by DEQD to output the results of trajectory integration. It controls output of the vehicle state on the system output tape, and accumulates state and transition matrix interpolation tables for use by the estimation link (ESTMAT).

The output on the system tape is accomplished by ØUTXPD (q.v.) and is controlled by the keys NPRØUT and NDPØUT. The values of the keys and their effect are:

NPRØUT	= 1 :	output vehicle state during propagation processes only.
	= 2 :	output during propagation and during integration of the nominal trajectory up to the start of the interpolation table accumulation.
	≥ 3 :	output during any integration, including accumulation of interpolation tables.
NDPØUT	= 1 :	output state in single precision (8 significant figures).
	2 :	output state in double precision (16 figures).

The accumulation of interpolation tables starts when the next integration interval will cause time (XDEQ(1)) to exceed the value stored in TBF. The interval for table storage (HBF) is set not to exceed the upper limit for integration interval size.

At the initial point, t_o, of the interpolation interval, TBF and TFF are replaced by t_o and t_o + 7HBF, respectively, and the vector RBF is loaded as described below. At each point, t_o + nHBF, n = 0, 1, ... ,7, the n +1st column of VBF is loaded. When all 8 columns have been loaded, the columns of VBF are converted to interpolation coefficients (see Reference 1, Appendix D), and NDEQ(8) is set to stop the integration.

Let

$R, \dot{R}$ = Vehicle position and velocity with respect to Earth

$\begin{bmatrix} \varphi_1 & \varphi_2 \\ \dot{\varphi}_1 & \dot{\varphi}_2 \end{bmatrix}$ = (each 3X3) state transition matrix

$\begin{bmatrix} \varphi_u \\ \dot{\varphi}_u \end{bmatrix}$ = (each 3XNEMPS) sensitivities of vehicle state to equation of motion parameters

$\delta\Psi, \delta\varepsilon$ = nutations in longitude and obliquity

Δt = transmission delay, from vehicle to receiving station

$R_E, \dot{R}_E$ = Position and velocity of Earth with respect to central body.

The components of RBF and the (n+1)st column of VBF are

RBF	=	R	(3 components)
		φ_1, φ_2	(18 components)
		φ_u	3XNEMPS components
		Not used	3 components
		R_E	3 components

at time t_o and

$$VBF(\quad,n+1) = \begin{matrix} \dot{R} \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \varphi_u \\ \delta_\psi \\ \delta_\varepsilon \\ \Delta t \end{matrix}$$

at time t_o + nHBF.

```
$IBFTC MC1327  XR3,M94,NODD,LIST
      SUBROUTINE OUTTRJ
C     CONTROLS OUTPUT FOR THE PROPAGATION/ESTIMATION LINK
C     ACCUMULATES INTERPOLATION TABLES
C
      DOUBLE PRECISION   DCOS ,DSIN
C
      COMMON             /DCPCOM/CDCP(900)
      DOUBLE PRECISION   SN(13,20) ,SPDLT
      EQUIVALENCE              (CDCP(111),IERR   ) ,(CDCP(676),NPROUT)
     1                         ,(CDCP(758),INDSTA) ,(CDCP(780),NTBL  )
     2     ,(CDCP(112),ETAPE ) ,(CDCP(686),NDPOUT) ,(CDCP(143),SN    )
     3     ,(CDCP(777),ICB   ) ,(CDCP(700),NPEND ) ,(CDCP( 15),SPDLT )
C
      COMMON             /ESTCOM/CEST(804)
      DOUBLE PRECISION   DELDAN(2) ,SE(14,20)
      EQUIVALENCE        (CEST( 97),DELDAN) ,(CEST(245),SE     )
     1                  ,(CEST( 61),EFEDAN)
C
      COMMON             /SBFCOM/CSBF(12),RBF(45),VBF(45,8)
      DOUBLE PRECISION   RBF,VBF
     1                  ,HBF,TBF,TFF
      EQUIVALENCE              (CSBF( 1),KBF    ) ,(CSBF( 9),TBF    )
     1     ,(CSBF( 7),HBF    ) ,(CSBF( 2),NTP    ) ,(CSBF(11),TFF    )
C
      COMMON             /TRJCOM/CTRJ(246)
      DOUBLE PRECISION   ETFMS ,RC(3) ,RCONIC(3) ,PCONIC(6,6)
     1                  ,UTIMV ,VC(3) ,VCONIC(3)
      EQUIVALENCE              (CTRJ(151),RC     ) ,(CTRJ(163),RCONIC)
     1     ,(CTRJ(  1),ETFMS ) ,(CTRJ(157),VC     ) ,(CTRJ(169),VCONIC)
     2     ,(CTRJ(  3),UTIMV ) ,(CTRJ( 23),TC2D   ) ,(CTRJ(175),PCONIC)
C
      COMMON             /DFMCOM/IFM(14),RFM(6,12)
     1                         ,DFM (4),BFM (577),SNT(2,102)
      DOUBLE PRECISION   BFM,DFM,RFM
C
      COMMON             /DQDCOM/NDEQ(10),CDEQ(10),XDEQ (4)
     1                         ,ADEQ(44),RDEQ(44),VDEQ(44),FDEQ(44,10)
      DOUBLE PRECISION   ADEQ,FDEQ,RDEQ,VDEQ,XDEQ
      DOUBLE PRECISION   ETIMV
      EQUIVALENCE        (XDEQ,ETIMV)
C
      COMMON             D(20)
      DOUBLE PRECISION   D
C
      DOUBLE PRECISION   A(8,8)
      DATA A/ 2520.D0,   -6534.D0,   13132.D0,  -20307.D0
     1               ,   23520.D0,  -19320.D0,   10080.D0,   -2520.D0
     2       ,    0.D0,   17640.D0,  -56196.D0,  107184.D0
     2               , -139860.D0,  123900.D0,  -68040.D0,   17640.D0
     3       ,    0.D0,  -26460.D0,  110754.D0, -247527.D0
     3               ,  357840.D0, -340200.D0,  196560.D0,  -52920.D0
     4       ,    0.D0,   29400.D0, -132860.D0,  326760.D0
     4               , -511980.D0,  518700.D0, -315000.D0,   88200.D0
     5       ,    0.D0,  -22050.D0,  103320.D0, -267225.D0
     5               ,  443520.D0, -474600.D0,  302400.D0,  -88200.D0
     6       ,    0.D0,   10584.D0,  -50652.D0,  135072.D0
     6               , -233100.D0,  260820.D0, -173880.D0,   52920.D0
     7       ,    0.D0,   -2940.D0,   14266.D0,  -38829.D0
     7               ,   68880.D0,  -79800.D0,   55440.D0,  -17640.D0
     8       ,    0.D0,     360.D0,   -1764.D0,    4872.D0
     8               ,   -8820.D0,   10500.D0,   -7560.D0,    2520.D0 /
C
  601 FORMAT(9H0ETIMV = ,D24.16,15H SEC,  UTIMV = ,D24.16,4H SEC)
C
C     POSITION AND VELOCITY
C
    1 IF (ETFMS.EQ.ETIMV) GO TO 2
C     PLANETARY POSITIONS AND VELOCITIES
      CALL DPFMRS (ETIMV,EFEDAN,ICB,IERR,ETAPE)
      IF (IERR.NE.0)       GO TO 998
      ETFMS = ETIMV
      UTIMV = ETIMV-DELDAN(1)-DELDAN(2)*ETIMV
C     REFERENCE CONIC
      CALL STEPDT (ETIMV,RCONIC)
```

```
      CALL STEPDP (PCONIC)
C     CENTRAL BODY ORIENTATION
    2 CALL GTR2BD (ICB,DFM,1)
C     POSITION AND VELOCITY RELATIVE TO CENTRAL BODY
      DO 3 I=1,3
      RC(I) = RDEQ(I)+RCONIC(I)
      VC(I) = VDEQ(I)+VCONIC(I)
    3 CONTINUE
      GO TO ( 5, 4,20) ,NTP
    4 S = TBF-ETIMV
      S = S/CDEQ(1)
      IF (CDEQ(1).LT.0.) S = S+1.
      IF (S.GE.2.)        GO TO 70
      CDEQ(4) = CDEQ(1)
      IF (S.GE.1.)        GO TO 70
      GO TO 998
    5 IF (NDEQ(8).EQ.0)   GO TO 70
C
C     INITIALIZE BUFFER
C
   20 IF (KBF.NE.0)       GO TO 30
      HBF = CDEQ(1)*FLOAT(NTBL)
      TBF = ETIMV
      TFF = ETIMV+7.D0*HBF
      KTBL = 0
      L = NPEND
      DO 21 I=1,3
      RBF(I) = RC(I)-RFM(I,3)
      L = L+1
   21 CONTINUE
      L = 4
      DO 22 J=1,6
      DO 22 I=1,3
      RBF(L) = RDEQ(L)+PCONIC(I,J)
   22 L = L+1
      IF (NPEND.EQ.22) GO TO 30
      DO 23 I=23,NPEND
   23 RBF(I-1) = RDEQ(I-1)
C
C     STORE DERIVATIVES
C
   30 KTBL = KTBL-1
      IF (KTBL.GT.0)    GO TO 70
      KTBL = NTBL
      KBF = KBF+1
      L = NPEND
      DO 31 I=1,3
      D(I) = RC(I)-RFM(I,3)
      VBF(I,KBF) = VC(I)-RFM(I+3,3)
      L = L+1
   31 CONTINUE
      L = 4
      DO 32 J=1,6
      DO 32 I=1,3
      VBF(L,KBF) = VDEQ(L)+PCONIC(I+3,J)
   32 L = L+1
      IF (NPEND.EQ.22) GO TO 34
      DO 33 I=23,NPEND
   33 VBF(I-1,KBF) = VDEQ(I-1)
   34 CONTINUE
      VBF(NPEND  ,KBF) = DFM(1)
      VBF(NPEND+1,KBF) = DFM(2)
      D(7) = UTIMV+VBF(NPEND+2,KBF)
      CALL DEHA (D(7),DFM(4),D(8),D(9))
      D(7) = DSIN(D(8))
      D(8) = DCOS(D(8))
      D(4) = SN(2,INDSTA)*D(8)+SN(3,INDSTA)*D(7)
      D(5) = SN(3,INDSTA)*D(8)-SN(2,INDSTA)*D(7)
      D(6) = SN(4,INDSTA)
      CALL DMVTRN (TC2D,D(4),D(7),2,1)
      DO 35 I=1,3
   35 D(I) = D(I)-D(I+6)
      VBF(NPEND+2,KBF) = DNORM(D)/(SPDLT+SE(6,INDSTA))
      IF (KBF.EQ.8)    GO TO 50
      VBF(NPEND+2,KBF+1) = VBF(NPEND+2,KBF)
```

```
      GO TO 70
C
C     CONVERT TO INTERPOLATION COEFFICIENTS
C
   50 NDELT = NPEND+2
      DO 52 I=1,NDELT
      DO 51 J=1,8
      D(J) = 0.D0
      DO 51 K=1,8
   51 D(J) = D(J)+A(J,K)*VBF(I,K)
      DO 52 J=1,8
   52 VBF(I,J) = D(J)/A(1,1)
C
C     CONVERT TIME SCALE FOR TRANSMISSION DELAY
C
   60 D( 9) =-(1.D0+VBF(NDELT,2)/HBF)
      D(10) =-VBF(NDELT,1)
      D(11) = 0.D0
      D(13) = 1.D0
      DO 64 I=1,8
      DO 63 J=1,2
      D( I) = D(10)-D(11)
      D(14) = D(10)/HBF
      D(12) = 2.D0
      DO 61 K=1,6
      D(K+14) = D(K+13)*D(14)/D(12)
   61 D(12) = D(12)+1.D0
      DO 62 K=1,8
   62 D( I) = D(I)+VBF(NDELT,K)*D(K+12)
   63 D(10) = D(10)+D(I)/D(9)
      D( I) = D(11)-D(10)
      D(10) = D(10)+HBF
   64 D(11) = D(11)+HBF
      DO 66 I=1,8
      VBF(NDELT,I) = 0.D0
      DO 65 J=1,8
   65 VBF(NDELT,I) = VBF(NDELT,I)+A(I,J)*D(J)
   66 VBF(NDELT,I) = VBF(NDELT,I)/A(1,1)
      NDEQ(8) = 3
C
C     RUNNING OUTPUT
C
   70 IF (NPROUT-NTP+1) 999,71,72
   71 IF (KBF.NE.1.OR.KTBL.NE.NTBL) GO TO 999
   72 WRITE (6,601) ETIMV,UTIMV
      CALL DATOUP (ETIMV,D(11),0)
      CALL OUTXPD (NDPOUT)
      GO TO 999
C
  998 NDEQ(8) = 3
  999 RETURN
      END
```

Subroutine: ØUTTR3

Purpose: To control output on the system output tape and the accumulation of interpolation coefficients during trajectory integration for the DCP residual link.

Calling Sequence: CALL ØUTTR3

Common Storages Used: //40 cells,/DCPCØM/,/DFMCØM/,/DQ3CØM/,/ESRCØM/, /SB3CØM/,/TRJCØM/

Subroutines Required: DATØUP, DEHA, DMVTRN, DNØRM, DPFMRS, GTR2BD, ØUTXPD, STEPDT

Method

ØUTTR3 is logically identical to ØUTTRJ, except that only the spacecraft position and velocity and the nutations are stored in SB3CØM.

```
$IBFTC MC1328  XR3,M94,NODD,LIST
      SUBROUTINE OUTTR3
C     CONTROLS OUTPUT FOR THE RESIDUAL LINK
C     ACCUMULATES INTERPOLATION TABLES
C
      DOUBLE PRECISION   DCOS ,DSIN
C
      COMMON            /DCPCOM/CDCP(900)
      DOUBLE PRECISION   SN(13,20) ,SPDLT
      EQUIVALENCE             (CDCP(111),IERR   ) ,(CDCP(677),NRSOUT)
     1                        ,(CDCP(758),INDSTA) ,(CDCP(780),NTBL  )
     2    ,(CDCP(112),ETAPE ) ,(CDCP(686),NDPOUT) ,(CDCP(143),SN    )
     3    ,(CDCP(777),ICB   )                     ,(CDCP( 15),SPDLT )
C
      COMMON            /ESRCOM/CESR(304)
      DOUBLE PRECISION   DELDAR(2) ,SER(14,2)
      EQUIVALENCE       (CESR( 97),DELDAR) ,(CESR(245),SER    )
     1                 ,(CESR( 61),EFEDAR)
C
      COMMON            /SB3COM/CSBF(12),RBF(6),VBF(6,8)
      DOUBLE PRECISION   RBF,VBF
     1                  ,HBF,TBF,TFF
      EQUIVALENCE             (CSBF( 1),KBF    ) ,(CSBF( 9),TBF    )
     1    ,(CSBF( 7),HBF    ) ,(CSBF( 2),NTP    ) ,(CSBF(11),TFF    )
C
      COMMON            /TRJCOM/CTRJ(246)
      DOUBLE PRECISION   ETFMS ,RC(3) ,RCONIC(3)
     1                  ,UTIMV ,VC(3) ,VCONIC(3)
      EQUIVALENCE             (CTRJ(151),RC     ) ,(CTRJ(163),RCONIC)
     1    ,(CTRJ(  1),ETFMS ) ,(CTRJ(157),VC     ) ,(CTRJ(169),VCONIC)
     2    ,(CTRJ(  3),UTIMV ) ,(CTRJ( 23),TC2D   )
C
      COMMON            /DFMCOM/IFM(14),RFM(6,12)
     1                        ,DFM (4),BFM (577),SNT(2,102)
      DOUBLE PRECISION   BFM,DFM,RFM
C
      COMMON            /DQ3COM/NDEQ(10),CDEQ(10),XDEQ (4)
     1                        ,ADEQ (8),RDEQ (8),VDEQ (8),FDEQ (8,10)
      DOUBLE PRECISION   ADEQ,FDEQ,RDEQ,VDEQ,XDEQ
      DOUBLE PRECISION   ETIMV
      EQUIVALENCE       (XDEQ,ETIMV)
C
      COMMON             D(20)
      DOUBLE PRECISION   D
C
      DOUBLE PRECISION   A(8,8)
      DATA A/ 2520.D0,   -6534.D0,   13132.D0,  -20307.D0
     1             ,   23520.D0,  -19320.D0,   10080.D0,   -2520.D0
     2      ,   0.D0,   17640.D0,  -56196.D0,  107184.D0
     2             , -139860.D0,  123900.D0,  -68040.D0,   17640.D0
     3      ,   0.D0,  -26460.D0,  110754.D0, -247527.D0
     3             ,  357840.D0, -340200.D0,  196560.D0,  -52920.D0
     4      ,   0.D0,   29400.D0, -132860.D0,  326760.D0
     4             , -511980.D0,  518700.D0, -315000.D0,   88200.D0
     5      ,   0.D0,  -22050.D0,  103320.D0, -267225.D0
     5             ,  443520.D0, -474600.D0,  302400.D0,  -88200.D0
     6      ,   0.D0,   10584.D0,  -50652.D0,  135072.D0
     6             , -233100.D0,  260820.D0, -173880.D0,   52920.D0
     7      ,   0.D0,   -2940.D0,   14266.D0,  -38829.D0
     7             ,   68880.D0,  -79800.D0,   55440.D0,  -17640.D0
     8      ,   0.D0,     360.D0,   -1764.D0,    4872.D0
     8             ,   -8820.D0,   10500.D0,   -7560.D0,    2520.D0 /
C
  601 FORMAT(9H0ETIMV = ,D24.16,15H SEC,  UTIMV = ,D24.16,4H SEC)
C
C     POSITION AND VELOCITY
C
    1 IF (ETFMS.EQ.ETIMV) GO TO 2
C     PLANETARY POSITIONS AND VELOCITIES
      CALL DPFMRS (ETIMV,EFEDAR,ICB,IERR,ETAPE)
      IF (IERR.NE.0)       GO TO 998
      ETFMS = ETIMV
      UTIMV = ETIMV-DELDAR(1)-DELDAR(2)*ETIMV
C     REFERENCE CONIC
      CALL STEPDT (ETIMV,RCONIC)
```

```
C     CENTRAL BODY ORIENTATION
    2 CALL GTR2BD (ICB,DFM,1)
C     POSITION AND VELOCITY RELATIVE TO CENTRAL BODY
      DO 3 I=1,3
      RC(I) = RDEQ(I)+RCONIC(I)
      VC(I) = VDEQ(I)+VCONIC(I)
    3 CONTINUE
      GO TO (70, 4,20) ,NTP
    4 S = TBF-ETIMV
      S = S/CDEQ(1)
      IF (CDEQ(1).LT.0.) S = S+1.
      IF (S.GE.2.)        GO TO 70
      CDEQ(4) = CDEQ(1)
      IF (S.GE.1.)        GO TO 70
      GO TO 998
C
C     INITIALIZE BUFFER
C
   20 IF (KBF.NE.0)       GO TO 30
      HBF = CDEQ(1)*FLOAT(NTBL)
      TBF = ETIMV
      TFF = ETIMV+7.D0*HBF
      KTBL = 0
      DO 21 I=1,3
      RBF(I) = RC(I)-RFM(I,3)
   21 CONTINUE
C
C     STORE DERIVATIVES
C
   30 KTBL = KTBL-1
      IF (KTBL.GT.0)    GO TO 70
      KTBL = NTBL
      KBF = KBF+1
      DO 31 I=1,3
      D(I) = RC(I)-RFM(I,3)
      VBF(I,KBF) = VC(I)-RFM(I+3,3)
   31 CONTINUE
      VBF(4,KBF) = DFM(1)
      VBF(5,KBF) = DFM(2)
      D(7) = UTIMV+VBF(6,KBF)
      CALL DEHA (D(7),DFM(4),D(8),D(9))
      D(7) = DSIN(D(8))
      D(8) = DCOS(D(8))
      D(4) = SN(2,INDSTA)*D(8)+SN(3,INDSTA)*D(7)
      D(5) = SN(3,INDSTA)*D(8)-SN(2,INDSTA)*D(7)
      D(6) = SN(4,INDSTA)
      CALL DMVTRN (TC2D,D(4),D(7),2,1)
      DO 35 I=1,3
   35 D(I) = D(I)-D(I+6)
      VBF(6,KBF) = DNORM(D)/(SPDLT+SER(6,1))
      IF (KBF.EQ.8)     GO TO 50
      VBF(6,KBF+1) = VBF(6,KBF)
      GO TO 70
C
C     CONVERT TO INTERPOLATION COEFFICIENTS
C
   50 DO 52 I=1,6
      DO 51 J=1,8
      D(J) = 0.D0
      DO 51 K=1,8
   51 D(J) = D(J)+A(J,K)*VBF(I,K)
      DO 52 J=1,8
   52 VBF(I,J) = D(J)/A(1,1)
C
C     CONVERT TIME SCALE FOR TRANSMISSION DELAY
C
   60 D( 9) =-(1.D0+VBF(6,2)/HBF)
      D(10) =-VBF(6,1)
      D(11) = 0.D0
      D(13) = 1.D0
      DO 64 I=1,8
      DO 63 J=1,2
      D( I) = D(10)-D(11)
      D(14) = D(10)/HBF
      D(12) = 2.D0
```

```
      DO 61 K=1,6                                                       OUT30150
      D(K+14) = D(K+13)*D(14)/D(12)                                     OUT30151
   61 D(12) = D(12)+1.D0                                                OUT30152
      DO 62 K=1,8                                                       OUT30153
   62 D( I) = D(I)+VBF(6,K)*D(K+12)                                     OUT30154
   63 D(10) = D(10)+D(I)/D(9)                                           OUT30155
      D( I) = D(11)-D(10)                                               OUT30156
      D(10) = D(10)+HBF                                                 OUT30157
   64 D(11) = D(11)+HBF                                                 OUT30158
      DO 66 I=1,8                                                       OUT30159
      VBF(6,I) = 0.D0                                                   OUT30160
      DO 65 J=1,8                                                       OUT30161
   65 VBF(6,I) = VBF(6,I)+A(I,J)*D(J)                                   OUT30162
   66 VBF(6,I) = VBF(6,I)/A(1,1)                                        OUT30163
      NDEQ(8) = 3                                                       OUT30164
C                                                                       OUT30165
C     RUNNING OUTPUT                                                    OUT30166
C                                                                       OUT30167
   70 IF (NRSOUT-NTP+1) 999,71,72                                       OUT30168
   71 IF (KBF.NE.1.OR.KTBL.NE.NTBL) GO TO 999                           OUT30169
   72 WRITE (6,601) ETIMV,UTIMV                                         OUT30170
      CALL DATCUP (ETIMV,D(11),0)                                       OUT30171
      CALL OUTXPD (NDPOUT)                                              OUT30172
      GO TO 999                                                         OUT30173
C                                                                       OUT30174
  998 NDEQ(8) = 3                                                       OUT30175
  999 RETURN                                                            OUT30176
      END
```

Subroutine: ØUTXPD

Purpose: Writes vehicle state relative to the central body and/or Earth in a variety of coordinate systems.

Calling Sequence: CALL OUTXPD (NPRØUT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NPRØUT				Output option key = 1 : output in single precision = 2 : output in double precision

Common storages used: 94 cells,/DCPCØM/,/ESTCØM/,/TRJCØM/,/DFMCØM/

Subroutines required: DEHA,DMVTRN,X2ØRBD

Usage

ØUTXPD accepts as input the data referenced in the labelled commons and writes the state on SYSØU1. The state relative to the central body (R,V) is assumed stored in RC(TRJCØM) in Earth's equator, equinox of 1950.0.

The output is controlled by NPRØUT and the array KEYØUT. The components of KEYØUT are examined in turn, and the state in output is single precision (NPRØUT=1) or in double precision (NPRØUT=2) in the coordinate system specified. The components of KEYØUT are the three-digit integers, CBA. The interpretations of the digits are:

A = 0 Cartesian coordinates
    1 Spherical coordinates
    2 Geographic coordinates
    3 Orbital elements

B = 0 Earth's equator, equinox of 1950.0
    1 Central body's equator of date
    2 Central body-fixed

C = 0 Coordinates relative to the central body ICB.
    1 Coordinates relative to Earth

```
$IBFTC MC13AK  XR3,M94,NODD,LIST
      SUBROUTINE OUTXPD (NPROUT)
C     OUTPUTS POSITION AND VELOCITY
C
      DOUBLE PRECISION   DATAN2 ,DCOS ,DSIN ,DSQRT
C
      COMMON            /DCPCOM/CDCP(900)
      DIMENSION          CBODY(8,11) ,KEYOUT(10)
      DOUBLE PRECISION   RTD
      EQUIVALENCE       (CDCP( 17),CBODY ) ,(CDCP(761),KEYOUT)
     1                  ,(CDCP(777),ICB   ) ,(CDCP(  7),RTD    )
C
      COMMON            /ESTCOM/CEST(804)
      DOUBLE PRECISION   EFEDAN(14)
      EQUIVALENCE       (CEST(61),EFEDAN)
C
      COMMON            /TRJCOM/CTRJ(246)
      DOUBLE PRECISION   RC(6),UTIMV
      EQUIVALENCE       (CTRJ(151),RC     ) ,(CTRJ( 23),TC2D   )
     1                  ,(CTRJ( 59),TB2C   ) ,(CTRJ(  3),UTIMV  )
C
      COMMON            /DFMCOM/IFM(14),RFM(6,12)
     1                         ,DFM (4),BFM (577),SNT(2,102)
      DOUBLE PRECISION   BFM,DFM,RFM
C
      COMMON             C,X,XO,XX,Y,A,B,CB
      DOUBLE PRECISION   C(3,3),X,XO(9),XX(6),Y(10),CB(10)
C
      DIMENSION          A(3),B(3)
      DOUBLE PRECISION   COBL50,SOBL50
      DIMENSION          AREF(6,5),BREF(3,4),CREF(2),KP(6),S(4)
      EQUIVALENCE       (KP(1),KCP),(KP(2),KBP),(KP(3),KOUNT)
     1                  ,(CP(4),KC ),(KP(5),KB ),(KP(6),KA )
      EQUIVALENCE       (S(1),COBL50),(S(3),SOBL50)
C
      DATA  AREF(1,1)/36H   X      Y      Z     DX     DY     DZ    /
     1     ,AREF(1,2)/36H   R    LAT    LON      V    PTH    AZM    /
     2     ,AREF(1,3)/36HALT    LAT    LON      V    PTH    AZM    /
     3     ,AREF(1,4)/36HSMA    ECC    INC    LAN    APF    THE    /
     4     ,AREF(1,5)/36HSLR    HEV    RCA                         /
      DATA  BREF(1,1)/18H    EQUATOR OF 1950/
     1     ,BREF(1,2)/18H    EQUATOR OF DATE/
     2     ,BREF(1,3)/18H            ECLIPTIC/
     3     ,BREF(1,4)/18H          BODY-FIXED/
      DATA  CREF(1)   /12H CENTERED    /
      DATA  S/O200677026177,O145771525474
     1      ,O177763136126,O144514406311/
  601 FORMAT(1X,3A6,83X,3A6)
  611 FORMAT(6(2X,A3,E15.8))
  602 FORMAT(1H )
  612 FORMAT(9X,3(4X,A3,D24.16),2X,3A6)
C
    1 CONTINUE
      DO 2 I=1,5
      KP(I) = 0
      CB(I)    = CBODY(I,ICB)
    2 CB(I+5) = CBODY(I,3)
      CB(2) = EFEDAN(ICB)
      CB(7) = EFEDAN(3)
    4 CONTINUE
      KOUNT = KOUNT+1
      KA = KEYOUT(KOUNT)
      IF (KA.LT.0) RETURN
      KA = KA+111
      KC = KA/100
      KA = KA-KC*100
      KB = KA/10
      KA = KA-KB*10
C     SET UP LABELS
   20 CONTINUE
      IF (KC.NE.KCP) GO TO 24
      IF (KB.NE.KBP) GO TO 26
      DO 22 I=1,3
      A(I) = AREF(4,5)
   22 B(I) = AREF(4,5)
```

```
      N = NPROUT+2
      GO TO 30
   24 CONTINUE
      N = NPROUT
      III = 5*KC-4
      KCP = KC
   26 KBP = KB
      DO 27 I=1,3
      A(I) = CREF(I-1)
   27 B(I) = BREF(I,KB)
      A(1) = CB(III)
C     COMPUTE STATE RELATIVE TO DESIRED BODY
   30 CONTINUE
      M = 6
      IF (KC.NE.1) GO TO 34
      DO 32 I=1,6
   32 XO(I) = RC(I)
      GO TO 38
   34 CONTINUE
      DO 36 I=1,6
   36 XO(I) = RC(I)-RFM(I,3)
   38 GO TO (40,50,60,70) ,KB
C     EQUATOR OF 1950
   40 CONTINUE
      DO 42 I=1,6
   42 XX(I) = XO(I)
      GO TO 99
C     EQUATOR OF DATE
   50 CONTINUE
      CALL DMVTRN (TC2D,XO,XX,1,2)
      IF (KB.EQ.4) GO TO 71
      GO TO 99
C     ECLIPTIC
   60 CONTINUE
      DO 62 I=1,4,3
      XX(I) = XO(I)
      X        = XO(I+1)
      XX(I+1) = X*COBL50+XO(I+2)*SOBL50
   62 XX(I+2) =-X*SOBL50+XO(I+2)*COBL50
      GO TO 99
C     BODY-FIXED
   70 CONTINUE
      IF (KC.NE.1.AND.ICB.NE.3) GO TO 50
      CALL DMVTRN (TB2C,XO,XX,2,2)
      IF (KA.EQ.4) GO TO 99
      GO TO 73
   71 CALL DEHA (UTIMV,DFM(4),X,CB(10))
      Y(1) = DSIN(X)
      Y(2) = DCOS(X)
      DO 72 I=1,4,3
      Y(3) = XX(I)
      XX(I) = Y(2)*XX(I)+Y(1)*XX(I+1)
   72 XX(I+1) = Y(2)*XX(I+1)-Y(1)*Y(3)
   73 XX(4) = XX(4)+XX(2)*CB(III+4)
      XX(5) = XX(5)-XX(1)*CB(III+4)
   99 CONTINUE
      GO TO (100,200,300,400) ,KA
C     CARTESIAN
  100 CONTINUE
      DO 102 I=1,6
  102 XO(I) = XX(I)
      GO TO 500
C     SPHERICAL
  200 CONTINUE
C     GEOGRAPHIC
  300 CONTINUE
      Y(3) = XX(1)*XX(1)+XX(2)*XX(2)
      Y(4) = XX(3)*XX(3)
      Y(6) = DSQRT(Y(3))
      X = 0.D0
      IF (Y(3).EQ.0.D0) GO TO 310
      C(1,3) =-XX(2)/Y(6)
      C(2,3) = XX(1)/Y(6)
      IF (KA.EQ.2) GO TO 308
      X = XX(3)/Y(6)
```

```
C     COEFFICIENTS OF G(X)
      Y(8) = CB(III+3)*CB(III+3)/Y(7)
      Y(9) = 1.D0-Y(8)
      Y(1) = Y(9)*Y(9)
      Y(2) =-Y(9)*Y(6)/2.D0
      Y(5) =-Y(7)*Y(3)
      Y(3) = Y(3)+Y(8)*Y(4)-Y(7)*Y(1)
      Y(4) =-Y(7)*Y(2)
      Y(7) = CB(III+2)*CB(III+2)
      X = CB(III+2)/DSQRT(1.D0+X*X/Y(8))
      Y(10) = ((4.D0*Y(1)*X+3.D0*Y(2))*X+2.D0*Y(3))*X+Y(4)
C     ITERATE FOR G(X) = 0
      DO 304 I=1,4
      Y(7) = (((Y(1)*X+Y(2))*X+Y(3))*X+Y(4))*X+Y(5)
  304 X = X-Y(7)/Y(10)
C     COMPUTE COORDINATES
      IF (X.LE.CB(III+2)*1.D-14) GO TO 310
  308 Y(1) = XX(3)/(Y(6)-Y(9)*X)
      C(3,2) = 1.D0/DSQRT(1.D0+Y(1)*Y(1))
      C(3,1) = Y(1)*C(3,2)
      Y(2) = Y(1)*Y(8)*X
      GO TO 312
  310 C(3,2) = X/XX(3)
      C(3,1) = 1.D0
      Y(2) = CB(III+3)
  312 XO(1) = DSQRT((Y(6)-X)**2+(XX(3)-Y(2))**2)
      XO(2) = RTD*DATAN2(C(3,1),C(3,2))
      XO(3) = RTD*DATAN2(XX(2),XX(1))
      C(1,1) = C(2,3)*C(3,2)
      C(1,2) =-C(2,3)*C(3,1)
      C(2,1) =-C(1,3)*C(3,2)
      C(2,2) = C(1,3)*C(3,1)
      C(3,3) = 0.D0
      CALL DMVTRN (C,XX(4),Y,2,1)
      XO(6) = 0.D0
      XO(5) = 90.D0
      Y(4) = Y(2)*Y(2)+Y(3)*Y(3)
      XO(4) = DSQRT(Y(4)+Y(1)*Y(1))
      Y(5) = DSQRT(Y(4))
      IF (Y(5).LE.1.D-14*XO(4)) GO TO 314
      XO(6) = RTD*DATAN2(Y(3),Y(2))
      XO(5) = RTD*DATAN2(Y(1),Y(5))
  314 IF (X.GE.Y(6)) XO(1) = -XO(1)
      GO TO 500
C     ORBITAL
  400 CONTINUE
      M = 9
      CALL X2ORBD (XX,XO,CB(III+1))
      DO 402 I=3,6
  402 XO(I) = RTD*XO(I)
C
  500 GO TO (501,502,511,502) ,N
  501 WRITE (6,601) A,B
  511 WRITE (6,611) (AREF(I,KA),XO(I),I=1,M)
      GO TO 4
  502 WRITE (6,602)
      WRITE (6,612) (AREF(I,KA),XO(I),I=1,3),A
     1              ,(AREF(I,KA),XO(I),I=4,6),B
      IF (M.LE.6) GO TO 4
      WRITE (6,612) (AREF(I,KA),XO(I),I=7,9)
      GO TO 4
      END
```

Subroutine: ØVRLYD

Purpose: To allow a convenient means whereby data may be entered into double-precision arrays. A blank card or a negative index terminates the read-in process.

Calling Sequence: CALL ØVRLYD(C)

Input and Output:

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	C	d( )			Array into which double-precision data are to be read.

Common storages used: None

Subroutines required: None

Use of ØVRLYD

Double-precision input consists of two numbers, an integer k and a double-precision number x. The integer k is a subscript indicating the relative location in array C in which the value x is to be stored; thus C(k) = x. The integer field width is 3, (I3; integers must be right adjusted) and the value field width is 21, (D21.16). There may be up to three such pairs per card, running from Column 1 through Column 72, (3(I3,D21.16)).

If k = 0 and x ≠ 0, x will be stored in the next successive location of C following the previous entry; if there were no previous entry, storage will start in C(1). If k and x are both zero, reading is terminated if they are the first pair on a card; otherwise the next card is processed. If k is negative, k is set positive, x is properly stored, and reading is terminated. Thus reading may be terminated by a card on which k_1 and x_1 are both zero, or by a negative value of k in any of the three integer fields.

All data read by ØVRLYD are written on the system output tape with zero or negative values of k replaced by the relative locations actually used.

```
$IBFTC MC13SY  XR3,M94,NODD,LIST
      SUBROUTINE OVRLYD(C)
C     OVERLAYS DOUBLE PRECISION ARRAYS
      DOUBLE PRECISION  C(1),      BUFF(3)
      DIMENSION         IND(3)
  501 FORMAT(3(I3,D21.16))
  601 FORMAT(53X,13HOVERLAY INPUT)
  602 FORMAT(16X,I3,D25.16,5X,I3,D25.16,5X,I3,D25.16)
      WRITE(6,601)
      K=1
      LOC=0
    1 KOUNT=0
      READ(5,501) (IND(I),BUFF(I),I=1,3)
      IF(BUFF(1).EQ.0..AND.IND(1).EQ.0) GO TO 7
      DO 4 I=1,3
      IF(IND(I).NE.0) GO TO 2
      IF(BUFF(I).EQ.0.) GO TO 6
      IND(I)=LOC+1
    2 LOC=IABS(IND(I))
      KOUNT=KOUNT+1
    3 C(LOC)=BUFF(I)
      IF(IND(I).LT.0) GO TO 5
    4 CONTINUE
      GO TO 6
    5 K=0
    6 WRITE(6,602) (IND(I),BUFF(I),I=1,KOUNT)
      IF(K.EQ.0) GO TO 7
      GO TO 1
    7 RETURN

      END
```

Subroutine: ØYAL

Purpose: To output the station, beacon data read in by LAYØ. Output is in input units rather than internal units.

Calling Sequence: CALL ØYAL,(NCH,IRA,J,VARVEC)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NCH				NCH = 1, station data =2, beacon data
I	IRA	(11)			Measurement treatment keys.
I	J				Index in IRA indicationg which measurement error to output (J = 1 for stations not allowed).
I	VARVEC	(24)			Random, bias errors in the measurement.

Common storages used: None

Subroutines required: None

Organization of OYAL

The keys input to OYAL have different meanings for station and beacon entries.

1. STATION MEASUREMENTS (NCH=1)

IRA(1) = 1 C-Band measurements
2 Goddard Range - Range Rate measurements
3 S-Band measurements
4 DSIF measurements

IRA(2) = range treatment key
(3) = doppler treatment key
(4) = angle 1 treatment key
(5) = angle 2 treatment key
(6) = northing error treatment key
(7) = easting error treatment key
(8) = down error treatment key
(9) = clock bias treatment key
(10-11)= ignored by OYAL (J=10 or 11 not allowed)

J is the index on the entry in IRA(2-9) to be interrogated.

IRA(I)= 0 quantity not measured
1 bias and random for measurements
bias only for location errors
2 random only (measurements only)

VARVEC(J+6) = standard deviation for random error
(J+10)= bias error value

2. BEACON MEASUREMENTS (NCH=2)

IRA(1) = range treatment key

(2) = range rate key

(3) = angle 1 key

(4) = angle 2 key

(5) = time bias key

J is the index of the indicator to be considered.

IRA(I) = 0 no measurement

1 random and bias errors

2 random errors only

VARVEC(1) = period of range or range rate observations

(2) = period of angular measurements

(J+2) = random error standard deviation

(J+6) = bias error value

```
$IBFTC MC130Y  NOREF,M94,NODD,XR3
CMC130Y   SUBROUTINE OYAL (=LAYO BACKWARDS FOR OUTPUT)
      SUBROUTINE OYAL(NCH,IRA,J,VARVEC)
      DIMENSION   IRA(11),       VARVEC(24),    ABC(4,20)
      DATA (ABC(1,I),I=1,12)/
     124HERROR J=1 IN OYAL           ,
     224HRANGE (MET)                 ,
     324HDOPPLER (CYCLES/SEC)        ,
     424HAZIMUTH (MR)                ,
     524HX-ANGLE (MR)                ,
     624HX-ANGLE (MR)                ,
     724HHOUR ANGLE (MR)             ,
     824HELEVATION (MR)              ,
     924HY-ANGLE (MR)                ,
     124HY-ANGLE (MR)                ,
     224HDECLINATION (MR)            ,
     324HSTATION LATITUDE (MET-N)/
      DATA (ABC(1,I),I=13,20)/
     124HSTATION LONG. (MET-EAST),
     224HSTATION ALTITUDE (MET-D),
     324HCLOCK BIAS (SEC)            ,
     424HRANGE (MET)                 ,
     524HRANGE RATE (MET/SEC)        ,
     624HANGLE 1 (MR)                ,
     724HANGLE 2 (MR)                ,
     824HTIME ERROR (SEC)            /
      IF(IRA(J)) 1,1000,1
    1 CONTINUE
      RANDOM=0.
      BIAS=0.
      OPSEC=0.
      GO TO (21,41),NCH
   21 CONTINUE
      IF((IRA(1).EQ.4.AND.J.EQ.2).OR.(IRA(1).EQ.1.AND.J.EQ.3)) GO TO 997
      IF(IRA(J).NE.2) GO TO 101
   92 Q=VARVEC(J+6)
      RANDOM=Q
      IF(J.EQ.3) GO TO 110
      RANDOM=Q*1000.
      GO TO 110
  101 CONTINUE
      Q=VARVEC(J+10)
      BIAS=Q
      IF(J.EQ.9) GO TO 110
      IF(J.GT.5) GO TO 105
      IF(J.EQ.3) GO TO 92
  102 BIAS=Q*1000.
      GO TO 92
  105 CONTINUE
      BIAS=Q*1000.
  110 CONTINUE
      NAB=J
      IF(J.GT.3) NAB=4*NAB+IRA(1)-13
      IF(J.GT.5) NAB=J+6
      GO TO 999
   41 CONTINUE
      NAB=J+15
      IF(IRA(J).EQ.2) GO TO 43
   42 Q=VARVEC(J+6)
      BIAS=Q
      IF(J.EQ.5) GO TO 999
      BIAS=Q*1000.
   43 CONTINUE
      Q=VARVEC(J+2)
      RANDOM=Q*1000.
   44 CONTINUE
      GO TO (45,45,46,46),J
   45 OPSEC=VARVEC(1)
      GO TO 999
   46 OPSEC=VARVEC(2)
  999 CONTINUE
      IF(OPSEC.EQ.0.) GO TO 51
      WRITE(6,600) (ABC(I,NAB),I=1,4),RANDOM,BIAS,OPSEC
  600 FORMAT(1H ,30X,4A6,10X,F8.5,7X,F8.4,8X,F9.2)
      GO TO 1000
```

```
   51 WRITE(6,600) (ABC(I,NAB),I=1,4),RANDOM,BIAS                    OYAL0075
      GO TO 1000                                                    OYAL0076
  997 IRA(J)=0                                                      OYAL0077
 1000 RETURN                                                        OYAL0078
      END                                                           OYAL0079
```

Subroutine: PARAB

Purpose: To fit a parabola through three points.

Calling Sequence: CALL PARAB(X,Y,A)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	(3)			X coordinates of the three points.
I	Y	(3)			Y coordinates of the three points.
Ø	A	(2)			Coefficients of the equation of the parabola $Y = A_2(X - X_2)^2 + A_1(X - X_2)+Y_2$

Common storages used: None

Subroutines required: None

```
$IBFTC MC13PB   NOREF,M94,NODD,XR3
CMC13PB     PARAB
      SUBROUTINE  PARAB( X, Y, A)
      DIMENSION X(3), Y(3),A(2)
      DX1= X(1)-X(2)
      DX2= X(3)-X(2)
      D1= Y(1) -Y(2)
      D2= Y(3)- Y(2)
      DX12=DX1* DX1
      DX22= DX2* DX 2
      DET =DX1 *DX22 - DX2* DX12
      A(1)=(DX22*D1 - DX12* D2)/ DET
      A(2)=(-DX2*D1 + DX1*D2)/ DET
      RETURN
      END
```

Subroutine: PERTD

Purpose: To compute the perturbing acceleration due to the celestial bodies other than the central body (see also PERTDP).

Calling Sequence: CALL PERTD (ICB, IBC, R, U, P, A)

I/Ø	Symbolic Name of Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ICB				Central body number
I	IBC	(11)			Include effect of body i only if IBC(i) ≠ 0
I	R	d(3)	R	km	Position relative to the central body
I	U	d(12)	μ_i	km^3/sec^2	Gravitational constant array
I	P	d(6,12)	P_i	km	Positions of the celestial bodies relative to the central body
IØ	A	d(3)	A	km/sec^2	Perturbation acceleration

Common storages used: 18 cells.
Subroutines required: DDØT

```
$IBFTC MC132P  XR3,M94,NODD,LIST
      SUBROUTINE PERTD (ICB,IBC,R,U,P,A)
C     N-BODY PERTURBATION ACCELERATION
      DOUBLE PRECISION  A(3),P(6,12),R(3),U(12)
     1                 ,C,D,Q,DDOT,DSQRT
      DIMENSION         IBC(11)
      COMMON            C(3),D(3),Q(3)
C
    1 DO 4 I=1,11
      IF (IBC(I).EQ.0.OR.I.EQ.ICB) GO TO 4
      DO 2 J=1,3
      Q(J) = P(J,I)
    2 D(J) = R(J)-Q(J)
      C(3) = DDOT(D,D)
      C(2) = DDOT(Q,Q)
      C(2) = C(3)/C(2)
      C(2) = C(2)*DSQRT(C(2))
      C(1) = C(3)*DSQRT(C(3))
      C(1) =-U(I)/C(1)
      DO 3 J=1,3
    3 A(J) = A(J)+C(1)*(D(J)+C(2)*Q(J))
    4 CONTINUE
  999 RETURN
      END
```

Subroutine: PERTDP

Purpose: To compute the n-body perturbation acceleration, excluding the central body, the gradient of the acceleration, and partial derivatives of the acceleration with respect to the physical constants.

Calling Sequence: Call PERTDP (ICB,IBC,R, U, P, A, AP, G, N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ICB				Central body number.
I	IBC	(11)			Include effect of body i only if IBC(i) $\neq$ 0. Compute $\partial A/\partial \mu$ if IBC(i) < 0.
I	R	d(3)	R	km	Position relative to the central body.
I	U	d(12)	μ_i	km^3/sec^2	Gravitational constant array.
I	P	d(6,12)	P_i	km	Positions of the celestial bodies relative to the central body.
IØ	A	d(3)	A	km/sec^2	Perturbation acceleration.
Ø	AP	d(3,11)	$\partial A/\partial \mu_i$	$1/km^2$	Partial derivatives
Ø	G	d(3,3)	$\nabla_R A$	$1/sec^2$	Gradient of the acceleration.
I	N				= 0 compute acceleration only. $\neq$ 0 compute acceleration and gradient.

Common storages used: 54 cells.
Subroutines required: DDØT

Method:

The acceleration vector, A_i, of a space vehicle relative to the central body (body ICB) due to the inverse-square attraction of central body i is

$$A_i = - \mu_i \left(\frac{\Delta_i}{\delta^3} + \frac{P_i}{p^3} \right)$$

$$\Delta_i = R - P_i$$

where

μ_i = gravitational constant for body i

P_i = position of body i relative to the central body

R = position of the vehicle relative to the central body

$p = | P_i |$

$\delta = | \Delta_i |$

The subroutine sums the A_i for all bodies for which IBC(i) $\neq$ 0 except i = IBC, and returns the sum in A. If N $\neq$ 0, the gradient of A with respect to R, $\nabla_R A$, is returned in G. It is computed from

$$\nabla_R A = \sum \nabla_R A_i$$

$$\nabla_R A_i = - \frac{\mu_i}{\delta^3} \left(I - \frac{3\Delta_i \Delta_i^T}{\delta^2} \right)$$

where the summation includes all bodies (including the central body) for which IBC(i) $\neq$ 0.

For all i $\neq$ 3, 10 for which IBC(i) < 0, the partial derivatives

$$\frac{\partial A}{\partial \mu_i} = \frac{1}{\mu_i} A_i$$

are returned in consecutive columns of AP. If IBC(10) < 0, the a.u. is assumed to vary with μ_{10} in such a way that the period of the Earth remains constant. That is,

$$\frac{\partial A}{\partial \mu_{10}} = \frac{1}{\mu_{10}} A_{10} + \frac{\partial A}{\partial (a.u.)} \frac{a.u.}{3\mu_{10}}$$

Similarly, if IBC(3) < 0, the lunar ephemeris scale factor, R_E, is assumed to vary with μ_3 in such a way thatthe lunar period is constant. That is,

$$\frac{\partial A}{\partial \mu_3} = \frac{1}{\mu_3} A_3 + \frac{\partial A}{\partial (R_E)} \frac{R_E}{3\mu_3}$$

The partial derivatives with respect to the scaling factors are

$$\frac{(a.u.)}{3} \frac{\partial A}{\partial (a.u.)} = \sum_{i\neq 3,11} \mu_i \{(a_i - b_i)P_i + b_i P_{12} + c_i \Delta_i\}$$

$$\frac{(R_E)}{3} \frac{\partial A}{\partial (R_E)} = \sum_{i\neq 3,11} \mu_i \left\{ \frac{1}{p^5}(P_i \cdot P_{12})P_i + b_i P_{12} - \frac{1}{\delta^5}(\Delta_i \cdot P_{12})\Delta_i \right\}$$

$$+ \mu_{11} \left\{ \left(\frac{1}{p^3} - b_{11}\right) P_{11} + \frac{1}{\delta^5}(\Delta_{11} \cdot P_{11})\Delta_{11} \right\}$$

where

$$a_i = \frac{1}{p^5}(p^2 - P_i \cdot P_{12})$$

$$b_i = \frac{1}{3}\left(\frac{1}{p^3} - \frac{1}{\delta^3}\right)$$

$$c_1 = \frac{1}{\delta^5}(\Delta_i \cdot P_i - \Delta_i \cdot P_{12})$$

```
$IBFTC MC132Q  XR3,M94,NODD,LIST
      SUBROUTINE PERTDP (ICB,IBC,R,U,P,A,AP,G,N)                        PRTP0001
C     N-BODY PERTURBATION ACCELERATION                                  PRTP0002
      DOUBLE PRECISION    A(3),AP(3,11),G(3,3),P(6,12),R(3),U(12)       PRTP0003
      DIMENSION           IBC(11)                                       PRTP0004
      COMMON              C(12),D(3),Q(3),T(3,3)                        PRTP0005
      DOUBLE PRECISION    C,D,Q,T,DDOT,DSQRT                            PRTP0006
C                                                                       PRTP0007
    1 IF (N.EQ.0) GO TO 3                                               PRTP0008
      C(12) = 0.D0                                                      PRTP0009
      DO 2 I=1,3                                                        PRTP0010
      DO 2 J=1,3                                                        PRTP0011
    2 T(I,J) = 0.D0                                                     PRTP0013
    3 L = 1                                                             PRTP0014
      L3 = 0                                                            PRTP0015
      L10 = 0                                                           PRTP0016
C                                                                       PRTP0017
   10 DO 49 M=1,11                                                      PRTP0018
      IF (IBC(M).EQ.0) GO TO 49                                         PRTP0019
      DO 11 I=1,3                                                       PRTP0020
      Q(I) = P(I,M)                                                     PRTP0021
   11 D(I) = R(I)-Q(I)                                                  PRTP0022
      C(7) = DDOT(D,D)                                                  PRTP0023
      C(8) = DDOT(Q,Q)                                                  PRTP0024
      C(5) = C(7)/C(8)                                                  PRTP0025
      C(6) = C(7)*DSQRT(C(7))                                           PRTP0026
      C(9) = C(5)*DSQRT(C(5))                                           PRTP0027
      C(1) = C(9)-1.D0                                                  PRTP0028
      C(2) =-1.D0/C(6)                                                  PRTP0029
      C(3) = U(M)/C(6)                                                  PRTP0030
      C(4) = C(3)/C(7)                                                  PRTP0031
C                                                                       PRTP0032
   20 DO 21 I=1,3                                                       PRTP0033
   21 T(I,1) = C(2)*(R(I)+C(1)*Q(I))                                    PRTP0034
      IF (IBC(M).GE.0) GO TO 23                                         PRTP0035
      DO 22 I=1,3                                                       PRTP0036
   22 AP(I,L) = T(I,1)                                                  PRTP0037
      L = L+1                                                           PRTP0038
   23 IF (M.EQ.ICB) GO TO 25                                            PRTP0039
      DO 24 I=1,3                                                       PRTP0040
   24 A(I) = A(I)+U(M)*T(I,1)                                           PRTP0041
   25 IF (N.EQ.0) GO TO 49                                              PRTP0042
C                                                                       PRTP0043
   30 C(11) = C(3)*C(1)/3.D0                                            PRTP0044
      C( 1) = C(3)*C(9)                                                 PRTP0045
      C( 2) = C(4)*DDOT(D,Q)                                            PRTP0046
      C( 6) =-C(3)*DDOT(D,P(1,12))/C(7)                                 PRTP0047
      C( 5) = C(1)*DDOT(Q,P(1,12))/C(8)                                 PRTP0048
      C( 1) = C(1)-C(11)                                                PRTP0049
      IF (M.EQ.3) GO TO 32                                              PRTP0050
      IF (M.EQ.11) GO TO 31                                             PRTP0051
      C(12) = C(12)+C(11)                                               PRTP0052
      C( 1) = C(1)-C(5)                                                 PRTP0053
      C( 2) = C(2)+C(6)                                                 PRTP0054
      IF (M.NE.10) GO TO 33                                             PRTP0055
      IF (IBC(10).GE.0) GO TO 33                                        PRTP0056
      L10 = L-1                                                         PRTP0057
      GO TO 33                                                          PRTP0058
   31 C( 5) = C(1)                                                      PRTP0059
      C( 6) = C(2)                                                      PRTP0060
      C( 1) = 0.D0                                                      PRTP0061
      C( 2) = 0.D0                                                      PRTP0062
      GO TO 33                                                          PRTP0063
   32 IF (IBC(3).GE.0) GO TO 40                                         PRTP0064
      L3 = L-1                                                          PRTP0065
      GO TO 40                                                          PRTP0066
   33 DO 34 I=1,3                                                       PRTP0067
      T(I,2) = T(I,2)+C(1)*Q(I)+C(2)*D(I)                               PRTP0068
   34 T(I,3) = T(I,3)+C(5)*Q(I)+C(6)*D(I)                               PRTP0069
C                                                                       PRTP0070
   40 DO 41 I=1,3                                                       PRTP0071
      C(5) = 3.D0*C(4)*D(I)                                             PRTP0072
      G(I,I) = G(I,I)-C(3)                                              PRTP0073
      DO 41 J=1,3                                                       PRTP0074
   41 G(I,J) = G(I,J)+C(5)*D(J)                                         PRTP0075
```

```
   49 CONTINUE                                                          PRTP0076
C                                                                       PRTP0077
   50 L=L10                                                             PRTP0078
      C(11) = U(10)                                                     PRTP0079
      DO 53 I=1,2                                                       PRTP0080
      IF (L.EQ.0) GO TO 52                                              PRTP0081
      DO 51 J=1,3                                                       PRTP0082
   51 AP(J,L) = AP(J,L)+(T(J,I+1)+C(12)*P(J,12))/C(11)                  PRTP0083
   52 L = L3                                                            PRTP0084
   53 C(11) = U(3)                                                      PRTP0085
C                                                                       PRTP0086
  999 RETURN                                                            PRTP0087
      END
```

Subroutine: PØLYFT

Purpose: Tests an arc (station pass) of edited data with walking polynomials;tags outliers. Outputs data in edited-data format on tape 12.

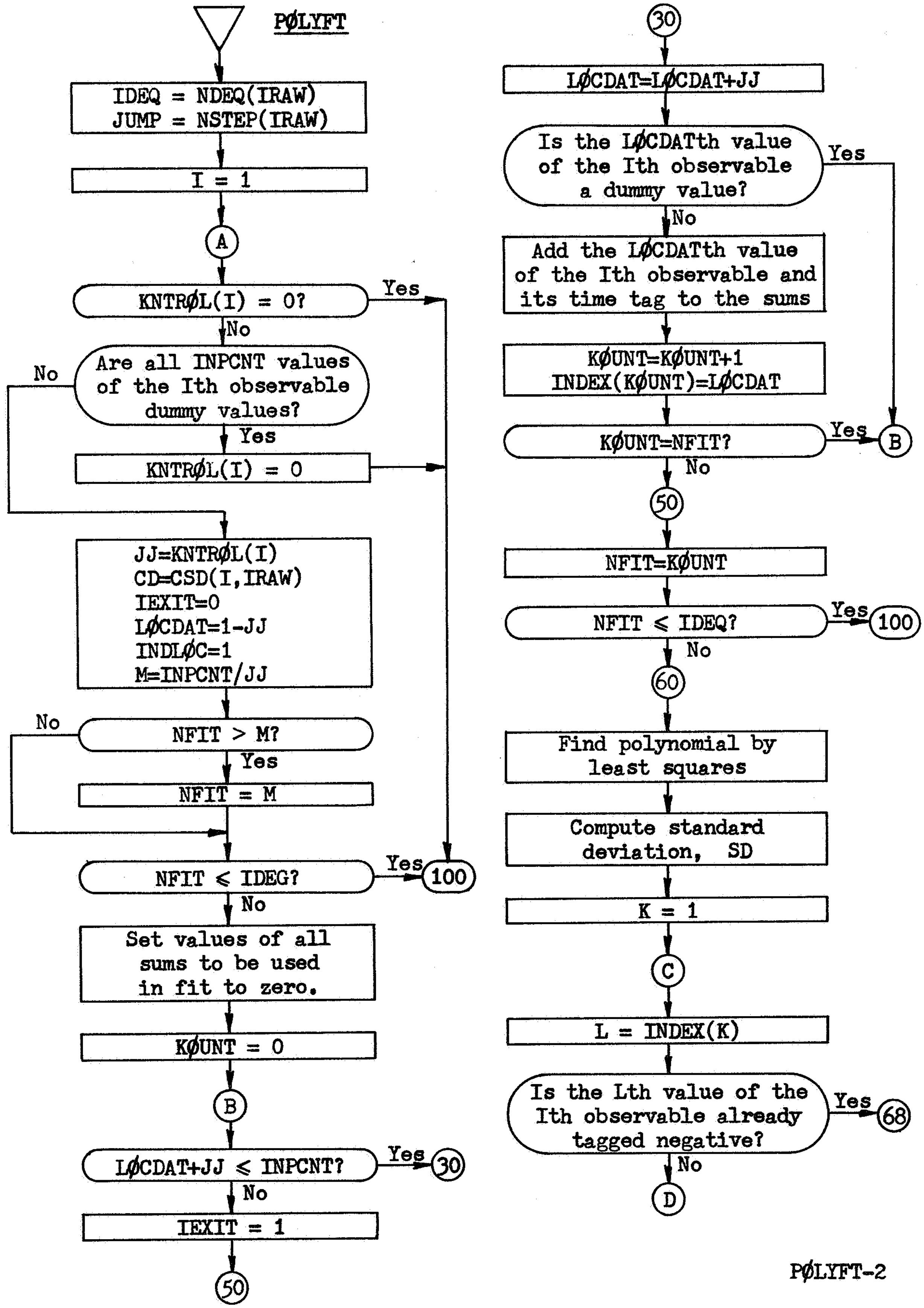
Calling Sequence: CALL PØLYFT(NPCNT)

Input and Output

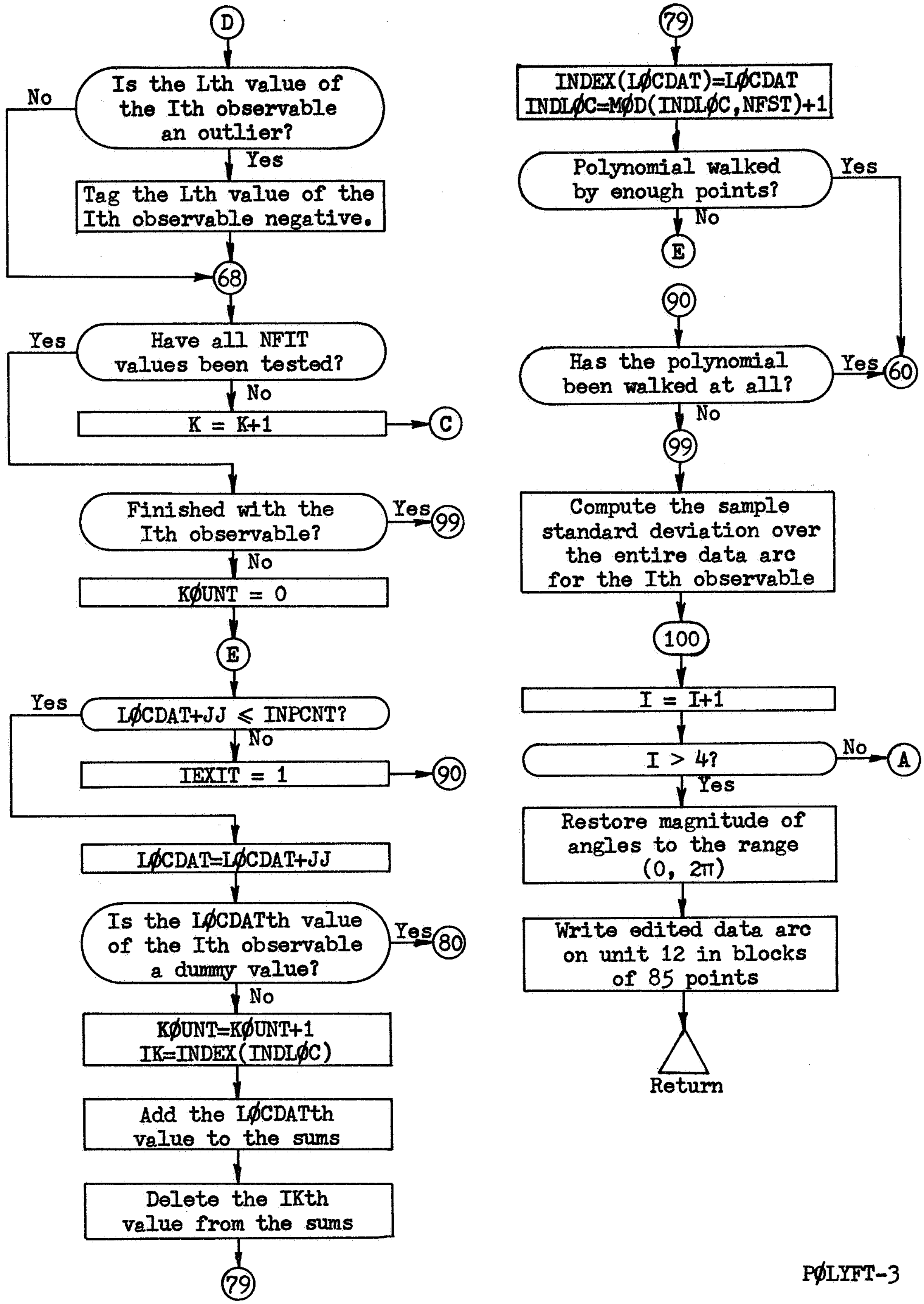
I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NPCNT				Number of time points for which data exists.

Common storages used: /TSTCØM/,/ØUTCØM/,/DATCØM/,/MESCØM/,/CRØCØM/

Subroutines required: CRØSIM


PØLYFT
IDEQ = NDEQ(IRAW)
JUMP = NSTEP(IRAW)
I = 1
A
KNTRØL(I) = 0?
Yes
No
Are all INPCNT values of the Ith observable dummy values?
No
Yes
KNTRØL(I) = 0
JJ=KNTRØL(I)
CD=CSD(I,IRAW)
IEXIT=0
LØCDAT=1-JJ
INDLØC=1
M=INPCNT/JJ
NFIT > M?
No
Yes
NFIT = M
NFIT ≤ IDEG?
Yes
100
No
Set values of all sums to be used in fit to zero.
KØUNT = 0
B
LØCDAT+JJ ≤ INPCNT?
Yes
30
No
IEXIT = 1
50
30
LØCDAT=LØCDAT+JJ
Is the LØCDATth value of the Ith observable a dummy value?
Yes
No
Add the LØCDATth value of the Ith observable and its time tag to the sums
KØUNT=KØUNT+1
INDEX(KØUNT)=LØCDAT
KØUNT=NFIT?
Yes
B
No
50
NFIT=KØUNT
NFIT ≤ IDEQ?
Yes
100
No
60
Find polynomial by least squares
Compute standard deviation, SD
K = 1
C
L = INDEX(K)
Is the Lth value of the Ith observable already tagged negative?
Yes
68
No
D

PØLYFT-2


D
Is the Lth value of the Ith observable an outlier?
No
Yes
Tag the Lth value of the Ith observable negative.
68
Have all NFIT values been tested?
Yes
No
K = K+1
C
Finished with the Ith observable?
Yes
99
No
KØUNT = 0
E
LØCDAT+JJ ≤ INPCNT?
Yes
No
IEXIT = 1
90
LØCDAT=LØCDAT+JJ
Is the LØCDATth value of the Ith observable a dummy value?
Yes
80
No
KØUNT=KØUNT+1
IK=INDEX(INDLØC)
Add the LØCDATth value to the sums
Delete the IKth value from the sums
79
79
INDEX(LØCDAT)=LØCDAT
INDLØC=MØD(INDLØC,NFST)+1
Polynomial walked by enough points?
Yes
No
E
90
Has the polynomial been walked at all?
Yes
60
No
99
Compute the sample standard deviation over the entire data arc for the Ith observable
100
I = I+1
I > 4?
No
A
Yes
Restore magnitude of angles to the range (0, 2π)
Write edited data arc on unit 12 in blocks of 85 points
Return

PØLYFT-3

```
$IBFTC MC134R  M94,NODD,XR3
CMC134R   TESTS DATA WITH WALKING POLYNOMIAL AND TAGS OUTLIERS
      SUBROUTINE POLYFT(NPCNT)
      COMMON  /TSTCOM/CTEST(400)
      DOUBLE PRECISION  FINISH(10)
      DIMENSION    NYR(10),        NPTS(10),      NSTEP(10),     NDEG(10)
     1,            CSD(4,10),      INTROL(4)
      EQUIVALENCE                (CTEST(1),NYR),        (CTEST(11),NPTS)
     1,  (CTEST(21),NSTEP),      (CTEST(31),NDEG),      (CTEST(41),CSD)
     2,  (CTEST(81),IRAW),       (CTEST(82),INTROL),    (CTEST(87),FINISH)
C
      COMMON  /OUTCOM/COUT(40)
      DOUBLE PRECISION  ONTIME,TFIRST,TLAST
      DIMENSION    NB(4),          SSD(4)
      EQUIVALENCE                (COUT(5),NRCD),        (COUT(6),NPTBLK)
     1,  (COUT(7),KONT),         (COUT(13),ONTIME),     (COUT(15),TFIRST)
     2,  (COUT(17),TLAST),       (COUT(33),NB),         (COUT(37),SSD)
C
      COMMON  /DATCOM/CDAT(2400)
      DOUBLE PRECISION TIMTAG(300)
      DIMENSION    TXTRA(300),     IQ(300),       XANG(300),     YANG(300)
     1,            RANGE(300),     DOP(300),      RDAT(300,4)
      EQUIVALENCE                (CDAT(1),TIMTAG),      (CDAT(601),TXTRA)
     1,  (CDAT(901),IQ),         (CDAT(1201),XANG),     (CDAT(1501),YANG)
     2,  (CDAT(1801),RANGE),     (CDAT(2101),DOP),      (XANG,RDAT)
C
      COMMON  /MESCOM/CMES(510)
      DIMENSION    IIQ(85),        TOUT(85),      ANG1(85),      ANG2(85)
     1,            ROUT(85),       DOUT(85)
      EQUIVALENCE                (CMES(1),IIQ),         (CMES(86),TOUT)
     1,  (CMES(171),ANG1),       (CMES(256),ANG2),      (CMES(341),ROUT)
     2,  (CMES(426),DOUT)
C
      COMMON  /CROCOM/CCRO(86)
      DOUBLE PRECISION  B(6,7)
      EQUIVALENCE                (CCRO(1),B),           (CCRO(85),KK)
     1,  (CCRO(86),ISING)
C
      DOUBLE PRECISION  SUMX(10),       SUMY(2),       SUMXY(5)
     1,  COEFF(6)
      DOUBLE PRECISION  AM,       AP,       CD,       DIFF,       SD
     1,         TXM,      TXP,      GSD,      PTS
      DIMENSION    INDEX(20),      KNTROL(4)
      DOUBLE PRECISION DSQRT
C
      DATA TEST/-.12345678E20/
      INPCNT=NPCNT
      IDEG=NDEG(IRAW)
      JUMP=NSTEP(IRAW)
C     FIT DATA COLUMNS, ONE AT A TIME
      DO 100 I=1,4
      KNTROL(I)=INTROL(I)
      SSD(I)=TEST
      IF(KNTROL(I).EQ.0) GO TO 100
      JJ=KNTROL(I)
C     TEST FOR VALID DATA.  RESET KNTROL IF NECESSARY
      DO 10 K=1,INPCNT,JJ
      IF(RDAT(K,I).NE.TEST) GO TO 20
   10 CONTINUE
      KNTROL(I)=0
      GO TO 100
   20 CONTINUE
      CD=CSD(I,IRAW)
      IEXIT=0
      LOCDAT=1-JJ
      INDLOC=1
      NFIT=NPTS(IRAW)
C     TEST WHETHER NFIT POINTS ARE AVAILABLE
      M=INPCNT/JJ
      IF(NFIT.GT.M) NFIT=M
      IF(NFIT.LE.IDEG) GO TO 100
C     ENOUGH POINTS ARE AVAILABLE IF VALID
      SUMY(1)=0.D0
      SUMY(2)=0.D0
      DO 21 J=1,IDEG
```

```
      SUMXY(J)=0.D0
      SUMX(2*J)=0.D0
   21 SUMX(2*J-1)=0.D0
C     FORM INITIAL SUMS ON FIRST NFIT POINTS
      KOUNT=0
      DO 40 K=1,INPCNT,JJ
      IF(LOCDAT+JJ.LE.INPCNT) GO TO 30
      IEXIT=1
      GO TO 50
   30 LOCDAT=LOCDAT+JJ
      IF(RDAT(LOCDAT,I).EQ.TEST) GO TO 40
      KOUNT=KOUNT+1
      INDEX(KOUNT)=LOCDAT
      AP=ABS(RDAT(LOCDAT,I))
      SUMY(1)=SUMY(1)+AP
      SUMY(2)=SUMY(2)+AP*AP
      TXP=TXTRA(LOCDAT)
      SUMXY(1)=SUMXY(1)+TXP*AP
      SUMX(1)=SUMX(1)+TXP
      SUMX(2)=SUMX(2)+TXP*TXP
      IF(IDEG.EQ.1) GO TO 39
      DO 31 J=2,IDEG
      TXP=TXP*TXTRA(LOCDAT)
      SUMXY(J)=SUMXY(J)+TXP*AP
      SUMX(2*J)=SUMX(2*J)+TXP*TXP
   31 SUMX(2*J-1)=SUMX(2*J-1)+TXP*TXTRA(LOCDAT)
   39 IF(KOUNT.EQ.NFIT) GO TO 50
   40 CONTINUE
C     TEST IF ENOUGH VALID POINTS WERE FOUND
   50 NFIT=KOUNT
      IF(NFIT.LE.IDEG) GO TO 100
C     WE HAVE AT LEAST IDEG+1 VALID POINTS.  (EXACTLY NFIT POINTS.)
      PTS=0.
      GSD=0.
   60 CONTINUE
      KK=IDEG+1
      DO 61 K=1,KK
      DO 61 L=K,KK
      M=K+L-2
      IF(M.EQ.0) GO TO 61
      B(K,L)=SUMX(M)
      B(L,K)=SUMX(M)
   61 CONTINUE
      DO 62 K=2,KK
   62 B(K,KK+1)=SUMXY(K-1)
      B(1,1)=NFIT
      B(1,KK+1)=SUMY(1)
C     SOLVE FOR POLYNOMIAL COEFFICIENTS
C     THIS SUBROUTINE USES CROUT-S METHOD
      CALL CROSIM
      IF(ISING.NE.0) GO TO 69
      DO 63 K=1,KK
   63 COEFF(K)=B(K,KK+1)
C     COMPUTE STANDARD DEVIATIONS
      AP=NFIT
      SD=SUMY(2)+COEFF(1)*(AP*COEFF(1)-2.*SUMY(1))+COEFF(2)*(COEFF(2)*SU
     1MX(2)+2.*(COEFF(1)*SUMX(1)-SUMXY(1)))
      IF(IDEG.EQ.1) GO TO 65
      SD=SD+COEFF(3)*(COEFF(3)*SUMX(4)+2.*(COEFF(2)*SUMX(3)+COEFF(1)*SUM
     1X(2)-SUMXY(2)))
      IF(IDEG.EQ.2) GO TO 65
      SD=SD+COEFF(4)*(COEFF(4)*SUMX(6)+2.*(COEFF(3)*SUMX(5)+COEFF(2)*SUM
     1X(4)+COEFF(1)*SUMX(3)-SUMXY(3)))
      IF(IDEG.EQ.3) GO TO 65
      SD=SD+COEFF(5)*(COEFF(5)*SUMX(8)+2.*(COEFF(4)*SUMX(7)+COEFF(3)*SUM
     1X(6)+COEFF(2)*SUMX(5)+COEFF(1)*SUMX(4)-SUMXY(4)))
      IF(IDEG.EQ.4) GO TO 65
      SD=SD+COEFF(6)*(COEFF(6)*SUMX(10)+2.*(COEFF(5)*SUMX(9)+COEFF(4)*SU
     1MX(8)+COEFF(3)*SUMX(7)+COEFF(2)*SUMX(6)+COEFF(1)*SUMX(5)-SUMXY(5))
     2)
   65 GSD=GSD+SD
      PTS=PTS+AP
      SD=DSQRT(DABS(SD/AP))
C     TEST FOR OUTLIERS
      DO 68 K=1,NFIT
```

```
      L=INDEX(K)
      IF(RDAT(L,I).LT.0.) GO TO 68
      DIFF=RDAT(L,I)-COEFF(1)
      TXM=1.D0
      DO 66 M=1,IDEG
      TXM=TXM*TXTRA(L)
   66 DIFF=DIFF-TXM*COEFF(M+1)
      IF(DABS(DIFF).GT.SD*CD) RDAT(L,I)=-RDAT(L,I)
   68 CONTINUE
   69 IF(IEXIT.EQ.1) GO TO 99
C     HERE WE WALK THE POLYNOMIAL
      KOUNT=0
      DO 80 K=1,INPCNT
      IF(LOCDAT+JJ.LE.INPCNT) GO TO 70
      IEXIT=1
      GO TO 90
   70 LOCDAT=LOCDAT+JJ
      IF(RDAT(LOCDAT,I).EQ.TEST) GO TO 80
      KOUNT=KOUNT+1
      IK=INDEX(INDLOC)
      AP=ABS(RDAT(LOCDAT,I))
      AM=ABS(RDAT(IK,I))
      SUMY(1)=SUMY(1)+AP-AM
      SUMY(2)=SUMY(2)+AP*AP-AM*AM
      TXP=TXTRA(LOCDAT)
      TXM=TXTRA(IK)
      SUMXY(1)=SUMXY(1)+AP*TXP-AM*TXM
      SUMX(1)=SUMX(1)+TXP-TXM
      SUMX(2)=SUMX(2)+TXP*TXP-TXM*TXM
      IF(IDEG.EQ.1) GO TO 79
      DO 71 J=2,IDEG
      TXP=TXP*TXTRA(LOCDAT)
      TXM=TXM*TXTRA(IK)
      SUMXY(J)=SUMXY(J)+TXP*AP-TXM*AM
      SUMX(2*J)=SUMX(2*J)+TXP*TXP-TXM*TXM
   71 SUMX(2*J-1)=SUMX(2*J-1)+TXP*TXTRA(LOCDAT)-TXM*TXTRA(IK)
   79 INDEX(INDLOC)=LOCDAT
      INDLOC=MOD(INDLOC,NFIT)+1
      IF(KOUNT.EQ.JUMP) GO TO 60
   80 CONTINUE
   90 IF(KOUNT.NE.0) GO TO 60
   99 SSD(I)=DSQRT(DABS(GSD/PTS))
  100 CONTINUE
C     RESTORE ANGLES TO RANGE ZERO TO TWO PI
      DO 106 I=1,2
      JJ=KNTROL(I)
      IF(JJ.EQ.0) GO TO 106
      DO 105 K=1,INPCNT,JJ
      IF(RDAT(K,I).EQ.TEST) GO TO 105
      RDAT(K,I)=RDAT(K,I)-FLOAT(IFIX(RDAT(K,I)/6.28318531))*6.28318531
  105 CONTINUE
C     FINISHED FITTING.  NOW OUTPUT DATA IN SUITABLE BLOCKS
  106 CONTINUE
      DO 110 I=1,4
  110 NB(I)=0
      KOUNT=0
      NRCD=0
      KONT=0
      I=0
      TFIRST=TIMTAG(1)
  111 I=I+1
      KOUNT=KOUNT+1
      IIQ(KOUNT)=IQ(I)
      TOUT(KOUNT)=TIMTAG(I)-ONTIME
      ANG1(KOUNT)=XANG(I)
      ANG2(KOUNT)=YANG(I)
      ROUT(KOUNT)=RANGE(I)
      DOUT(KOUNT)=DOP(I)
      IF(KOUNT.LT.85) GO TO 140
      ASSIGN 129 TO KOP
      GO TO 141
  129 TFIRST=TIMTAG(I+1)
      KONT=1
      KOUNT=0
      DO 130 L=1,4
```

```
130 NB(L)=0                                                 POLY0225
140 IF(I.LT.INPCNT) GO TO 111                               POLY0226
    IF(KOUNT.EQ.0) GO TO 999                                POLY0227
    ASSIGN 999 TO KOP                                       POLY0228
141 NRCD=NRCD+1                                             POLY0229
    NPTBLK=KOUNT                                            POLY0230
    TLAST=TIMTAG(I)                                         POLY0231
    DO 150 L=1,KOUNT                                        POLY0232
    IF(ANG1(L).LT.0.) NB(1)=NB(1)+1                         POLY0233
    IF(ANG2(L).LT.0.) NB(2)=NB(2)+1                         POLY0234
    IF(ROUT(L).LT.0.) NB(3)=NB(3)+1                         POLY0235
150 IF(DOUT(L).LT.0.) NB(4)=NB(4)+1                         POLY0236
    WRITE(12) COUT                                          POLY0237
    WRITE(12) CMES                                          POLY0238
    GO TO KOP,(129,999)                                     POLY0239
999 RETURN                                                  POLY0240
    END                                                     POLY0241
```

Subroutine: PSEUDØ

Purpose: Given a real symmetric matrix, finds the unique pseudo-inverse matrix described in the discussion. (As programmed, PSEUDØ will handle matrices with dimension no greater than 6 × 6. This matrix may be imbedded as the upper left corner of a larger matrix.)

Calling Sequence: CALL PSEUDØ(ATA,N,M,EPS).

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I/Ø	ATA	d(N,N)	A or $A^{\#}$		Input: Matrix to be inverted A Output: Pseudo-inverse matrix $A^{\#}$
I/Ø	N				Input: Dimension of A Output: Rank of A
I	M				Number of rows of matrix of which A constitutes the upper left corner.
I	EPS	d	ϵ		Significance test value for determining rank of A.

Common storages used: //254 cells

Subroutines required: DINVRT,DMPLY

References:

1. Ben-Israel, Charness: "Contributions to Theory of Generalized Inverses," JOUR. SOC. INDUST. APPL. MATH., September, 1963.
2. Philco-Ford: SECOND QUARTERLY REPORT FOR COMPUTER PROGRAMS, WDL-TR3052, 31 August 1966, Appendix B.

Discussion:

In dealing with systems of linear equations one repeatedly encounters the following problem: Given the equation

$$y = Ax$$

where the vector y and the matrix A are known, solve for the vector x.

This problem admits of three distinct possibilities:

1. The system is overdetermined in the sense that no x exists that will exactly satisfy the equation. In this case, a common practice is to accept the least-squares estimate of x given by

$$\hat{x} = (A^T A)^{-1} A^T y.$$

In subroutine PSEUDØ we are not concerned with this case.

2. The system is exactly determined in the sense that a unique solution exists and is given by

$$x = A^{-1} y.$$

3. The system is underdetermined in the sense that A is singular and an infinite set of solutions exists.

Subroutine PSEUDØ is concerned with the latter two cases with the restriction that A is symmetric (and real). When A^{-1} exists, x is unique and no decision need be made as to which solution to choose. (A is inverted by subroutine DINVRT, which uses a Gauss-Jordan reduction). When A is singular, PSEUDØ computes the PSEUDØ inverse $A^{\#}$, which, using the equation

$$\hat{x} = A^{\#} y,$$

yields the estimate vector $\hat{x}$ described by the second of the following theorems, which we state without proof.

Theorem 1:

For any real symmetric matrix A there exists a unique matrix $A^{\#}$ that satisfies the following:

a. $AA^{\#}A = A$

b. $A^{\#}AA^{\#} = A^{\#}$

c. $(AA^{\#})^T = AA^{\#}$

d. $(A^{\#}A)^T = A^{\#}A$

Corollary: $A^{\#} = A^{-1}$ if A^{-1} exists.

Theorem 2:

If $\hat{x} = A^{\#}y$, then for any x

$$|y - A\hat{x}| \leq |y - Ax|$$

and if

$$|y - A\hat{x}| = |y - Ax|$$

then

$$|\hat{x}| \leq |x|$$

Method:

Since considerable interchanging of the column vectors of a matrix occurs during the computations, some amount of confusion arising from this fact may be avoided by stating the following:

Let U be any square matrix. The product UU^T is necessarily symmetric. Moreover, let V be a matrix obtained fom U by any number of column interchanges. Then, of course, VV^T is symmetric, and also $UU^T = VV^T$. Some definitions may also be useful at this point. If a symmetric matrix A be considered as a linear transformation from one vector space to another, the entire space on which A operates is called the <u>domain space</u> of A. The space into which A transforms the domain space is called the <u>range space</u> of A. The dimension of the range space is less than or equal to the dimension of the domain space. For example, if the domain is a Euclidean three space, and A transforms every vector in the domain into a plane, then the range space has dimension 2, in which case the inverse transform A^{-1} does not exist, since many vectors in the domain may transform into the same vector in the range. (PSEUDØ was developed to deal with just this situation.) The rank of A, denoted $\rho(A)$, is defined as the dimension of the range space of A, denoted R(A). If we denote the domain of A as D(A), then the set of all vectors d such that $d \in D(A)$ and such that $Ad = 0$ is called the <u>null space</u> of A, denoted N(A). The set of all vectors perpendicular to N(A), denoted $N^{\perp}(A)$, is called the <u>perpendicular null space</u> of A. Note that the trivial vector 0 belongs to both N(A) and $N^{\perp}(A)$. With such trivial exceptions, the dimension of R(A) is equal to the dimension of $N^{\perp}(A)$. Moreover, the dimension of the domain of A is equal to the dimension of R(A) plus the dimension of N(A).

The point of all the previous discussion is that PSEUDØ begins by attempting to construct an orthonormal set of basis vectors for $N^{\perp}(A)$. If n such vectors can be found, where n is the dimension of A, then A^{-1} exists and is found directly by subroutine DINVRT. The number of such basis vectors that can be formed is, of course, equal to the number of linearly independent (column) vectors of A; i.e., the rank of A.

Construction of the set of orthonormal basis vectors proceeds as follows (ignoring for the moment certain refinements added for maximum numerical accuracy):

First we construct a second matrix B having the same dimension as A and whose columns are the corresponding columns of A after having been normalized. That is, $b_i = a_i / |a_i|$, where the subscript denotes the i-th column.

We are now ready to start construction of a third matrix U whose columns will be the desired orthonormal basis vectors. The first vector u_1 we set equal to b_1. (Assume that a_1 has norm greater than the significance criterion ϵ.) We next compute u_2 by subtracting from b_2 the projection of b_2 onto u_1. That is, $u_2 = (I - u_1 u_1^T) b_2$. We now test the norm of u_2. If the norm is significant ($|u_2| > \epsilon$) we accept u_2 as being independent of u_1 and accept it as a second basis vector after normalizing it. Otherwise, we assume that b_2 is a linear multiple of b_1 and set all the elements of u_2 to zero. We continue this general scheme, computing the k-th basis vector as

$$u_k = (I - u_{k-1} u_{k-1}^T)(I - u_{k-2} u_{k-2}^T) \ldots (I - u_1 u_1^T) b_k ,$$

normalizing and accepting u_k if the norm is significant, or setting $u_k = 0$ if it is not, until all n vectors of B have been used. We next note the number r of non-zero basis vectors in U. If $r = n$, A is invertable and we set $A^\# = A^{-1}$.

If $r < n$, we note which columns of U contain zero vectors and delete from A the corresponding rows and columns, leaving an r-dimensional matrix C which is invertable. We compute C^{-1} with subroutine DINVRT and distribute the result into an $n \times n$ matrix C^* as follows: We fill the rows of C^* corresponding to the deleted rows and columns of A with zeros. We then place the elements of C^{-1} into the remaining spaces of C^* in such a way that the natural row and column order of C^{-1} is preserved. Then

$$A^\# = UU^T C^* UU^T .$$

Note that if $r = n$ and if we were to proceed in this way, $UU^T = I$ and $C^* = A^{-1}$ so that $A^{\#} = IA^{-1} I = A^{-1}$ as expected.

We now present a tutorial example. Let

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} \quad , \quad \epsilon = n \cdot 10^{-4} = 3 \cdot 10^{-4}$$

Then

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Therefore

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & x & x \\ \frac{1}{\sqrt{2}} & x & x \\ 0 & x & x \end{pmatrix}$$

where x denotes an as yet undetermined element. We next compute u_2 as

$$u_2 = (I - u_1 u_1^T) b_2 = \left(\frac{1}{2\sqrt{2}} \; , \; -\frac{1}{2\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right)^T$$

and note that the norm is significant so we normalize u_2 and accept it so that

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & x \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & x \\ 0 & \frac{2}{\sqrt{6}} & x \end{pmatrix}$$

We now compute u_3 as

$$u_3 = (I - u_2u_2^T)(I - u_1u_1^T)b_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

which is not significant, so that

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 \\ 0 & \frac{2}{\sqrt{6}} & 0 \end{pmatrix}$$

We know that n = 3, r = 2, so we note that it was column 3 of B that was not linearly independent of columns 1 and 2. Accordingly we form C by deleting the third row and column from A to leave

$$C = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

By direct computation

$$C^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix}$$

We now construct

$$C^* = \left(\begin{array}{cc|c} 0 & 1 & 0 \\ 1 & -1 & 0 \\ \hline 0 & 0 & 0 \end{array}\right)$$

By direct computation

$$UU^T = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{pmatrix}$$

whence

$$A^{\#} = UU^T C^* UU^T = \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

As a check, we compute

$$AA^{\#} = \frac{1}{3}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{pmatrix}$$

We note in passing that $AA^{\#} = UU^T$. This will always be true.

By computation

$$AA^{\#}A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} = A \text{ , and}$$

$$A^{\#}AA^{\#} = \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} = A^{\#}\text{, as promised.}$$

We now introduce some numerical devices used to obtain the greatest numerical accuracy. We explain by example.

Again, let

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

so that

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Instead of merely taking $u_1 = b_1$, let us take as u_1 the column of B corresponding to the column of A having the largest norm. Since, in this example, the three norms are identical, let us pretend that a_1 has the largest norm so that $u_1 = b_1$

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & x & x \\ \frac{1}{\sqrt{2}} & x & x \\ 0 & x & x \end{pmatrix}$$

Let us now alter B by subtracting from the unused vectors b_2 and b_3 their projections onto u_1 so

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

As u_2 we now select from b_2 and b_3 the one whose projection into u_1 was smallest; i.e., the one that was originally most perpendicular to those vectors of U already computed. Again we must pretend, because of equality, so let us pretend that b_3 is the choice so that, after normalizing b_2 and b_3

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & x \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & x \\ 0 & -\frac{2}{\sqrt{6}} & x \end{pmatrix} \quad \text{and}$$

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{pmatrix}$$

For the u_3 we are forced to use b_2 and, having already subtracted its projection into u_1 we need merely compute

$$u_3 = (I - u_2u_2^T)b_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{which is not significant so}$$

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & 0 \\ 0 & -\frac{2}{\sqrt{6}} & 0 \end{pmatrix}$$

and we note that it was column 2 of B that was not significant. We always put the r vectors we can compute into the first r columns of U so that in forming UU^T we need only multiply an $n \times r$ matrix by an $r \times n$ matrix. (Remember that the column arrangement has no effect on the product.)

Once again

$$UU^T = \frac{1}{3}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ 1 & -1 & 2 \end{pmatrix}$$

However,

$$C = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{so that}$$

$$C^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad , \text{ and}$$

$$C^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

(Remember, the second row and column of A were deleted to form C).

and once again

$$A^\# = UU^T C^* UU^T = \frac{1}{3}\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} \quad \text{as before.}$$

```
$IBFTC MC13Z5  XR3,M94,LIST,NODD
      SUBROUTINE PSEUDO(ATA,N,M,EPS)
      DOUBLE PRECISION DSQRT
      DOUBLE PRECISION ATA(1),EPS,EP
      DIMENSION B(6,6),BB(6),C(6,6),D(6,6),NI(6),MI(6),TP(6)
      DOUBLE PRECISION B,BB,C,D,TP,TPP,EPN
      EQUIVALENCE (B(1,6),BB),(TP(2),TPP)
      COMMON SAVE(14),TP,D,B,C,NI,MI
      DATA EPN/0.D0/
C     SET UP INDICES
      NN = N
      ND = NN
      EP = FLOAT(NN)*EPS*EPS
      DO 10 I=1,NN
   10 NI(I) = I
C     SET UP ARRAYS
      JN = 1
      DO 14 J=1,NN
      K = JN
      DO 12 I=1,NN
      D(I,J) = ATA(K)
      C(I,J) = D(I,J)
   12 K = K+1
   14 JN = JN+M
      DO 28 J=1,NN
      TP = 0.D0
      DO 22 I=1,NN
   22 TP = TP+C(I,J)**2
      IF( TP.GT.EPN ) GO TO 24
      NI(J) = NI(ND)
      NI(ND) = J
      ND = ND-1
      GO TO 28
   24 BB(J) = DSQRT(TP)
      DO 26 I=1,NN
   26 C(I,J) = C(I,J)/BB(J)
   28 CONTINUE
C     SELECT FIRST BASE VECTOR
      TP = 0.D0
      DO 30 J=1,ND
      JN = NI(J)
      IF( BB(JN).LT.TP ) GO TO 30
      TP = BB(JN)
      NKN = JN
      K = J
   30 BB(JN) = 0.D0
      NK = 1
   40 CONTINUF
      NI(K) = NI(NK)
      NI(NK) = NKN
   41 CONTINUE
C     COMPUTE BASE VECTOR
      DO 42 I=1,NN
   42 B(I,NK) = C(I,NKN)
      IF( NK.LE.1 ) GO TO 62
      DO 44 J=2,NK
      JN = NI(J-1)
      DO 44 I=1,NN
   44 B(I,NK) = B(I,NK)-B(I,J-1)*C(NKN,JN)
      TP = 0.D0
      DO 46 I=1,NN
   46 TP = TP+B(I,NK)**2
C     TEST FOR SIGNIFICANCE
      IF( TP.GE.EP ) GO TO 48
      K = NI(ND)
      NI(ND) = NKN
      NI(NK) = K
      NKN = K
      ND = ND-1
      IF( ND.LT.NK ) GO TO 200
      GO TO 41
   48 TP = DSQRT(TP)
C     NORMALIZE
      DO 60 I=1,NN
   60 B(I,NK) = B(I,NK)/TP
```

```
   62 NK = NK+1
      IF( NK.GT.ND ) GO TO 200
C     COMPUTE PROJECTION OF REMAINING COLUMNS ONTO NEW BASE VECTOR
      TPP = 2.D0
      DO 66 J=NK,ND
      JN = NI(J)
      TP = 0.D0
      DO 64 I=1,NN
   64 TP = TP+B(I,NK-1)*C(I,JN)
      C(JN,NKN) = TP
      TP = BB(JN)+TP**2
C     SELECT NEXT BASE VECTOR
      IF( TPP.LT.TP ) GO TO 66
      K = J
      TPP = TP
   66 BB(JN) = TP
      NKN = NI(K)
      GO TO 40
C     ORTHONORMAL BASIS FOR PERPENDICULAR NULL SPACE COMPLETED
  200 CONTINUE
      IF( ND.LT.NN ) GO TO 202
      CALL DINVRT(ATA,NN,M)
      GO TO 999
  202 CONTINUE
      DO 204 I=1,NN
  204 MI(I) = 1
      K = NN-1
      DO 206 I=ND,K
      IN = NI(I+1)
  206 MI(IN) = 0
      JN = 1
      DO 222 J=1,NN
      IF( MI(J).EQ.0 ) GO TO 222
      IN = 1
      DO 220 I=1,NN
      IF( MI(I).EQ.0 ) GO TO 220
      C(IN,JN) = D(I,J)
      IN = IN+1
  220 CONTINUE
      JN = JN+1
  222 CONTINUE
      CALL DINVRT(C,ND,6)
      CALL DMPLY(B,B,D,NN,ND,NN,6,6,6,2)
      DO 229 I=1,NN
      JN=1
      DO 228 J=1,NN
      TP=0.D0
      IF(MI(J).EQ.0) GO TO 228
      NK=1
      DO 226 K=1,NN
      IF(MI(K).EQ.0) GO TO 226
      TP=TP+C(JN,NK)*D(K,I)
      NK=NK+1
  226 CONTINUE
      JN=JN+1
  228 B(J,I)=TP
  229 CONTINUE
      CALL DMPLY(D,B,ATA,NN,NN,NN,6,6,M,0)
  999 N = ND
      RETURN
      END
```

Subroutine: QUARTC

Purpose: To find the solutions of the quadratic equation

$$a_2X^2 + a_1X + c = d$$

Calling Sequence: CALL QUARTC(C,D,A,ANS,K)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	C		c		
I	D		d		
I	A	(2)	a_1, a_2		
Ø	ANS	(2)			Solutions ANS(1) = $(-a_1-F)/2a_2$ ANS(2) = $(-a_1+F)/2a_2$ where F is the square root of the discriminant $G = a_1^2 - 2a_2(c-d)$
Ø	K				K = - 1 if G < 0 (no solution) K = 0 if G ≥ 0

Common storages used: None

Subroutines required: None

```
$IBFTC MC13QC  NOREF,M94,NODD,XR3
CMC13QC     QUARTC
      SUBROUTINE  QUARTC(O,OD,A,DT,K)
      DIMENSION DT(2) , A(2)
      K= 0
      B = A(1)*A(1) - 4.*A(2)*(O-OD)
      IF(B) 1,1,2
1     K= -1
      GO TO  3
2     CONTINUE
      B=  SQRT(B)
      C= 2. * A (2)
      DT(1)= (-A(1) - B) / C
      DT(2) = (-A(1) + B)/  C
3     CONTINUE
      RETURN
      END
```

Subroutine: RESØUT

Purpose: Controls output of the residuals of the data from a single station from a specified estimate of state.

Calling Sequence: CALL RESØUT (LTRAJ)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	LTRAJ	ℓ			.TRUE. except on initial entry and final return.

Common storages used: //146 cells,/DATCØM/,/DCPCØM/,/EDTCØM/,/ESRCØM/, /PØTCØM/,/RSCØM/,/RSLCØM/,/SB3CØM/

Subroutines required: CBDAT, DEHA, DEQTR, DGTRN, DGTSN, DMPLY, DMVTRN, DATØUP, DSDAT, GRDAT, RSPLØT, SBDAT

Method

The subroutine initialization, state interpolation, and residual computation parallels that of ESTMAT (q.v.).

The plotting of residuals is accomplished as each time point set of residuals is computed. The residuals and interpolated state are stored in the buffer BUFRSL. When the buffer is filled, the record pair INDRSL, BUFRSL is written on the residual tape if a tape is desired, or if the residuals are being plotted. The residuals and state are listed at this point if the residuals are not being plotted, and if the list has been requested on the process control card.

On completion of the residual plotting, if any, the residual tape is read and the residuals and state are listed. If no residual tape is desired and the residuals have been written for temporary storage, the tape is again repositioned at the start of the process, and an end-of-tape record is written over the first INDRSL record.

```
$IBFTC MC133K  XR3,M94,NOCD,LIST
      SUBROUTINE RESOUT (NRSPOS,LTRAJ)
C     CONTROL ROUTINE FOR RESIDUAL OUTPUT
C
      LOGICAL             LTRAJ
      DOUBLE PRECISION    DCOS ,DSIN ,DSQRT
C
      COMMON             /DCPCOM/CDCP(900)
      DIMENSION           IFSKIP(4) ,IPROC(5) ,MSKIP(4,4) ,NPROC(22)
      DOUBLE PRECISION    CON(8) ,SN(13,20) ,STIMR(2)
      EQUIVALENCE              (CDCP(881),MSKIP ) ,(CDCP(143),SN     )
     1     ,(CDCP(  1),CON     ) ,(CDCP(116),NRSEND) ,(CDCP(781),STIMR )
     2     ,(CDCP(756),IPROC ) ,(CDCP(731),NPROC  ) ,(CDCP(110),YTEST )
      EQUIVALENCE         (IPROC( 5),IFTSTP) ,(IPROC( 2),ITRSTA)
     1                   ,(IPROC( 3),INDSTA)
      EQUIVALENCE              (NPROC(19),IFOUTL) ,(NPROC(13),NPSKIP)
     1                        ,(NPROC(14),IFSKIP) ,(NPROC(22),NRLIST)
     2                        ,(NPROC(10),NPREST) ,(NPROC(21),NRPLOT)
     3     ,(NPROC(18),IFBADD) ,(NPROC( 3),NPRSTA) ,(NPROC(20),NRTAPE)
C
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)
      DOUBLE PRECISION    DATIND(20),ONTIME
      EQUIVALENCE              (INDDAT( 7),KONT   ) ,(INDDAT( 6),NPTS   )
     1                        ,(INDDAT(12),KTAU   ) ,(INDDAT( 3),NRSTA  )
     2   ,(INDDAT( 1),DATIND) ,(INDDAT( 8),MTYPE  ) ,(INDDAT( 4),NTSTA  )
     3   ,(INDDAT(11),DELT   ) ,(INDDAT( 2),NEOT   ) ,(INDDAT(13),ONTIME)
C
      COMMON             /ESRCOM/CESR(304)
      DOUBLE PRECISION    CRF    (2) ,ETIMVA ,SER(14,2)
     1                   ,DELDAR(2)         ,SPCDAR(6)
      EQUIVALENCE              (CESR( 29),ETIMVA) ,(CESR(245),SER     )
     1     ,(CESR(301),CRF     ) ,(CESR(   2),NBY    ) ,(CESR( 49),SPCDAR)
     2     ,(CESR( 97),DELDAR) ,(CESR(   1),NESPOS)
C
      COMMON             /SB3COM/CSBF(12),RBF(6),VBF(6,8)
      DOUBLE PRECISION    RBF ,VBF ,HBF ,TBF ,TFF
      EQUIVALENCE         (CSBF( 7),HBF     ) ,(CSBF( 2),NTP     )
     1                   ,(CSBF( 5),LCB     ) ,(CSBF( 9),TBF     )
     2                   ,(CSBF( 3),NPT1    ) ,(CSBF(11),TFF     )
C
      COMMON             /RSLCOM/INDRSL(16),BUFRSL(18,14)
      DIMENSION           IRSL(14)
      DOUBLE PRECISION    DBUFRL(9,14) ,TRFRST ,TRON
     1                   ,DINDRL     (8) ,TRLAST ,TROFF
      EQUIVALENCE              (INDRSL( 2),NREOT ) ,(INDRSL( 6),NRSSTA)
     1   ,(INDRSL( 1),DINDRL) ,(INDRSL( 7),NREST ) ,(INDRSL(13),TRFRST)
     2   ,(INDRSL( 3),IRSL   ) ,(INDRSL( 3),NRKUNT) ,(INDRSL(15),TRLAST)
     3   ,(INDRSL( 8),MRTYPE) ,(INDRSL( 1),NRPOS ) ,(INDRSL(11),TROFF )
     4   ,(INDRSL( 4),NRBLK ) ,(INDRSL( 5),NRPTS ) ,(INDRSL( 9),TRON   )
      EQUIVALENCE         (BUFRSL(1,1),DBUFRL)
C
      COMMON             /RSCOM/KPTIM(8),RLSAVE(56)
      LOGICAL             KOUTL ,LSKIP(4) ,KPMAX ,KPTIME
      DIMENSION           NASGN(18) ,PMAX(5) ,SUM   (4)
     1                   ,NSUM   (4)          ,SUMSQ(4)
      DOUBLE PRECISION    OBIAS(4) ,TFIRST ,YMOD(4)
      EQUIVALENCE         (KPTIM( 1),KPTIME) ,(KPTIM( 2),PTIME )
     1                   ,(KPTIM( 3),KPMAX ) ,(KPTIM( 4),PMAX   )
      EQUIVALENCE              (RLSAVE(23),LSKIP ) ,(RLSAVE(31),SUM    )
     1                        ,(RLSAVE(39),NASGN ) ,(RLSAVE(35),SUMSQ )
     2                        ,(RLSAVE(27),NSUM  ) ,(RLSAVE(17),TFIRST)
     3                        ,(RLSAVE( 1),OBIAS ) ,(RLSAVE(20),TREF  )
     4    ,(RLSAVE(22),KOUTL ) ,(RLSAVE(19),STPTIM) ,(RLSAVE( 9),YMOD  )
      EQUIVALENCE              (NASGN( 5),N203   ) ,(NASGN(12),N406   )
     1                        ,(NASGN( 6),N204   ) ,(NASGN(13),N410   )
     2                        ,(NASGN( 7),N290   ) ,(NASGN(14),N425   )
     3     ,(NASGN( 1),N122   ) ,(NASGN( 8),N300   ) ,(NASGN(15),N435   )
     4     ,(NASGN( 2),N129   ) ,(NASGN( 9),N302   ) ,(NASGN(16),N440   )
     5     ,(NASGN( 3),N130   ) ,(NASGN(10),N319   ) ,(NASGN(17),N442   )
     6     ,(NASGN( 4),N200   ) ,(NASGN(11),N404   ) ,(NASGN(18),N825   )
C
      COMMON             /POTCOM/PLOTC(42)
      LOGICAL             KDATA(4)
      DIMENSION           DATA   (4) ,SYMBOL(4)
     1                   ,DATMAX(4) ,LABEL(12)
```

```
      EQUIVALENCE              (PLOTC(11),DATMAX) ,(PLOTC(15),LABEL )   ROUT0075
     1                        ,(PLOTC(10),DT     ) ,(PLOTC(27),SYMBOL)   ROUT0076
     2     ,(PLCTC( 2),DATA   ) ,(PLOTC( 6),KDATA ) ,(PLOTC( 1),TIME   )   ROUT0077
C                                                                       ROUT0078
      COMMCN            /CATCOM/CDAT(299)                               ROUT0079
      DIMENSION          IDATA(7)                                       ROUT0080
      DOUBLE PRECISION   BIAS  (2) ,OMEGA    ,TAU        ,TT2BT(9)      ROUT0081
     1                  ,FTR        ,SPDLT    ,TB2C(9,4) ,XV   (12)     ROUT0082
     2                  ,OBS(16,4) ,STA(10) ,TT2BC (9)                  ROUT0083
      EQUIVALENCE              (CDAT(293),IDATA ) ,(CDAT(159),TAU    )  ROUT0084
     1                        ,(CDAT(  5),OBS   ) ,(CDAT(161),TB2C   )  ROUT0085
     2                        ,(CDAT(135),OMEGA ) ,(CDAT(233),TT2BO )   ROUT0086
     3     ,(CDAT(   1),BIAS  ) ,(CDAT(137),SPDLT ) ,(CDAT(251),TT2BT )   ROUT0087
     4     ,(CDAT(133),FTR    ) ,(CDAT(139),STA   ) ,(CDAT(269),XV     )  ROUT0088
      EQUIVALENCE        (IDATA( 2),MODE   ) ,(IDATA( 5),NALIGN)         ROUT0089
     1                  ,(IDATA( 7),NFRAC )                             ROUT0090
C                                                                       ROUT0091
      COMMCN             SAVE(60) ,D(33) ,STT(5,2)                      ROUT0092
      DCUBLE PRECISICN   C ,STT                                         ROUT0093
C                                                                       ROUT0094
      LCGICAL            L ,LOUTL                                       ROUT0095
      DIMENSICN          IDUAL(4) ,JDATA (7) ,LBTYP(4) ,NQUAL(4)        ROUT0096
     1                  ,IQL   (4) ,LBM(12,4) ,NM    (4) ,PRCD (5)      ROUT0097
     2                                                   ,SYM(4,4)      ROUT0098
      EQUIVALENCE        (QUAL,IQUAL)                                   ROUT0099
      DCUBLE PRECISION   ETIMR ,ETIMV  ,GHA                             ROUT0100
     1                  ,ETIMT ,FLT(8) ,UTIMR                           ROUT0101
C                                                                       ROUT0102
      DATA    FLT  /1.D0,2.D0,3.D0,4.D0,5.D0,6.D0,7.D0,8.D0/            ROUT0103
      DATA    NM   /-3,1,3,-1/                                          ROUT0104
      DATA    JDATA/ 2, 0, 2, 2, 0, 1, 0/                               ROUT0105
      DATA    NQUAL/ 1C00, 100,  10,   1/                               ROUT0106
      DATA    LBM   /6HAZIMUT,6HELEVAT,6HRANGE ,6H     ,6HH      ,6HION  ROUT0107
     1              ,6H      ,6H      ,6HRAD    ,6HRAD    ,6HKM     ,6H    ROUT0108
     2              ,6HX     ,6HY     ,6HRANGE  ,6HDOPPLE,6H       ,6H    ROUT0109
     3              ,6H      ,6HR     ,6HRAD    ,6HRAD    ,6HSEC    ,6HCYCLESROUT0110
     3              ,6HX     ,6HY     ,6HRANGE  ,6HDOPPLE,6H       ,6H    ROUT0111
     2              ,6H      ,6HR     ,6HRAD    ,6HRAD    ,6HKM     ,6HCYCLESROUT0112
     4              ,6HHOUR A,6HDECLIN,6H       ,6HDOPPLE,6HNGLE    ,6HATION ROUT0113
     4              ,6H      ,6HR     ,6HRAD    ,6HRAD    ,6H       ,6HCYCLESROUT0114
     5              /                                                   ROUT0115
      DATA    LBTYP/6HC-BAND,6HG-RR  ,6HS-BAND,6HDSIF   /                ROUT0116
      DATA    IDUAL/6HIDUALS,6H AND V,6HEHICLE,6H STATE/                ROUT0117
      DATA    PRCD /6H, PROC,6HEED TO,6H THE N,6HEXT PR,6HOCESS /       ROUT0118
      DATA    START ,STCP   /6HOSTART,6H STOP /                         ROUT0119
      DATA    SYM  /1HA,1HE,1HR,1HZ ,1HX,1HY,1HR,1HD                    ROUT0120
     1              ,1HX,1HY,1HR,1HD ,1HH,1HC,1HR,1HD /                 ROUT0121
C                                                                       ROUT0122
  601 FCRMAT(//23HORECEIVING STATION      ,A6/23H TRANSMITTING STATION   ,ROUT0123
     1A6/10H DATA TYPE,13X,A6,2H (,3(2A6,2H, ),2A6/32X,4(A6,8X),1H))    ROUT0124
  603 FORMAT(/41HOTIME TAGS ARE REFERRED TO FIRST ONTIME =,D24.16,23H SEROUT0125
     1CONDS (ST) FROM 1950)                                             ROUT0126
  604 FCRMAT(A6,14H PRCCESSING AT,F12.2,13H SECONDS (ST))               ROUT0127
  605 FORMAT(35H STOP PROCESSING AT END OF DATA ARC)                    ROUT0128
  606 FORMAT(6HOSKIP ,I3,31H DATA PTS BETWEEN PROCESSED PTS)            ROUT0129
  607 FORMAT(27HOESTIMATE TAPE RECORD PAIR ,I3,14H DEFINES STATE)       ROUT0130
  608 FORMAT(33HO** NO RESIDUAL PROCESS SPECIFIED,5A6)                  ROUT0131
  609 FORMAT(73HO** WRITE RESIDUALS AND VEHICLE STATE ON TAPE, STARTING ROUT0132
     1WITH RECCRD PAIR ,I3)                                             ROUT0133
  610 FORMAT(33HO** RESIDUALS NOT WRITTEN ON TAPE)                      ROUT0134
  611 FORMAT(18H1** PLCT RESIDUALS)                                     ROUT0135
  612 FORMAT(31HO** ALL MEASUREMENTS SUPPRESSED,5A6)                    ROUT0136
  641 FORMAT(22H1** LIST RESIDUAL TAPE)                                 ROUT0137
  642 FORMAT(23H     INCRSL RECORDS ONLY)                               ROUT0138
  643 FCRMAT(28H     VEHICLE STATE SUPPRESSED)                          ROUT0139
  644 FORMAT(//119HORECORD PAIR     END OF TAPE      CONTINUATION      BLOROUT0140
     1CK CCUNT      POINTS      STATION      EST TAPE RECORD PAIR     MTYPEROUT0141
     2/9X,I3,14X,I2,16X,I1,13X,I3,9X,I2,6X,A6,15X,2I10//16X,15HPROCESS OROUT0142
     3N TIME,9X,16HPRCCESS OFF TIME,8X,16HBLOCK FIRST TIME,8X,15HBLOCK LROUT0143
     4AST TIME/11X,4D24.16)                                             ROUT0144
  645 FCRMAT(12H1** LIST RES,4A6)                                       ROUT0145
  646 FORMAT(/14HOQUALITY  TIME,8X,4(4X,2A6)/10X,5H(SEC),3X,4(8X,1H(,A6,ROUT0146
     11H))/1H )                                                         ROUT0147
  647 FORMAT(26X,61HVEHICLE STATE IN EARTH CENTERED EQUATOR OF 1950.0 COROUT0148
     1ORDINATES/14X,6(12X,2HX(,I1,1H))/1H )                             ROUT0149
```

```
  648 FORMAT(2X,I4,5E16.8)
  649 FORMAT(22X,6E16.8/1H )
  681 FORMAT(20H0** RESIDUAL SUMMARY,4X,4(4X,2A6)/20X,4(8X,1H(,A6,1H)))
  682 FORMAT(/6X,15HFIRST  MOMENT =,3X,4E16.8/6X,15HSECOND MOMENT =,3X,4
     1E16.8/6X,15HSAMPLE ST DEV =,3X,4E16.8/6X,15HPTS INCLUDED  =,3X,4I1
     26)
C
C**** INITIALIZATION
C
    1 IF (LTRAJ)              GO TO 90
C
C     HERE FOR INITIAL ENTRY FOR EACH PROCESS
C     POSITION TAPES
   10 CONTINUE
      NRPOS = NRSPOS
      NREST = NESPOS
      IF (NPREST.EQ.0)        GO TO 11
      READ  (12) SKIP
      NPREST = 0
   11 N = NRSEND-NRPOS
      IF (N.EQ.0)             GO TO 13
      DO 12 I=1,N
      READ  (11) NRPOS,NREOT
      IF (NREOT.LE.0)         GO TO 12
      BACKSPACE 11
      NRSEND = NRPOS
      GO TC 13
   12 READ  (11) SKIP
   13 CONTINUE
      DO 14 I=1,4
   14 INDRSL(I+1) = 0
      IF (NRSEND.EQ.0)        NREOT =-1
      NRSSTA = NRSTA
C
C     SET UP COUNTERS AND REFERENCES
   20 CONTINUE
      III = NPSKIP
      NPSKIP = NPSKIP+1
      TFIRST = ONTIME
      STPTIM = STIMR(2)-ONTIME
      DO 21 I=1,7
   21 IDATA(I) = JDATA(I)
      ASSIGN 801 TO N122
      ASSIGN  22 TO N129
      GO TO 110
   22 ASSIGN 800 TO N122
      ASSIGN 130 TO N129
      ASSIGN  23 TO N130
   23 CONTINUE
      STP = STIMR(1)-ONTIME
      DO 24 NPT=1,NPTS
   24 IF (STP.LE.BUFDAT(NPT,2)) GO TO 25
      NPT = NPTS+1
      GO TO 100
   25 ASSIGN 200 TO N130
      TRON  = CNTIME+BUFDAT(NPT,2)
      TROFF = STIMR(2)
      NPT1  = NPT-NPSKIP
      YMOD(1) = CON(3)
      YMOD(2) = CON(3)
      YMOD(4) = YTEST
      OBIAS(4) = 0.D0
      DO 26 I=1,4
      NSUM (I) = 0
      SUM  (I) = 0.
   26 SUMSQ(I) = 0.
C
C     WRITE HEADING DATA
   30 CONTINUE
      WRITE (6,601) NRSTA ,NTSTA ,LBTYP(MTYPE)
     1            ,(LABEL(I),LABEL(I+4),I=1,4) ,(LABEL(I),I=9,12)
      WRITE (6,603) TFIRST
      CALL DATOUP (TFIRST,D,0)
      WRITE (6,604) START ,BUFDAT(NPT,2)
      GO TC (31,32) ,IFTSTP
```

```
   31 WRITE (6,604) STOP ,STPTIM                                               ROUT0225
      ASSIGN 201 TO N200                                                      ROUT0226
      GO TO 33                                                                ROUT0227
   32 WRITE (6,605)                                                           ROUT0228
      ASSIGN 202 TO N200                                                      ROUT0229
   33 CONTINUE                                                                ROUT0230
      IF (III.NE.0)              WRITE (6,606) III                            ROUT0231
      IF (NREST.NE.0)            WRITE (6,607) NREST                          ROUT0232
C                                                                             ROUT0233
C     SET SWITCHES                                                            ROUT0234
   40 CONTINUE                                                                ROUT0235
      IF (IFOUTL.NE.0)           GO TO 41                                     ROUT0236
      ASSIGN 203 TO N203                                                      ROUT0237
      ASSIGN 319 TO N302                                                      ROUT0238
      KOUTL = .TRUE.                                                          ROUT0239
      GO TO 42                                                                ROUT0240
   41 ASSIGN 204 TO N203                                                      ROUT0241
      ASSIGN 303 TO N302                                                      ROUT0242
      KOUTL = .FALSE.                                                         ROUT0243
   42 IF (IFBACD.NE.0)           GO TO 43                                     ROUT0244
      ASSIGN 100 TO N204                                                      ROUT0245
      ASSIGN 301 TO N300                                                      ROUT0246
      GO TO 44                                                                ROUT0247
   43 ASSIGN 205 TO N204                                                      ROUT0248
      ASSIGN 302 TO N300                                                      ROUT0249
   44 CONTINUE                                                                ROUT0250
      ASSIGN 400 TO N319                                                      ROUT0251
      ASSIGN 500 TO N404                                                      ROUT0252
      ASSIGN 410 TO N406                                                      ROUT0253
      ASSIGN 450 TO N410                                                      ROUT0254
      ASSIGN 825 TO N425                                                      ROUT0255
      ASSIGN 440 TO N435                                                      ROUT0256
      ASSIGN 442 TO N440                                                      ROUT0257
      ASSIGN 825 TO N442                                                      ROUT0258
      ASSIGN 420 TO N825                                                      ROUT0259
      IF (NRPLCT.NE.0)           GO TO 46                                     ROUT0260
      ASSIGN 100 TO N404                                                      ROUT0261
      ASSIGN 420 TO N410                                                      ROUT0262
      ASSIGN 440 TO N425                                                      ROUT0263
      ASSIGN 455 TO N442                                                      ROUT0264
      IF (NRLIST.GT.3)           ASSIGN 450 TO N425                           ROUT0265
      IF (NRTAPE.NE.0)           GO TO 47                                     ROUT0266
      ASSIGN 430 TO N406                                                      ROUT0267
      IF (NRLIST.NE.0)           GO TO 45                                     ROUT0268
      WRITE (6,608) PRCD                                                      ROUT0269
      GO TO 801                                                               ROUT0270
   45 IF (NRLIST.GE.3)           GO TO 48                                     ROUT0271
      ASSIGN 100 TO N319                                                      ROUT0272
      GO TO 49                                                                ROUT0273
   46 IF (NRTAPE.NE.0)           GO TO 47                                     ROUT0274
      IF (NRLIST.GE.3)           GO TO 48                                     ROUT0275
      ASSIGN 500 TO N319                                                      ROUT0276
      GO TO 49                                                                ROUT0277
   47 N = NRPOS+1                                                             ROUT0278
      WRITE (6,609) N                                                         ROUT0279
      IF (NRLIST.LT.3)           GO TO 49                                     ROUT0280
      ASSIGN 450 TO N442                                                      ROUT0281
   48 IF (NRLIST.LT.4)           GO TO 49                                     ROUT0282
      ASSIGN 436 TO N435                                                      ROUT0283
      ASSIGN 441 TO N440                                                      ROUT0284
   49 CONTINUE                                                                ROUT0285
      IF (NRTAPE.NE.0)           GO TO 50                                     ROUT0286
      ASSIGN 430 TO N825                                                      ROUT0287
      WRITE (6,610)                                                           ROUT0288
C                                                                             ROUT0289
C     LOAD FIXED DATA                                                         ROUT0290
   50 CONTINUE                                                                ROUT0291
      DO 51 I=1,9                                                             ROUT0292
      TT2BC(I) = SN(I+4,INDSTA)                                               ROUT0293
   51 TT2BT(I) = SN(I+4,ITRSTA)                                               ROUT0294
      CALL DMVTRN (TT2BC,SER(6,1),STA    ,1,1)                                ROUT0295
      CALL DMVTRN (TT2BT,SER(6,2),STA(6),1,1)                                 ROUT0296
      DO 52 I=1,3                                                             ROUT0297
      OBIAS(I) = SER(I+6,1)                                                   ROUT0298
      STA(I)   = SN(I+1,INDSTA)+STA(I)                                        ROUT0299
```

```
   52 STA(I+5) = SN(I+1,ITRSTA)+STA(I+5)
      SPDLT = CON(8)+SER(6,1)
C
C     INITIALIZE PLOT
   80 CONTINUE
      IF (NRPLOT.EQ.0)        GO TO 90
      DO 82 I=1,4
      IF (LSKIP(I))           GO TO 82
      DATMAX(I) = PMAX(I+1)
      IF (.NOT.KPMAX)         DATMAX(I) = PMAX*SER(I+10,1)
   82 CONTINUE
      DT = PTIME
      IF (KPTIME)             GO TO 84
      DT = AINT(DELT)
      IF (DT.GT.0.)           GO TO 84
      DT = 0.1
   84 WRITE (6,611)
      CALL RSPLOT(1)
C
C     HERE FOR REPEATED ENTRIES
   90 CONTINUE
      LTRAJ = .FALSE.
      NPT = NPT1
C
C**** DATA INPUT CONTROL
C
  100 NPT = NPT+NPSKIP
      IF (NPT.LE.NPTS)        GO TO 200
      NPT = NPT-NPTS
C
C     LOCATE NEXT DATA RECORD
      ASSIGN 102 TO N101
  101 READ  (10) INDDAT
      IF (NEOT.EQ.0)          GO TO N101, (102,103)
      GO TO 800
  102 IF (KCNT.NE.0)          GO TO 130
      ASSIGN 103 TO N101
      NPT = 1
      GO TO (103,800) ,IFTSTP
  103 IF (STIMR(2).LT.CNTIME) GO TO 800
      IF (NPRSTA.EQ.NRSTA)    GO TO 110
      READ  (10) SKIP
      GO TO 101
C
C     SET DATA TAPE DEPENDENT VARIABLES
  110 CONTINUE
      MRTYPE   = MTYPE
      TREF     = ONTIME-TFIRST
      NALIGN   = INDDAT( 9)
      MODE     = INDDAT(10)
      TAU      = DATIND(10)
      FTR      = DATIND(11)
      STA(4)   = DATIND(12)
      STA(5)   = DATIND(13)
      YMOD(3)  = DATIND(14)
      BIAS(1)  = DATIND(15)+SER(10,1)/TAU
      BIAS(2)  = DATIND(16)
      GO TO (111,112,113,114) ,MTYPE
  111 ASSIGN 291 TO N290
      MODE = 0
      GO TO 120
  112 ASSIGN 292 TO N290
      GO TO 120
  113 ASSIGN 293 TO N290
      GO TO 120
  114 ASSIGN 294 TO N290
C
C     SET DATA TAPE DEPENDENT SWITCHES
  120 CONTINUE
      L = .FALSE.
      DO 122 I=1,4
      LSKIP(I) = .TRUE.
      KDATA(I) = .FALSE.
      DO 121 J=I,12,4
  121 LABEL(J)   = LBM(J,MTYPE)
```

```
      SYMBCL(I) = SYM(I,MTYPE)
      IF (MSKIP(I,MTYPE)+IFSKIP(I).GT.0) GO TO 122
      LSKIP(I) = .FALSE.
      KDATA(I) = .TRUE.
      L = .TRUE.
  122 CONTINUE
      IF (L) GO TO 123
      WRITE (6,612) PRCD
      GO TO N122, (800,801)
  123 NFRAC  = 1
      IF (CRF(1)) 124,126,125
  124 NFRAC  = 2
      GO TO 126
  125 STA(4) = CRF(1)
      STA(5) = CRF(2)
  126 CONTINUE
      STA( 9) = STA(4)
      STA(10) = STA(5)
  129 GO TO N129, ( 22,130)
C
C     READ NEXT DATA RECORD
  130 READ  (10) BUFDAT
      GO TO N130, ( 23,200)
C
C**** COMPUTE OBSERVABLES
C
C     SET UP WORKING ARRAYS
  200 CONTINUE
      TIME = BUFDAT(NPT,2)+TREF
      GO TO N200, (201,202)
  201 IF (STPTIM.LT.TIME)      GO TO 801
  202 QUAL = BUFDAT(NPT,1)
      LOUTL = KOUTL
      DO 204 I=1,4
      DATA(I) = BUFDAT(NPT,I+2)
      GO TO N203, (203,204)
  203 IF (DATA(I).GE.0.)       LOUTL = .FALSE.
  204 IQL(I) = 0
      IF (LOUTL)               GO TO 100
      IF (IQUAL.EQ.0)          GO TO 206
      IF (IQUAL.EQ.7)          GO TO N204, (100,205)
  205 IQL(4) = IQUAL/4
      IQL(1) = IQUAL-4*IQL(4)
      IQL(3) = IQL(1)/2
      IQL(2) = IQL(1)-2*IQL(3)
      IQL(1) = IQL(2)
  206 CONTINUE
      IF (KTAU.NE.0)           GO TO 207
      TAU     = DATA(4)
      DATA(4) = DATIND(10)
      BIAS(1) = DATIND(15)+SER(10,1)/TAU
  207 CONTINUE
C
C     COMPUTE VEHICLE TIME AND STATE
C
C     RECEPTION TIME
  210 CONTINUE
      STIMR = TFIRST+TIME
      UTIMR = STIMR+SER(4,1)+SER(5,1)*STIMR
      ETIMR = UTIMR+DELDAR(1)+DELDAR(2)*UTIMR
      III = 1
      D(2) = (ETIMR-TBF)/HBF
C
C     T**N/N-FACTORIAL
  220 D(1) = FLT(1)
      DO 221 I=2,8
  221 D(I+1) = D(2)*D(I)/FLT(I)
      GO TO (230,251,262,251) ,III
C
C     COMPUTE DOWN-LEG DELAY TIME
  230 D(9) = 0.D0
      DO 231 I=1,8
  231 D(9) = D(9)+D(I)*VBF(6,I)
      ETIMV = ETIMR-D(9)
      ETIMT = ETIMV-D(9)
```

```
      IF (ETIMV.LE.TFF)       GO TO 250
C
C     SET ICS FOR NEW INTERPOLATION TABLES
  240 LTRAJ = .TRUE.
      III = 4
      D(2) = FLT(7)
      GO TO 220
  241 CONTINUE
      DO 242 I=1,3
      SPCDAR(I)   = STT(I,2)
  242 SPCDAR(I+3) = STT(I,1)
      ETIMVA = TFF
      TBF = ETIMVA
      VBF(6,1) = D(9)
      NBY = 100+LCB
      GO TO 999
C
C     INTERPOLATE FOR VEHICLE STATE
  250 III = 2
      D(2) = (ETIMV-TBF)/HBF
      GO TO 220
  251 CONTINUE
      CALL DMPLY (VBF,D,STT,5,8,2,6,1,5,0)
      DO 252 I=1,3
  252 STT(I,2) = RBF(I)+HBF*STT(I,2)
      GO TO (999,253,999,241) ,III
  253 CONTINUE
      CALL DEQTR (ETIMV,D(10))
      D(2) = STT(5,1)+D(12)
      D(3) = STT(4,1)
      D(4) = D(12)
      D(5) = D(3)*DCOS(D(2))
      D(6) = ETIMV-DELDAR(1)-DELDAR(2)*ETIMV
      CALL DEHA (D(6),D(5),D(1),OMEGA)
      GHA = D(1)
      CALL DGTRN (D(16),NM,D,4)
      CALL DGTSN (D(25),0,D(11),D(10),D(13))
      CALL DMVTRN (D(25),D(16),D,2,3)
      DO 254 I=1,9
  254 D(I+9) = 0.D0
      DO 255 I=1,9,4
  255 D(I+9) = 1.D0
      D(19) = ETIMV-ETIMR
      D(21) = ETIMV-ETIMT
      D(20) = D(19)+TAU
      D(22) = D(21)+TAU
      DO 256 I=1,4
      D(10) = DCOS(OMEGA*D(I+18))
      D(13) = DSIN(OMEGA*D(I+18))
      D(11) =-D(13)
      D(14) = D(10)
  256 CALL DMVTRN (D,D(10),TB2C(1,I),1,3)
C
C     LOAD SPACECRAFT STATE
  260 CONTINUE
      DO 261 I=1,3
      XV(I)   = STT(I,2)
  261 XV(I+3) = STT(I,1)
      D(2) = (ETIMV-TBF-TAU)/HBF
      III = 3
      GO TO 220
  262 CONTINUE
      CALL DMPLY (VBF,D(2),XV(7),3,8,2,6,-1,3,0)
      DO 263 I=1,3
  263 XV(I+6) = XV(I+6)*HBF+RBF(I)
C
C     COMPUTE OBSERVABLES
  290 CONTINUE
      GO TO N290, (291,292,293,294)
C     C-BAND
  291 CALL CBDAT
      GO TO 300
C     GODDARD
  292 CALL GRDAT
      GO TO 300
```

```
C     S-BAND
  293 CALL SBDAT
      GO TO 300
C     DSIF
  294 CALL DSDAT (GHA)
      GO TO 300
C
C**** COMPUTE RESIDUALS AND MOMENTS
C
C     RESIDUALS
  300 CONTINUE
      IQUAL = 10000
      DO 319 I=1,4
      KDATA(I) = .FALSE.
      IQUAL = IQUAL+IQL(I)*NQUAL(I)
      IF (LSKIP(I))             GO TO 319
      IF (DATA(I).EQ.YTEST)     GO TO 319
      GO TO N300, (301,302)
  301 IF (IQL(I).GT.0)          GO TO 319
  302 IF (DATA(I).GE.0.)        GO TO 304
      GO TO N302, (303,319)
  303 DATA(I)  =-DATA(I)
  304 DATA(I)   = DATA(I)-OBS(1,I)-OBIAS(I)
      KDATA(I) = .TRUE.
      STP       = DATA(I)/YMOD(I)+0.5
      IF (STP.LT.0.)            STP = STP-1.
      STP       = AINT(STP)
      DATA(I)   = DATA(I)-STP*YMOD(I)
C
C     MOMENTS
  310 CONTINUE
      NSUM (I) = NSUM (I)+1
      SUM  (I) = SUM  (I)+DATA(I)
      SUMSQ(I) = SUMSQ(I)+DATA(I)*DATA(I)
  319 CONTINUE
      GO TO N319, (100,400,500)
C
C**** TAPE WRITE AND LIST
C
C     LOAD TIME POINT INTO BUFFER
  400 CONTINUE
      NRPTS = NRPTS+1
      BUFRSL(1,NRPTS) = QUAL
      BUFRSL(2,NRPTS) = TIME
      DO 402 I=1,4
  402 BUFRSL(I+2,NRPTS) = DATA(I)
      DO 404 I=1,6
  404 DBUFRL(I+3,NRPTS) = XV(I)
      IF (NRPTS.LT.14)          GO TO N404, (100,500,810)
  406 GO TO N406, (410,430,440)
C
C     SET UP INDRSL
  410 CONTINUE
      NRPOS = NRPOS+1
      NRBLK = NRBLK+1
      TRFRST = TFIRST+BUFRSL(2,1)
      TRLAST = TFIRST+TIME
      GO TO N410, (420,425,450,455)
C
C     TITLE TAPE LISTING
  420 CONTINUE
      ASSIGN 425 TO N410
      ASSIGN 425 TO N825
      WRITE (6,641)
      IF (NRLIST-3)             421,422,425
  421 WRITE (6,642)
      GO TO 425
  422 WRITE (6,643)
C
C     LIST  ,INDRSL,
  425 WRITE (6,644) (INDRSL(I),I=1,8),(DINDRL(I),I=5,8)
      GO TO N425, (440,450,830)
C
C     TITLE ,BUFRSL,
  430 ASSIGN 440 TO N406
```

```
      ASSIGN 440 TO N825
      N = 1
      IF (NRLIST.GE.4)         N = 4
      WRITE (6,645) (IDUAL(I),I=1,N)
  435 CONTINUE
      WRITE (6,646) (LABEL(I),LABEL(I+4),I=1,4),(LABEL(I),I=9,12)
      GO TO N435, (436,440)
  436 WRITE (6,647) (I,I=1,6)
C
C     LIST  ,BUFRSL,
  440 CONTINUE
      DO 442 J=1,NRPTS
      WRITE (6,648) (BUFRSL(I,J),I=1,6)
      GO TO N440, (441,442)
  441 WRITE (6,649) (BUFRSL(I,J),I=7,18,2)
  442 CONTINUE
      GO TO N442, (450,455,825)
C
C     WRITE TAPE RECORD PAIR
  450 WRITE (11) INDRSL
      WRITE (11) BUFRSL
      NRKONT = 1
      NRECT  = 0
      WRITE (11) NRPOS,NQUAL(4),IRSL
      BACKSPACE 11
  455 NRPTS = 0
      GO TO N404, (100,500,810)
C
C**** PLOT RESIDUALS
C
  500 CONTINUE
      CALL RSPLOT(0)
      GO TO 100
C
C**** POSTLOGUE
C
  800 BACKSPACE 10
  801 NTP = 4
C
C     DUMP INCOMPLETE BUFFER
  802 IF (NRPTS.EQ.0)          GO TO 810
      ASSIGN 810 TO N404
      GO TO 406
C
C     TERMINATE PLOT
  810 IF (NRPLCT.EQ.0)         GO TO 830
      CALL RSPLOT(-1)
      IF (NRLIST-2)            830,811,820
  811 IF (NRTAPE.EQ.0)         GO TO 830
C
C     PREPARE TAPE FOR READING
  820 CONTINUE
      DO 822 I=1,NRBLK
      BACKSPACE 11
  822 BACKSPACE 11
  825 CONTINUE
      READ  (11) INDRSL
      IF (NRECT.GT.0)          GO TO 826
      READ  (11) BUFRSL
      GO TO N825, (420,425,430,440)
  826 BACKSPACE 11
C
C     REPOSITION TAPE
  830 IF (NRTAPE.NE.0)         GO TO 840
      IF (NRPOS.EQ.NRSEND)     GO TO 840
      DO 832 I=1,NRBLK
      BACKSPACE 11
  832 BACKSPACE 11
      NRPOS = NRSEND
      WRITE (11) NRPOS,NQUAL(4),IRSL
      BACKSPACE 11
C
C     OUTPUT MOMENTS
  840 IF (NRLIST.EQ.0)         GO TO 999
      WRITE (6,681) (LABEL(I),LABEL(I+4),I=1,4),(LABEL(I),I=9,12)
```

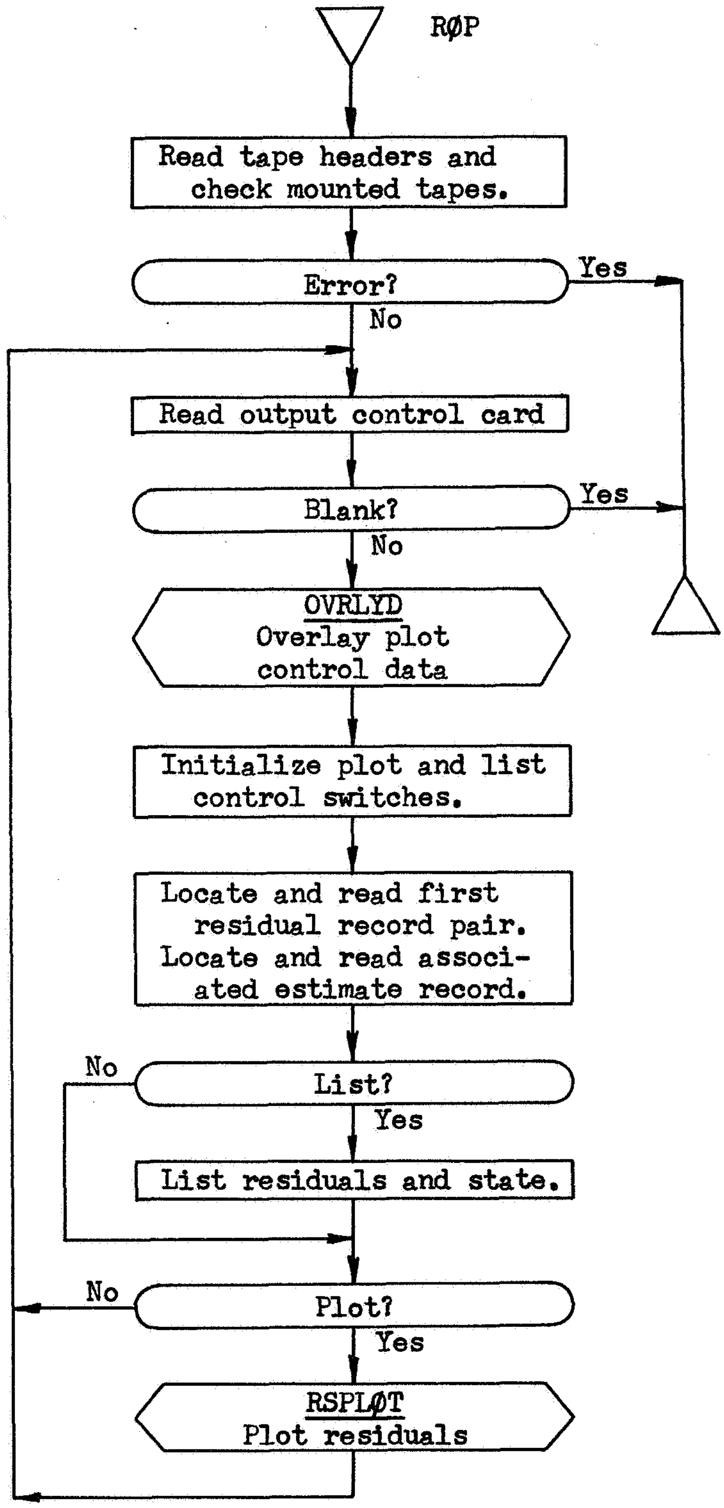
```
      DO 842 I=1,4
      STP = NSUM(I)
      SUM  (I) = SUM(I)/STP
      DATA (4) = (SUMSQ(I)-SUM(I)*SUM(I))/(STP-1.)
      DATA (I) = DSQRT(DATA(4))
  842 SUMSQ(I) = SUMSQ(I)/STP
      WRITE (6,682) SUM ,SUMSQ ,DATA ,NSUM
C
C     EXIT
  999 NRSPCS = NRPOS
      NRSEND = NRPOS
      RETURN
      END
```

Subroutine: RØP

Purpose: Residual output program. Accepts the residual and estimate tapes written by the DCP and lists and/or plots residuals.

Common storages required: /ESTCØM/,/PØTCØM/,/RSLCØM/

Subroutines required: DATØUP, ØVRLYD, RSPLØT


RØP
Read tape headers and check mounted tapes.
Error?
Yes
No
Read output control card
Blank?
Yes
No
OVRLYD
Overlay plot control data
Initialize plot and list control switches.
Locate and read first residual record pair. Locate and read associated estimate record.
List?
No
Yes
List residuals and state.
Plot?
No
Yes
RSPLØT
Plot residuals

```
$IBFTC MC138I  XR3,M94,NODD,LIST
C     ROP - RESIDUAL OUTPUT PROGRAM
C
C     DEVELOPED BY  PHILCO-FORD CORPORATION
C               FOR GODDARD SPACE FLIGHT CENTER
C               CONTRACT NAS5-9939
C
C     REFERENCE TR-DA1508, PROGRAM DESCRIPTION AND THEORETICAL BASIS,
C                          ORBIT DETERMINATION PROGRAM
C               TR-DA1509, SUBROUTINE DESCRIPTIONS AND LISTINGS,
C                          ORBIT DETERMINATION PROGRAM
C               TR-DA1510, INPUT-OUTPUT SUMMARY,
C                          ORBIT DETERMINATION PROGRAM
C
C     DECEMBER 1967
C
      COMMON            /ESTCOM/CEST(804)
      DIMENSION          NAMSTA(20)
      DOUBLE PRECISION   SPCDAN (6) ,EFEDAN(14) ,PREDAN (4) ,DELDAN (2)
     1                  ,EHADAN(24) ,MHADAN(24) ,XHADAN(24) ,SE (14,20)
     2                  ,ETIMVA     ,TX2Z (3,3)
      EQUIVALENCE                (CEST(   1),NESPOS) ,(CEST(   2),NBY    )
     1                          ,(CEST(   3),NBD    ) ,(CEST(   4),NBX    )
     2                          ,(CEST(   5),NBH    ) ,(CEST(   6),NUMSTA)
     3       ,(CEST(   7),NAMSTA) ,(CEST( 27),KOUNTN) ,(CEST( 29),ETIMVA)
     4       ,(CEST( 31),TX2Z   ) ,(CEST( 49),SPCDAN) ,(CEST( 61),EFEDAN)
     5       ,(CEST( 89),PREDAN) ,(CEST( 97),DELDAN) ,(CEST(101),EHADAN)
     6       ,(CEST(149),MHADAN) ,(CEST(197),XHADAN) ,(CEST(245),SE     )
C
      COMMON            /POTCOM/PLOTC(42)
      LOGICAL            KDATA(4)
      DIMENSION          DATA  (4) ,SYMBOL(4)
     1                  ,DATMAX(4) ,LABEL(12)
      EQUIVALENCE              (PLOTC(11),DATMAX) ,(PLOTC(15),LABEL )
     1                        ,(PLOTC(10),DT     ) ,(PLOTC(27),SYMBOL)
     2     ,(PLOTC( 2),DATA   ) ,(PLOTC( 6),KDATA ) ,(PLOTC( 1),TIME  )
C
      COMMON            /RSLCOM/INDRSL(16),BUFRSL(18,14)
      DOUBLE PRECISION   DBUFRL(9,14)
      EQUIVALENCE              (INDRSL( 2),NREOT ) ,(INDRSL( 1),NRPOS )
     1                        ,(INDRSL( 7),NREST ) ,(INDRSL( 5),NRPTS )
     2   ,(INDRSL( 8),MRTYPE) ,(INDRSL( 3),NRKONT) ,(INDRSL( 6),NRSSTA)
      EQUIVALENCE        (BUFRSL(1,1),DBUFRL)
C
      LOGICAL            KPLOT     ,LINIT ,LPLOT
     1                  ,LDATA(4) ,LLIST ,LTAPE
      DOUBLE PRECISION   PCNTRL(6),PMAX(4)
      DIMENSION          BODNAM(11) ,HEAD     (4) ,IDSTAT (7) ,PRENAM (4)
     1                  ,D        (10) ,HEADER(11) ,LBM (12,4) ,SPCNAM (6)
     2                  ,DELNAM (2) ,HEADES(11) ,NN       (6) ,STBNAM(10)
     3                  ,EFENAM(14) ,HEADRL(11) ,NPLOT    (4) ,SYM    (4,4)
     4                  ,HARNAM(24) ,ICNTRL (8) ,NQUAL    (4) ,YN       (2)
      EQUIVALENCE        (QUAL,IQL)
      EQUIVALENCE              (ICNTRL(5),NPLOT  ) ,(ICNTRL(8),NPLOT4)
     1                        ,(ICNTRL(5),NPLOT1) ,(ICNTRL(1),NRECRD)
     2                        ,(ICNTRL(6),NPLOT2) ,(ICNTRL(2),NRLIST)
     3     ,(ICNTRL(4),NPLBAD) ,(ICNTRL(7),NPLOT3) ,(ICNTRL(3),NROVER)
      EQUIVALENCE        (PCNTRL(3),PMAX   ) ,(PCNTRL(1),PTIME )
     1                  ,(PCNTRL(2),PSCALE)
C
      DATA     BODNAM/6HMERCRY,6HVENUS ,6HEARTH ,6HMARS   ,6HJUPTER
     1              ,6HSATURN,6HURANUS,6HNEPTUN,6HPLUTO ,6HSUN
     2              ,6HMOON   /
      DATA     BLANK /6H      /
      DATA     HEAD  /6HSTATE ,6HAND    ,6HRESIDU,6HALS    /
      DATA     YN    /6HYES    ,6HNO     /
      DATA     LBM   /6HAZIMUT,6HELEVAT,6HRANGE ,6H
     1              ,6HH       ,6HION    ,6H      ,6H
     1              ,6HRAD    ,6HRAD    ,6HKM     ,6H
     2              ,6HX       ,6HY       ,6HRANGE ,6HDOPPLE
     2              ,6H        ,6H        ,6H      ,6HR
     2              ,6HRAD    ,6HRAD    ,6HKM     ,6HCYCLES
     3              ,6HX       ,6HY       ,6HRANGE ,6HDOPPLE
     3              ,6H        ,6H        ,6H      ,6HR
     3              ,6HRAD    ,6HRAD    ,6HSEC    ,6HCYCLES
```

```
    4              ,6HHOUR A,6HDECLIN,6H       ,6HDOPPLE
    4              ,6HNGLE   ,6HATION ,6H      ,6HR
    4              ,6HRAD    ,6HRAD    ,6H      ,6HCYCLES/
     DATA     SYM   /1HA,1HE,1HR,1HZ  ,1HX,1HY,1HR,1HD
    1               ,1HX,1HY,1HR,1HD  ,1HH,1HC,1HR,1HD/
     DATA     IDSTAT/6HSPCRFT,6HEFEMRS,6HPRESUR,6HDELTIM,6HHARM03
    1               ,6HHARM11,6HHARMXX/
     DATA     SPCNAM/3HX   ,3HY   ,3HZ   ,3HXD  ,3HYD  ,3HZD  /
     DATA     EFENAM/3HUME,3HUVE,3HUEA,3HUMA,3HUJU,3HUSA,3HUUR,3HUNE
    1               ,3HUPL,3HUSU,3HUMO,3HUBA,3HAU  ,3HMER/
     DATA     PRENAM/3HSPR,3HDR1,3HDR2,3HVNT/
     DATA     DELNAM/3HDT  ,3HDTD/
     DATA     HARNAM/3HJ20,3HJ30,3HJ40,3HJ50,3HJ60,3HJ70,3HJ21,3HL21
    1               ,3HJ22,3HL22,3HJ31,3HL31,3HJ32,3HL32,3HJ33,3HL33
    2               ,3HJ41,3HL41,3HJ42,3HL42,3HJ43,3HL43,3HJ44,3HL44/
     DATA     STBNAM/3HNOR,3HFST,3HDWN,3HTM1,3HTM2,3HLIT,3HAN1,3HAN2
    1               ,3HRNG,3HDOP/
     DATA     NN    / 3, 1, 1, 4, 1, 4/
     DATA     NQUAL / 1000, 100,  10,    1/
     DATA     PCNTRL/ 2*0.D0  ,2*0.1D0  ,2*1.D2 /
     DATA     YTEST /-0.12345678E20/
C
 5001 FORMAT(I6,11A6)
 5100 FORMAT(I4,34I2)
C
 6001 FORMAT(23H1RESIDUAL TAPE HEADER ',11A6,1H')
 6002 FORMAT(23H ESTIMATE TAPE HEADER ',11A6,1H')
 6003 FORMAT(20H DOES NOT AGREE WITH/23H INPUT HEADER CARD     ',11A6,1H'
     1/15H CANNOT PROCEED)
 6100 FORMAT(89H1NRECRD    NRLIST    OVERLAY    PLOT BAD PTS    PLOT ANG1
     1PLOT ANG2    PLOT RNGE    PLOT DOPL/1X,I5,I5,7X,A3,2(10X,A3),3(9X,A3
     2))
 6220 FORMAT(/69H0PARAMETERS USED FOR THIS RESIDUAL RECORD ARE TAKEN FRO
     1M RECORD PAIR ,I3,21H OF THE ESTIMATE TAPE/4X,5HGROUP,3X,4HNAME,12
     2X,5HVALUE,2(25X,4HNAME,12X,5HVALUE))
 6221 FORMAT(1H0,A6,23H IS THE CENTRAL BODY ON)
 6222 FORMAT(25H0COORDINATES RELATIVE TO ,A6,33H, MEAN EQUATOR, EQUINOX
     1OF 1950.0)
 6223 FORMAT(1H0,2X,A6,4X,A3,D26.16,2(7X,A3,D26.16)/(13X,A3,D26.16,7X,A3
     1,D26.16,7X,A3,D26.16))
 6224 FORMAT(26H0DRAG PARAMETERS APPLY TO ,A6)
 6225 FORMAT(30H0GRAVITY TERMS BELOW APPLY TO ,A6)
 6226 FORMAT(21H0RECEIVING STATION IS)
 6300 FORMAT(//6H0LIST ,2(A6,A4))
 6301 FORMAT(19H0NPT QUAL          TIME,5X,4(4X,2A6)/15X,5H(SEC),4(10X,A6))
 6302 FORMAT(35X,61HVEHICLE STATE IN EARTH CENTERED EQUATOR OF 1950.0 CO
     1ORDINATES/18X,6(12X,2HX(,I1,1H))/1H )
 6311 FORMAT(1X,I3,I5,F11.2,4X,4E16.8)
 6312 FORMAT(24X,6E16.8)
 6400 FORMAT(15H1PLOT RESIDUALS)
C
C**** INITIALIZE
C
C     CHECK MOUNTED TAPE IDENTIFICATION
    1 CONTINUE
      NESPOS = 0
      NRPOS  = 0
      READ  (5,5001) IFTEST,HEADER
      READ  (11)     HEADRL
      WRITE (6,6001) HEADRL
      IF (IFTEST.NE.0)                GO TO 3
      DO 2 I=1,11
    2 IF (HEADER(I).NE.HEADRL(I))     GO TO 9
    3 READ  (5,5001) IFTEST,HEADER
      LTAPE = .TRUE.
      IF (IFTEST.NE.0)                GO TO 5
      DO 4 I=1,11
    4 IF (HEADER(I).NE.BLANK)         GO TO 5
      LTAPE = .FALSE.
      GO TO 100
    5 READ  (12)     HEADES
      WRITE (6,6002) HEADES
      IF (IFTEST.NE.0)                GO TO 100
      DO 6 I=1,11
    6 IF (HEADER(I).NE.HEADES(I))     GO TO 9
```

```
      GO TO 100
C
C     ERRCR
    9 WRITE (6,6003) HEADER
      GO TO 999
C
C**** INPUT CONTROL
C
C     READ CONTROL CARD AND OVERLAY DATA
  100 CONTINUE
      READ  (5,5100) ICNTRL
      IF (NRECRD.EQ.0)          GO TO 999
      WRITE (6,6100) NRECRD ,NRLIST
     1              ,YN(NROVER+1) ,YN(NPLBAD+1) ,YN(NPLOT1+1)
     2              ,YN(NPLOT2+1) ,YN(NPLOT3+1) ,YN(NPLOT4+1)
      IF (NROVER.NE.0)          CALL OVRLYD (PCNTRL)
      LLIST = .TRUE.
      LPLOT = .FALSE.
      DO 102 I=1,4
      LDATA(I) = .FALSE.
      IF (NPLOT(I).EQ.0)        GO TO 102
      LDATA(I) = .TRUE.
      LPLCT    = .TRUE.
  102 CONTINUE
C
C     LOCATE FIRST RECORD PAIR
  110 LINIT = .TRUE.
      N = NRECRD-NRPOS-1
      IF (N) 111,120,113
  111 N =-N
      DO 112 I=1,N
      BACKSPACE 11
  112 BACKSPACE 11
      GO TO 120
  113 DO 114 I=1,N
      READ  (11) SKIP
  114 READ  (11) SKIP
C
C     READ AND IDENTIFY RECORD PAIR
  120 CONTINUE
      READ  (11) INDRSL
      IF (NRECT.LE.0)           GO TO 121
      BACKSPACE 11
      GO TO 122
  121 READ  (11) BUFRSL
      IF (LINIT)                GO TO 200
      IF (NRKONT.NE.0)          GO TO 310
  122 IF (.NOT.LLIST)           GO TO 420
      LLIST = .FALSE.
      GC TO 110
C
C**** INITIALIZE OUTPUT
C
C     SET UP HEADINGS, LABELS
  200 CONTINUE
      LINIT = .FALSE.
      IF (.NOT.LLIST)           GO TO 400
      DO 201 I=1,12
  201 LABEL(I)  = LBM(I,MRTYPE)
      DO 202 I=1,4
      DATMAX(I) = PMAX(I)
  202 SYMBCL(I) = SYM(I,MRTYPE)
      IF (NRLIST.EQ.0)          GO TO 400
C
C     LOCATE AND READ ESTIMATE RECORD
  210 IF (NRLIST.LT.4)          GO TO 300
      NRLIST = NRLIST-4
      IF (.NOT.LTAPE)           GO TO 300
      IF (NREST.EQ.0)           GO TO 300
      N = NREST-NESPOS-1
      IF (N) 211,215,213
  211 N =-N
      DO 212 I=1,N
      BACKSPACE 12
  212 BACKSPACE 12
```

```
      GO TO 215
  213 DO 214 I=1,N
      READ  (12) SKIP
  214 READ  (12) SKIP
  215 CONTINUE
      READ  (12) CEST
      READ  (12) SKIP
C
C     OUTPUT STATE ESTIMATE
  220 CONTINUE
      WRITE (6,6220) NESPOS
      I = NBY/100
      J = NBY-100*I
      WRITE (6,6221) BODNAM(J)
      CALL DATOUP (ETIMVA,D,0)
      IF (I.NE.0)    J = 3
      WRITE (6,6222) BODNAM(J)
      WRITE (6,6223) IDSTAT(1),(SPCNAM(I),SPCDAN(I),I=1,6)
      WRITE (6,6223) IDSTAT(2),(EFENAM(I),EFEDAN(I),I=1,14)
      WRITE (6,6224) BODNAM(J)
      WRITE (6,6223) IDSTAT(3),(PRENAM(I),PREDAN(I),I=1,4)
      WRITE (6,6223) IDSTAT(4),(DELNAM(I),DELDAN(I),I=1,2)
      WRITE (6,6225) BODNAM(3)
      WRITE (6,6223) IDSTAT(5),(HARNAM(I),EHADAN(I),I=1,24)
      WRITE (6,6225) BODNAM(11)
      WRITE (6,6223) IDSTAT(6),(HARNAM(I),MHADAN(I),I=1,24)
      IF (NBX.EQ.0) GO TO 222
      WRITE (6,6225) BODNAM(NBX)
      WRITE (6,6223) IDSTAT(7),(HARNAM(I),XHADAN(I),I=1,24)
  222 CONTINUE
      DO 223 I=1,NUMSTA
  223 IF (NRSSTA.EQ.NAMSTA(I)) INDSTA = I
      WRITE (6,6226) NRSSTA,(STENAM(I),SE(I,INDSTA),I=1,10)
      IF (PSCALE.LE.0.)          GO TO 225
      DO 224 I=1,4
  224 DATMAX(I) = PSCALE*SE(I+10,INDSTA)
  225 IF (NRLIST.EQ.0)           GO TO 400
C
C**** LIST RESIDUALS/STATE
C
C     HEADING
  300 CONTINUE
      ASSIGN 312 TO N310
      ASSIGN 319 TO N312
      N = NN(NRLIST)
      M = NN(NRLIST+3)
      WRITE (6,6300) (HEAD(I),I=N,M)
      IF (NRLIST.EQ.2)           GO TO 301
      ASSIGN 311 TO N310
      WRITE (6,6301) (LABEL(I),LABEL(I+4),I=1,4) ,(LABEL(I),I=9,12)
      IF (NRLIST.EQ.1)           GO TO 302
  301 ASSIGN 313 TO N312
      WRITE (6,6302) (I,I=1,6)
  302 CONTINUE
C
C     LIST
  310 IF (.NOT.LLIST)            GO TO 410
      DO 319 I=1,NRPTS
      GO TO N310, (311,312)
  311 WRITE (6,6311) I,(BUFRSL(J,I),J=1,6)
  312 GO TO N312, (313,319)
  313 WRITE (6,6312) (DBUFRL(J,I),J=4,9)
  319 CONTINUE
      GO TO 120
C
C**** PLOT RESIDUALS
C
C     INITIALIZE
  400 CONTINUE
      IF (.NOT.LPLOT)            GO TO 100
      IF (NRPTS.LE.1)            GO TO 100
      LLIST = .FALSE.
      DT = PTIME
      IF (DT.GE.0.)              GO TO 401
      TP = BUFRSL(1,2)-BUFRSL(1,1)
```

```
      CT = 1.
      IF (TP.LT.1.) CT = 10.
      DT = AINT(CT*TP)/CT
  401 WRITE (6,6400)
      CALL RSPLOT(1)
      ASSIGN 412 TO N410
      IF (NPLBAD.EQ.0)         GO TO 410
      ASSIGN 411 TO N410
C
C     CONTINUE PLOTTING
  410 DO 419 I=1,NRPTS
      TIME = BUFRSL(1,I)
      QUAL = BUFRSL(2,I)
      DO 412 J=1,4
      KDATA(I) = .FALSE.
      DATA(I)  = BUFRSL(J+2,I)
      IF (IQL.LT.NQUAL(J))     GO TO 411
      IQL = IQL-NQUAL(J)
      GO TO N410, (411,412)
  411 IF (.NOT.LDATA(J))       GO TO 412
      IF (DATA(I).EQ.YTEST)    GO TO 412
      KDATA(I) = .TRUE.
      KPLOT    = .TRUE.
  412 CONTINUE
      IF (.NOT.KPLOT)          GO TO 419
      CALL RSPLOT(0)
  419 CONTINUE
      GO TO 120
C
C     TERMINATE PLOT
  420 IF (LINIT)               GO TO 100
      CALL RSPLOT(-1)
      GO TO 100
C
  999 STOP
      END
```

Subroutine: RØTEQ

Purpose: RØTEQ evaluates elements of the rotation matrix which relates the general precession of the Earth's equator and the consequent retrograde motion of the equinox on the ecliptic. It is used to provide the transformation from mean equator, equinox of 1950.0 to mean equator, equinox of date.

Calling Sequence: CALL RØTEQ(TIME,A)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TIME		T	days	Total number of days from 1950.0
Ø	A	(3,3)	A		Rotation matrix.

Common storages used: None

Subroutines required: None

```
$IBFTC MC1350  XR3
CMC1350  SUBROUTINE ROTEQ                                                ROTQ
      SUBROUTINE ROTEQ(TIME,A)                                           ROTQ0000
      DIMENSION A(3,3)                                                   ROTQ0010
      T = TIME/36525.                                                    ROTQ0020
      T2 = T*T                                                           ROTQ0030
      T3 = T2*T                                                          ROTQ0040
      A(1,1) = 1. - .00029697*T2 - .00000013*T3                          ROTQ0050
      A(1,2) = -.02234988*T - .00000676*T2 + .00000221*T3                ROTQ0060
      A(2,1) = -A(1,2)                                                   ROTQ0070
      A(1,3) = -.00971711*T + .00000207*T2 + .00000096*T3                ROTQ0080
      A(3,1) = -A(1,3)                                                   ROTQ0090
      A(2,2) = 1. - .00024976*T2 - .00000015*T3                          ROTQ0100
      A(2,3) = -.00010859*T2 - .00000003*T3                              ROTQ0110
      A(3,2) = A(2,3)                                                    ROTQ0120
      A(3,3) = 1. - .00004721*T2 + .00000002*T3                          ROTQ0130
      RETURN                                                             ROTQ0140
      END                                                                ROTQ
```

Subroutine: RØVLEY

Purpose: To read fixed, floating and alphanumeric data into core. Certain conversions of the floating point data may be made automatically. A blank card terminates a set of data.

Calling Sequence: CALL RØVLEY (C,I,H)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	C				Array into which floating (decimal) data is to be read.
Ø	I				Array into which fixed (integer) data is to be read.
Ø	H				Array into which alphanumeric data is to be read.

Common storages used: None

Subroutines required: None

The function of subroutine RØVLEY is to centralize and facilitate the input of data to the various portions of the program requiring data. The data may be fixed point integers, floating point, or alphanumeric, with a given set of data including any combination of the three types.

A floating point input consists of two numbers, an integer k and a number x. The integer k is a subscript indicating the relative location in C in which the value x is to be stored, thus C(k) = x. The integer field width is 3 (I3) (integers must always be to the right of their field) and the value field width is 12 (E12.8). There may be up to four such pairs/card.

If k=0 and x≠0, x will be stored in the next successive location of C following the previous entry. If both are zero, reading is terminated if they are the first set on the card; otherwise the next card is processed.

Certain alphanumeric flags may be entered, starting in column 61 of a card. The legal flags and their meaning is given below.

Flag	Action
FT	Convert data from ft. to km.
FT/SEC	Convert data from ft/sec to km/sec.
NM	Convert data from n.m. to km.
NM/SEC	Convert data from n.m./sec to km/sec.
SQUARE	Square the value prior to storing in the C array.

It is important to note that these conversions apply to every value on the card, but do not carry over to the next card.

An I in column 61 of a card indicates that fixed point data will be on the following card(s). The first two k values on the I card (col 1-3 and col 16-18) specify the starting and finishing locations in I for data storage. The integer data on the following card(s) is entered with format 12I6, and are stored sequentially.

The word HEAD in columns 61-64 indicates that the next card contains 12 words of alphanumeric data. If the first k value is zero, the 12 words are stored in H(1) through H(12); if k≠0, they are stored in H(13) through H(24).

All data read by RØVLEY is written on the output tape.

```
$IBFTC MC139Q  XR3
      SUBROUTINE ROVLEY(C,IC,HED)                                       ROVL0010
CMC13RL   SUBROUTINE FOR INPUT USAGE TO C ARRAY  WITH INTEGERS, HEADER. ROVL0020
C                                                                       ROVL0030
      DIMENSION    C(1),     BUF(4),       IND(4),       UNIT(6)        ROVL0040
     1,   FACT(5),       IC(1),       HED(24)                           ROVL0050
      DATA (UNIT(I),I=1,6)/2HFT,6HFT/SEC,2HNM,6HNM/SEC,3HDEG,6HSQUARE/  ROVL0060
      DATA (FACT(I),I=1,5)/ 3.048E-4,3.048E-4,1.8519963,1.8519963       ROVL0070
     1,.0174532925/                                                     ROVL0080
      DATA  HEAD/4HHEAD/,HINT/1HI/                                      ROVL0090
      K=-1                                                              ROVL0120
10    L=1                                                               ROVL0130
      READ (5,501) (IND(I),BUF(I),I=1,4),UNT                            ROVL0140
501   FORMAT(4(I3,E12.8),A6)                                            ROVL0150
      IF(UNT .EQ. HEAD) GO TO 100                                       ROVL0160
      IF(UNT .EQ. HINT) GO TO 200                                       ROVL0170
      DO 11 I=1,5                                                       ROVL0180
      CVF=FACT(I)                                                       ROVL0190
      IF(UNT-UNIT(I))11,13,11                                           ROVL0200
11    CONTINUE                                                          ROVL0210
      L = 0                                                             ROVL0220
      IF(UNT - UNIT(6))13,12,13                                         ROVL0230
12    L = -1                                                            ROVL0240
13    CONTINUE                                                          ROVL0250
      IF (BUF) 15,14,15                                                 ROVL0260
14    IF(IND) 15,90,15                                                  ROVL0270
15    IF(K) 16,17,17                                                    ROVL0280
16    K =0                                                              ROVL0290
      WRITE (6,603)                                                     ROVL0300
603   FORMAT(1H1,40X,13HOVERLAY INPUT)                                  ROVL0310
   17 WRITE (6,602) (IND(I),BUF(I),I=1,4),UNT                           ROVL0320
      IF(L) 18,22,20                                                    ROVL0330
18    DO 19 I=1,4                                                       ROVL0340
19    BUF(I) = BUF(I)*BUF(I)                                            ROVL0350
602   FORMAT(4(5X,I3,E17.8),4X,A6)                                      ROVL0360
      GO TO 22                                                          ROVL0370
20    CONTINUE                                                          ROVL0380
      DO 21 I=1,4                                                       ROVL0390
21    BUF(I) = BUF(I)*CVF                                               ROVL0400
22    DO 25 I=1,4                                                       ROVL0410
      IF(IND(I))24,23,24                                                ROVL0420
23    IND(I) = LOC+1                                                    ROVL0430
      IF(BUF(I))24,10,24                                                ROVL0440
24    LOC = IND(I)                                                      ROVL0450
      C(LOC) = BUF(I)                                                   ROVL0460
25    CONTINUE                                                          ROVL0470
      GO TO 10                                                          ROVL0480
90    CONTINUE                                                          ROVL0490
      RETURN                                                            ROVL0500
 100  K1=IND(1)*12+1                                                    ROVL0510
      K2=K1+11                                                          ROVL0520
      READ (5,700) (HED(I),I=K1,K2)                                     ROVL0530
      IF(K) 101,102,102                                                 ROVL0540
101   K=0                                                               ROVL0550
      WRITE (6,603)                                                     ROVL0560
  102 WRITE (6,700) (HED(I),I=K1,K2)                                    ROVL0570
700   FORMAT(12A6)                                                      ROVL0580
      GO  TO 10                                                         ROVL0590
200   L=IND                                                             ROVL0600
      LK = IND(2)                                                       ROVL0610
      READ (5,701) (IC(J),J=L,LK)                                       ROVL0620
701   FORMAT(12I6)                                                      ROVL0630
      IF(K)201,202,202                                                  ROVL0640
201   K=0                                                               ROVL0650
      WRITE (6,603)                                                     ROVL0660
  202 WRITE (6,702) L,LK,(IC(J),J=L,LK) .                               ROVL0670
702   FORMAT(14H INTEGERS FROM ,I5,3H TO,I5/ (12I6))                    ROVL0680
      GO TO 10                                                          ROVL0690
      END                                                               ROVL0700
```

Subroutine: RSIDUL

Purpose: To control the order of trajectory integration and residual output for the residual link.

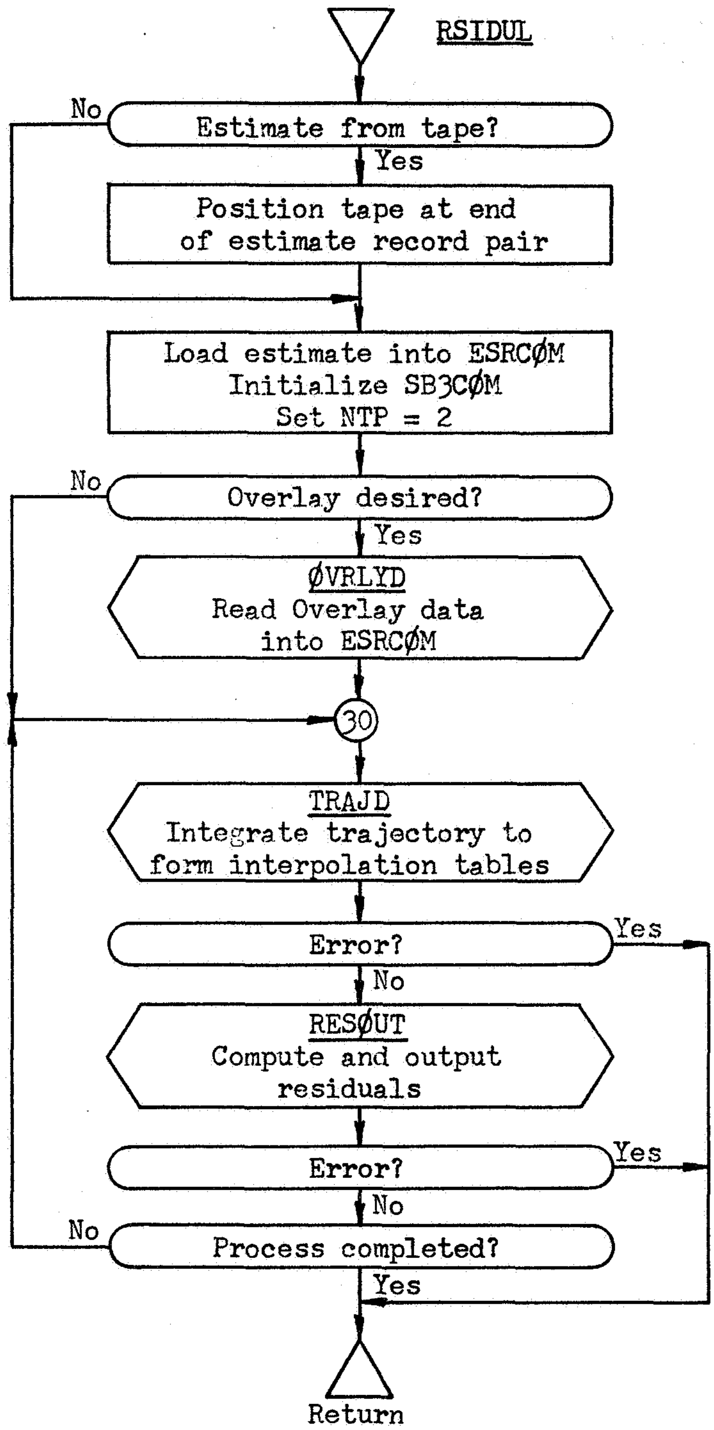
Calling Sequence: CALL RSIDUL (NRSPØS)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	NRSPØS				Residual tape position. The residual tape is positioned between record pairs NRSPØS and NRSPØS+1

Common storages used: /DCPCØM/,/ESRCØM/,/ESTCØM/,/SB3CØM/

Subroutines required: ØVRLYD,RESØUT,TRAJD


RSIDUL
No
Estimate from tape?
Yes
Position tape at end of estimate record pair
Load estimate into ESRCØM
Initialize SB3CØM
Set NTP = 2
No
Overlay desired?
Yes
ØVRLYD
Read Overlay data into ESRCØM
30
TRAJD
Integrate trajectory to form interpolation tables
Error?
Yes
No
RESØUT
Compute and output residuals
Error?
Yes
No
No
Process completed?
Yes
Return

```
$IBFTC MC133I  XR3,M94,NODD,LIST
      SUBROUTINE RSIDUL (NRSPOS)
C     CONTROL ROUTINE FOR DCP RESIDUAL LINK
C
      COMMON            /DCPCOM/CDCP(900)
      EQUIVALENCE       (CDCP(111),IERR   ) ,(CDCP(740),NPREST)
     1                 ,(CDCP(758),INDSTA) ,(CDCP(732),NPROVR)
     2                 ,(CDCP(757),ITRSTA)
C
      COMMON            /ESTCOM/CEST(804)
      DOUBLE PRECISION   SE(14,20)
      EQUIVALENCE       (CEST(245),SE     )
C
      COMMON            /ESRCOM/CESR(304)
      DOUBLE PRECISION   CRF(2),SER(14,20)
      EQUIVALENCE       (CESR(301),CRF    ) ,(CESR(245),SER    )
C
      COMMON            /SB3COM/CSBF(12),RBF(6),VBF(6,8)
      DOUBLE PRECISION   RBF,VBF
C
      LOGICAL            LTRAJ
C
  601 FORMAT(1HO//48X,21H*** RESIDUAL LINK ***)
  602 FORMAT(1HO//46X,26H*** EXIT RESIDUAL LINK ***)
C
C     READ ESTIMATE AND OVERLAY DATA
    1 WRITE (6,601)
      IF (NPREST.EQ.0) GO TO 2
      READ  (12) SKIP
      NPREST = 0
    2 CONTINUE
      DO 3 I=1,244
    3 CESR(I) = CEST(I)
      DO 4 I=1,14
      SER(I,1) = SE(I,INDSTA)
    4 SER(I,2) = SE(I,ITRSTA)
      CRF(1) = 0.
      CRF(2) = 0.
      IF (NPROVR.NE.0) CALL OVRLYD(CESR)
C
C     INITIALIZE STATE BUFFER
   10 DO 11 I=1,12
   11 CSBF(I) = 0.
      LTRAJ = .FALSE.
C
C     INTEGRATE TRAJECTORY AND OUTPUT RESIDUALS
   30 NTP = 2
      CALL TRAJD
      IF (IERR.NE.0) GO TO 900
      CALL RESCUT (NRSPOS,LTRAJ)
      IF (IERR.NE.0) GO TO 900
      IF (NTP.NE.4)  GO TO 30
C
C     RETURN
  900 WRITE (6,602)
  999 RETURN
      END
```

Subroutine: RSPLØT

Purpose: To produce a plot of data on the system output tape. The vertical axis for the plot must be time (in seconds); any four quantities (or less) are plotted along the horizontal axis.

Calling Sequence: CALL RSPLØT (KEY)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	KEY				KEY > 0 to initialize a plot. = 0 to plot a time point < 0 to terminate a plot.

Common storages used: //117 cells, /PØTCØM/

Subroutines required: None.

Usage:

1. Initialization (KEY > 0)

(On a plot entry, the data to be plotted at a given time point will be in DATA(1) - DATA(4)).

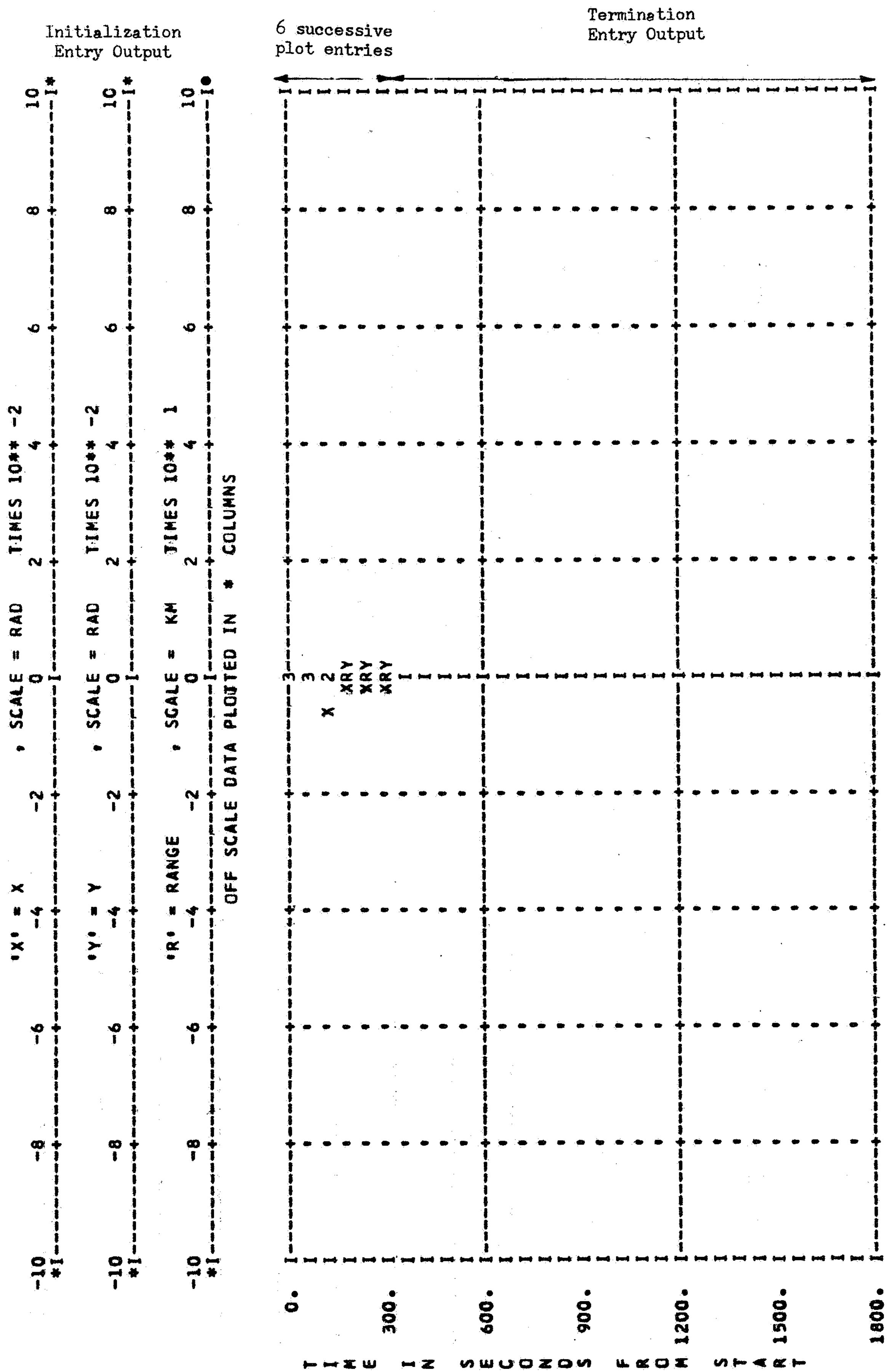
DATMAX (I)	Maximum value of DATA(I) to be plotted on scale. (Off scale values will be plotted in either the far right or far left column of the graph).	
KDATA (I)	To plot DATA type (I), set KDATA (I) $\neq$ 0.	
LABEL (I)	First word of BCI label for DATA (I)	See example
LABEL (I+4)	Second word of BCI label for DATA (I)	See example
LABEL (I+8)	BCI label of units for DATA (I)	See example
SYMBOL (I)	BCI character used to plot DATA (I)	See example
DT	Time scale increment in seconds	See example

2. Plot Entry (KEY = 0)

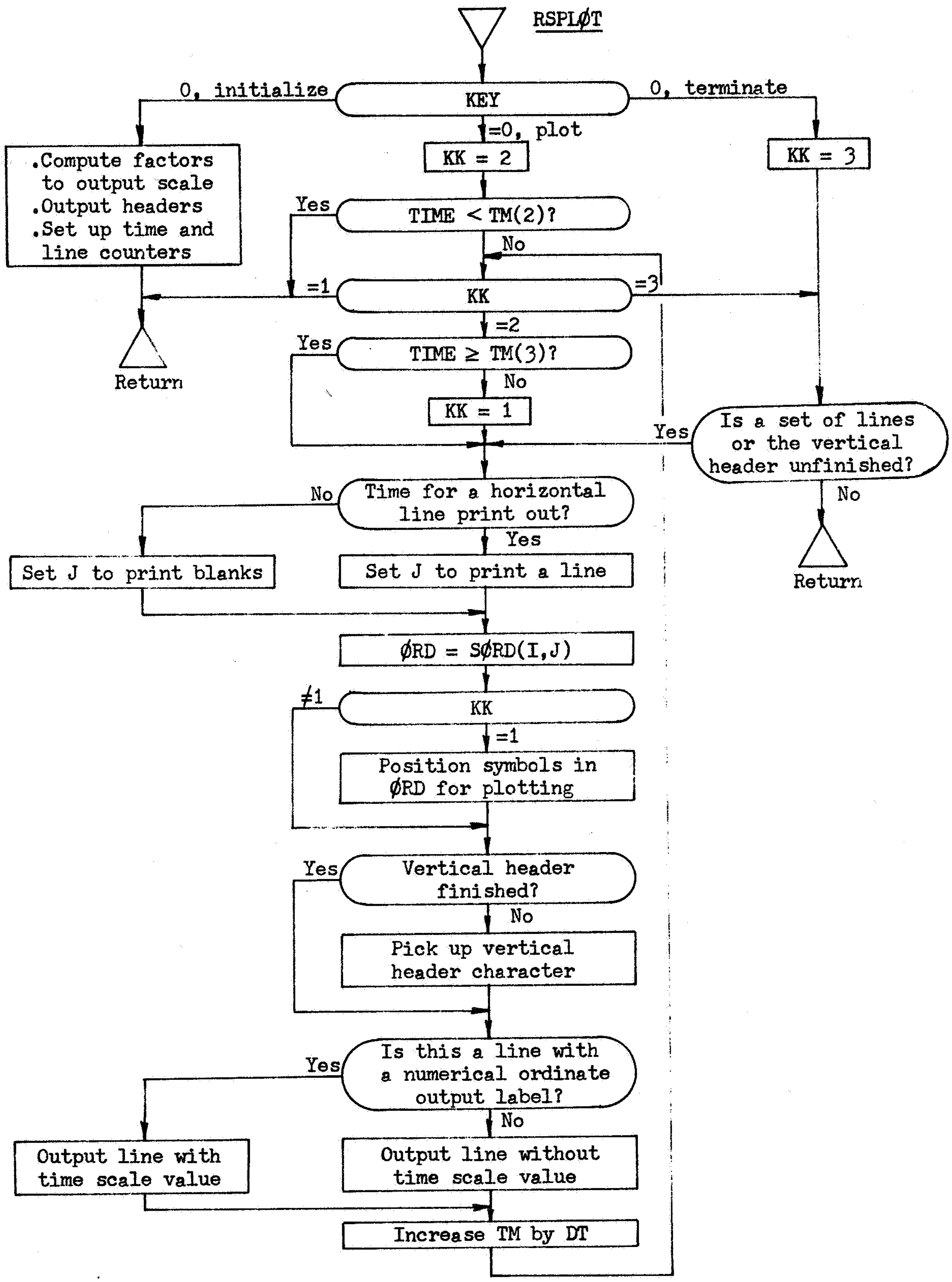
TIME	Time, in seconds from initialization time, of point to be plotted.
DATA (I)	Data to be plotted at this time.
KDATA (I)	Data types to be plotted at this call. All DATA (I) will be plotted for which KDATA (I) $\neq$ 0.
DT	Same as above.

3. Terminate plot (KEY < 0)

DT	Same as above.


Initialization Entry Output
6 successive plot entries
Termination Entry Output
'X' = X , SCALE = RAD TIMES 10** -2
'Y' = Y , SCALE = RAD TIMES 10** -2
'R' = RANGE , SCALE = KM TIMES 10** 1
OFF SCALE DATA PLOTTED IN * COLUMNS
-10 -8 -6 -4 -2 0 2 4 6 8 10
XRY
TIME IN SECONDS FROM START
0. 300. 600. 900. 1200. 1500. 1800.

RSPLØT-3


RSPLØT
0, initialize
KEY
0, terminate
=0, plot
.Compute factors to output scale
.Output headers
.Set up time and line counters
KK = 2
KK = 3
Yes
TIME < TM(2)?
No
=1
KK
=3
=2
Return
Yes
TIME ≥ TM(3)?
No
KK = 1
Yes
Is a set of lines or the vertical header unfinished?
No
Return
No
Time for a horizontal line print out?
Yes
Set J to print blanks
Set J to print a line
ØRD = SØRD(I,J)
≠1
KK
=1
Position symbols in ØRD for plotting
Yes
Vertical header finished?
No
Pick up vertical header character
Yes
Is this a line with a numerical ordinate output label?
No
Output line with time scale value
Output line without time scale value
Increase TM by DT

```
$IBFTC MC13ZT  XR3,M94,LIST,NODD
      SUBROUTINE RSPLOT (KEY)
C     PLOTS DATA ON THE OFF-LINE PRINTER
C     (KEY .GT. 0) INITIALIZE
C       DATMAX(I)     MAXIMUM VALUE OF DATA(I) TO BE PLOTTED ON SCALE
C       KDATA(I)      INITIALIZE FOR PLOTTING DATA(I), YES=1, NO=0
C       LABEL(I)      FIRST  WORD OF BCI LABEL FOR DATA(I)
C       LABEL(I+4)    SECOND WORD OF BCI LABEL FOR DATA(I)
C       LABEL(I+8)    BCI LABEL OF UNITS OF DATA(I)
C       SYMBOL(I)     BCI CHARACTER USED TO PLOT DATA(I)
C       DT            TIME SCALE INCREMENT IN SECONDS
C     (KEY .EQ. 0) PLOT 'DATA' AT TIME'TIME'
C       TIME          TIME IN SECONDS FROM INITIALIZATION TIME
C       DATA(I)       DATA TO BE PLOTTED AT TIME 'TIME'
C       KDATA(I)      DATA(I) TO BE PLOTTED THIS CALL, YES=1, NO=0
C       DT            SAME AS ABOVE
C       SYMBOL        SAME AS ABOVE
C     (KEY .LT. 0) TERMINATE PLOT
C       DT            SAME AS ABOVE
C
C     REMAINING /POTCOM/ CELLS ARE USED FOR STORAGE BETWEEN CALLS
      COMMON /POTCOM/  TIME,DATA(4),KDATA(4),DT,DATMAX(4)
     1                 ,LABEL(12),SYMBOL(4)
     2                 ,DORD(4),TM(3),LINE,LINE2,LINE3,LTM,N1
      LOGICAL          KDATA
      COMMON           ORD(103),TP,NORD(4),KKDATA(4),I,J,K,KK,M
      LOGICAL          KKDATA
      DIMENSION        SORD(103,2),SYMBLX(3),IORD(11)
     1                 ,JORD(11)
      EQUIVALENCE      (ORD(90),JORD)
      DATA             SYMBLX/1H2,1H3,1H4/,ALOG5/.69897/,ASTR/1H*/
      DATA IORD/-5,-4,-3,-2,-1,0,1,2,3,4,5/
      DATA SORD/1H ,1HI,9*1H-,1H+,9*1H-,1H+,9*1H-,1H+,9*1H-,1H+,9*1H-
     1     ,1HI,9*1H-,1H+,9*1H-,1H+,9*1H-,1H+,9*1H-,1H+,9*1H-,1HI,1H
     3     ,1H ,1HI,9*1H ,1H',9*1H ,1H',9*1H ,1H',9*1H ,1H',9*1H ,1HI
     4         ,9*1H ,1H',9*1H ,1H',9*1H ,1H',9*1H ,1H',9*1H ,1HI,1H /
C
C
C     'LTIME' IS VERTICAL AXIS LABEL ARRAY, LAST CELL MUST BE BCI BLANK
      DIMENSION        LTIME(28)
      DATA LTIME /1H ,1HT,1HI,1HM,1HE,1H ,1HI,1HN,1H ,1HS,1HE,1HC,1HO
     1     ,1HN,1HD,1HS,1H ,1HF,1HR,1HO,1HM,1H ,1HS,1HT,1HA,1HR,1HT
     2     ,1H /
C
  600 FORMAT(/41X,1H',A1,4H' = ,2A6,1H,,1X,8HSCALE = ,A6,11H TIMES 10**
     1      ,I3/6X,11(7X,I3))
  602 FORMAT(6X,A1,1X,F5.0,1X,103A1)
  604 FORMAT(6X,A1,7X,103A1)
  606 FORMAT(46X,37HOFF SCALE DATA PLOTTED IN  *  COLUMNS/1H0)
C
      IF( KEY ) 90,50,1
C     INITIALIZE,  KEY=1
    1 CONTINUE
      DO 4 M=1,4
      IF( .NOT.KDATA(M) ) GO TO 4
      TP = ( ALOG10( ABS(DATMAX(M)) )-0.0000001 ) - 50.0
      K = IFIX(TP)+49
      IF( K.EQ.0 ) K=0
      J = 1
      TP = AMOD(TP+130.,1.)
      IF( TP.GT.ALOG5 .OR. TP.LE.1.E-5 )  J=2
      DORD(M) = FLOAT(J) * 10.**(K-1)
      DO 2 I=1,11
    2 JORD(I) = IORD(I)*J
      WRITE(6,600) SYMBOL(M),LABEL(M),LABEL(M+4),LABEL(M+8),K
     1        ,(JORD(I),I=1,11)
      WRITE(6,604)  LTIME(28),ASTR,(SORD(I,1),I=2,102),ASTR
    4 CONTINUE
      WRITE(6,606)
      TM(1) = 0.
      TM(3) = DT/2.
      TM(2) = -TM(3)
      ASSIGN 72 TO N1
      LINE = 10000
      LINE2 = 10000
```

```
      LINE3 = 0
      GO TO 99
C     PLOT,  KEY=0
   50 CONTINUE
      IF( TIME .LT. TM(2) ) GO TO 99
   51 KK=1
      IF( TIME .GE. TM(3) ) KK=2
C
C       SET FOR NUMBER OF LINES PER HORIZONAL LINE PRINT
C                         V
   52 IF(  LINE.GE.10 ) GO TO 54
      LINE = LINE+1
      J = 2
      GO TO 56
   54 LINE = 1
      J = 1
   56 CONTINUE
      DO 58 I=1,103
   58 ORD(I) = SORD(I,J)
      GO TO (60,70,70), KK
   60 CONTINUE
      DO 62 M=1,4
      KKDATA(M) = KDATA(M)
      IF( .NOT.KKDATA(M) ) GO TO 62
      K = IFIX( DATA(M)/DORD(M) + SIGN( 0.5,DATA(M) )   ) + 52
      K = MAX0(K,1)
      K = MIN0(K,103)
      NORD(M) = K
   62 CONTINUE
      DO 68 M=1,4
      IF( .NOT.KKDATA(M) ) GO TO 68
      J = NORD(M)
      ORD(J) = SYMBOL(M)
      K = 0
      I = M
   64 IF( I.GE.4 ) GO TO 68
      IF( J.NE.NORD(I+1) .OR. .NOT.KKDATA(I+1) ) GO TO 66
      K = K+1
      ORD(J) = SYMBLX(K)
      KKDATA(I+1) = .FALSE.
   66 I=I+1
      GO TO 64
   68 CONTINUE
   70 CONTINUE
      GO TO N1, (72,74)
   72 LINE3 = LINE3+1
      LTM = LTIME(LINE3)
C
C       SET FOR NUMBER OF CELLS IN LTIME
C                         V
      IF( LINE3.NE.28 ) GO TO 74
      ASSIGN 74 TO N1
      LINE3 = 0
   74 CONTINUE
C
C       SET FOR NUMBER OF LINES PER TIME LABEL PRINT
C                         V
      IF( LINE2.GE. 5 ) GO TO 76
      LINE2 = LINE2+1
      WRITE(6,604) LTM,ORD
      GO TO 78
   76 LINE2 = 1
      WRITE(6,602) LTM,TM(1),ORD
   78 CONTINUE
      DO 82 I=1,3
   82 TM(I) = TM(I) + DT
      GO TO (99,51,92), KK
C     TERMINATE PLOT,  KEY=-1
   90 CONTINUE
      KK = 3
   92 IF( LINE.NE.1  .OR.  LINE3.NE.0 ) GO TO 52
   99 RETURN
      END
```

Subroutine: SBDAT

Purpose: Computes four Unified S-Band measurements (X, Y, range, doppler).

Calling Sequence: CALL SBDAT

Common storages used: /DATCØM/

Subroutines required: DCRØSS,DDØT,DMVTRN,DNØRM

COMMON LOCATIONS

Common	Location	Name	Dimension	Description
DATCØM	C(1)	BIAS	d(2)	BIAS(1) Doppler bias frequency (cps). BIAS(2) Transponder retransmission ratio.
	C(133)	FTR	d	Doppler transmitter frequency (cps).
	C(135)	OMEGA	d	Earth rotation rate (rad/sec).
	C(137)	SPDLT	d	Speed of light (km/sec).
	C(139)	STA	d(10)	STA(1-3) Receiving station position in B-frame (km). STA(4-5) Refraction constants. STA(6-10) As above, but for transmitting station if three-way doppler.
	C(159)	TAU	d	Doppler count interval (sec).
	C(161)	TB2CØ	d(18)	TB2CØ(1-9) B-frame to C-frame transform at end of doppler count interval at signal reception. TB2CØ(10-18) Same as above, but at beginning of doppler count interval.
	C(197)	TB2CT	d(18)	Same as TB2CØ, but at time of signal transmission.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors at receiving station in B-frame.
	C(251)	TT2BT	d(9)	Same as TT2BØ, but for transmitting station if doppler is three-way.
	C(269)	XV	d(12)	XV(1-6) Spacecraft position and velocity in C-frame at end of doppler count interval as signal leaves spacecraft (km, km/sec). XV(7-12) As above, but at beginning of doppler count interval.
	C(294)	MØDE		Doppler mode = 2, two-way = 3, three-way
	C(297)	NALIGN		Antenna principal axis alignment +1, North +2, East
	C(298)	NANG		Angle inclusion option*.
	C(299)	NFRAC		Refraction correction option*. *+1, include option. +2, omit option.
	C(5)	ØBS	d(64)	Measurements. See Table 1, description of CBDAT.

```
$IBFTC MC13BN  M94,NODD,XR3
CMC13BN    S-BAND MEASUREMENTS
      SUBROUTINE SBDAT
      COMMON  /DATCOM/    BIAS(2),     OBS(64),     FTR,         OMEGA
     1,                   SPDLT,       STA(10),     TAU,         TB2CO(18)
     2,                   TB2CT(18),   TT2BO(9),    TT2BT(9),    XV(12)
     3,                   MLT,         MODE,        MSTA,        MTIM
     4,                   NALIGN,      NANG,        NFRAC
      DOUBLE PRECISION    CROB(3),     DCXSOB(3),   DCXSTR(3),   DELTE(2)
     1,                   DNOB(3),     DNTR(3),     DRDT(2),     DUM1(3)
     2,                   DUM2(3),     E(2),        EAOB(3),     EATR(3)
     3,                   EDOT(2),     ENOB(3),     ENTR(3),     OMG(3)
     4,                   RM(2),       RTMG(2),     RTOB(6),     RTOBX(6)
     5,                   RTTR(6),     RTTRX(6),    SXOB(3),     SXTR(3)
     6,                   TT2CO(9),    TT2CT(9),    XSOB(6),     XSOBX(6)
     7,                   XSTR(6),     XSTRX(6),    Z1(6)
      DOUBLE PRECISION    BIAS,      CE,        CONST,     CX,        CY
     1,                   OBS,       DDOT,      DEN1,      DEN2,      DNORM
     2,                   F,         FTR,       OMEGA,     RATOB,     RATOBX
     3,                   RATTR,     RATTRX,    RSPLT,     R2OB,      R2TR
     4,                   SA,        SB,        SE,        SPDLT,     STA
     5,                   SX,        SY,        T,         TAU,       TB2CO
     6,                   TB2CT,     TT2BO,     TT2BT,     XTDOB,     XTDTR
     7,                   XTEOB,     XTETR,     XTNOB,     XTNTR,     XV
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION          DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE  (TT2CO(1),ENOB),   (TT2CO(4),EAOB),   (TT2CO(7),DNOB)
     1,            (TT2CT(1),ENTR),   (TT2CT(4),EATR),   (TT2CT(7),DNTR)
     2,            (SXOB(1),XTNOB),   (SXOB(2),XTEOB),   (SXOB(3),XTDOB)
     3,            (SXTR(1),XTNTR),   (SXTR(2),XTETR),   (SXTR(3),XTDTR)
     4,            (RM(1),RATOB),     (RM(2),RATTR)
      IF(MODE.EQ.3) GO TO 9
      DO 1 I=1,5
    1 STA(I+5)=STA(I)
      DO 2 I=1,9
    2 TT2BT(I)=TT2BO(I)
    9 RSPLT=1.D0/SPDLT
C     N,E,D VECTORS IN C FRAME AT RECEIVING AND TRANSMITTING TIMES
      CALL DMVTRN(TB2CO,TT2BO,TT2CO,1,3)
      CALL DMVTRN(TB2CT,TT2BT,TT2CT,1,3)
C     STATION VECTORS AT START AND END OF DOPPLER INTERVAL
      CALL DMVTRN(TB2CO,STA,RTOB,1,1)
      CALL DMVTRN(TB2CO(10),STA,RTOBX,1,1)
      CALL DMVTRN(TB2CT,STA(6),RTTR,1,1)
      CALL DMVTRN(TB2CT(10),STA(6),RTTRX,1,1)
      DO 10 I=1,3
   10 OMG(I)=TB2CO(I+6)*OMEGA
      CALL DCROSS(OMG,RTOB,RTOB(4))
      CALL DCROSS(OMG,RTTR,RTTR(4))
      CALL DCROSS(OMG,RTOBX,RTOBX(4))
      CALL DCROSS(OMG,RTTRX,RTTRX(4))
C     RANGE AND RANGE-RATE VECTORS AND MAGNITUDES
      DO 20 I=1,6
      XSOB(I)=XV(I)-RTOB(I)
      XSTR(I)=XV(I)-RTTR(I)
      XSOBX(I)=XV(I+6)-RTOBX(I)
   20 XSTRX(I)=XV(I+6)-RTTRX(I)
      R2OB=DDOT(XSOB,XSOB)
      R2TR=DDOT(XSTR,XSTR)
      RATOB=DSQRT(R2OB)
      RATTR=DSQRT(R2TR)
      RATOBX=DNORM(XSOBX)
      RATTRX=DNORM(XSTRX)
C     RANGE VECTORS IN TOPOCENTRIC COORDINATES
      CALL DMVTRN(TT2CO,XSOB,SXOB,2,1)
      CALL DMVTRN(TT2CT,XSTR,SXTR,2,1)
C     MAGNITUDE OF STATION VECTOR
      RTMG(1)=DNORM(RTOB)
      RTMG(2)=DNORM(RTTR)
      GO TO (30,60),NANG
C     X AND Y ANGLES
   30 GO TO (40,50),NALIGN
C     PRINCIPAL AXIS NORTH
```

```
   40 CALL DCROSS(ENOB,XSOB,CROB)
      OBS(1)=DATAN2(XTEOB,-XTDOB)
      OBS(17)=DATAN(XTNOB/DNORM(CROB))
      GO TO 60
C     PRINCIPAL AXIS EAST
   50 CALL DCROSS(EAOB,XSOB,CROB)
      OBS(1)=DATAN2(-XTNOB,-XTDOB)
      OBS(17)=DATAN(XTEOB/DNORM(CROB))
C     RANGING OBSERVABLE (TOTAL PATH LENGTH IN KM)
   60 OBS(33)=(RATOB+RATTR)
C     DIFFERENCED DOPPLER OBSERVABLE
      CONST=BIAS(2)*FTR*RSPLT
      OBS(49)=BIAS(1)*TAU+CONST*(OBS(33)-RATOBX-RATTRX)
      GO TO (70,160),NFRAC
C     REFRACTION CORRECTIONS
C     FIRST, ELEVATIONS AND ELEVATION CORRECTIONS
   70 E(1)=DATAN(-XTDOB/DSQRT(XTNOB*XTNOB+XTEOB*XTEOB))
      E(2)=DATAN(-XTDTR/DSQRT(XTNTR*XTNTR+XTETR*XTETR))
      DO 110 I=1,2
      DELTE(I)=0.
      IF(E(I).LT..01) GO TO 110
      SE=DSIN(E(I))
      CE=DCOS(E(I))
      IF(E(I).GE..17453293D0) GO TO 100
      T=(1.03585796D0-(.01072014D0-(.1279119D-7-.1227363D-7/E(I))/E(I))/
     1E(I))*STA(5*I-1)*CE/SE
   80 F=RTMG(I)/RM(I)
   90 DELTE(I)=T-F*((STA(5*I-1)+T*T/2.)*CE-T*SE)
      GO TO 110
  100 DELTE(I)=STA(5*I-1)*CE/SE
  110 CONTINUE
      GO TO (120,130),NANG
C     ANGLE CORRECTIONS
  120 SX=DSIN(OBS(1))
      CX=DCOS(OBS(1))
      SY=DSIN(OBS(17))
      CY=DCOS(OBS(17))
      DEN1=DSQRT(1.D0-CX*CX*CY*CY)
      OBS(1)=OBS(1)-SX*DELTE(1)/(CY*DEN1)
      OBS(17)=OBS(17)-CX*SY*DELTE(1)/DEN1
C     RANGING OBSERVABLE CORRECTION
  130 OBS(33)=OBS(33)+(STA(4)/(DSIN(E(1)+DELTE(1))*STA(5))+STA(9)/(DSIN(
     1E(2)+DELTE(2))*STA(10)))*1.D-3
C     DOPPLER CORRECTION
C     FIRST, ELEVATION TIME DERIVATIVES
      CALL DCROSS(DNOB,XSOB,DCXSOB)
      CALL DCROSS(DNTR,XSTR,DCXSTR)
      CALL DCROSS(DCXSOB,XSOB,DUM1)
      CALL DCROSS(DCXSTR,XSTR,DUM2)
      CALL DCROSS(XSOB,OMG,Z1)
      CALL DCROSS(XSTR,OMG,Z1(4))
      DEN1=R2OB*DNORM(DCXSOB)
      DEN2=R2TR*DNORM(DCXSTR)
      DO 140 I=1,3
      DUM1(I)=DUM1(I)/DEN1
      DUM2(I)=DUM2(I)/DEN2
      Z1(I)=Z1(I)+XSOB(I+3)
  140 Z1(I+3)=Z1(I+3)+XSTR(I+3)
      EDOT(1)=DDOT(DUM1,Z1)
      EDOT(2)=DDOT(DUM2,Z1(4))
      DO 150 I=1,2
      SA=DSIN(E(I)+DELTE(I))
      SB=DSIN(E(I)+DELTE(I)-EDOT(I)*TAU)
  150 DRDT(I)=STA(5*I-1)/STA(5*I)*(1.D0/SA-1.D0/SB)
      OBS(49)=OBS(49)+CONST*(DRDT(1)+DRDT(2))*1.D-3
  160 CONTINUE
      IF(OBS(1).LT.0.D0) OBS(1)=OBS(1)+6.2831853071795864
      IF(OBS(17).LT.0.D0) OBS(17)=OBS(17)+6.2831853071795864
      RETURN
      END
```

Subroutine: SBDATP

Purpose: Computes four Unified S-Band measurements and their partials.

Calling Sequence: CALL SBDATP(R2)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	R2	d	S^2	km^2	Slant range squared.

Common storages used: /DATCØM/

Subroutines required: DCRØSS,DDØT,DMVTRN,DNØRM

COMMON LOCATIONS

Common	Location	Name	Dimension	Description
DATCØM	C(1)	BIAS	d(2)	BIAS(1), Doppler bias frequency (cps) BIAS(2), Transponder retransmission ratio.
	C(133)	FTR	d	Doppler transmitter frequency (cps).
	C(135)	ØMEGA	d	Earth rotation rate (rad/sec).
	C(137)	SPDLT	d	Speed of light (km/sec).
	C(139)	STA	d(10)	STA(1-3) Receiving station postion in B-frame (km). STA(4-5) Refraction costants. STA(6-10) As above, but for transmitting station if three-way doppler
	C(159)	TAU	d	Doppler count interval (sec).
	C(161)	TB2CØ	d(18)	TB2CØ(1-9), B-frame to C-frame transform at end of doppler count interval at signal reception. TB2CØ(10-18) As above, but at beginning of doppler count interval.
	C(197)	TB2CT	d(18)	Same as TB2CØ, but at time of signal transmission.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors in B-frame at receiving station.
	C(251)	TT2BT	d(9)	Same as TT2BØ, but for transmitting station if three-way doppler.
	C(269)	XV	d(12)	XV(1-6), Spacecraft position and velocity at end of doppler count interval as signal leaves spacecraft. XV(7-12), Same as above, but at beginning of count interval (km,km/sec).
	C(293)	MLT		Speed of light partial option key*.
	C(294)	MØDE		Doppler mode +2, two-way +3, three-way
	C(295)	MSTA		Station position partial option key*.
	C(296)	MTIM		Station clock partial option key*.
	C(297)	NALIGN		Antenna principal axis alignment +1, North +2, East

COMMON LOCATIONS

Common	Location	Name	Dimension	Description
	C(298)	NANG		Angle inclusion option key*.
	C(299)	NFRAC		Refraction effects option key*. *+1, include +2, omit If angles are omitted, no angle partials are computed.
	C(5)	ØBS	d(64)	Measurements and partials. See Table 1, description of CBDAT.

```
$IBFTC MC13BL  XR3,M94,NODD,LIST
CMC13BL   UNIFIED S-BAND MEASUREMENTS AND PARTIALS
      SUBROUTINE SBDATP(R2OB)
      COMMON  /DATCOM/  BIAS(2),    OBS(64),    FTR,        OMEGA
     1,                 SPDLT,      STA(10),    TAU,        TB2CO(18)
     2,                 TB2CT(18),  TT2BO(9),   TT2BT(9),   XV(12)
     3,                 MLT,        MODE,       MSTA,       MTIM
     4,                 NALIGN,     NANG,       NFRAC
      DOUBLE PRECISION  CROB(3),    DCXSOB(3),  DCXSTR(3),  DELTE(2)
     1,                 DNOB(3),    DNTR(3),    DRDT(2),    DUM1(3)
     2,                 DUM2(3),    E(2),       EAOB(3),    EATR(3)
     3,                 EDOT(2),    ENOB(3),    ENTR(3),    OMG(3)
     4,                 RM(2),      RTMG(2),    RTOB(6),    RTOBX(6)
     5,                 RTTR(6),    RTTRX(6),   SXOB(3),    SXTR(3)
     6,                 TT2CO(9),   TT2CT(9),   XSOB(6),    XSOBX(6)
     7,                 XSTR(6),    XSTRX(6),   Z1(6)
      DOUBLE PRECISION  BIAS,     CE,       CONST,    CX,       CY
     1,                 OBS,      DDOT,     DEN1,     DEN2,     DNORM
     2,                 F,        FTR,      OMEGA,    RATOB,    RATOBX
     3,                 RATTR,    RATTRX,   RDOT,     RSPLT,    RSPLT2
     4,                 R2OB,     R2TR,     SA,       SB,       SE
     5,                 SPDLT,    STA,      SX,       SY,       S1
     6,                 S2,       T,        TAU,      TB2CO,    TB2CT
     7,                 TT2BO,    TT2BT,    XTDOB,    XTDTR,    XTEOB
     8,                 XTETR,    XTNOB,    XTNTR,    XV
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION          DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE  (TT2CO(1),ENOB),  (TT2CO(4),EAOB),  (TT2CO(7),DNOB)
     1,            (TT2CT(1),ENTR),  (TT2CT(4),EATR),  (TT2CT(7),DNTR)
     2,            (SXOB(1),XTNOB),  (SXOB(2),XTEOB),  (SXOB(3),XTDOB)
     3,            (SXTR(1),XTNTR),  (SXTR(2),XTETR),  (SXTR(3),XTDTR)
     4,            (RM(1),RATOB),    (RM(2),RATTR)
      IF(MODE.EQ.3) GO TO 9
      DO 1 I=1,5
    1 STA(I+5)=STA(I)
      DO 2 I=1,9
    2 TT2BT(I)=TT2BO(I)
    9 RSPLT=1.D0/SPDLT
      RSPLT2=RSPLT*RSPLT
C     N,E,D VECTORS IN C FRAME AT RECEIVING AND TRANSMITTING TIMES
      CALL DMVTRN(TB2CO,TT2BO,TT2CO,1,3)
      CALL DMVTRN(TB2CT,TT2BT,TT2CT,1,3)
C     STATION VECTORS AT START AND END OF DOPPLER INTERVAL
      CALL DMVTRN(TB2CO,STA,RTOB,1,1)
      CALL DMVTRN(TB2CO(10),STA,RTOBX,1,1)
      CALL DMVTRN(TB2CT,STA(6),RTTR,1,1)
      CALL DMVTRN(TB2CT(10),STA(6),RTTRX,1,1)
      DO 10 I=1,3
   10 OMG(I)=TB2CO(I+6)*OMEGA
      CALL DCROSS(OMG,RTOB,RTOB(4))
      CALL DCROSS(OMG,RTTR,RTTR(4))
      CALL DCROSS(OMG,RTOBX,RTOBX(4))
      CALL DCROSS(OMG,RTTRX,RTTRX(4))
C     RANGE AND RANGE-RATE VECTORS AND MAGNITUDES
      DO 20 I=1,6
      XSOB(I)=XV(I)-RTOB(I)
      XSTR(I)=XV(I)-RTTR(I)
      XSOBX(I)=XV(I+6)-RTOBX(I)
   20 XSTRX(I)=XV(I+6)-RTTRX(I)
      R2OB=DDOT(XSOB,XSOB)
      R2TR=DDOT(XSTR,XSTR)
      RATOB=DSQRT(R2OB)
      RATTR=DSQRT(R2TR)
      RATOBX=DNORM(XSOBX)
      RATTRX=DNORM(XSTRX)
C     RANGE VECTORS IN TOPOCENTRIC COORDINATES
      CALL DMVTRN(TT2CO,XSOB,SXOB,2,1)
      CALL DMVTRN(TT2CT,XSTR,SXTR,2,1)
C     MAGNITUDE OF STATION VECTOR
      RTMG(1)=DNORM(RTOB)
      RTMG(2)=DNORM(RTTR)
      GO TO (30,90),NANG
C     X AND Y ANGLES
```

```
   30 GO TO (40,50),NALIGN
C     PRINCIPAL AXIS NORTH
   40 CALL DCROSS(ENOB,XSOB,CROB)
      GO TO 60
C     PRINCIPAL AXIS EAST
   50 CALL DCROSS(EAOB,XSOB,CROB)
   60 S2=DDOT(CROB,CROB)
      S1=DSQRT(S2)
      GO TO (70,80),NALIGN
   70 OBS(1)=DATAN2(XTEOB,-XTDOB)
      OBS(17)=DATAN(XTNOB/S1)
      GO TO 90
   80 OBS(1)=DATAN2(-XTNOB,-XTDOB)
      OBS(17)=DATAN(XTEOB/S1)
C     RANGING OBSERVABLE (TOTAL PATH LENGTH IN KM)
   90 OBS(33)=RATOB+RATTR
C     DIFFERENCED DOPPLER OBSERVABLE
      CONST=BIAS(2)*FTR*RSPLT
      OBS(49)=BIAS(1)*TAU+CONST*(OBS(33)-RATOBX-RATTRX)
C     ANGLE-RATE OPERATOR (ANY ANGLE)
      CALL DCROSS(XSOB,OMG,Z1)
      CALL DCROSS(XSTR,OMG,Z1(4))
      DO 100 I=1,3
      Z1(I)=Z1(I)+XSOB(I+3)
  100 Z1(I+3)=Z1(I+3)+XSTR(I+3)
      GO TO (110,200),NFRAC
C     REFRACTION CORRECTIONS
C     FIRST, ELEVATIONS AND ELEVATION CORRECTIONS
  110 E(1)=DATAN(-XTDOB/DSQRT(XTNOB*XTNOB+XTEOB*XTEOB))
      E(2)=DATAN(-XTDTR/DSQRT(XTNTR*XTNTR+XTETR*XTETR))
      DO 150 I=1,2
      DELTE(I)=0.
      IF(E(I).LT..01) GO TO 150
      SE=DSIN(E(I))
      CE=DCOS(E(I))
      IF(E(I).GE..17453293D0) GO TO 140
      T=(1.03585796D0-(.01072014D0-(.1279119D-7-.1227363D-7/E(I))/E(I))/
     1E(I))*STA(5*I-1)*CE/SE
  120 F=RTMG(I)/RM(I)
  130 DELTE(I)=T-F*((STA(5*I-1)+T*T/2.)*CE-T*SE)
      GO TO 150
  140 DELTE(I)=STA(5*I-1)*CE/SE
  150 CONTINUE
      GO TO (160,170),NANG
C     ANGLE CORRECTIONS
  160 SX=DSIN(OBS(1))
      CX=DCOS(OBS(1))
      SY=DSIN(OBS(17))
      CY=DCOS(OBS(17))
      DEN1=DSQRT(1.D0-CX*CX*CY*CY)
      OBS(1)=OBS(1)-SX*DELTE(1)/(CY*DEN1)
      OBS(17)=OBS(17)-CX*SY*DELTE(1)/DEN1
C     RANGING OBSERVABLE CORRECTION
  170 OBS(33)=OBS(33)+(STA(4)/(DSIN(E(1)+DELTE(1))*STA(5))+STA(9)/(DSIN(
     1E(2)+DELTE(2))*STA(10)))*1.D-3
C     DOPPLER CORRECTION
C     FIRST, ELEVATION TIME DERIVATIVES
      CALL DCROSS(DNOB,XSOB,DCXSOB)
      CALL DCROSS(DNTR,XSTR,DCXSTR)
      CALL DCROSS(DCXSOB,XSOB,DUM1)
      CALL DCROSS(DCXSTR,XSTR,DUM2)
      DEN1=R2OB*DNORM(DCXSOB)
      DEN2=R2TR*DNORM(DCXSTR)
      DO 180 I=1,3
      DUM1(I)=DUM1(I)/DEN1
  180 DUM2(I)=DUM2(I)/DEN2
      EDOT(1)=DDOT(DUM1,Z1)
      EDOT(2)=DDOT(DUM2,Z1(4))
      DO 190 I=1,2
      SA=DSIN(E(I)+DELTE(I))
      SB=DSIN(E(I)+DELTE(I)-EDOT(I)*TAU)
  190 DRDT(I)=STA(5*I-1)/STA(5*I)*(1.D0/SA-1.D0/SB)
      OBS(49)=OBS(49)+CONST*(DRDT(1)+DRDT(2))*1.D-3
C     PARTIAL DERIVATIVES
  200 GO TO (210,290),NANG
```

```
C     ANGLE PARTIALS
C     PARTIALS WRT VEHICLE STATE
  210 CONTINUE
      IF(OBS(1).LT.0.D0) OBS(1)=OBS(1)+6.2831853071795864
      IF(OBS(17).LT.0.D0) OBS(17)=OBS(17)+6.2831853071795864
      CALL DCROSS(XSOB,CROB,OBS(18))
      DEN1=R2OB*S1
      DO 220 I=1,3
      OBS(I+1)=CROB(I)/S2
      OBS(I+4)=0.D0
      OBS(I+17)=OBS(I+17)/DEN1
  220 OBS(I+20)=0.D0
C     PARTIALS WRT MEASUREMENT BIAS
      OBS(8)=1.D0
      OBS(24)=1.D0
      GO TO (230,240),MTIM
C     PARTIALS WRT OBSERVING STATION CLOCK BIAS
  230 OBS(9)=DDOT(OBS(2),Z1)
      OBS(25)=DDOT(OBS(18),Z1)
  240 GO TO (250,260),MLT
C     PARTIALS WRT SPEED OF LIGHT
  250 OBS(10)=DDOT(OBS(2),XV(4))*RATOB*RSPLT2
      OBS(26)=DDOT(OBS(18),XV(4))*RATOB*RSPLT2
  260 GO TO (270,290),MSTA
C     PARTIALS WRT STATION LOCATION ERRORS
  270 CALL DMVTRN(TB2CO,OBS(2),OBS(11),2,1)
      CALL DMVTRN(TB2CO,OBS(18),OBS(27),2,1)
      DO 280 I=11,13
      OBS(I+3)=0.D0
      OBS(I+19)=0.D0
      OBS(I)=-OBS(I)
  280 OBS(I+16)=-OBS(I+16)
C     RANGING AND DOPPLER OBSERVABLE PARTIALS
C     PARTIALS WRT VEHICLE STATE
C     NORMALIZE RANGE VECTORS
  290 DO 300 I=1,3
      XSOB(I)=XSOB(I)/RATOB
      XSTR(I)=XSTR(I)/RATTR
      XSOBX(I)=XSOBX(I)/RATOBX
  300 XSTRX(I)=XSTRX(I)/RATTRX
C     PARTIALS WRT TRANSMITTING STATION LOCATION ARE USED IN THE VEHICLE
C     STATE PARTIALS AND WILL BE CALCULATED HERE
      CALL DCROSS(OMG,XSTRX,OBS(62))
      CALL DCROSS(OMG,OBS(62),DUM1)
      CALL DCROSS(OMG,DUM1,DUM2)
      DO 310 I=1,3
  310 DUM2(I)=(TAU*(OBS(I+61)+.5D0*TAU*(DUM1(I)+TAU*DUM2(I)/3.D0))+XSTRX
     1(I)-XSTR(I))*CONST
      DEN1=DDOT(XSTR,RTTR(4))*RSPLT
      CX=DDOT(DUM2,RTTR(4))*RSPLT
      CY=DDOT(DUM2,RTTRX(4))*RSPLT
      DO 320 I=1,3
      DUM1(I)=XSOB(I)+XSTR(I)
      OBS(I+33)=DUM1(I)*(1.D0-DEN1)
      OBS(I+36)=0.D0
      OBS(I+49)=DUM1(I)*(CONST+CX)-(XSOBX(I)+XSTRX(I))*(CONST+CY)
  320 OBS(I+52)=CONST*TAU*(XSOBX(I)+XSTRX(I))
C     PARTIALS WRT MEASUREMENT BIAS
      OBS(40)=1.D0
      OBS(56)=TAU
      GO TO (330,340),MTIM
C     PARTIALS WRT OBSERVING STATION CLOCK BIAS
  330 RDOT=DDOT(XSOB,XSOB(4))+DDOT(XSTR,XSTR(4))
      OBS(41)=RDOT
      OBS(57)=CONST*(RDOT-DDOT(XSOBX,XSOBX(4))-DDOT(XSTRX,XSTRX(4)))
  340 GO TO (350,360),MLT
C     PARTIALS WRT SPEED OF LIGHT
  350 OBS(42)=RATOB*RSPLT2*DDOT(OBS(34),XV(4))-RSPLT*(RATOB+RATTR)*(DEN1
     1+1.D0)
      OBS(58)=RSPLT2*(DDOT(OBS(50),XV(4))*RATOB-DDOT(OBS(50),XV(10))*RATS
     1OBX+DDOT(DUM2,RTTR(4))*(RATOB+RATTR)-DDOT(DUM2,RTTRX(4))*(RATOBX+RS
     2ATTRX))-CONST*RSPLT*(RATOB+RATTR-RATOBX-RATTRX)
  360 GO TO (370,410),MSTA
C     PARTIALS WRT STATION LOCATION ERRORS
  370 CALL DMVTRN(TB2CO,XSOB,OBS(43),2,1)
```

```
      CALL DMVTRN(TB2CT,XSTR,OBS(46),2,1)                               SBDP0225
      DO 380 I=43,48                                                    SBDP0226
  380 OBS(I)=-OBS(I)                                                    SBDP0227
      CALL DMVTRN(TB2CT,DUM2,OBS(62),2,1)                               SBDP0228
      CALL DCROSS(OMG,XSOBX,OBS(59))                                    SBDP0229
      CALL DCROSS(OMG,OBS(59),DUM1)                                     SBDP0230
      CALL DCROSS(OMG,DUM1,DUM2)                                        SBDP0231
      DO 390 I=1,3                                                      SBDP0232
  390 DUM2(I)=(TAU*(OBS(I+58)+.5D0*TAU*(DUM1(I)+TAU*DUM2(I)/3.D0))+XSOBXSBDP0233
     1(I)-XSOB(I))*CONST                                                SBDP0234
      CALL DMVTRN(TB2CO,DUM2,OBS(59),2,1)                               SBDP0235
      IF(MODE.EQ.3) GO TO 410                                           SBDP0236
C     ADD STATION LOCATION PARTIALS FOR TWO-WAY DOPPLER                 SBDP0237
      DO 400 I=59,61                                                    SBDP0238
      OBS(I-16)=OBS(I-16)+OBS(I-13)                                     SBDP0239
  400 OBS(I)=OBS(I)+OBS(I+13)                                           SBDP0240
  410 RETURN                                                            SBDP0241
      END
```

Subroutine: SBEV2

Purpose: Computes station and beacon critical events and occulting times. Creates an ordered array of those times when the vehicle comes into view or goes out of view for tracking stations and beacons.

Calling Sequence: CALL SBEV2 (XI,TI,KI,TIM,S,NRS,NCS,ISS,SECR,ISEE, ISTIM,STIME,MBØD)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	XI	(6)		km, km/sec	State vector of vehicle with respect to central body, equator and equinox of 1950.0
I	TI	(2)		days	Whole and fractional days from 1950 Jan 0.0
I	KI				Current central body number.
I	TIM	(2)		seconds	Minimum and maximum times (from epoch) defining interval of investigation.
I	S	(NRS,NCS)			Data array for stations/beacons
I	NRS,NCS				Dimensions of S.
I	ISS	(12)			Numbers of stations or beacons to be considered. List is terminated by a zero.
Ø	SECR	(12)		seconds	Time from epoch to make observations for stations or beacons given in corresponding ISS.
Ø	ISEE	(12)			Inview indicator, for station given in ISS. 0, in view -1, out of view.
Ø	ISTIM	(50)			Event indicator, in conjunction with STIME times. +, station into view -, station our of view 13, occulting starts 14, occulting stops

Input and Output - Continued

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	STIME	(40)		seconds	Ordered array of times from epoch at which ISTIME events occur.
I	MBØD	(4)			(1) Stations, set = 0. Beacons, number of body on which beacons are located. (2) = 0, no occulting bodies. (2)-(3) Occulting bodies. (4) Total number of critical events found.

Common storages used: /INPCØM/

Subroutines required: CRITA, CRITØ, DØT, EXINST, FNØRM, GHA, MNA, MTRN, NUTAIT, PARAB, QUARTC, RØTEQ, SHIF2, SØRDR, STAT, TFRAC, VNØRM, VTRN, VTRT

```
$IBFTC MC13JW   NOREF,M94,NODD,XR3
CMC13JW   SBEV2 A VERSION OF SBEV1
      SUBROUTINE SBEV2(XI,TI,KI,TIM,S,NRS,NCS,ISS,SECR,ISEE,ISTIM,STIME,
     1MBOD)
C
C SBEV2 DETERMINES AQUISITION TIMES FOR TRACKING STATIONS AND BEACONS
C       SUBROUTINES REQUIRED BY SBEV
C           CRITA,     CRITO,     DOT,       GHA,       MNA,       NUTAIT
C          PARAB,     QUARTC,    ROTEQ,     SETN,      SHIFT
C           SORDR,     STAT,      STEPD,     TFRAC,     VNORM,   MTRN,VTRN,VTRT
C
C       MBOD(1)=0 FOR TRACKING STATIONS.  =BODY ON WHICH BEACONS ARE MOUNT
C       MBOD(2) TO (4) = OCCULTING BODIES TO BE CONSIDERED
      DIMENSION     IMAX(15)
      DIMENSION  XI(6),       TI(3),       S(NRS,NCS)
     1,              ISS(12),       ISEE(12),      ISTIM(50),       SECR(12)
     2,            STIME(50),        AN(3,3),          A(3,3),        EN(3,3)
     3,                             ISEK(15),      XOUT(6,10),          XO(6)
     4,                TO(2),         TST(3),      STOR(3,15),           R(3)
     5,                RS(3),         RT(3),           SC(5),        TDUM(3)
     6,               DUM(3),         DT(2),           BB(2),        BRAD(2)
     7,             MBOD (4),         U1(3),           U2(3),        ISC(15)
     8,               UM(10),      BNAM(10),      BODC(10,8)
     9,               TIM(2),         XX(6),                          TT(2)
     1,           XKO(6,10),         AA(2)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)/EXIC/W(36),CRX(9,2)
      EQUIVALENCE  (C(11),BNAM),      (C(21),UM),      (C(11),BODC)
     1,            (C(5),DTR),        (C(6),SPMSD),    (C(7),RSPMSD)
      EQUIVALENCE              (NOR,POR)
      EQUIVALENCE  (IW(489),ITRIG),  (IW(490),KOUNT)
      DOUBLE PRECISION TWR
      LOGICAL NOCUL,IMOO,IBEA
      NOR=KI
      TSECO=TIM(1)
      IBEA=NCS.EQ.10
      LL=3
      LH=6
      NB=1
      IF(NCS.EQ.12) GO TO 1
      LL=2
      LH=5
      NB=MBOD(1)
1     CONTINUE
      IMOO=NB.EQ.2
      NN=-2
      TSTP=TSECO
      TSEC=TSECO
      FLTIM=TIM(2)
      DELTX=(FLTIM - TSECO)/3.3
      TISEC=TI(2)*SPMSD
      TIME=TI(1)+TI(2)
      CALL ROTEQ(TIME,A)
      CALL NUTAIT(TIME,WM,CR,DA,EN,EPSIL)
      DO 3 I=1,NCS
      IF(ISS(I)) 4,4,2
2     ISC(I)=ISS(I)
3     CONTINUE
      I=NCS+1
4     NKK=I-1
      NOCUL=MBOD(2).EQ.0
      IF(NOCUL) GO TO 7
      DO 6 I=2,4
      IF(MBOD(I)) 7,7,5
5     NKK=NKK+1
      ISC(NKK)=-MBOD(I)
6     CONTINUE
7     CONTINUE
      DO 17 I=1,NKK
17    IMAX(I)=1
      IF(IMOO) GO TO 8
      CALL GHA(TISEC,TI(1),GHAR,EN(2,1),WET)
      GHAR=GHAR*DTR
      GO TO 9
8     CONTINUE
      CALL MTRN(EN,A,AN)
```

```
      CALL MNA(TIME,WM,CR,DA,EPSIL,RR,G,GP,WW,EN)
      CALL MTRN(EN,AN,A)
      GHAR=0.
9     CONTINUE
      DO 11 I=1,3
      DO 11 J=1,3
11    AN(I,J)=A(I,J)
      KULT=-1
      BRAD(1)=BODC(NB,3)
      BRAD(2)=BODC(NB,4)
      WE=BODC(NB,7)
      NTT=2
      IF(NOCUL) NTT=4
      KKK=0
      NHEAD=1
       GHAM=0.
      GHAN=0.
      DO 111 I=1,6
 111  XO(I)=XI(I)
      TO(1)=TI(1)
      TO(2)=TI(2)
      GO TO 119
10    CONTINUE
 112  TWR=TSTP
      N=3
113   CONTINUE
      M=1
      CALL EXINST(TWR,M,N,XO,XO(4),W,CRX,10,ITRIG,KOUNT)
      IF(N.EQ.3) GO TO 118
      IF(M.EQ.0.OR.N.EQ.1) GO TO 113
      N=-N
      POR=CRX(7,N)
      NOR=NOR
      GO TO 112
  118 CALL TFRAC(TI(1),TI(2)+(TSTP-TSECO)*RSPMSD,TO(1),TO(2))
      TSEC=TSTP
 119  CONTINUE
      CALL SHIF2(NTT,NOR,TO,XO,NB,XOUT)
      GHAN=GHAR+(TSEC-TSECO)*WE
      CALL VTRN(AN,XOUT(1,NB),R)
      V2=DOT(XO(4),XO(4))
      RDV=DOT(XO,XO(4))
      R2=DOT(XO,XO)
      RN=SQRT(R2)
      DELTY=C(10)/(WE+SQRT(V2-RDV*RDV/R2)/RN)
      IF(DELTY.GT.DELTX) DELTY=DELTX
      TSTP=TSTP+DELTY
      NN=NN+1
      TST(3)=TST(2)
      TST(2)=TST(1)
      TST(1)=TSEC
      IF(NOCUL) GO TO 20
      KULT=-1
      RTB=VNORM(XOUT(1,NB),U1)
20    CONTINUE
      DO 80 II=1,NKK
      NST=ISC(II)
      IF(NST) 23,80,21
21    CONTINUE
      CALL STAT(S(LL,NST),GHAN,EN,RT,SC,BRAD)
      DO 22 I=1,3
22    RT(I)=R(I)-RT(I)
      CALL VTRT(EN,RT,RS)
      EL=ATAN(-RS(3)/SQRT(RS(1)*RS(1)+RS(2)*RS(2)))-S(LH,NST)
      GO TO 24
23    CONTINUE
      KB=-NST
      RFB=VNORM(XOUT(1,KB),U2)
      RB=BODC(KB,3)
      EL=DOT(U1,U2)-SQRT(RFB*RFB-RB*RB)/RFB
      DO 230 I=1,3
230   U2(I)=XOUT(I,NB)-XOUT(I,KB)
      RFEO=FNORM(U2)
      KULT=-1
      RBB=1.1*RTB
```

```
      IF(RFEO.GT.RBB) KULT=0                                            SBV20150
24    CONTINUE                                                          SBV20151
      STOR(3,II)=STOR(2,II)                                             SBV20152
      STOR(2,II)=STOR(1,II)                                             SBV20153
      STOR(1,II)=EL                                                     SBV20154
      IF(NN) 25,80,40                                                   SBV20155
25    IF(EL) 29,29,26                                                   SBV20156
26    IF(NST)27,28,28                                                   SBV20157
27    KKK=KKK+1                                                         SBV20158
      ISTIM(KKK)=13                                                     SBV20159
      STIME(KKK)=TSEC                                                   SBV20160
      ISEK(II)=3                                                        SBV20161
      CALL CRITA(TSEC,BNAM(KB),BNAM(NB),6    ,1)                        SBV20162
      DO 270 I=1,NCS                                                    SBV20163
270   SECR(I)=FLTIM                                                     SBV20164
      GO TO 80                                                          SBV20165
29    CONTINUE                                                          SBV20166
      ISEK(II)=1                                                        SBV20167
      IF(NST) 80,80,30                                                  SBV20168
28    CONTINUE                                                          SBV20169
      ISEK(II)=3                                                        SBV20170
      IF(IBEA) GO TO 295                                                SBV20171
      ISEK(II)=2                                                        SBV20172
      IF(EL.GT.S(7,NST)-S(LH,NST))GO TO 296                             SBV20173
295   ISEE(II)=0                                                        SBV20174
      SECR(II)=TSEC                                                     SBV20175
      KSW=2                                                             SBV20176
      GO TO 31                                                          SBV20177
296   IMAX(II)=2                                                        SBV20178
30    CONTINUE                                                          SBV20179
      ISEE(II)=-1                                                       SBV20180
      KSW=1                                                             SBV20181
31    CONTINUE                                                          SBV20182
      EL=EL+S(LH,NST)                                                   SBV20183
      CALL CRITO(TST(1),S(1,NST),KSW,RS,EL,6    ,NHEAD)                 SBV20184
      GO TO 80                                                          SBV20185
40    CONTINUE                                                          SBV20186
      IF(KULT.EQ.0) GO TO 80                                            SBV20187
      CALL PARAB(TST,STOR(1,II),AA)                                     SBV20188
      AD=0.                                                             SBV20189
      JJ=ISEK(II)                                                       SBV20190
      KSAV=IMAX(II)                                                     SBV20191
      IF(JJ.EQ.2) AD=S(7,NST)-S(6,NST)                                  SBV20192
41    CONTINUE                                                          SBV20193
      CALL QUARTC(STOR(2,II),AD,AA,DT,KK)                               SBV20194
      IF(KK) 42,44,44                                                   SBV20195
42    CONTINUE                                                          SBV20196
      IF(NST.LT.0.OR.IBEA) GO TO 79                                     SBV20197
      IF(JJ+KSAV.EQ.3) GO TO 43                                         SBV20198
      IF(JJ.EQ.3.AND.AA(1).GT.0.) JJ=2                                  SBV20199
      GO TO 79                                                          SBV20200
43    JJ=3                                                              SBV20201
      AD=0.                                                             SBV20202
      GO TO 41                                                          SBV20203
44    CONTINUE                                                          SBV20204
      GO TO (445,440,445),JJ                                            SBV20205
440   IF(STOR(2,II).LT.0..AND.KSAV.EQ.1) GO TO 43                       SBV20206
445   CONTINUE                                                          SBV20207
      DELT=TST(2)-TST(3)                                                SBV20208
      DTEST=DT(2)                                                       SBV20209
      IF(JJ+KSAV.EQ.4) DTEST=DT(1)                                      SBV20210
      IF(DTEST) 45,45,79                                                SBV20211
45    CONTINUE                                                          SBV20212
      IF(DELT+DTEST.LT.0.) GO TO 42                                     SBV20213
      GO TO(47,46,47),JJ                                                SBV20214
46    DSAV=DT(1)                                                        SBV20215
 47   CONTINUE                                                          SBV20216
48    CONTINUE                                                          SBV20217
      DUM(1)=STOR(2,II)                                                 SBV20218
      DUM(3)=STOR(3,II)                                                 SBV20219
      TDUM(1)=TST(2)- TST(1)                                            SBV20220
      TDUM(2)=TDUM(1) + DTEST                                           SBV20221
      TDUM(3)=TST(3)- TST(1)                                            SBV20222
      NJAZ=1                                                            SBV20223
49    CONTINUE                                                          SBV20224
```

```
      NJAZ=NJAZ+1                                                       SBV20225
      TWR=TDUM(2)+TSEC                                                  SBV20226
  490 CONTINUE                                                          SBV20227
      N=3                                                               SBV20228
  491 CONTINUE                                                          SBV20229
      M=1                                                               SBV20230
      CALL EXINST(TWR,M,N,XX,XX(4),W,CRX,10,ITRIG,KOUNT)                SBV20231
      IF(N.EQ.3) GO TO 495                                              SBV20232
      IF(M.EQ.0.OR.N.EQ.1) GO TO 491                                    SBV20233
      N=-N                                                              SBV20234
      POR=CRX(7,N)                                                      SBV20235
      NOR=NOR                                                           SBV20236
      GO TO 490                                                         SBV20237
  495 CONTINUE                                                          SBV20238
      CALL TFRAC(TO(1),TO(2)+TDUM(2)*RSPMSD,TT(1),TT(2))                SBV20239
      CALL SHIF2(NTT,NOR,TT,XX,NB,XKO)                                  SBV20240
      IF(NST) 50,51,51                                                  SBV20241
   50 CONTINUE                                                          SBV20242
      RFB=VNORM(XKO(1,KB),U2)                                           SBV20243
      EL=DOT(U1,U2)-SQRT(RFB*RFB-RB*RB)/RFB                             SBV20244
      CON=.0001                                                         SBV20245
      GO TO 55                                                          SBV20246
   51 CONTINUE                                                          SBV20247
      CALL VTRN(AN,XKO(1,NB),XX)                                        SBV20248
      GHAM=GHAN + TDUM(2)*WE                                            SBV20249
      CALL STAT(S(LL,NST),GHAM,EN,RT,SC,BRAD)                           SBV20250
      DO 54 I=1,3                                                       SBV20251
   54 RT(I)=XX(I)-RT(I)                                                 SBV20252
      CALL VTRT(EN,RT,RS)                                               SBV20253
      EL=ATAN(-RS(3)/SQRT(RS(1)*RS(1)+RS(2)*RS(2)))-S(LH,NST)           SBV20254
      CON=.002                                                          SBV20255
   55 IF(ABS(EL-AD)-CON) 61,56,56                                       SBV20256
   56 CONTINUE                                                          SBV20257
      IF(NJAZ.GT.10) GO TO 61                                           SBV20258
      DUM(2)=EL                                                         SBV20259
      CALLPARAB(TDUM,DUM,BB)                                            SBV20260
      CALL QUARTC(DUM(2),AD,BB,DT,KK)                                   SBV20261
      IF(KK) 42,57,57                                                   SBV20262
   57 CONTINUE                                                          SBV20263
      DTEST=DT(2)                                                       SBV20264
      IF(JJ+KSAV.EQ.4) DTEST=DT(1)                                      SBV20265
      K=1                                                               SBV20266
      IF(DTEST.GT.0.) K=3                                               SBV20267
      DUM(K)=DUM(2)                                                     SBV20268
      TDUM(K)=TDUM(2)                                                   SBV20269
      TDUM(2)=TDUM(2)+DTEST                                             SBV20270
      GO TO 49                                                          SBV20271
   61 CONTINUE                                                          SBV20272
      KKK=KKK+1                                                         SBV20273
      STIME(KKK)=TST(1)+TDUM(2)                                         SBV20274
      GO TO(62,65,68),JJ                                                SBV20275
   62 CONTINUE                                                          SBV20276
      IF(NST) 63,64,64                                                  SBV20277
   63 ISTIM(KKK)=13                                                     SBV20278
      CALL CRITA(STIME(KKK),BNAM(KB),BNAM(NB),6    ,1)                  SBV20279
      JJ=3                                                              SBV20280
      IF(AA(2)) 41,79,79                                                SBV20281
   64 CONTINUE                                                          SBV20282
      ISTIM(KKK) = II                                                   SBV20283
      EL=EL+S(LH,NST)                                                   SBV20284
      CALL CRITO(STIME(KKK),S(1,NST),2,       RS,EL,6    ,NHEAD)        SBV20285
      JJ=2                                                              SBV20286
      IF(IBEA) JJ=3                                                     SBV20287
      IF(AA(2).GT.0.) GO TO 79                                          SBV20288
      AD=S(7,NST)-S(6,NST)                                              SBV20289
      IF(IBEA) AD=0.                                                    SBV20290
      GO TO 41                                                          SBV20291
   65 CONTINUE                                                          SBV20292
      GO TO(66,75),KSAV                                                 SBV20293
   66 ISTIM(KKK)=-II                                                    SBV20294
      EL=EL+S(LH,NST)                                                   SBV20295
      CALL CRITO(STIME(KKK),S(1,NST),1,       RS,EL,6    ,NHEAD)        SBV20296
      KSAV=2                                                            SBV20297
      IF(TST(2)+DSAV.LE.TDUM(2)+TST(1).OR.TST(2)+DSAV.GT.TST(1))GO TO 79SBV20298
      IF(DSAV.GT.0.) GO TO 67                                           SBV20299
```

```
      DTEST=DSAV                                                        SBV20300
      GO TO 48                                                          SBV20301
   67 CONTINUE                                                          SBV20302
      DUM(1)=STOR(1,II)                                                 SBV20303
      DUM(3)=STOR(2,II)                                                 SBV20304
      TDUM(1)=0.                                                        SBV20305
      TDUM(3)=TST(2)-TST(1)                                             SBV20306
      TDUM(2)=TDUM(3)+DSAV                                              SBV20307
      GO TO 49                                                          SBV20308
   68 CONTINUE                                                          SBV20309
      IF(NST) 69,70,70                                                  SBV20310
   69 ISTIM(KKK)=14                                                     SBV20311
      CALL CRITA(STIME(KKK),BNAM(KB),BNAM(NB),6    ,2)                  SBV20312
      GO TO 71                                                          SBV20313
   70 CONTINUE                                                          SBV20314
      ISTIM(KKK)= -II                                                   SBV20315
      EL=EL+S(LH,NST)                                                   SBV20316
      CALL CRITO(STIME(KKK),S(1,NST),1,       RS,EL,6    ,NHEAD)        SBV20317
   71 JJ=1                                                              SBV20318
      IF(AA(2)) 79,79,41                                                SBV20319
   75 CONTINUE                                                          SBV20320
      KSAV=1                                                            SBV20321
      ISTIM(KKK) = II                                                   SBV20322
      EL=EL+S(LH,NST)                                                   SBV20323
      CALL CRITO(STIME(KKK),S(1,NST),2,       RS,EL,6    ,NHEAD)        SBV20324
      JJ=3                                                              SBV20325
      AD=0.                                                             SBV20326
      GO TO 41                                                          SBV20327
   79 CONTINUE                                                          SBV20328
      ISEK(II)=JJ                                                       SBV20329
      IMAX(II)=KSAV                                                     SBV20330
   80 CONTINUE                                                          SBV20331
      IF(NN) 10,10,81                                                   SBV20332
   81 IF(TST(2).GE.FLTIM) GO TO 82                                      SBV20333
      IF(KKK-38) 10,83,83                                               SBV20334
   82 KKK=KKK+1                                                         SBV20335
      STIME(KKK)=FLTIM + 99999.                                         SBV20336
   83 CONTINUE                                                          SBV20337
      MBOD(4)=KKK                                                       SBV20338
      CALL SORDR(STIME,ISTIM,KKK)                                       SBV20339
      RETURN                                                            SBV20340
      END                                                               SBV20341
```

Subroutine: SBTEST

Purpose: To read Unified S-Band System raw data tapes, decode the data, convert units, and, using subroutine PØLYFT, test for outliers and write the edited data in the DCP format on Unit 12. (Raw data are assumed to be in the IBM Preprocessor format.)

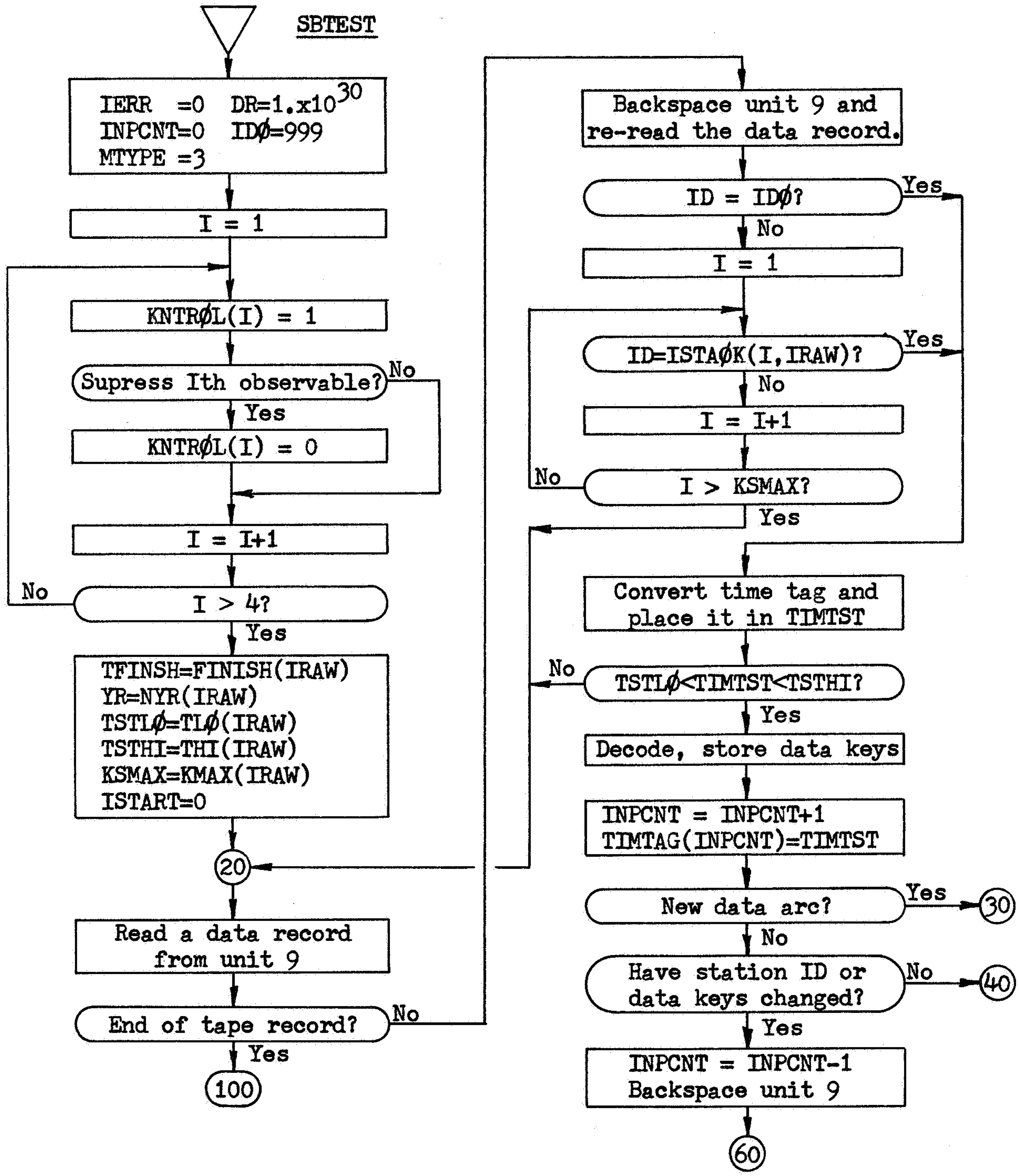
Calling Sequence: CALL SBTEST(IERR)

Input and Output

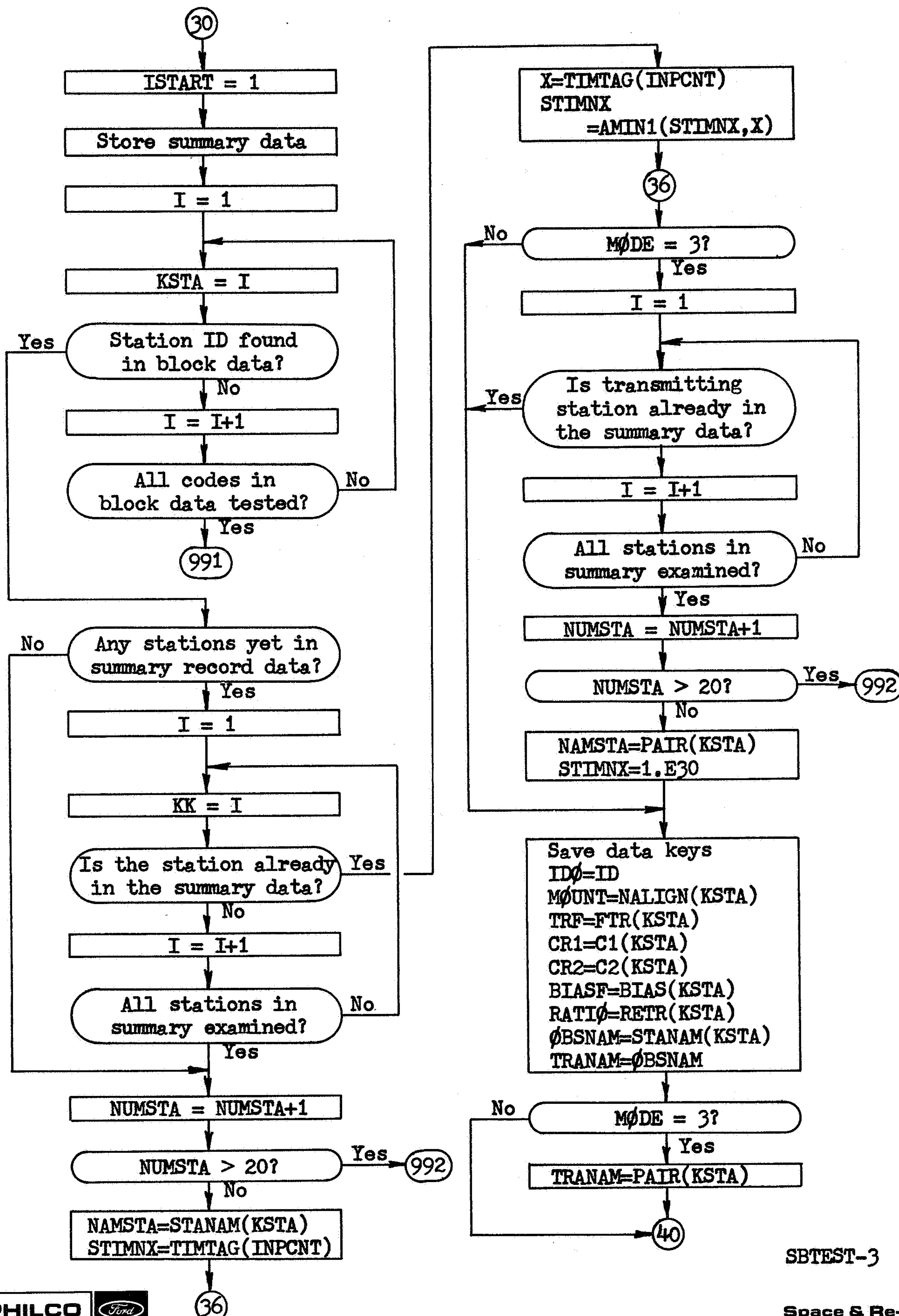
I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	IERR				Error flag. Set = 2 by SBTEST if more than 20 stations are accumlated. Set = 0 otherwise.

Common storages used: /TRKCØM/,/ØUTCØM/,/TSTCØM/,/DATCØM/,/SUMCØM/

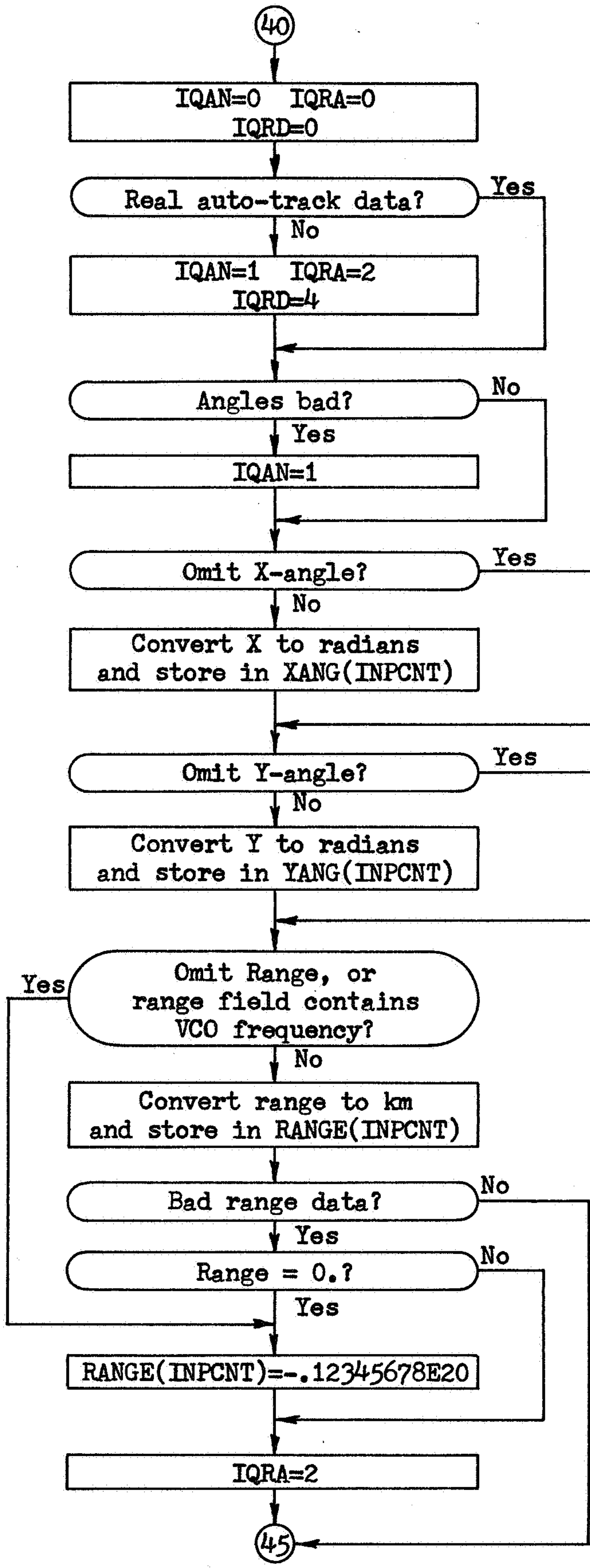
Subroutines required: DATINP, PØLYFT


SBTEST
IERR =0 DR=1.x10^30
INPCNT=0 IDØ=999
MTYPE =3
I = 1
KNTRØL(I) = 1
Supress Ith observable?
No
Yes
KNTRØL(I) = 0
I = I+1
No
I > 4?
Yes
TFINSH=FINISH(IRAW)
YR=NYR(IRAW)
TSTLØ=TLØ(IRAW)
TSTHI=THI(IRAW)
KSMAX=KMAX(IRAW)
ISTART=0
20
Read a data record
from unit 9
End of tape record?
No
Yes
100
Backspace unit 9 and
re-read the data record.
ID = IDØ?
Yes
No
I = 1
ID=ISTAØK(I,IRAW)?
Yes
No
I = I+1
No
I > KSMAX?
Yes
Convert time tag and
place it in TIMTST
No
TSTLØ<TIMTST<TSTHI?
Yes
Decode, store data keys
INPCNT = INPCNT+1
TIMTAG(INPCNT)=TIMTST
New data arc?
Yes
30
No
Have station ID or
data keys changed?
No
40
Yes
INPCNT = INPCNT-1
Backspace unit 9
60

SBTEST-2

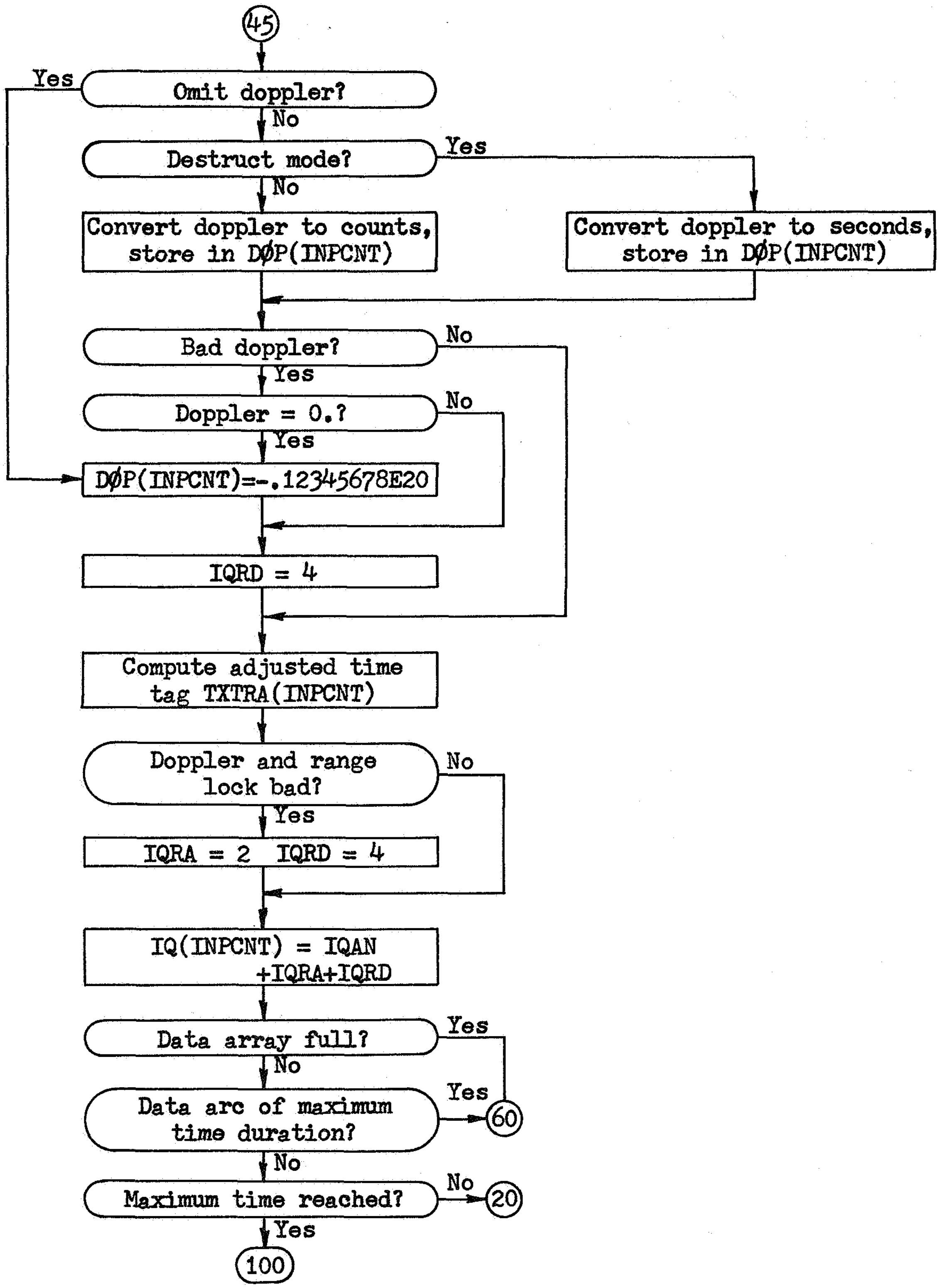

30
ISTART = 1
Store summary data
I = 1
KSTA = I
Station ID found in block data?
Yes
No
I = I+1
All codes in block data tested?
No
Yes
991
Any stations yet in summary record data?
No
Yes
I = 1
KK = I
Is the station already in the summary data?
Yes
No
I = I+1
All stations in summary examined?
No
Yes
NUMSTA = NUMSTA+1
NUMSTA > 20?
Yes
992
No
NAMSTA=STANAM(KSTA)
STIMNX=TIMTAG(INPCNT)
36
X=TIMTAG(INPCNT)
STIMNX =AMIN1(STIMNX,X)
36
MØDE = 3?
No
Yes
I = 1
Is transmitting station already in the summary data?
Yes
I = I+1
All stations in summary examined?
No
Yes
NUMSTA = NUMSTA+1
NUMSTA > 20?
Yes
992
No
NAMSTA=PAIR(KSTA)
STIMNX=1.E30
Save data keys
IDØ=ID
MØUNT=NALIGN(KSTA)
TRF=FTR(KSTA)
CR1=C1(KSTA)
CR2=C2(KSTA)
BIASF=BIAS(KSTA)
RATIØ=RETR(KSTA)
ØBSNAM=STANAM(KSTA)
TRANAM=ØBSNAM
MØDE = 3?
No
Yes
TRANAM=PAIR(KSTA)
40

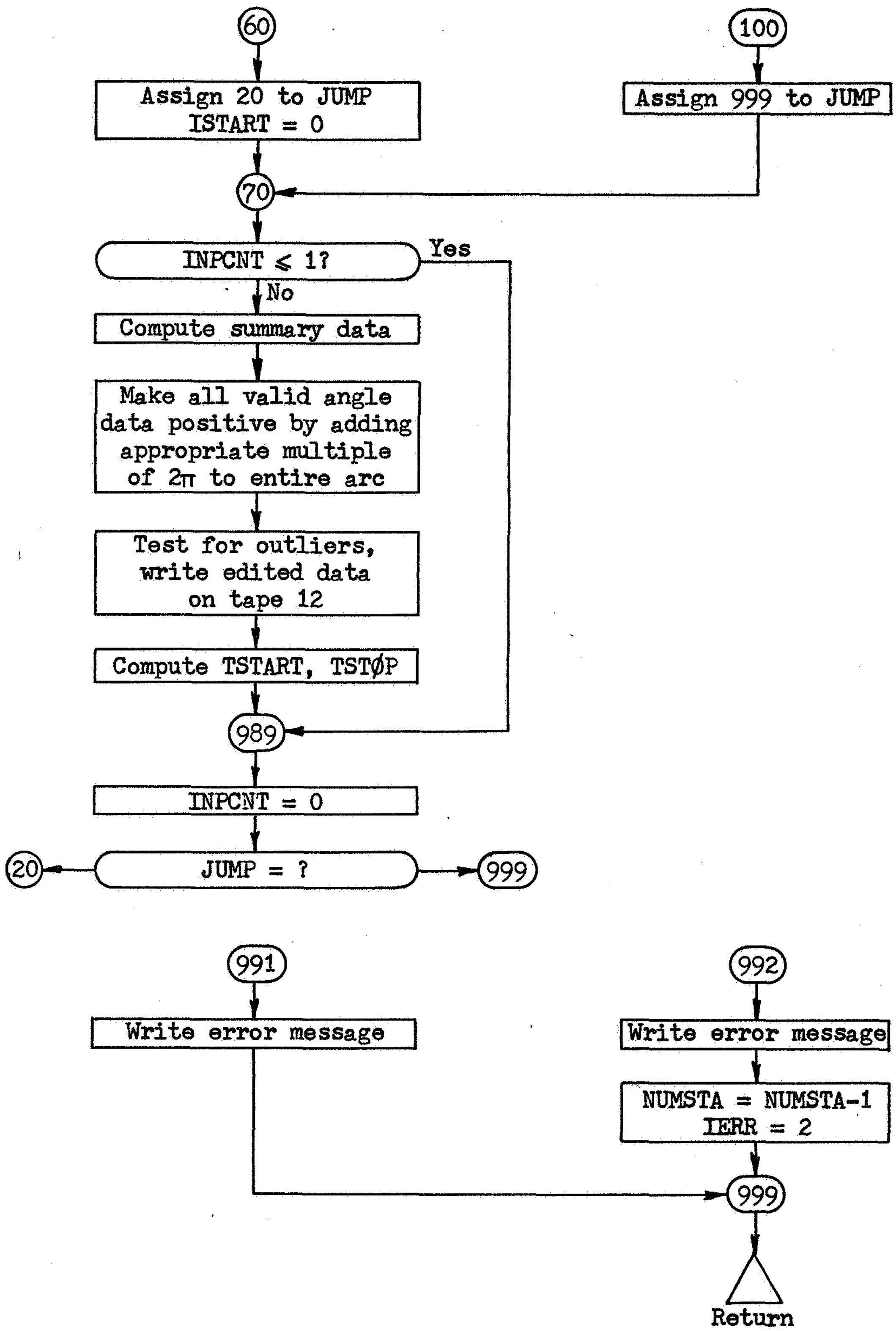
SBTEST-3



40
IQAN=0 IQRA=0
IQRD=0
Real auto-track data?
Yes
No
IQAN=1 IQRA=2
IQRD=4
Angles bad?
No
Yes
IQAN=1
Omit X-angle?
Yes
No
Convert X to radians
and store in XANG(INPCNT)
XANG(INPCNT)=-.12345678E20
IQAN=1
Omit Y-angle?
Yes
No
Convert Y to radians
and store in YANG(INPCNT)
YANG(INPCNT)=-.12345678E20
IQAN=1
Omit Range, or
range field contains
VCO frequency?
Yes
No
Convert range to km
and store in RANGE(INPCNT)
Bad range data?
No
Yes
Range = 0.?
No
Yes
RANGE(INPCNT)=-.12345678E20
IQRA=2
45

SBTEST-4


45
Omit doppler?
Yes
No
Destruct mode?
Yes
No
Convert doppler to counts, store in DØP(INPCNT)
Convert doppler to seconds, store in DØP(INPCNT)
Bad doppler?
No
Yes
Doppler = 0.?
No
Yes
DØP(INPCNT)=-.12345678E20
IQRD = 4
Compute adjusted time tag TXTRA(INPCNT)
Doppler and range lock bad?
No
Yes
IQRA = 2 IQRD = 4
IQ(INPCNT) = IQAN +IQRA+IQRD
Data array full?
Yes
No
Data arc of maximum time duration?
Yes
60
No
Maximum time reached?
No
20
Yes
100


60
Assign 20 to JUMP
ISTART = 0
100
Assign 999 to JUMP
70
INPCNT ≤ 1?
Yes
No
Compute summary data
Make all valid angle data positive by adding appropriate multiple of 2π to entire arc
Test for outliers, write edited data on tape 12
Compute TSTART, TSTØP
989
INPCNT = 0
20
JUMP = ?
999
991
Write error message
992
Write error message
NUMSTA = NUMSTA-1
IERR = 2
999
Return

SBTEST-6

```
$IBFTC MC134W  M94,NODD,XR3
CMC134W    S-BAND PROCESSOR, IBM FORMAT
      SUBROUTINE SBTEST(IERR)
      COMMON  /TRKCOM/CTRK(700)
      DOUBLE PRECISION  FTR(50),  C1(50),   C2(50),   BIAS(50), RETR(50)
      DIMENSION   STANAM(50),   KODSTA(50),   NALIGN(50),   PAIR(50)
      EQUIVALENCE              (CTRK(1),STANAM),     (CTRK(51),KODSTA)
     1,  (CTRK(101),NALIGN),   (CTRK(151),FTR),      (CTRK(251),C1)
     2,  (CTRK(351),C2),       (CTRK(451),BIAS),     (CTRK(551),RETR)
     3,  (CTRK(651),PAIR)
C
      COMMON  /OUTCOM/COUT(40)
      DOUBLE PRECISION  ONTIME,   TFIRST,   TLAST,    TAU,      TRF
     1,        CR1,      CR2,      DR,       BIASF,    RATIO
      EQUIVALENCE              (COUT(1),NPAIR),      (COUT(2),NEOT)
     1,  (COUT(3),OBSNAM),     (COUT(4),TRANAM),     (COUT(5),NRCD)
     2,  (COUT(6),NPTBLK),     (COUT(7),KONT),       (COUT(8),MTYPE)
     3,  (COUT(9),MOUNT),      (COUT(10),MOAD),      (COUT(11),DELT)
     4,  (COUT(12),KTAU),      (COUT(13),ONTIME),    (COUT(15),TFIRST)
     5,  (COUT(17),TLAST),     (COUT(19),TAU),       (COUT(21),TRF)
     6,  (COUT(23),CR1),       (COUT(25),CR2),       (COUT(27),DR)
     7,  (COUT(29),BIASF),     (COUT(31),RATIO),     (COUT(33),NB1)
     8,  (COUT(34),NB2),       (COUT(35),NB3),       (COUT(36),NB4)
C
      COMMON  /TSTCOM/CTEST(400)
      DOUBLE PRECISION  FINISH(10),    TLO(10),       THI(10)
      DIMENSION   NYR(10),       NPTS(10),      NSTEP(10),     NDEG(10)
     1,           CSD(4,10),     KNTROL(4),     IFOMIT(4,10), KMAX(10)
     2,           ISTAOK(20,10)
      EQUIVALENCE              (CTEST(1),NYR),       (CTEST(11),NPTS)
     1,  (CTEST(21),NSTEP),    (CTEST(31),NDEG),     (CTEST(41),CSD)
     2,  (CTEST(81),IRAW),     (CTEST(82),KNTROL),   (CTEST(87),FINISH)
     3,  (CTEST(107),IFOMIT), (CTEST(147),KMAX),     (CTEST(157),ISTAOK)
     4,  (CTEST(357),TLO),     (CTEST(377),THI)
C
      COMMON  /DATCOM/CDAT(2400)
      DOUBLE PRECISION TIMTAG(300)
      DIMENSION   TXTRA(300),   IQ(300),      XANG(300),     YANG(300)
     1,           RANGE(300),   DOP(300),     RDAT(300,4)
      EQUIVALENCE              (CDAT(1),TIMTAG),     (CDAT(601),TXTRA)
     1,  (CDAT(901),IQ),       (CDAT(1201),XANG),    (CDAT(1501),YANG)
     2,  (CDAT(1801),RANGE),   (CDAT(2101),DOP),     (XANG,RDAT)
C
      COMMON  /SUMCOM/SUMARY(56)
      DOUBLE PRECISION TSTART,TSTOP
      DIMENSION   HEADER(11),   STIMNX(20)
      REAL NAMSTA(20)
      EQUIVALENCE              (SUMARY(1),HEADER),   (SUMARY(12),NUMSTA)
     1,  (SUMARY(13),NAMSTA), (SUMARY(33),TSTART), (SUMARY(35),TSTOP)
     2,  (SUMARY(37),STIMNX)
C
      DIMENSION   IDOP(2),      IDC(5),       DI(3)
      DOUBLE PRECISION  TMAX,     TFINSH,   TIMTST,   TSTLO,    TSTHI
C
C     THESE TWO ARRAYS MUST HAVE THE SAME DIMENSION AS THE ARRAYS IN
C     DATCOM.  THE DIMENSION IS EQUAL TO THE VARIABLE INPMAX.
C     THE VARIABLE INPMAX IS SET BY A DATA STATEMENT
C     IN THIS SUBROUTINE.
C     TO AVOID TRUNCATING GODDARD SYSTEM DATA FRAMES, INPMAX MUST
C     BE A MULTIPLE OF FOUR
      DIMENSION   HOLD(300),    ITEM(300)
C
C
  602 FORMAT(A2)
  603 FORMAT(23X,2O6)
  605 FORMAT(I2,5I1,F3.0,2F2.0,1X,F4.1,2X,O6,2X,O6,2X,F11.4,2X,2O6)
  901 FORMAT(18H0*** STATION CODE,I3,29H DOES NOT EXIST IN BLOCK DATA/
     16X,34HPROCESSING OF THIS TAPE ABANDONED./
     26X,38HPROGRAM PROCEEDS TO NEXT TAPE, IF ANY.)
  902 FORMAT(24H0***  TOO MANY STATIONS./
     16X,49HPROGRAM CONTROL PASSES TO TERMINATION OPERATIONS.)
      DATA EOTIND/2HED/
      DATA INPMAX/300/
      DATA TMAX/.432D5/
      DATA TEST/-.12345678E20/
```

```
      IERR=0                                                            SBTS0075
      INPCNT=0                                                          SBTS0076
      MTYPE=3                                                           SBTS0077
      IDO=999                                                           SBTS0078
C     SET RANGE AMBIGUITY LARGE                                         SBTS0079
C     IF RANGE REALLY DOES RECYCLE, YOU WILL CHANGE THIS                SBTS0080
      DR=1.D30                                                          SBTS0081
      DO 1 I=1,4                                                        SBTS0082
      KNTROL(I)=1                                                       SBTS0083
      IF(IFOMIT(I,IRAW).NE.0) KNTROL(I)=0                               SBTS0084
    1 CONTINUE                                                          SBTS0085
      YR=NYR(IRAW)                                                      SBTS0086
      TFINSH=FINISH(IRAW)                                               SBTS0087
      TSTLO=TLO(IRAW)                                                   SBTS0088
      TSTHI=THI(IRAW)                                                   SBTS0089
      KSMAX=KMAX(IRAW)                                                  SBTS0090
      ISTART=0                                                          SBTS0091
C     TEST FOR END OF TAPE INDICATOR                                    SBTS0092
   20 READ(9,602) ARCIND                                                SBTS0093
      IF(ARCIND.EQ.EOTIND) GO TO 100                                    SBTS0094
      BACKSPACE 9                                                       SBTS0095
C     READ A LINE OF DATA                                               SBTS0096
      READ(9,605) ID,IDC,DAY,HR,XMIN,SEC,IX,IY,RAN,IDOP                 SBTS0097
C     FIND IF STATION NUMBER IS UNCHANGED                               SBTS0098
      IF(ID.EQ.IDO) GO TO 22                                            SBTS0099
C     TEST IF STATION ID VALID                                          SBTS0100
      DO 21 I=1,KSMAX                                                   SBTS0101
      IF(ID.EQ.ISTAOK(I,IRAW)) GO TO 22                                 SBTS0102
   21 CONTINUE                                                          SBTS0103
      GO TO 20                                                          SBTS0104
   22 CONTINUE                                                          SBTS0105
C     DECODE DATA KEYS AND CONVERT TIME TAG                             SBTS0106
      DI(1)=YR*100.+1.                                                  SBTS0107
      DI(2)=(DAY*100.+HR)*100.+XMIN                                     SBTS0108
      DI(3)=SEC                                                         SBTS0109
      CALL DATINP(DI,TIMTST)                                            SBTS0110
C     TEST TIME TAG FOR VALID RANGE OF TIMES                            SBTS0111
      IF(TIMTST.LT.TSTLO.OR.TIMTST.GT.TSTHI) GO TO 20                   SBTS0112
      MODE=IDC(3)/2+1                                                   SBTS0113
      IF(MODE.EQ.4) MODE=3                                              SBTS0114
      IF(MODE.NE.2.AND.MODE.NE.3) GO TO 20                              SBTS0115
      INPCNT=INPCNT+1                                                   SBTS0116
      TIMTAG(INPCNT)=TIMTST                                             SBTS0117
      IQ(INPCNT)=0                                                      SBTS0118
      IDSRCT=IDC(1)/4                                                   SBTS0119
      I=IDC(1)-4*IDSRCT                                                 SBTS0120
      NCOUNT=I/2                                                        SBTS0121
      IRFREQ=I-2*NCOUNT                                                 SBTS0122
      IRA=IDC(5)/4                                                      SBTS0123
      I=IDC(5)-IRA*4                                                    SBTS0124
      IRAQ=I/2                                                          SBTS0125
      IDOPOK=I-2*IRAQ                                                   SBTS0126
      IF(ISTART.EQ.0) GO TO 30                                          SBTS0127
C     IF KEYS CHANGE, A NEW DATA ARC IS FORCED                          SBTS0128
      IF(ID.EQ.IDO.AND.IDSRCT.EQ.IDSRCO.AND.NCOUNT.EQ.NCOUNO.AND.MODE.EQSBTS0129
     1.MODEO) GO TO 40                                                  SBTS0130
      INPCNT=INPCNT-1                                                   SBTS0131
      BACKSPACE 9                                                       SBTS0132
      GO TO 60                                                          SBTS0133
C     NEW DATA ARC                                                      SBTS0134
   30 ISTART=1                                                          SBTS0135
      MOAD=MODE                                                         SBTS0136
      KTAU=0                                                            SBTS0137
      IF(IDSRCT.EQ.0) KTAU=1                                            SBTS0138
      IF(KTAU.EQ.1) GO TO 301                                           SBTS0139
      TAU=77824.                                                        SBTS0140
      IF(NCOUNT.EQ.0) TAU=778240.                                       SBTS0141
C     FIND STATION ID IN BLOCK DATA                                     SBTS0142
  301 DO 31 I=1,50                                                      SBTS0143
      KSTA=I                                                            SBTS0144
      IF(ID.EQ.KODSTA(I)) GO TO 32                                      SBTS0145
   31 CONTINUE                                                          SBTS0146
      GO TO 991                                                         SBTS0147
   32 IF(NUMSTA.EQ.0) GO TO 34                                          SBTS0148
C     IS STATION NAME ALREADY IN SUMMARY DATA                           SBTS0149
```

```
      DO 33 I=1,NUMSTA
      KK=I
      IF(NAMSTA(I).EQ.STANAM(KSTA)) GO TO 35
   33 CONTINUE
C     NEW STATION NAME
   34 NUMSTA=NUMSTA+1
      IF(NUMSTA.GT.20) GO TO 992
      NAMSTA(NUMSTA)=STANAM(KSTA)
      STIMNX(NUMSTA)=TIMTAG(INPCNT)
      GO TO 36
C     OLD STATION NAME
C     UPDATE STATION FIRST ON TIME
   35 X=TIMTAG(INPCNT)
      STIMNX(KK)=AMIN1(STIMNX(KK),X)
C     TEST FOR DISTINCT TRANSMITTING STATION
   36 IF(MODE.NE.3) GO TO 38
      DO 37 I=1,NUMSTA
      IF(NAMSTA(I).EQ.PAIR(KSTA)) GO TO 38
   37 CONTINUE
      NUMSTA=NUMSTA+1
      IF(NUMSTA.GT.20) GO TO 992
      NAMSTA(NUMSTA)=PAIR(KSTA)
      STIMNX(NUMSTA)=1.E30
   38 IDO=ID
      IDSRCO=IDSRCT
      NCOUNO=NCOUNT
      MODEO=MODE
      OBSNAM=STANAM(KSTA)
      TRANAM=OBSNAM
      IF(MODE.EQ.3) TRANAM=PAIR(KSTA)
      MOUNT=NALIGN(KSTA)
      TRF=FTR(KSTA)
      CR1=C1(KSTA)
      CR2=C2(KSTA)
      BIASF=BIAS(KSTA)
      RATIO=RETR(KSTA)
   40 CONTINUE
C     LOAD ROW OF DATA ARRAY
      IQAN=0
      IQRA=0
      IQRD=0
      IF(IDC(2)/2.EQ.3) GO TO 4000
      IQRD=4
      IQRA=2
      IQAN=1
 4000 IF(MOD(IDC(2),2).EQ.0) IQAN=1
      IF(KNTROL(1).NE.0) GO TO 4001
      XANG(INPCNT)=TEST
      IQAN=1
      GO TO 410
 4001 ISGN=IX/131072
      IF(ISGN.EQ.0) GO TO 401
      IX=IX-131072
      GO TO 41
  401 IX=-IX
   41 XANG(INPCNT)=FLOAT(IX)*1.19841288E-5
  410 IF(KNTROL(2).NE.0) GO TO 4002
      YANG(INPCNT)=TEST
      IQAN=1
      GO TO 420
 4002 ISGN=IY/131072
      IF(ISGN.EQ.0) GO TO 402
      IY=IY-131072
      GO TO 42
  402 IY=-IY
   42 YANG(INPCNT)=FLOAT(IY)*1.19841288E-5
  420 IF(KNTROL(3).EQ.0) GO TO 404
      IF(IRFREQ.NE.1) GO TO 404
   43 RU=1.048397E-3
      IF(IDC(4).EQ.7) RU=1.050694E-3
      RANGE(INPCNT)=RU*RAN
   44 IF(IRA.EQ.1) GO TO 45
      IF(RAN.EQ.0.) GO TO 404
      GO TO 405
  404 RANGE(INPCNT)=TEST
```

```
  405 IQRA=2                                                             SBTS0225
   45 IF(KNTROL(4).EQ.0) GO TO 470                                       SBTS0226
      IRDOP=IDOP(1)/131072                                               SBTS0227
      IDOP(1)=IDOP(1)-IRDOP*131072                                       SBTS0228
      IIDOP=IDOP(1)*262144+IDOP(2)                                       SBTS0229
      IF(IDSRCT.EQ.0) GO TO 46                                           SBTS0230
      DOP(INPCNT)=FLOAT(IIDOP)*1.E-8                                     SBTS0231
      GO TO 47                                                           SBTS0232
   46 DOP(INPCNT)=IIDOP-IIDOPO                                           SBTS0233
      IIDOPO=IIDOP                                                       SBTS0234
   47 IF(IDOPOK.EQ.1) GO TO 49                                           SBTS0235
      IF(IIDOP.NE.0) GO TO 48                                            SBTS0236
  470 DOP(INPCNT)=TEST                                                   SBTS0237
   48 IQRD=4                                                             SBTS0238
   49 TXTRA(INPCNT)=TIMTAG(INPCNT)-TIMTAG(1)                              SBTS0239
      IF(IRDOP.NE.0) GO TO 490                                           SBTS0240
      IQRA=2                                                             SBTS0241
      IQRD=4                                                             SBTS0242
  490 IQ(INPCNT)=IQAN+IQRA+IQRD                                          SBTS0243
      IF(INPCNT.EQ.INPMAX) GO TO 60                                      SBTS0244
      IF(TIMTAG(INPCNT)-TIMTAG(1).GE.TMAX) GO TO 60                      SBTS0245
      IF(TIMTAG(INPCNT).GE.TFINSH) GO TO 100                             SBTS0246
      GO TO 20                                                           SBTS0247
   60 ASSIGN 20 TO JUMP                                                  SBTS0248
      ISTART=0                                                           SBTS0249
      GO TO 70                                                           SBTS0250
  100 ASSIGN 999 TO JUMP                                                 SBTS0251
   70 IF(INPCNT.LE.1) GO TO 989                                          SBTS0252
C     FROM HERE TO EFN 705 IS ALL TO FIND THE MOST COMMON TIME INTERVAL  SBTS0253
C     BETWEEN MEASUREMENTS.  THIS WILL BE PUT INTO DELT                  SBTS0254
      MANY=1                                                             SBTS0255
      ITEM(1)=1                                                          SBTS0256
      HOLD(1)=TIMTAG(2)-TIMTAG(1)                                        SBTS0257
      IF(INPCNT.LE.2) GO TO 704                                          SBTS0258
      DO 703 I=3,INPCNT                                                  SBTS0259
      HERE=TIMTAG(I)-TIMTAG(I-1)                                         SBTS0260
      DO 701 J=1,MANY                                                    SBTS0261
      IF(HERE.EQ.HOLD(J)) GO TO 702                                      SBTS0262
  701 CONTINUE                                                           SBTS0263
      MANY=MANY+1                                                        SBTS0264
      HOLD(MANY)=HERE                                                    SBTS0265
      ITEM(MANY)=1                                                       SBTS0266
      GO TO 703                                                          SBTS0267
  702 ITEM(J)=ITEM(J)+1                                                  SBTS0268
  703 CONTINUE                                                           SBTS0269
  704 MOST=1                                                             SBTS0270
      ISRCH=1                                                            SBTS0271
      DO 705 I=1,MANY                                                    SBTS0272
      IF(MOST.GE.ITEM(I)) GO TO 705                                      SBTS0273
      MOST=ITEM(I)                                                       SBTS0274
      ISRCH=I                                                            SBTS0275
  705 CONTINUE                                                           SBTS0276
      DELT=HOLD(ISRCH)                                                   SBTS0277
      IF(KTAU.EQ.1) TAU=DELT                                             SBTS0278
      ONTIME=TIMTAG(1)                                                   SBTS0279
C     MAKE ALL ANGLES     POSITIVE IF NECESSARY                          SBTS0280
C     BUT RETAIN CONTINUITY                                              SBTS0281
      DO 710 I=1,2                                                       SBTS0282
      DO 710 K=1,INPCNT                                                  SBTS0283
      IF(RDAT(K,I).EQ.TEST) GO TO 710                                    SBTS0284
      IF(RDAT(K,I).GE.0.) GO TO 710                                      SBTS0285
      DO 71 L=1,INPCNT                                                   SBTS0286
      IF(RDAT(L,I).EQ.TEST) GO TO 71                                     SBTS0287
      RDAT(L,I)=RDAT(L,I)+6.28318531                                     SBTS0288
   71 CONTINUE                                                           SBTS0289
  710 CONTINUE                                                           SBTS0290
C     DO POLYNOMIAL FITS, TEST FOR OUTLIERS                              SBTS0291
C     PUT DATA ON TAPE 12                                                SBTS0292
      CALL POLYFT(INPCNT)                                                SBTS0293
      TSTART=DMIN1(TSTART,TIMTAG(1))                                     SBTS0294
      TSTOP=DMAX1(TSTOP,TIMTAG(INPCNT))                                  SBTS0295
  989 INPCNT=0                                                           SBTS0296
      GO TO JUMP,(20,999)                                                SBTS0297
  991 WRITE(6,901) ID                                                    SBTS0298
      GO TO 999                                                          SBTS0299
```

```
  992 WRITE(6,902)
      NUMSTA=NUMSTA-1
      IERR=2
  999 RETURN
      END
```

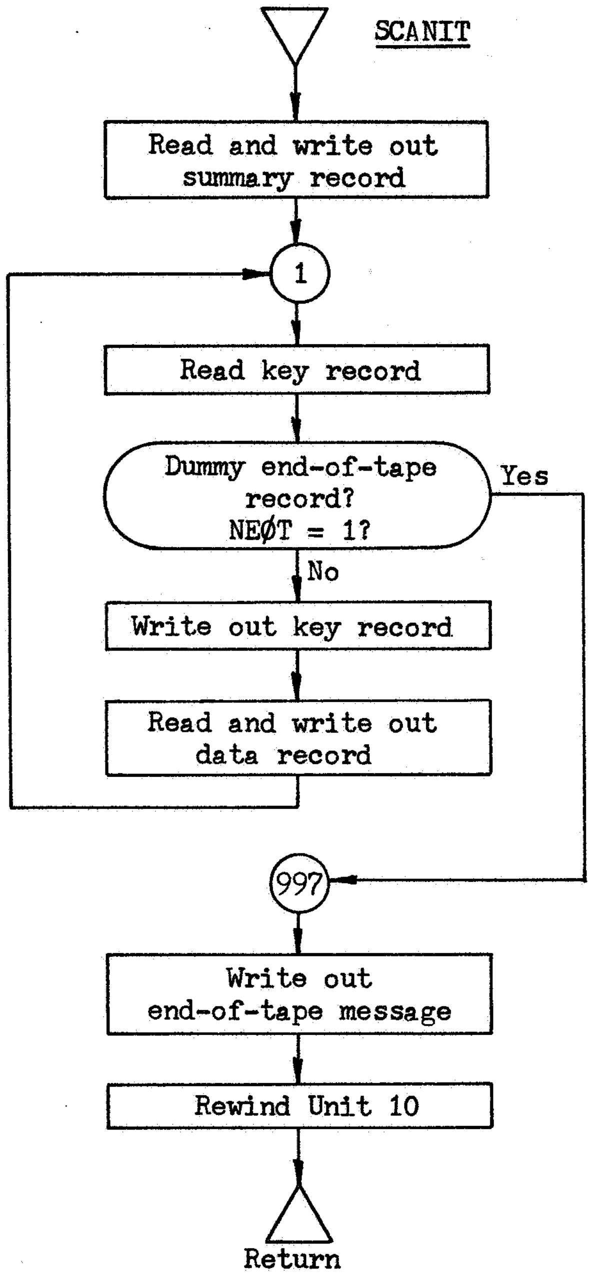
Subroutine: SCANIT

Purpose: To read the final edited data tape on Unit 10 and write all the data on the System Output Tape. May be used as a separate program for reading and writing out the contents of either edited or simulated data tapes.

Calling Sequence: CALL SCANIT

Common storages used: /SUMCØM/,/ØUTCØM/,/MESCØM/

Subroutines required: DATØUP


SCANIT
Read and write out summary record
1
Read key record
Dummy end-of-tape record? NEØT = 1?
Yes
No
Write out key record
Read and write out data record
997
Write out end-of-tape message
Rewind Unit 10
Return

```
$IBFTC MC134P  M94,NODD,XR3
CMC134P    PROGRAM TO READ EDITED DATA TAPE                              SCAN0001
C      THIS SUBROUTINE MAY BE USED AS A SEPARATE PROGRAM FOR             SCAN0002
C      READING EDITED OR SIMULATED DATA TAPES                            SCAN0003
C      (IT WAS ORIGINALLY WRITTEN FOR JUST THAT PURPOSE.  THAT IS WHY    SCAN0004
C      IT DUPLICATES SOME OF THE EFFORT OF SUBROUTINE EGRESS.)           SCAN0005
C      TO USE SEPARATELY, REMOVE SUBROUTINE AND RETURN CARDS             SCAN0006
C      BE SURE TO FURNISH A SUBROUTINE DATOUP                            SCAN0007
      SUBROUTINE SCANIT                                                  SCAN0008
      COMMON  /SUMCOM/SUMM(56)                                           SCAN0009
      DOUBLE PRECISION  TSTART,    TSTOP                                 SCAN0010
      DIMENSION   HEADER(11),    STIMNX(20)                              SCAN0011
      REAL        NAMSTA(20)                                             SCAN0012
      EQUIVALENCE             (SUMM(1),HEADER),     (SUMM(12),NUMSTA)    SCAN0013
     1,   (SUMM(13),NAMSTA),   (SUMM(33),TSTART),   (SUMM(35),TSTOP)     SCAN0014
     2,   (SUMM(37),STIMNX)                                              SCAN0015
      COMMON  /OUTCOM/BUFF(40)                                           SCAN0016
      DOUBLE PRECISION  ONTIME,    TFIRST,    TLAST,    TAU,       FTR   SCAN0017
     1,        CR1,       CR2,       DR,        BIAS,     RATIO,   DBUFF(10)SCAN0018
      DIMENSION   NB(4),        SSD(4)                                   SCAN0019
      EQUIVALENCE             (BUFF(1),NREC),       (BUFF(2),NEOT)       SCAN0020
     2,   (BUFF(3),OBSNAM),    (BUFF(4),TRANAM),    (BUFF(5),NRCD)       SCAN0021
     2,   (BUFF(6),NPTS),      (BUFF(7),KONT),      (BUFF(8),MTYPE)      SCAN0022
     3,   (BUFF(9),NALIGN),    (BUFF(10),MODE),     (BUFF(11),DELT)      SCAN0023
     4,   (BUFF(12),KTAU),     (BUFF(13),ONTIME),   (BUFF(15),TFIRST)    SCAN0024
     5,   (BUFF(17),TLAST),    (BUFF(19),TAU),      (BUFF(21),FTR)       SCAN0025
     6,   (BUFF(23),CR1),      (BUFF(25),CR2),      (BUFF(27),DR)        SCAN0026
     7,   (BUFF(29),BIAS),     (BUFF(31),RATIO),    (BUFF(33),NB)        SCAN0027
     8,   (BUFF(13),DBUFF),    (BUFF(37),SSD)                            SCAN0028
      COMMON  /MESCOM/DATA(85,6)                                         SCAN0029
      DIMENSION    DI(3)                                                 SCAN0030
  601 FORMAT(13H1TAPE HEADER,,11A6/10H0NUMSTA = ,I2/14H0STATION NAMES/(  SCAN0031
     16(3X,A6)))                                                         SCAN0032
  602 FORMAT(17H0STATION ON-TIMES/(6E20.8))                              SCAN0033
  603 FORMAT(10H0TSTART = ,D23.16)                                       SCAN0034
  604 FORMAT(9H0TSTOP = ,D23.16)                                         SCAN0035
  605 FORMAT(4H0   N,4X,3HEOT,3X,6HOBSNAM,3X,6HTRANAM,3X,4HNRCD,3X,4HNPTSSCAN0036
     1,3X,6HKONTIN,3X,5HMTYPE,3X,6HNALIGN,3X,4HMODE,3X,10HMEAS. INT.,3X,SCAN0037
     24HKTAU/2I5,4X,A6,3X,A6,I6,I7,2I8,I9,I8,F12.2,I7//10X,6HONTIME,18X,SCAN0038
     36HTFIRST,19X,5HTLAST,20X,3HTAU,21X,3HFTR/5D24.16//12X,2HC1,22X,2HCSCAN0039
     42,20X,7HR. AMB.,18X,4HBIAS,20X,5HRATIO/5D24.16/                    SCAN0040
     57H NB1 = ,I2,3X,6HNB2 = ,I2,3X,6HNB3 = ,I2,3X,6HNB4 = ,I2/         SCAN0041
     620H0STANDARD DEVIATIONS/4E20.8)                                    SCAN0042
  606 FORMAT(22H0ONTIME IN DATE FORMAT)                                  SCAN0043
  607 FORMAT(22H0TFIRST IN DATE FORMAT)                                  SCAN0044
  608 FORMAT(21H0TLAST IN DATE FORMAT)                                   SCAN0045
  609 FORMAT(5H0DATA/(I3,5E20.8))                                        SCAN0046
  610 FORMAT(18H0END OF TAPE, N = ,I4,9H, NEOT = ,I2,9H, NRCD = ,I2)     SCAN0047
      READ(10) SUMM                                                      SCAN0048
      WRITE(6,601) HEADER,NUMSTA,(NAMSTA(I),I=1,NUMSTA)                  SCAN0049
      WRITE(6,602) (STIMNX(I),I=1,NUMSTA)                                SCAN0050
      WRITE(6,603) TSTART                                                SCAN0051
      CALL DATOUP(TSTART,DI,0)                                           SCAN0052
      WRITE(6,604) TSTOP                                                 SCAN0053
      CALL DATOUP(TSTOP,DI,0)                                            SCAN0054
    1 READ(10) BUFF                                                      SCAN0055
      IF(NEOT.EQ.1) GO TO 997                                            SCAN0056
      READ(10) DATA                                                      SCAN0057
      WRITE(6,605) (BUFF(I),I=1,12),(DBUFF(I),I=1,10),NB,SSD             SCAN0058
      WRITE(6,606)                                                       SCAN0059
      CALL DATOUP(ONTIME,DI,0)                                           SCAN0060
      WRITE(6,607)                                                       SCAN0061
      CALL DATOUP(TFIRST,DI,0)                                           SCAN0062
      WRITE(6,608)                                                       SCAN0063
      CALL DATOUP(TLAST,DI,0)                                            SCAN0064
      WRITE(6,609) ((DATA(I,J),J=1,6),I=1,NPTS)                          SCAN0065
      GO TO 1                                                            SCAN0066
  997 WRITE(6,610) NREC,NEOT,NRCD                                        SCAN0067
      REWIND 40                                                          SCAN0068
      RETURN                                                             SCAN0069
      END                                                                SCAN0070
```

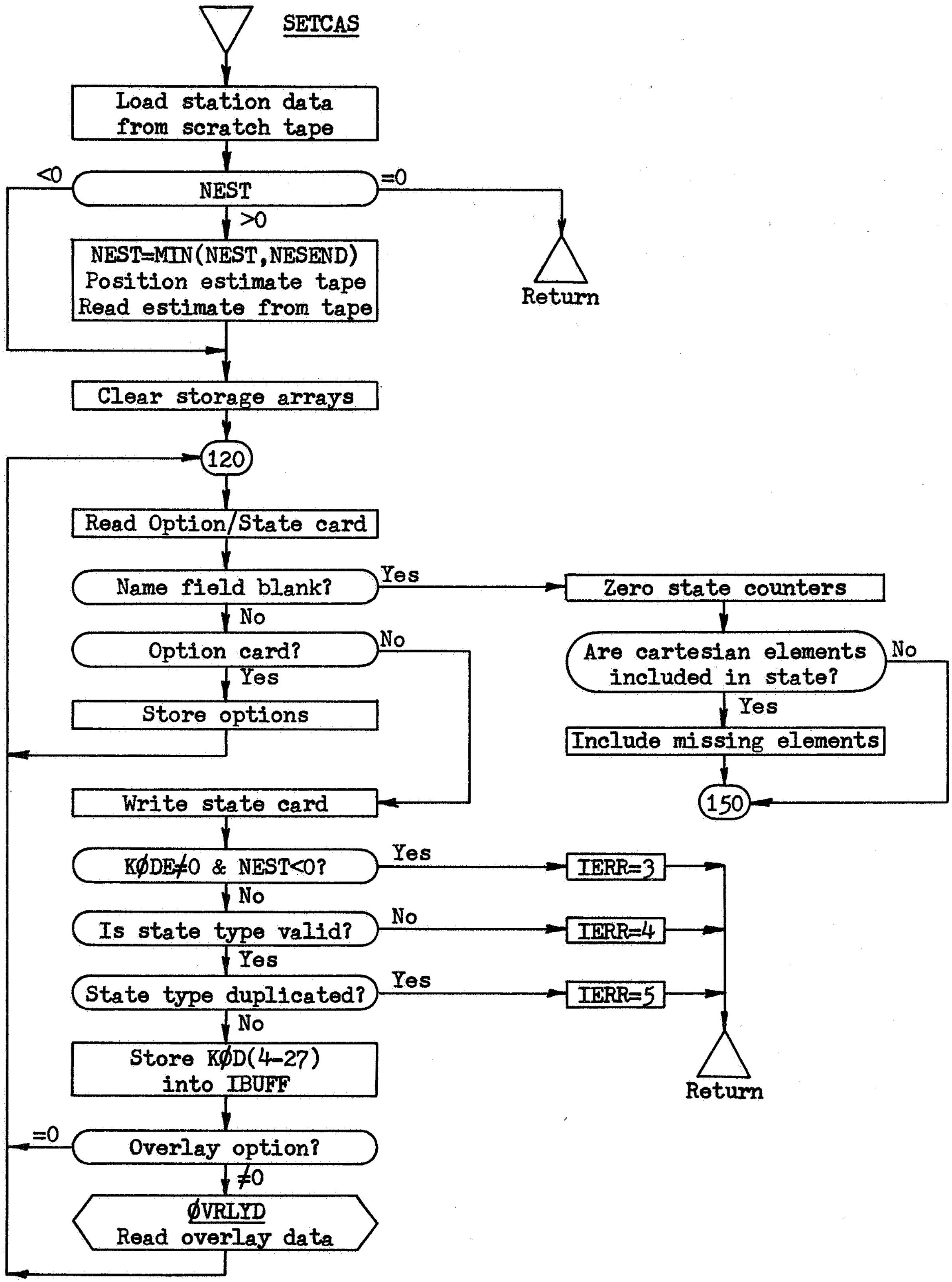
Subroutine: SETCAS

Purpose: Reads input data defining state elements and program control for a sequence of data analyses, and set up internal storage arrays. Constructs the *a priori* estimate of state and its covariance.

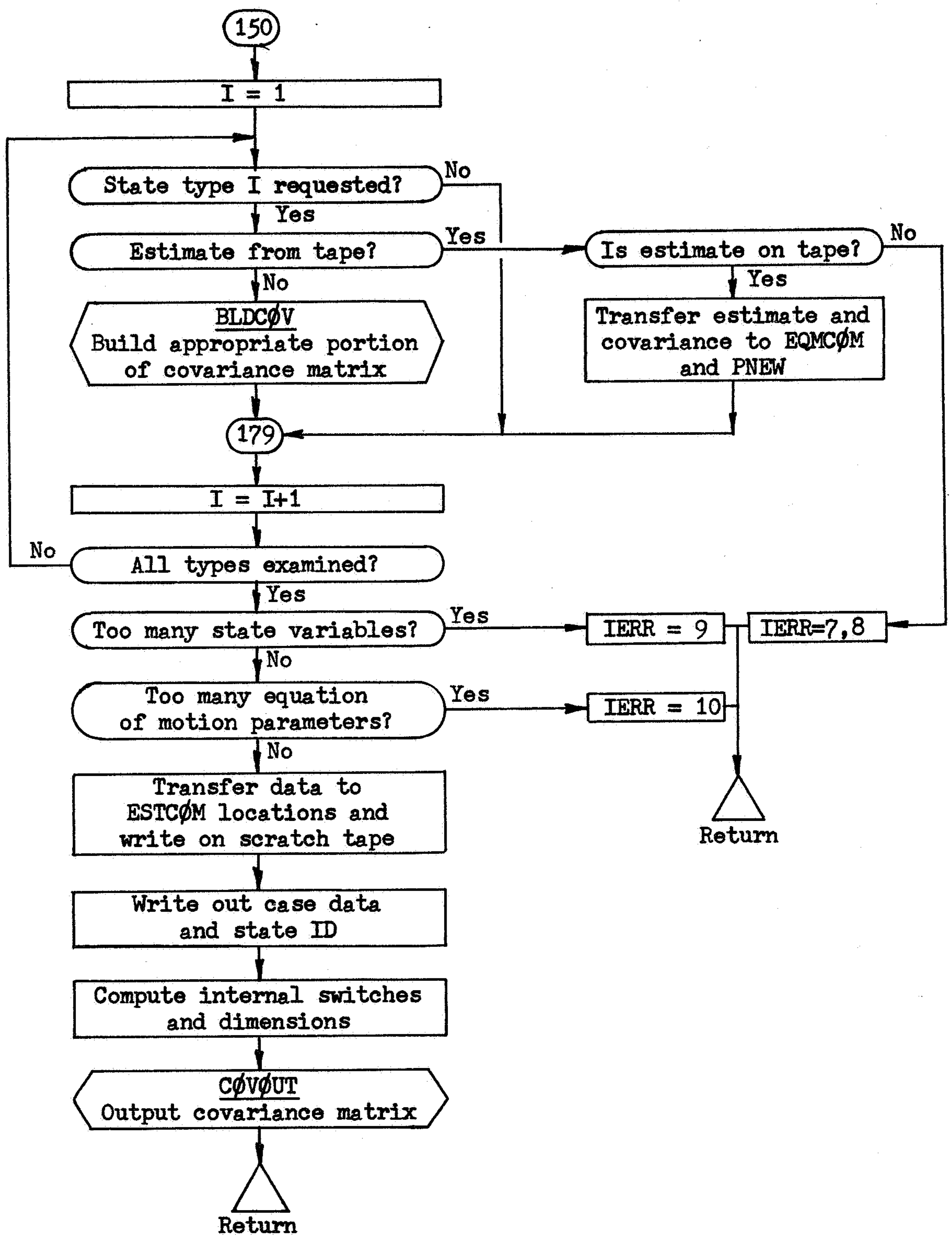
Calling Sequence: CALL SETCAS

Common storages used: //204 cells,/DCPCØM/,/EQMCØM/,/ESØCØM/,/ESTCØM/, /ES1CØM/

Subroutines required: BLDCØV,CØVØUT,DATINP,DATØUP,ØVRLYD


SETCAS
Load station data from scratch tape
NEST
<0
=0
>0
Return
NEST=MIN(NEST,NESEND)
Position estimate tape
Read estimate from tape
Clear storage arrays
120
Read Option/State card
Name field blank?
Yes
No
Zero state counters
Option card?
No
Yes
Are cartesian elements included in state?
No
Yes
Store options
Include missing elements
Write state card
150
KØDE≠0 & NEST<0?
Yes
IERR=3
No
Is state type valid?
No
IERR=4
Yes
State type duplicated?
Yes
IERR=5
No
Return
Store KØD(4-27) into IBUFF
Overlay option?
=0
≠0
ØVRLYD
Read overlay data

SETCAS-2


150
I = 1
State type I requested?
No
Yes
Estimate from tape?
Yes
Is estimate on tape?
No
Yes
Transfer estimate and covariance to EQMCØM and PNEW
No
BLDCØV
Build appropriate portion of covariance matrix
179
I = I+1
No
All types examined?
Yes
Too many state variables?
Yes
IERR = 9
IERR=7,8
No
Too many equation of motion parameters?
Yes
IERR = 10
Return
No
Transfer data to ESTCØM locations and write on scratch tape
Write out case data and state ID
Compute internal switches and dimensions
CØVØUT
Output covariance matrix
Return

```
$IBFTC MC13SW  XR3,M94,NODD,LIST
      SUBROUTINE SETCAS                                                 SCAS0001
C     CASE INPUT AND INITIALIZATION                                     SCAS0002
C                                                                       SCAS0003
      COMMON          /DCPCOM/CDCP(900)                                 SCAS0004
      DOUBLE PRECISION  DATEV1(3)                                       SCAS0005
      DIMENSION         CBODY(8,11) ,IFEMPS( 8) ,IFPLNT(11) ,NEMP  (8)  SCAS0006
     1,                 IDCP  (900) ,IFOUTP(11) ,IHM   (24)             SCAS0007
      EQUIVALENCE             (CDCP(687),IFEMPS) ,(CDCP(106),MAXQ   )   SCAS0008
     1    ,(CDCP( 17),CBODY ) ,(CDCP(676),IFOUTP) ,(CDCP(105),MAXZ   )   SCAS0009
     2    ,(CDCP(801),DATEV1) ,(CDCP(663),IFPLNT) ,(CDCP(695),NEMP   )   SCAS0010
     3    ,(CDCP(  1),IDCP  ) ,(CDCP(703),IHM   ) ,(CDCP(117),NESEND)   SCAS0011
     4    ,(CDCP(111),IERR  ) ,(CDCP(107),MAXD  ) ,(CDCP(118),NEST  )   SCAS0012
      EQUIVALENCE             (IFEMPS(7),IFDRAG) ,(IFEMPS(2),IFSOLR)   SCAS0013
     1                       ,(IFEMPS(1),IFGRAV) ,(IFEMPS(5),IFTIME)   SCAS0014
     2    ,(IFEMPS(3),IFATMD) ,(IFEMPS(6),IFHARM) ,(IFEMPS(4),IFVENT)   SCAS0015
      EQUIVALENCE             (NEMP  (7),NEMP3 ) ,(NEMP  (3),NSOLR )   SCAS0016
     1    ,(NEMP  (1),NDRAG ) ,(NEMP  (2),NHARM ) ,(NEMP  (5),NTIME )   SCAS0017
     2    ,(NEMP  (8),NEMPS ) ,(NEMP  (6),NPEND ) ,(NEMP  (4),NVENT )   SCAS0018
      EQUIVALENCE             (IFOUTP(3),ICVOUT)                        SCAS0019
      COMMON          /EQMCOM/EQMC(2214)                                SCAS0020
      DOUBLE PRECISION  SPCDAT (31),FFEDAT (83),PREDAT (15)             SCAS0021
     1                 ,DELDAT  (5),EHADAT(324),MHADAT(324)             SCAS0022
     2                 ,XHADAT(325)                                     SCAS0023
      EQUIVALFNCE             (EQMC(    1),SPCDAT) ,(FQMC(   63),EFFDAT) SCAS0024
     1                       ,(EQMC( 229),PREDAT) ,(EQMC( 259),DELDAT) SCAS0025
     2    ,(EQMC( 269),EHADAT) ,(EQMC( 917),MHADAT) ,(EQMC(1565),XHADAT) SCAS0026
C                                                                       SCAS0027
      COMMON          /ESOCOM/CESO(804)                                 SCAS0028
     1                    ,POLD(30,30) ,STNAMO(30) ,TRAKEO(30)          SCAS0029
     2                    ,ITRETO (30) ,ILOCO (30) ,KLOCO (54)          SCAS0030
     3                    ,NSO (12,20)                                  SCAS0031
      DOUBLE PRECISION  POLD                                            SCAS0032
      INTEGER           TRAKEO                                          SCAS0033
      DIMENSION         NAMSTO(20)                                      SCAS0034
      DIMENSION         DESO(2214)                                      SCAS0035
      EQUIVALENCE      (POLD,DESO)                                      SCAS0036
      DOUBLE PRECISION  SPCDAO (6) ,FFEDAO(14) ,PREDAO (4) ,DELDAO (2)  SCAS0037
     1                 ,EHADAO(24) ,MHADAO(24) ,XHADAO(24) ,SEO(14,20)  SCAS0038
     2                 ,ETIMVO      ,TX2ZO(3,3)                         SCAS0039
      EQUIVALENCE               (CESO(  1),NESPSO) ,(CESO(  2),NBYO  )  SCAS0040
     1                         ,(CESO(  3),NBDO  ) ,(CESO(  4),NBXO  )  SCAS0041
     2                         ,(CESO(  5),NBHO  ) ,(CESO(  6),NUMSTO)  SCAS0042
     3        ,(CESO(  7),NAMSTO) ,(CESO( 27),KOUNTO) ,(CESO( 29),ETIMVO) SCAS0043
     4        ,(CESO( 31),TX2ZO ) ,(CESO( 49),SPCDAO) ,(CESO( 61),EFEDAO) SCAS0044
     5        ,(CESO( 89),PREDAO) ,(CESO( 97),DELDAO) ,(CESO(101),EHADAO) SCAS0045
     6        ,(CESO(149),MHADAO) ,(CESO(197),XHADAO) ,(CESO(245),SEO   ) SCAS0046
C                                                                       SCAS0047
      COMMON          /ESTCOM/CEST(804)                                 SCAS0048
      DIMENSION         NAMSTA(20)                                      SCAS0049
      DOUBLE PRECISION  SPCDAN (6) ,EFEDAN(14) ,PREDAN (4) ,DELDAN (2)  SCAS0050
     1                 ,EHADAN(24) ,MHADAN(24) ,XHADAN(24) ,SE (14,20)  SCAS0051
     2                 ,ETIMVA      ,TX2Z (3,3)                         SCAS0052
      EQUIVALENCE               (CEST(  1),NESPOS) ,(CEST(  2),NBY   )  SCAS0053
     1                         ,(CEST(  3),NBD   ) ,(CEST(  4),NBX   )  SCAS0054
     2                         ,(CEST(  5),NBH   ) ,(CEST(  6),NUMSTA)  SCAS0055
     3        ,(CEST(  7),NAMSTA) ,(CEST( 27),KOUNTN) ,(CEST( 29),ETIMVA) SCAS0056
     4        ,(CEST( 31),TX2Z  ) ,(CEST( 49),SPCDAN) ,(CEST( 61),EFEDAN) SCAS0057
     5        ,(CEST( 89),PREDAN) ,(CEST( 97),DELDAN) ,(CEST(101),EHADAN) SCAS0058
     6        ,(CEST(149),MHADAN) ,(CEST(197),XHADAN) ,(CEST(245),SE    ) SCAS0059
C                                                                       SCAS0060
      COMMON          /ES1COM/PNEW(30,30) ,STNAMN(30) ,TRAKER(30)       SCAS0061
     1                    ,ITRETN (30) ,ILOCN (30) ,KLOCN (54)          SCAS0062
     2                    ,NSN (12,20)                                  SCAS0063
      DOUBLE PRECISION  PNEW                                            SCAS0064
      INTEGER           TRAKER                                          SCAS0065
      DIMENSION         DEST(2214)                                      SCAS0066
      EQUIVALENCE      (PNEW,DEST)                                      SCAS0067
C                                                                       SCAS0068
C                                                                       SCAS0069
      COMMON            SAVE(10),DUM(3),IUSED(27),KOD(27),KODE(27)      SCAS0070
C                                                                       SCAS0071
      DOUBLE PRECISION  SEN(69,20)                                      SCAS0072
      EQUIVALENCE      (SPSEN,SEN)                                      SCAS0073
      DIMENSION         NAMOPT(5) ,IDSTAT(15) ,SPCNAM (6) ,LOCSPC (5)   SCAS0074
```

```
     1                  ,NUMOPT(5) ,IBFCOL(15) ,EFENAM(14) ,LOCEFE(10)
     2                  ,LOCOPT(5) ,LOCDAT (7) ,PRENAM (4) ,LOCPRE (3)
     3    ,IBUFF(24,27)            ,MXST   (7) ,DFLNAM (2) ,LOCDEL (1)
     4    ,SPSEN (2760)            ,IHRM   (3) ,HARNAM(24) ,LOCHAR(23)
     5                                         ,STBNAM(10) ,LOCSTA (9)
C
      EQUIVALENCE    (BLANKS,IBLANK)
      DATA    BLANKS/6H      /
      DATA    HARMXX/6HHARMXX/
C     OPTIONS
      DATA    NAMOPT/6HIFPLNT,6HIFEMPS,6HIFOUTP,6HKEYOUT,6HNSCTRL/
      DATA    NUMOPT/ 11, 8, 11, 10, 14/
      DATA    LOCOPT/663,687,676,761,787/
C     STATE IDENTIFICATION
      DATA    IDSTAT/6HSPCRFT,6HEFEMRS,6HPRESUR,6HDFLTIM,6HHARM03
     1             ,6HHARM11,6HHARM01,6HHARM02,6HHARM04,6HHARM05
     2             ,6HHARM06,6HHARM07,6HHARM08,6HHARM09,6HHARM10/
      DATA    IBFCOL/   1,   2,   3,   4, 305,1106, 107, 207
     1              , 407, 507, 607, 707, 807, 907,1007/
      DATA    LOCDAT/   1,  63, 229, 259, 269, 917,1565/
      DATA    MXST  /   6,  11,   4,   2,  24,  24,  24/
      DATA    IHRM  /   3,  11,   0/
C     OUTPUT NAMES
      DATA    SPCNAM/3HX  ,3HY  ,3HZ  ,3HXD ,3HYD ,3HZD /
      DATA    EFENAM/3HUME,3HUVE,3HUEA,3HUMA,3HUJU,3HUSA,3HUUR,3HUNE
     1             ,3HUPL,3HUSU,3HUMO,3HUBA,3HAU ,3HMER/
      DATA    PRENAM/3HSPR,3HDR1,3HDR2,3HVNT/
      DATA    DELNAM/3HDT ,3HDTD/
      DATA    HARNAM/3HJ20,3HJ30,3HJ40,3HJ50,3HJ60,3HJ70,3HJ21,3HL21
     1             ,3HJ22,3HL22,3HJ31,3HL31,3HJ32,3HL32,3HJ33,3HL33
     2             ,3HJ41,3HL41,3HJ42,3HL42,3HJ43,3HL43,3HJ44,3HL44/
      DATA    STBNAM/3HNOR,3HEST,3HDWN,3HTM1,3HTM2,3HLIT,3HAN1,3HAN2
     1             ,3HRNG,3HDOP/
      DATA    LOCSPC/11,15,18,20,21/
      DATA    LOCEFE/27,36,44,51,57,62,66,69,71,72/
      DATA    LOCPRE/ 7, 9,10/
      DATA    LOCDEL/ 3/
      DATA    LOCHAR/ 47, 69, 90,110,129,147,164,180,195,209,222,234
     1             ,245,255,264,272,279,285,290,294,297,299,300/
      DATA    LOCSTA/19,27,34,40,45,49,52,54,55/
C
 5101 FORMAT(2I1,A6,24I2)
 6102 FORMAT(22H0 KODE   IOVRLY    GROUP,24I4)
 6103 FORMAT(I4,I7,5X,A6,24I4)
 6104 FORMAT(1H0/46X,28HPARAMETERS USED IN THIS CASE/4X,5HGROUP,3X,4HNAM
     1E,12X,5HVALUE,2(15X,4HNAME,12X,5HVALUE))
 6105 FORMAT(1H0,2X,A6,4X,A3,D26.16,2(7X,A3,D26.16)/(13X,A3,D26.16,7X,A3
     1,D26.16,7X,A3,D26.16))
 6106 FORMAT(1H0,A6,23H IS THE CENTRAL BODY ON)
 6107 FORMAT(26H0DRAG PARAMETERS APPLY TO ,A6)
 6108 FORMAT(30H0GRAVITY TERMS BELOW APPLY TO ,A6)
 6109 FORMAT(1H0/45X,29HSTATE VARIABLES FOR THIS CASE/38X,45HNAME, TREAT
     1MENT CODE, 0=CONSIDER, 1=SOLVE FOR/10(6X,A3,I3))
 6110 FORMAT(56X,31HMEASUREMENT STANDARD DEVIATIONS/27X,13HANGLE 1 (RAD)
     1,12X,13HANGLE 2 (RAD),14X,10HRANGE (KM),12X,16HDOPPLER (COUNTS)/20
     2X,4D25.16)
C
C     LOAD STATION DATA
  100 REWIND 9
      READ(9) SPSEN
C
C     POSITION ESTIMATION TAPE
      IF(NEST) 107,999,101
  101 IF(NEST.GT.NESEND) NEST=NESEND
      N=NEST-NESPOS-1
      IF(N) 102,106,104
  102 N=-N
      DO 103 I=1,N
      BACKSPACE 12
  103 BACKSPACE 12
      GO TO 106
  104 DO 105 I=1,N
      READ(12) SKIP
  105 READ(12) SKIP
C
```

```
C         READ ESTIMATE FROM TAPE
  106     READ(12) CESO
          READ(12) DESO
          NESPOS=NESPSO
C
C         CLEAR STORAGE ARRAYS
  107     DO 108 I=1,MAXZ
          STNAMN(I)=BLANKS
          TRAKER(I)=IBLANK
          ITRETN(I)=0
          ILOCN(I)=0
          DO 108 J=1,MAXZ
  108     PNEW(I,J)=0.D0
          DO 109 I=1,12
          DO 109 J=1,20
  109     NSN(I,J)=0
          DO 110 I=1,27
          KODE(I)=0
          IUSED(I)=0
          KLOCN(I)=0
          KLOCN(I+27)=0
          DO 110 J=1,24
  110     IBUFF(J,I)=0
          NBH=0
          DO 111 I=1,11
  111     IF(IFPLNT(I).LT.0) IFPLNT(I)=1
C
C         READ OPTION AND STATE CARDS
          ASSIGN 121 TO LLA
  120     READ(5,5101) KOD
          IF(KOD(3).EQ.IBLANK) GO TO 140
          GO TO LLA, (121,125)
  121     DO 123 I=1,5
          IF(KOD(3).NE.NAMOPT(I)) GO TO 123
          N=LOCOPT(I)
          M=NUMOPT(I)
          DO 122 J=1,M
          IDCP(N)=KOD(J+3)
  122     N=N+1
          GO TO 120
  123     CONTINUE
          ASSIGN 125 TO LLA
C
C         INTERPRET AND STORE STATE IDENTIFICATION CARD
  125     CONTINUE
          WRITE(6,6102) (J,J=1,24)
          WRITE(6,6103) KOD
          IERR=3
          IF(KOD.NE.0.AND.NEST.LE.0) GO TO 999
          DO 126 I=1,15
          LSTA=IBFCOL(I)/100
          KSTA=IBFCOL(I)-100*LSTA
          IF(KOD(3).EQ.IDSTAT(I)) GO TO 128
  126     CONTINUE
          DO 127 I=1,NUMSTA
          KSTA=I+7
          IF(KOD(3).EQ.NAMSTA(I)) GO TO 128
  127     CONTINUE
          IERR=4
          GO TO 999
  128     IERR=5
          IF(IUSED(KSTA).NE.0)     GO TO 999
          IF(KSTA.EQ.7) IHRM(3)=LSTA
          IUSED(KSTA)=1
          KODE(KSTA)=KOD
          DO 129 I=1,24
  129     IBUFF(I,KSTA)=KOD(I+3)
          IF(KOD(2).EQ.0) GO TO 120
          IF(KSTA.GT.7) GO TO 130
          L1=LOCDAT(KSTA)
          CALL OVRLYD(EQMC(L1))
          GO TO 120
  130     CALL OVRLYD(SEN(1,KSTA-7))
          GO TO 120
C
```

```
C       INITIALIZE STATE SET-UP                                         SCAS0225
  140  CONTINUE                                                         SCAS0226
       KOUNTO=0                                                         SCAS0227
       KOUNTN=0                                                         SCAS0228
       KO=1                                                             SCAS0229
       KN=1                                                             SCAS0230
       NBY=SPCDAT(28)                                                   SCAS0231
       DO 141 I=1,3                                                     SCAS0232
  141  DUM(I)=SPCDAT(I+28)                                              SCAS0233
       CALL DATINP(DUM,ETIMVA)                                          SCAS0234
       NBD=PREDAT(15)                                                   SCAS0235
       NBX=XHADAT(325)                                                  SCAS0236
       DO 142 I=1,6                                                     SCAS0237
  142  IF(IBUFF(I,1).NE.0) GO TO 143                                    SCAS0238
       GO TO 145                                                        SCAS0239
  143  DO 144 I=1,6                                                     SCAS0240
  144  IF(IBUFF(I,1).EQ.0) IBUFF=-1                                     SCAS0241
  145  CONTINUE                                                         SCAS0242
       IERR=6                                                           SCAS0243
       DO 148 I=5,7                                                     SCAS0244
       IF(IUSED(I).EQ.0) GO TO 148                                      SCAS0245
       DO 146 J=1,24                                                    SCAS0246
  146  IF(IBUFF(J,I).NE.0) GO TO 147                                    SCAS0247
       GO TO 148                                                        SCAS0248
  147  IF(NBH.NE.0) GO TO 999                                           SCAS0249
       NBH=IHRM(I-4)                                                    SCAS0250
  148  CONTINUE                                                         SCAS0251
C                                                                       SCAS0252
C       SET UP NON-STATION PORTION OF STATE                             SCAS0253
  150  DO 179 I=1,7                                                     SCAS0254
       KHOLD=KOUNTN                                                     SCAS0255
       IF(IUSED(I).EQ.0) GO TO 179                                      SCAS0256
       IF(KODE(I).EQ.0) GO TO 170                                       SCAS0257
C                                                                       SCAS0258
C       OVERLAY EQMCOM FROM TAPE ESTIMATE                               SCAS0259
  152  L1=KOUNTO                                                        SCAS0260
       KOUNTO=KOUNTO+KLOCO(KO+1)                                        SCAS0261
       IF(KLOCO(KO)-I) 153,155,154                                      SCAS0262
  153  KO=KO+2                                                          SCAS0263
       GO TO 152                                                        SCAS0264
  155  IERR=7                                                           SCAS0265
       GO TO 999                                                        SCAS0266
  154  KLOCN(KN)=I                                                      SCAS0267
       KLOCN(KN+1)=KLOCO(KO+1)                                          SCAS0268
       KOUNTN=KOUNTN+KLOCN(KN+1)                                        SCAS0269
       IF(KOUNTN.GT.MAXZ) GO TO 189                                     SCAS0270
       MXT=MXST(I)                                                      SCAS0271
       IF(I.EQ.2) MXT=14                                                SCAS0272
       IERR=11                                                          SCAS0273
       DO 168 J=1,MXT                                                   SCAS0274
       GO TO (161,162,163,164,165,166,167) ,I                           SCAS0275
  161  SPCDAT(J)=SPCDAO(J)                                              SCAS0276
       NBY=NBO                                                          SCAS0277
       ETIMVA=ETIMVO                                                    SCAS0278
       GO TO 168                                                        SCAS0279
  162  EFEDAT(J)=EFEDAO(J)                                              SCAS0280
       GO TO 168                                                        SCAS0281
  163  PREDAT(J)=PREDAO(J)                                              SCAS0282
       NBD=NBDO                                                         SCAS0283
       GO TO 168                                                        SCAS0284
  164  DELDAT(J)=DELDAO(J)                                              SCAS0285
       GO TO 168                                                        SCAS0286
  165  EHADAT(J)=FHADAO(J)                                              SCAS0287
       GO TO 168                                                        SCAS0288
  166  MHADAT(J)=MHADAO(J)                                              SCAS0289
       GO TO 168                                                        SCAS0290
  167  XHADAT(J)=XHADAO(J)                                              SCAS0291
       IF(IHRM(3).NE.NBXO) GO TO 999                                    SCAS0292
       NBX=NBXO                                                         SCAS0293
  168  CONTINUE                                                         SCAS0294
C                                                                       SCAS0295
C       SET COVARIANCE FROM TAPE                                        SCAS0296
       NUMBER=KOUNTN-KHOLD                                              SCAS0297
       DO 169 J=1,NUMBER                                                SCAS0298
       KKJ=KHOLD+J                                                      SCAS0299
```

```
      LLJ=L1+J
      ILOCN(KKJ)=ILOCO(LLJ)
      ITRETN(KKJ)=ITRETO(LLJ)
      STNAMN(KKJ)=STNAMO(LLJ)
      DO 169 M=1,NUMBER
      KKM=KHOLD+M
      LLM=L1+M
  169 PNEW(KKJ,KKM)=POLD(LLJ,LLM)
      KN=KN+2
      KO=KO+2
      GO TO 179
C
C     BUILD COVARIANCE FROM BLOCK DATA
  170 NUMBER=0
      MXT=MXST(I)
      DO 171 J=1,MXT
      IF(IBUFF(J,I).EQ.0) GO TO 171
      KOUNTN=KOUNTN+1
      IF(KOUNTN.GT.MAXZ) GO TO 189
      ILOCN(KOUNTN)=J
      NUMBER=NUMBER+1
      IF(IBUFF(J,I).GT.0) ITRETN(KOUNTN)=1
  171 CONTINUE
      IF(NUMBER.EQ.0) GO TO 179
      KLOCN(KN)=I
      KLOCN(KN+1)=NUMBER
      KN=KN+2
      GO TO (172,173,174,175,176,177,178) ,I
  172 CALL BLDCOV (SPCDAT,SPCNAM,KHOLD,NUMBER,LOCSPC)
      GO TO 179
  173 CALL BLDCOV (EFEDAT,EFENAM,KHOLD,NUMBER,LOCEFE)
      GO TO 179
  174 CALL BLDCOV (PREDAT,PRENAM,KHOLD,NUMBER,LOCPRE)
      GO TO 179
  175 CALL BLDCOV (DELDAT,DELNAM,KHOLD,NUMBER,LOCDEL)
      GO TO 179
  176 CALL BLDCOV (EHADAT,HARNAM,KHOLD,NUMBER,LOCHAR)
      GO TO 179
  177 CALL BLDCOV (MHADAT,HARNAM,KHOLD,NUMBER,LOCHAR)
      GO TO 179
  178 CALL BLDCOV (XHADAT,HARNAM,KHOLD,NUMBER,LOCHAR)
  179 CONTINUE
      IERR=10
      L1=KOUNTN
      IF(KLOCN(1).EQ.1) L1=L1-6
      IF(L1.GT.MAXQ) GO TO 999
C
C     SET UP STATION PORTION OF STATE
  180 DO 188 I=8,27
      KHOLD=KOUNTN
      IF(IUSED(I).EQ.0) GO TO 188
      IF(KODE(I).EQ.0) GO TO 185
C
C     OVERLAY SEN FROM TAPE
      KSTA=NAMSTO(I-7)
      DO 181 J=1,NUMSTO
      L1=J
      LSTA=NAMSTO(J)
      IF(KSTA.EQ.LSTA) GO TO 182
  181 CONTINUE
      IERR=8
      GO TO 999
  182 IF(NSO(1,L1).EQ.0) GO TO 188
      KLOCN(KN)=I+93
      NSN(1,I-7)=KHOLD+1
      NSN(2,I-7)=NSO(2,L1)
      KLOCN(KN+1)=NSN(2,I-7)
      DO 183 J=1,10
      SEN(J,I-7)=SEO(J,L1)
      NSN(J+2,I-7)=NSO(J+2,L1)
      IF(NSN(J+2,I-7).EQ.0) GO TO 183
      KOUNTN=KOUNTN+1
      IF(KOUNTN.GT.MAXZ) GO TO 189
      ILOCN(KOUNTN)=J
      STNAMN(KOUNTN)=STBNAM(J)
```

```
      TRAKER(KOUNTN)=LSTA
      IF(NSN(J+2,I-7).GT.0) ITRFTN(KOUNTN)=1
  183 CONTINUE
      KN=KN+2
C
C     SET COVARIANCE FROM TAPE
      NUMBER=KOUNTN-KHOLD
      L1=NSO(1,L1)-1
      DO 184 J=1,NUMBER
      KKJ=KHOLD+J
      LLJ=L1+J
      DO 184 M=1,NUMBER
      KKM=KHOLD+M
      LLM=L1+M
  184 PNEW(KKJ,KKM)=POLD(LLJ,LLM)
      GO TO 188
C
C     BUILD COVARIANCE FROM SEN
  185 NUMBER=0
      DO 186 J=1,10
      NSN(J+2,I-7)=IBUFF(J,I)
      IF(IBUFF(J,I).EQ.0) GO TO 186
      KOUNTN=KOUNTN+1
      IF(KOUNTN.GT.MAXZ) GO TO 189
      ILOCN(KOUNTN)=J
      NUMBER=NUMBER+1
      TRAKER(KOUNTN)=NAMSTA(I-7)
      IF(IBUFF(J,I).GT.0) ITRFTN(KOUNTN)=1
  186 CONTINUE
      IF(NUMBER.EQ.0) GO TO 188
      KLOCN(KN)=I+93
      KLOCN(KN+1)=NUMBER
      KN=KN+2
      NSN(1,I-7)=KHOLD+1
      NSN(2,I-7)=NUMBER
      CALL BLDCOV (SEN(1,I-7),STBNAM,KHOLD,NUMBER,LOCSTA)
  188 CONTINUE
      GO TO 200
  189 IERR=9
      GO TO 999
C
C     LOAD A PRIORI ESTIMATE
  200 CONTINUE
      DO 201 I=1,6
  201 SPCDAN(I)=SPCDAT(I)
      DO 202 I=1,14
  202 EFEDAN(I)=EFEDAT(I)
      DO 203 I=1,4
  203 PREDAN(I)=PREDAT(I)
      DO 204 I=1,2
  204 DELDAN(I)=DELDAT(I)
      DO 205 I=1,24
      EHADAN(I)=EHADAT(I)
      MHADAN(I)=MHADAT(I)
  205 XHADAN(I)=XHADAT(I)
      DO 207 J=1,20
      DO 206 I=1,10
  206 SE(I,J)=SEN(I,J)
      DO 207 I=11,14
  207 SE(I,J)=SEN(I+55,J)
C
C     OUBPUT CASE PARAMFTERS
  210 CONTINUE
      WRITE(6,6104)
      L1=MOD(NBY,100)
      WRITE(6,6106) CBODY(1,L1)
      CALL DATOUP(ETIMVA,DUM,0)
      DO 211 I=1,3
  211 DATEV1(I)=DUM(I)
      WRITE(6,6105) IDSTAT(1),(SPCNAM(I),SPCDAN(I),I=1,6)
      WRITE(6,6105) IDSTAT(2),(EFENAM(I),EFEDAN(I),I=1,14)
      WRITE(6,6107) CBODY(1,NBD)
      WRITE(6,6105) IDSTAT(3),(PRENAM(I),PREDAN(I),I=1,4)
      WRITE(6,6105) IDSTAT(4),(DELNAM(I),DELDAN(I),I=1,2)
      WRITE(6,6108) CBODY(1,3)
```

```
      WRITE(6,6105) IDSTAT(5),(HARNAM(I),EHADAN(I),I=1,24)
      WRITE(6,6108) CBODY(1,11)
      WRITE(6,6105) IDSTAT(6),(HARNAM(I),MHADAN(I),I=1,24)
      IF(NBX.EQ.0) GO TO 212
      WRITE(6,6108) CBODY(1,NBX)
      WRITE(6,6105) HARMXX,(HARNAM(I),XHADAN(I),I=1,24)
  212 CONTINUE
      DO 213 J=1,NUMSTA
      WRITE(6,6105) NAMSTA(J),(STBNAM(I),SE(I,J),I=1,10)
  213 WRITE(6,6110) (SE(I,J),I=11,14)
      WRITE(6,6109) (STNAMN(I),ITRETN(I),I=1,KOUNTN)
C
C     STORE P MATRIX AND RELATED DATA ON SCRATCH TAPE
      WRITE(9) DEST
      BACKSPACE 9
C
C     SET INTERNAL DIMENSIONS AND OPTIONS
      DO 220 I=1,24
  220 IHM(I)=0
      MAXD=3*MAXQ+26
      KHARM=0
      IFAT=0
      IFT=0
      NUMBER=0
      KO=1
  221 KHOLD=NUMBER
      KKJ=KLOCN(KO)
      IF(KKJ.EQ.0.OR.KKJ.GT.7) GO TO 227
      NUMBER=NUMBER+KLOCN(KO+1)
      KKL=KHOLD+1
      DO 226 I=KKL,NUMBER
      J=ILOCN(I)
      GO TO (226,222,223,224,225,225,225) ,KKJ
  222 IFPLNT(J)=-1
      GO TO 226
  223 IF(J.EQ.1) IFSOLR=-1
      IF(J.EQ.2) IFAT=IFAT-1
      IF(J.EQ.3) IFAT=IFAT-2
      IF(J.EQ.4) IFVENT=-1
      GO TO 226
  224 IFT=IFT-J
      GO TO 226
  225 IHM(J)=1
      KHARM=KHARM+1
  226 CONTINUE
      IF(IFAT.NE.0) IFATMD=IFAT
      IF(IFT.NE.0) IFTIME=IFT
      KO=KO+2
      GO TO 221
  227 NSOLR=22
      DO 228 I=1,11
  228 IF(IFPLNT(I).LT.0) NSOLR=NSOLR+3
      NDRAG=NSOLR
      IF(IFSOLR.LT.0) NDRAG=NDRAG+3
      NVENT=NDRAG
      IF(IFATMD.EQ.-3) NVENT=NVENT+3
      IF(IFATMD.LT.0) NVENT=NVENT+3
      NTIME=NVENT
      IF(IFVENT.LT.0) NTIME=NTIME+3
      NHARM=NTIME
      IF(IFTIME.EQ.-3) NHARM=NHARM+3
      IF(IFTIME.LT.0) NHARM=NHARM+3
      NPEND=NHARM+3*KHARM
      NEMP3=NPEND-22
      NEMPS=NEMP3/3
      DO 229 I=1,3
      DO 229 J=1,3
  229 TX2Z(I,J) = 0.
      DO 230 I=1,3
  230 TX2Z(I,I) = 1.D0
C
C     OUTPUT COVARIANCE MATRIX
      IERR=0
      CALL COVOUT(KOUNTN,SPCDAN,ICVOUT)
  999 RETURN
      END
```

Subroutine: SETN

Purpose: SETN sets the logical tape numbers for input and output to conform with the system unit table.

Calling Sequence: CALL SETN (NIN,NØUT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	NIN				Logical input tape number (5).
Ø	NØUT				Output tape number (6).

Common storages used: None

Subroutines required: None

```
$IBFTC SETN     M94,XR3,NODD                                    
      SUBROUTINE SETN(NIN,NOUT)                                 SETN0001
      NIN=5                                                     SETN0002
      NOUT=6                                                    SETN0003
      RETURN                                                    SETN0004
      END                                                       SETN0005
```

Subroutine: SETSTA

Purpose: Used at the beginning of a run to set up arrays for up to 20 stations (for stations found on the data tape) using the station block data values (up to 50 stations may be described in block data.)

Calling Sequence: CALL SETSTA

Common storages used: //16 cells,/DCPCØM/,/ESTCØM/,/STNCØM/

Subroutines required: DATØUP,DSTAT

Discussion

Because of the vast size of the Differential Correction Program, it is necessary to restrict the size of a number of arrays. One such array pertains to tracking stations that are considered active in any one run. This is the SEN array whose dimensions are 69 X 20. Since each column corresponds to a unique station, no more than twenty stations may be considered active for any one run of the program. The actual stations to be selected as active are those stations named on either the Edited Tracking Data tape or the Simulated Tracking Data tape.

It is convenient, however, to allow the stations to be selected from a larger group of 50 possible stations. SETSTA is used at the beginning of a run to examine each station name on the list made from the data tape, find this station among the 50 in the block data, assign an internal station number k, and use block data values to load up the k^{th} column of two working station arrays. After initialization, the working arrays are available to other parts of the program; the block data arrays are not.

The following two pairs of arrays are involved:

BLOCK DATA		BLOCK DATA	
NAME	DESCRIPTION	NAME	DESCRIPTION
STALØC d(3,50)	Station location	SN d(13,20)	Station name Body fixed position vector, local tangent plane to body-fixed transformation.
STANAM (50)	Station name		
STACØR d(69,50)	Station related biases, standard deviations, and correlations.	SEN d(69,20)	Station related biases, standard deviations, and correlations.

The list of station names to be examined has been already loaded into /ESTCØM/ by the main program.

The contents of STALØC are:

STALØC(I,J)   J = 1,50

I = 1 Station latitude, degrees

2 Station longitude, degrees

3 Station altitude, degrees

and the components of SN are:

SN(I,J) J = 1, 20

I =	Description	
1	Station name	
2, 3, 4	Position vector, body-fixed, km.	
5, 6, 7	Unit North vector	Local tangent plane to body-fixed frame orthogonal transform.
8, 9, 10	Unit East vector	
11, 12, 13	Unit Down vector	

```
$IBFTC MC13SV  XR3,M94,NODD,LIST
      SUBROUTINE SETSTA
C     SETS UP WORKING STATION ARRAYS
C
      COMMON            /DCPCOM/CDCP(900)
      DIMENSION          CBODY(8,11),ISN(26,20)
      DOUBLE PRECISION   DTR          ,SN(13,20) ,TBEGIN ,TEND
      EQUIVALENCE       (CDCP( 17),CBODY ) ,(CDCP(143),SN     )
     1                 ,(CDCP(  9),DTR   ) ,(CDCP(119),TBEGIN)
     2                 ,(CDCP(111),IERR  ) ,(CDCP(121),TEND  )
      EQUIVALENCE       (SN,ISN)
C
      COMMON            /ESTCOM/CEST(804)
      DIMENSION          NAMSTA(20)
      EQUIVALENCE       (CEST(  7),NAMSTA) ,(CEST(  6),NUMSTA)
C
      COMMON            /STNCOM/CSTN(7250)
      INTEGER            STANAM(50)
      DOUBLE PRECISION   STACOR(69,50),STALOC(3,50)
      EQUIVALENCE       (CSTN(351),STACOR) ,(CSTN(  1),STANAM)
     1                 ,(CSTN( 51),STALOC)
C
      COMMON             SAVE(6),A,B,DUM(6)
      DOUBLE PRECISION   A,B
C
      DOUBLE PRECISION   SEN (69,20)
      DIMENSION          SPSEN(2760)
      EQUIVALENCE       (SEN,SPSEN)
C
  600 FORMAT(1H0/44X,30HSTATIONS ON TRACKING DATA TAPE//35X,4HNAME,23X
     1,8HLOCATION)
  601 FORMAT(1H0,33X,A6,6X,8HLATITUDE,D25.16,9H  DEGREES/46X,9HLONGITUDE
     1,D24.16,9H  DEGREES/46X,8HALTITUDE,D25.16,8H  METERS)
  602 FORMAT(1H0,A6)
  603 FORMAT(21H0TRACKING DATA BEGINS)
  604 FORMAT(19H TRACKING DATA ENDS)
C
    1 A = CBODY(3,3)
      B = CBODY(4,3)
      WRITE (6,600)
      DO 10 I=1,NUMSTA
      DO  3 J=1,50
      IF (NAMSTA(I).NE.STANAM(J)) GO TO 3
      CALL DSTAT (A,B,DTR,STALOC(1,J),SN(2,I))
      ISN(1,I) = NAMSTA(I)
      WRITE (6,601) NAMSTA(I),(STALOC(K,J),K=1,3)
      DO  2 K=1,69
    2 SEN(K,I) = STACOR(K,J)
      GO TO 10
    3 CONTINUE
      WRITE (6,602) NAMSTA(I)
      IERR = 2
      GO TO 99
   10 CONTINUE
      WRITE (6,603)
      CALL DATOUP (TBEGIN,DUM,0)
      WRITE (6,604)
      CALL DATOUP (TEND,DUM,0)
      REWIND 9
      WRITE (9) SPSEN
   99 RETURN
      END
```

Subroutine: SETTAP

Purpose: Checks or writes tape headers and initializes tape positions.

Calling sequence: CALL SETTAP

Common storages used: // 11 cells, /DCPCØM/,/ESTCØM/

Subroutines required: None

Discussion:

Subroutine SETTAP reads the three tape identification cards. For each, if the tape has not been written by a previous run, the subroutine writes columns 7-72 of the card as a tape identification record. If the tape has been written previously, the tape identification record is compared with columns 7-72. Any difference will cause an immediate return with the error flag (IERR) set = 1.

For the data tape (unit 10) only, the header is always compared with the data card. The tape record contains, in addition to the 11-word header, 45 words describing the tape contents. These are retained in the arrays below

NUMSTA NAMSTA (20)	ESTCØM
TBEGIN d TEND d STIMNX (20)	DCPCØM

```
$IBFTC MC138E  XR3,M94,NODD,LIST
      SUBROUTINE SETTAP
C     CHECKS HEADERS AND INITIALIZES TAPE POSITIONS
C
      COMMON               /DCPCOM/CDCP(900)
      DIMENSION             STIMNX(20)
      DOUBLE PRECISION      TBEGIN,TEND
      EQUIVALENCE          (CDCP(111),IERR   ) ,(CDCP(123),STIMNX)
     1                    ,(CDCP(117),NESEND) ,(CDCP(119),TBEGIN)
     2                    ,(CDCP(116),NRSEND) ,(CDCP(121),TEND   )
C
      COMMON               /ESTCOM/CEST(804)
      DIMENSION             NAMSTA(20)
      EQUIVALENCE          (CEST(  7),NAMSTA) ,(CEST(  6),NUMSTA)
C
      COMMON                IHEAD(12),JHEAD(11)
C
      DIMENSION             FMT(6),T(2,3)
      DATA    FMT(1)/36H(11X,2A6,6X,5X,7HHEADER,,11A6)A6,I6,/
      DATA    T(1,1)/36H   DATA TAPE    NRSEND =     NESEND = /
C
  501 FORMAT(I6,11A6)
  600 FORMAT(1H1,45X,26H*** RUN INITIALIZATION ***/1H )
C
    1 WRITE (6,600)
      IERR = 1
      K = 2
      DO 9 I=1,3
      READ  (5,501)               (IHEAD(J),J=1,12)
      WRITE (6,FMT) T(1,I),T(2,I),(IHEAD(J),J=K,12)
      GO TO (2,3,4) ,I
C     DATA TAPE
    2 READ  (10) JHEAD,NUMSTA,NAMSTA,TBEGIN,TEND,STIMNX
      K = 1
      FMT(2) = FMT(6)
      GO TO 7
C     RESIDUAL TAPE
    3 NRSEND = IHEAD(1)
      IF (NRSEND.NE.0)    GO TO 5
      WRITE (11) (IHEAD(J),J=2,12)
      GO TO 9
C     ESTIMATION TAPE
    4 NESEND = IHEAD(1)
      IF (NESEND.NE.0)    GO TO 6
      WRITE (12) (IHEAD(J),J=2,12)
      GO TO 9
    5 READ  (11) JHEAD
      GO TO 7
    6 READ  (12) JHEAD
C     CHECK HEADERS
    7 DO 8 J=2,12
      IF (IHEAD(J).NE.JHEAD(J-1)) GO TO 999
    8 CONTINUE
    9 CONTINUE
      IERR = 0
  999 RETURN
      END
```

Subroutine: SHIF2

Purpose: Given the state of a body or vehicle with respect to a specified body, SHIF2 returns the state with respect to any other body(s) as requested by option key KØP.

Calling Sequence: CALL SHIF2 (KØP,NØR,DATE,X,NBØD,XØUT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	KØP				Option key.
I	NØR				Central body number.
I	DATE	(2)		days	Whole and fractional days from 1950 Jan 0.0
I	X	(6)		km, km/sec	Vehicle state with respect to NØR equator and equinox of 1950.0
I	NBØD				Any body number.
Ø	XØUT	(6,10)		km, km/sec	Requested state(s) KØP=1: state of all bodies wrt NØR KØP=2: vehicle state wrt all bodies KØP=3: NBØD state wrt NØR in column NBØD KØP=4: vehicle state wrt NBØD in column NBØD.

Common storages used: /INPCØM/,/WCØM/

Subroutines required: ANTR, STEPT, FNØRM

Procedure

Bodies are identified by the following numbers

1	Earth	6	Saturn
2	Moon	7	Jupiter
3	Sun	8	Extra body 1 with body center NØX(1)
4	Venus	9	Extra body 2 with body center NØX(2)
5	Mars	10	Extra body 3 with body center NØX(3)

The procedure used depends upon the quantities requested and the bodies involved. The relative positions of bodies one through seven are obtained from a call of ANTR (the subroutine which reads and interprets an ephemeris tape). Subroutine STEPT is used to obtain the relative positions of bodies whose ephemerides are not stored on tape. The required quantities are obtained by combining the relevant vectors.

It should be noted that NØR may be any body (1 to 10), but if NØR is an extra body (8, 9, or 10), then that body's orbit must be described with respect to the usual bodies (1-7). When an extra body is the central body of another central body, it must have the smaller extra body number. NBØD is not required unless KØP is 3 or 4; when used it may be any body, 1-10.

```
$IBFTC MC13S2  NOREF,M94,NODD,XR3
CMC13S2  SHIF2  CALLS STEPT FOR XTRA BODIES , SCOTTY LINK
      SUBROUTINE SHIF2(KOP,NOR,DATE,X,NBOD,XOUT)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)
      DIMENSION     DATE(2),  X(6),     XOUT(6,10),     PO(21)
     1,             VE(21),   TX(2,3),  RX(3,3),        VX(3,3)
     2,             NOX(3),   SHFC(18)
      EQUIVALENCE  (C(6),SPMSD)
     1,            (IW(4),NOX),    (IW(7),IXTRA)
     2,            (CW(1),RX),     (CW(10),VX),    (CW(19),TX)
     3,            (CW(1372),SHFC)
C     BODIES ARE    1 EARTH
C                   2 MOON
C                   3 SUN
C                   4 VENUS
C                   5 MARS
C                   6 SATURN
C                   7 JUPITER
C                   8 EXTRA BODY 1 GIVEN WRT BODY NOX(1)
C                   9 EXTRA BODY 2                NOX(2)
C                  10 EXTRA BODY 3                NOX(3)
C     NOR IS THE CENTRAL BODY NUMBER, X IS VEHICLE STATE WRT NOR
C     KOP OPTIONS  1 XOUT IS STATE OF ALL BODIES WRT NOR
C                  2 XOUT IS VEHICLE STATE WRT ALL BODIES
C                  3 XOUT(I,NBOD) IS NBOD STATE WRT NOR
C                  4 XOUT(I,NBOD) IS VEHICLE STATE WRT NBOD (ONLY)
      NSTP=1
      LXT = 0
      IF(NOR.LT.8) GO TO 10
      LXT = NOR-7
      NSAV= NOR
      NOR=NOX(LXT)
   10 IF(KOP.GT.2) GO TO 20
      I1=1
      I2=7
   11 CALL ANTR1(DATE,DATE(2),NOR-1,PO,1,VE,FNORM(X))
      DO 13 I=I1,I2
      JJ=3*I-3
      DO 13 J=1,3
      KK=JJ+J
      XOUT(J,I)=PO(KK)
   13 XOUT(J+3,I)=VE(KK)
      GO TO(14,120),NSTP
14    CONTINUE
      IF(IXTRA.GT.0) GO TO 90
      GO TO 120
   20 I1=NBOD
      I2=NBOD
      IF(NBOD.GT.7) GO TO 90
      IF(LXT.NE.0) GO TO 11
      IF(NOR.NE.3.OR.NBOD.EQ.2) GO TO 11
      NSTP=2
      GO TO 11
   90 I2 = 10
      DO 210 J=1,IXTRA
      MM=J+7
      L=NOX(J)
      KK=6*J-5
      TT=(DATE - TX(1,J) + DATE(2) - TX(2,J))*SPMSD
      CALL STEPT(TT,RX(1,J),VX(1,J),XOUT(1,MM),XOUT(4,MM),SHFC(KK))
      IF(NOR.EQ.L) GO TO 210
      IF(L.GT.7.OR.KOP.LT.3) GO TO 100
      IF(NBOD.LT.8) GO TO 101
      CALL ANTR1(DATE,DATE(2),NOR-1,PO,1,VE,FNORM(X))
  101 JJ=3*L -3
      DO 105 K=1,3
      KK=JJ+K
      XOUT(K,L)=PO(KK)
  105 XOUT(K+3,L)=VE(KK)
  100 DO 110 I=1,6
  110 XOUT(I,MM) = XOUT(I,MM) + XOUT(I,L)
  210 CONTINUE
      IF(LXT.EQ.0) GO TO 120
      NOR = NSAV
      DO 211 I=I1,I2
```

```
      DO 211 J=1,6                                                        SHI20750
  211 XOUT(J,I) = XOUT(J,I) - XOUT(J,NSAV)                                SHI20760
  120 GO TO (150,130,150,130),KOP                                         SHI20770
  130 DO 140 I=I1,I2                                                      SHI20780
      DO 140 J=1,6                                                        SHI20790
  140 XOUT(J,I)=X(J)-XOUT(J,I)                                            SHI20800
  150 RETURN                                                              SHI20810
      END                                                                 SHI20820
```

Subroutine: SKDØUT

Purpose: To write out the measurement schedule. SKDØUT serves principally as a driver for ØYAL.

Calling Sequence: CALL SKDØUT

Common storages used: /INPCØM/,/WCØM/

Subroutines required: ØYAL

```
$IBFTC MC13KD  NOREF,M94,NODD,XR3
CMC13KD  SKDOUT   BLDP TYPE MEASUREMENT BUILDUP AND OUTPUT                 SKD00001
      SUBROUTINE SKDOUT                                                    SKD00002
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)                           SKD00003
      DIMENSION     ISC(12),    IS(11,12),                     S(23,12)    SKD00004
     1,             IBC(10),    IB(36),                        B(91)       SKD00005
     2,             IEMP(24),                                  OB(18)      SKD00006
     3,             IOBR(4),                                   EMP(24)     SKD00007
     5,                         BODC(10,8)                                 SKD00008
     6,             DNAM(6),    IOBA(3,6),      REFS(3)                    SKD00009
      DIMENSION     EQNA(3,24)                                             SKD00010
      DIMENSION     TYPNAM(4),  MOUNT(3)                                   SKD00011
      EQUIVALENCE   (C(4),RTD),       (C(7),RSPMSD),      (C(11),BODC)     SKD00012
     1,             (C(200),S),       (C(476),B),         (C(567),OB)      SKD00013
     2,             (C(585),EMP)                                           SKD00014
      EQUIVALENCE   (IW(8),ISC),      (IW(20),IBC),       (IW(30),IEMP)    SKD00015
     1,             (IW(54),IOBR),    (IW(58),IOBA)                        SKD00016
     2,                               (IW(180),IS),       (IW(312),IB)     SKD00017
      EQUIVALENCE   (CW(1157),REFS)                                        SKD00018
      DATA (DNAM(J),J=1,6)/4HNONE,6HSUBTEN,6HRA-REF,6HDECREF,6HSEXTIP,     SKD00019
     16HSEXTOP/                                                            SKD00020
      DATA RTS/.20626481E6/                                                SKD00021
      DATA TYPNAM/6HC-BAND,6HGODDRD,6H USBS ,6H DSIF /                     SKD00022
      DATA MOUNT/6H 30-FT,6H 85-FT,1H /                                    SKD00023
      DATA (EQNA(1,J),J=1,12)/                                             SKD00024
     118HASTRO UNIT KM/AU   ,                                              SKD00025
     218HEARTH M FRAC/SUN   ,                                              SKD00026
     318HMOON   M FRAC/SUN  ,                                              SKD00027
     418HVENUS M FRAC/SUN   ,                                              SKD00028
     518HMARS   M FRAC/SUN  ,                                              SKD00029
     618HJUPTR M FRAC/SUN   ,                                              SKD00030
     718HSATRN M FRAC/SUN   ,                                              SKD00031
     818HMRCRY M FRAC/SUN   ,                                              SKD00032
     918H2ND ZONAL HARM     ,                                              SKD00033
     118H3RD ZONAL HARM     ,                                              SKD00034
     218H4TH ZONAL HARM     ,                                              SKD00035
     318H5TH ZONAL HARM     /                                              SKD00036
      DATA(EQNA(1,J),J=13,24)/                                             SKD00037
     418H1ST LONG HARM RAD ,                                               SKD00038
     518H2ND LONG HARM RAD ,                                               SKD00039
     618H3RD LONG HARM RAD ,                                               SKD00040
     718H4TH LONG HARM RAD ,                                               SKD00041
     818H1ST LUNAR KM**2   ,                                               SKD00042
     918H2ND LUNAR KM**2   ,                                               SKD00043
     118H3RD LUNAR KM**2   ,                                               SKD00044
     218H1ST DRAG   1/(KM) ,                                               SKD00045
     318H2ND DRAG   1/(KM) ,                                               SKD00046
     418HSOLAR RAD K**3/S*S,                                               SKD00047
     518HVENT. THR. KM/SEC ,                                               SKD00048
     618HSPEED LGT ERROR   /                                               SKD00049
      IF(ISC) 30,30,20                                                     SKD00050
20    WRITE(  6,101)                                                       SKD00051
  101 FORMAT(1H0,41X,38H*** EARTH-BASED TRACKING IN EFFECT ***/7H NUMBERSKD00052
     1,4X,4HNAME,6X,4HTYPE,18X,8HLOCATION,38X,8HOBSERVES)                  SKD00053
      DO 28 I=1,12                                                         SKD00054
      ISTAT=ISC(I)                                                         SKD00055
      IF(ISTAT)30,30,21                                                    SKD00056
   21 CONTINUE                                                             SKD00057
      SLAT=S(3,ISTAT)*RTD                                                  SKD00058
      SLON=S(4,ISTAT)*RTD                                                  SKD00059
      AH=S(6,ISTAT)*RTD                                                    SKD00060
      SALT=S(5,ISTAT)*1000.                                                SKD00061
      EM=S(7,ISTAT)*RTD                                                    SKD00062
      MTYPE=IS(1,ISTAT)                                                    SKD00063
      NALIGN=3                                                             SKD00064
      IF(MTYPE.EQ.3) NALIGN=IS(10,ISTAT)                                   SKD00065
      WRITE(6,102) ISTAT,S(1,ISTAT),TYPNAM(MTYPE),SLAT,S(2,ISTAT),MOUNT(SKD00066
     1NALIGN),SLON,AH,SALT,EM                                              SKD00067
  102 FORMAT(1H0,2X,I2,6X,A6,3X,A6,9X,11HLATITUDE  =,F10.5,4H DEG,16X,15SKD00068
     1HAT INTERVALS OF,F9.2,4H SEC/20X,A6, 9X,11HLONGITUDE =,F10.5,4H DESKD00069
     2G,16X,23HWHEN ELEVATION IS ABOVE,F7.2,4H DEG/35X,11HALTITUDE  =,F1SKD00070
     30.5,4H MET,16X,13HBUT LESS THAN,F7.2,4H DEG)                         SKD00071
      IF(MTYPE.EQ.1) GO TO 1020                                            SKD00072
      FTR=S(22,ISTAT)*1.E-6                                                SKD00073
      OMG3=S(20,ISTAT)*1.E-6                                               SKD00074
```

```
      TAU=S(23,ISTAT)                                                   SKD00075
      OMG4=S(21,ISTAT)                                                  SKD00076
      WRITE(6,1021) FTR,OMG3,TAU,OMG4                                   SKD00077
 1021 FORMAT(1H0,28X,18HDOPPLER PARAMETERS/31X,22HTRANSMISSION FREQUENCYSKD00078
     1,9X,F11.5,15H MEGACYCLES/SEC/31X,14HBIAS FREQUENCY,20X,F8.5,15H MESKD00079
     2GACYCLES/SEC/31X,14HCOUNT INTERVAL,20X,F8.5,8H SECONDS/31X,20HRETRSKD00080
     3ANSMISSION RATIO,15X,F7.5)                                        SKD00081
 1020 WRITE(6,103)                                                      SKD00082
  103 FORMAT(1H0,28X,25HMEASUREMENT ERROR SOURCES,11X,6HRANDOM,9X,4HBIASSKD00083
     1)                                                                 SKD00084
      DO 27 J=2,9                                                       SKD00085
      IF(IS(J,ISTAT).EQ.0) GO TO 27                                     SKD00086
      CALL  OYAL(1,IS(1,ISTAT),J,S(1,ISTAT)      )                      SKD00087
   27 CONTINUE                                                          SKD00088
   28 CONTINUE                                                          SKD00089
   30 CONTINUE                                                          SKD00090
      IF(IBC.EQ.0) GO TO 40                                             SKD00091
      NBD=IB(36)                                                        SKD00092
      WRITE(   6,104)                                                   SKD00093
  104 FORMAT(1H0,47X,27H*** BEACON MEASUREMENTS ***/29X,12HMEASUREMENTS,SKD00094
     124X,6HRANDOM, 9X,4HBIAS,13X,23HOBSERVATION PERIOD(SEC))           SKD00095
      DO 391 I=31,35                                                    SKD00096
      J=I-30                                                            SKD00097
      CALL  OYAL(2,IB(31),J,B(81)      )                                SKD00098
  391 CONTINUE                                                          SKD00099
      WRITE(   6,105)                                                   SKD00100
  105 FORMAT(1H0, 48X,26H*** BEACONS CONSIDERED ***/7H NUMBER,6X,4HNAME,SKD00101
     122X,8HLOCATION,20X,20HLOCATION UNCERTAINTY,14X,4HBIAS)            SKD00102
      DO 36 I=1,10                                                      SKD00103
      IBEAC=IBC(I)                                                      SKD00104
      IF(IBEAC) 40,40,31                                                SKD00105
   31 CONTINUE                                                          SKD00106
      NBEAC=3*(IBEAC-1)                                                 SKD00107
      K=8*(IBEAC-1)+1                                                   SKD00108
      SLAT=B(K+1)*RTD                                                   SKD00109
      SLON=B(K+2)*RTD                                                   SKD00110
      SALT=B(K+3)*1000.                                                 SKD00111
      AH  =B(K+4)*RTD                                                   SKD00112
      BLAT=      B(K+5) *1000.                                          SKD00113
      BLON=      B(K+6) *1000.                                          SKD00114
      BALT=      B(K+7) *1000.                                          SKD00115
      IF(IB(NBEAC+1).EQ.0) BLAT=0.                                      SKD00116
      IF(IB(NBEAC+2).EQ.0) BLON=0.                                      SKD00117
      IF(IB(NBEAC+3).EQ.0) BALT=0.                                      SKD00118
      WRITE(   6,106) IBEAC,B(K),SLAT,BLAT,SLON,BLON,SALT,BALT,AH       SKD00119
  106 FORMAT(1H0,2X,I2,8X,A6,12X,11HLATITUDE  =,F10.5,4H DEG,11X,21HBEACSKD00120
     1ON NORTHING (MET),13X,F10.5/31X,11HLONGITUDE =,F10.5,4H DEG,11X,  SKD00121
     221HBEACON EASTING  (MET),13X,F10.5/31X,11HALTITUDE  =,F10.5,4H METSKD00122
     3,11X,21HBEACON ALTITUDE (MET),13X,F10.5/31X,11HHORIZON   =,F10.5, SKD00123
     44H DEG,F10.5)                                                     SKD00124
   36 CONTINUE                                                          SKD00125
   40 CONTINUE                                                          SKD00126
      IF(IOBR .LE.0) GO TO 549                                          SKD00127
      NOB = IOBR                                                        SKD00128
      WRITE(   6,113) BODC(NOB,1),OB(6),OB(7)                           SKD00129
113   FORMAT(1H0,12X,17H***RADAR MEAS TO ,A6,1X,16HIN EFFECT BELOW ,E15.SKD00130
     18,1X,22HKM ALT, OBSERVED EVERY,F9.0,9H SECS*** )                  SKD00131
      IF(IOBR(2) .EQ.0) GO TO 545                                       SKD00132
      A =      OB(1)  * 1000.                                           SKD00133
      BB=     OB(3) *1000.                                              SKD00134
      IF(IOBR(2).EQ.2) BB=0.                                            SKD00135
      WRITE (   6,114) A,BB                                             SKD00136
114   FORMAT(21H HEIGHT (RANDOM) M    ,E15.8/8X,6H(BIAS),7X,E15.8)      SKD00137
545   IF(IOBR(3) .EQ. 0)GO TO 546                                       SKD00138
      A =      OB(2) *1000.                                             SKD00139
      BB=     OB(4) *1000.                                              SKD00140
      IF(IOBR(3).EQ.2) BB=0.                                            SKD00141
      WRITE (   6,115) A,BB                                             SKD00142
115   FORMAT(24H HEIGHT RT.(RANDOM) M/S ,E15.8/ 11X,6H(BIAS),7X,E15.8)  SKD00143
546   IF(IOBR(4) .EQ.0) GO TO 549                                       SKD00144
      WRITE(6,116) OB(5)                                                SKD00145
116   FORMAT( 7H TBIAS ,E15.8, 4H SEC)                                  SKD00146
549   CONTINUE                                                          SKD00147
      IF(IOBA .LE. 0) GO TO 580                                         SKD00148
C     REFS=REFERENCE VECTOR FOR ONBOARD OPTICAL OBSERVATIONS            SKD00149
```

```
      SD=SIN(OB(16))                                                    SKD00150
      CD=COS(OB(16))                                                    SKD00151
      SR=SIN(OB(15))                                                    SKD00152
      CR=COS(OB(15))                                                    SKD00153
      REFS(1)=CD*CR                                                     SKD00154
      REFS(2)=CD*SR                                                     SKD00155
      REFS(3)=SD                                                        SKD00156
      WRITE (   6,107)                                                  SKD00157
107   FORMAT(1H0,39X,40H***OPTICAL ONBOARD TRACKING IN EFFECT***)       SKD00158
      PER=OB(14)*RSPMSD                                                 SKD00159
      TBIAS=      OB(5)                                                 SKD00160
      IF(IOBR(4) .EQ.0) TBIAS =0.                                       SKD00161
      RA = OB(15)*RTD                                                   SKD00162
      DEC= OB(16)*RTD                                                   SKD00163
      WRITE(   6,108) PER,TBIAS,RA,DEC                                  SKD00164
108   FORMAT(7H0PERIOD,E17.8,5H DAYS/7H  TBIAS,E17.8,4H SEC/7H REFRA.,  SKD00165
     1E17.8,4H DEG/7H REFDEC,E17.8,4H DEG)                              SKD00166
      WRITE(   6,109)                                                   SKD00167
109   FORMAT(1H0,20X,76HRANDOM ERROR MODEL = (K1)**2+(K2**2)*(2.*ASIN(RASKD00168
     1DIUS BODY/RANGE TO BODY))**2/8X,4HBODY,11X,12HNO. OF MEAS.,12X,4HTSKD00169
     2YPE,15X,6HK1 SEC,11X, 2HK2)                                       SKD00170
      DO 575 I=1,6                                                      SKD00171
      IF(IOBA(1,I) .EQ. 0) GO TO 580                                    SKD00172
      NB = IOBA(3,I)                                                    SKD00173
      IT = IOBA(2,I)                                                    SKD00174
      KK = IT+1                                                         SKD00175
      IF (KK .GT. 3) KK=KK+1                                            SKD00176
      GO TO (550,555,560,565,560,565),KK                                SKD00177
550   A=0.                                                              SKD00178
      BB=0.                                                             SKD00179
      GO TO 570                                                         SKD00180
555   A=RTS*      OB(12)                                                SKD00181
      BB=      OB(13)                                                   SKD00182
      GO TO 570                                                         SKD00183
560   A=RTS*      OB(8)                                                 SKD00184
      BB=      OB(9)                                                    SKD00185
      GO TO 570                                                         SKD00186
565   A=RTS*      OB(10)                                                SKD00187
      BB=      OB(11)                                                   SKD00188
570   WRITE (   6,110) BODC(NB,1),IOBA(1,I),IT,A,BB                     SKD00189
110   FORMAT(4X,A6,2I20,5X,2E20.8)                                      SKD00190
      KK = KK+1                                                         SKD00191
      IF(KK .EQ. 4) GO TO 565                                           SKD00192
575   CONTINUE                                                          SKD00193
580   CONTINUE                                                          SKD00194
      K=0                                                               SKD00195
      DO 590 I=1,24                                                     SKD00196
      IF(IEMP(I)) 585,590,585                                           SKD00197
585   IF(K.GT.0) GO TO 586                                              SKD00198
      K=1                                                               SKD00199
      WRITE (   6,111)                                                  SKD00200
586   WRITE (   6,112) (EQNA(J,I),J=1,3),EMP(I)                         SKD00201
590   CONTINUE                                                          SKD00202
111   FORMAT(1H0,42X,35H***OTHER DET. UNKNOWNS IN EFFECT***)            SKD00203
112   FORMAT(1H ,3A6,2X,E15.8)                                          SKD00204
      CALL LOCAT                                                        SKD00205
      RETURN                                                            SKD00206
      END                                                               SKD00207
```

Subroutine: SKDUL

Purpose: Reads control times. Reads and writes measurement schedule.

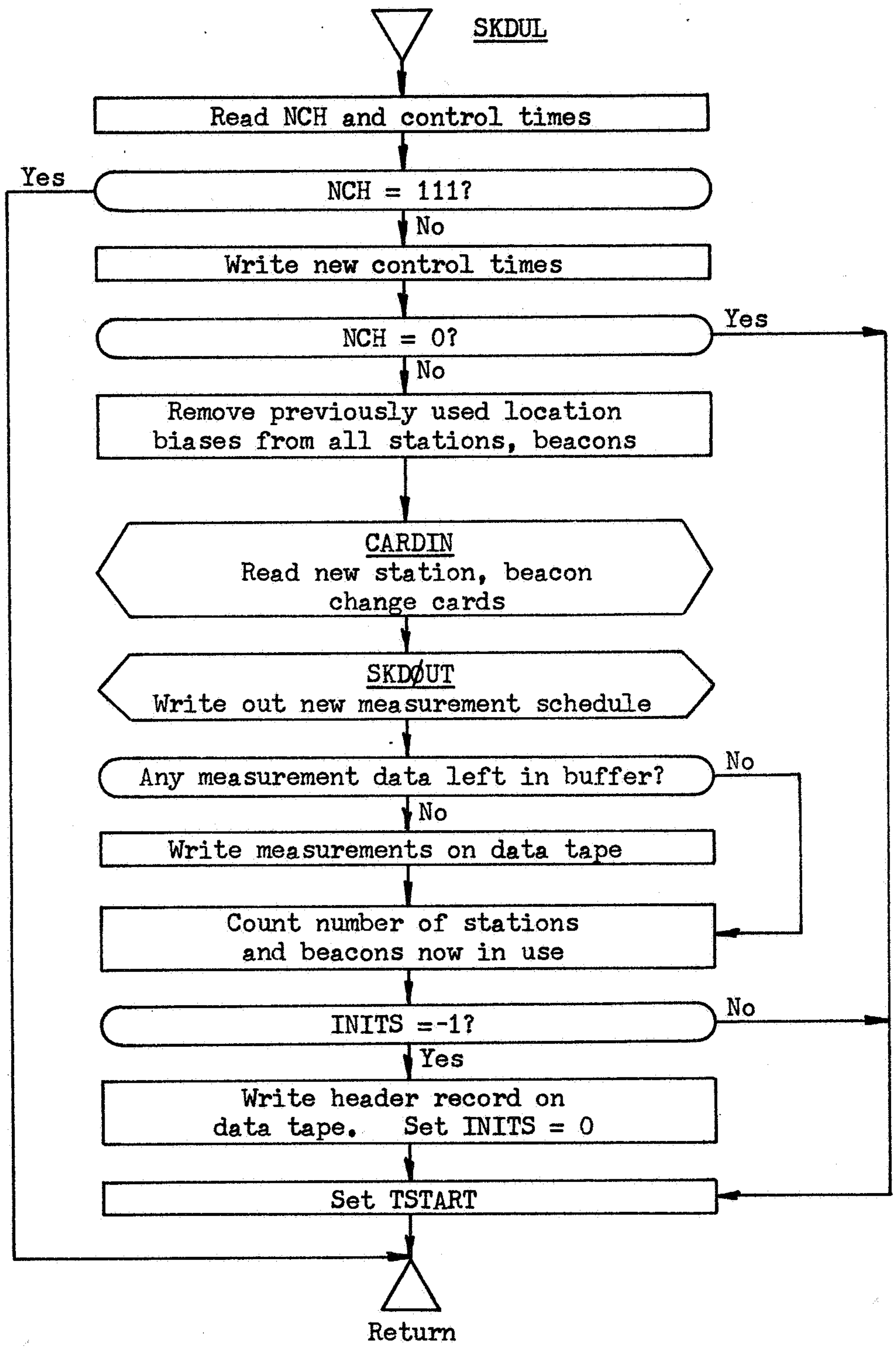
Calling Sequence: CALL SKDUL (INITS,NCH)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	INITS				Initialization key. Set -1 on first entry to initialize, internally reset to 0.
Ø	NCH				NCH=111 indicates end of run; NCH=0 indicates change in TSTART and TSTØP only.

Common storages used: /INPCØM/,/WCØM/,/BUFCØM/

Subroutines required: CARDIN, ERRØUT, SKDØUT, TIMED, TIMES


SKDUL
Read NCH and control times
NCH = 111?
Yes
No
Write new control times
NCH = 0?
Yes
No
Remove previously used location biases from all stations, beacons
CARDIN
Read new station, beacon change cards
SKDØUT
Write out new measurement schedule
Any measurement data left in buffer?
No
No
Write measurements on data tape
Count number of stations and beacons now in use
INITS =-1?
No
Yes
Write header record on data tape. Set INITS = 0
Set TSTART
Return

```
$IBFTC MC13UL  M94,NODD,XR3
CMC13UL   SKDUL  READS CONTROL TIMES, READS AND WRITES MEAS SCHEDULE
      SUBROUTINE SKDUL(INITS,NCH)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)
      COMMON  /BUFCOM/BUFF1(40) ,KEY
      DOUBLE PRECISION  DBUFF(10)
      DOUBLE PRECISION  ONTM,      TFST,      TLST,      TAU,      FTR
     1,                 C1,        C2,        RAMB,      BIAS,     OMG4
      EQUIVALENCE  (BUFF1(1),IKY),    (BUFF1(2),KEE),    (BUFF1(3),STANAM)
     1,            (BUFF1(4),STNAM1),(BUFF1(5),NRCD),   (BUFF1(6),NKOUNT)
     2,            (BUFF1(7),KONTIN),(BUFF1(8),MTYPE),  (BUFF1(9),NALIGN)
     3,            (BUFF1(10),MODE), (BUFF1(11),FMS),   (BUFF1(12),KTAU)
     4,            (BUFF1(13),DBUFF,ONTM)
      EQUIVALENCE  (DBUFF(2),TFST),  (DBUFF(3),TLST),   (DBUFF(4),TAU)
     1,            (DBUFF(5),FTR),   (DBUFF(6),C1),     (DBUFF(7),C2)
     2,            (DBUFF(8),RAMB),  (DBUFF(9),BIAS),   (DBUFF(10),OMG4)
      DIMENSION   STNM(20),    TI(3)
      DIMENSION    TFORM(6), TST(2),   TWT(2),   ONAME(2)
      DIMENSION     S(23,12), B(8,10),  ISC(12),  IBC(10)
     1,             IS(11,12),     IB(3,12),       BIASL(3,22)
      DIMENSION  ISEM(5,13),  XMES(5, 85)
      DIMENSION   HEAD(24),      STIMNX(20)
      EQUIVALENCE   (C(200),S),          (C(476),B),          (IW(8),ISC)
     1,             (IW(20),IBC),        (IW(180),IS),        (IW(312),IB)
     2,       (CW(156),BIASL),    (CW(522),XMES),       (CW(1160),TI)
     3,            (CW(1140),TSEC),(CW(1178),TSTART), (CW(1188),TWT)
      EQUIVALENCE  (C(6),SPMSD),      (CW(1179),FLTIM), (CW(1172),TSECO)
      EQUIVALENCE  (C(675),HEAD)
      EQUIVALENCE  (C(7),RSPMSD),     (C(115),XTROUT)
      DOUBLE PRECISION  TSTDP,TSPDP
      DIMENSION   TYPE(4)
      DATA TYPE/6HC-BAND,6HGODDRD,6HS-BAND,6H DSIF /
      DATA QYES,QNO/3HYES,2HNO/
C  SUBRS REQUIRED
C     TIME - TIMES, TIMED
C     ERROUT
C     CARDIN
C     SKDOUT
      DATA (ONAME(J),J=1,2) /6H START, 6H  STOP/
      DATA IQUAL/0/
      CALL TIMES(TSEC,D,TFORM)
      READ(  5,706) NCH,TST
706   FORMAT(I3,1E12.8,  3X,1E12.8)
      IF(NCH .EQ. 111) GO TO 230
      WRITE (   6,707) TFORM
707   FORMAT(21HONEW CONTROL TIMES AT,6A6)
      DO 21 I=1,2
      CALL TIMED(TST(I),TWT(I))
      CALL TIMES(TWT(I),D,TFORM)
21    WRITE(   6,708) ONAME(I),TFORM
708   FORMAT(1H ,1A6, 1X,6A6)
      IF(TWT(1).GT.TWT(2))                CALL ERROUT(1,31H(17HOERROR I
     1N INPUT  2E20.8,I4),TWT(1),TWT(2),NCH)
      IF(NCH .EQ. 0) GO TO 22
      IF(INITS.EQ.-1) GO TO 50
      DO 10 I=1,12
      K=ISC(I)
      IF(K) 11,11,1
    1 DO 10 J=6,8
      IF(IS(J,K)) 2,10,2
    2 L=J-3
      M=J-5
      S(L,K)=S(L,K)-BIASL(M,K)
10    CONTINUE
   11 DO 20 I=1,10
      K=IBC(I)
      IF(K) 50,50,12
12    DO 20 J=1,3
      IF(IB(J,K)) 13,20,13
13    L=J+1
      B(L,K)=B(L,K)-BIASL(J,K+12)
20    CONTINUE
50    CONTINUE
      CALL CARDIN(INITS)
      CALL SKDOUT
```

```
      IF(IKY.EQ.0) GO TO 61
      WRITE(12) BUFF1
      WRITE(12) (IQUAL,I=1,85),((XMES(I,J),J=1,85),I=1,5)
C***  CHECK BINARY OUTPUT
      QOUT=QNO
      IF(KONTIN.EQ.1) QOUT=QYES
      WRITE(6,601) IKY,STANAM,STNAM1,TYPE(MTYPE),QOUT,TFST
  601 FORMAT(//24H ***  RECORD PAIR NUMBER,I5//75H RECEIVING STATION   T
     1RANSMITTING STATION   MEASUREMENT TYPE   CONTINUATION/7X,A6,15X,A6
     2,15X,A6,12X,A3//12H DATA BEGINS,D24.16,20H SECONDS FROM 1950.0)
      TEMP=(TFST -.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,602) TLST
  602 FORMAT(12H0 DATA ENDS ,D24.16,20H SECONDS FROM 1950.0)
      TEMP=(TLST -.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,606) NKOUNT
  606 FORMAT(12H0TIME POINTS/I7)
      IF(XTROUT.EQ.0.) GO TO 61
      WRITE(6,607) KEE,NRCD,NALIGN,MODE,FMS,KTAU,ONTM, (DBUFF(I),I=4,10)
     1,(IQUAL,(XMES(I,J),I=1,5),J=1,NKOUNT)
  607 FORMAT(5H0NEOT,3X,4HNRCD,3X,6HNALIGN,3X,4HMODE,7X,8HMEAS INT,6X
     1,4HKTAU,11X,6HONTIME,22X,3HTAU/I4,I7,2I8,E19.8,I6,D27.16,D26.16//
     213X,3HFTR,23X,2HC1,24X,2HC2,24X,2HDR/4D26.16//12X,4HBIAS,21X,5HRAT
     3IO/2D26.16//3H IQ,11X,4HTIME,15X,5HANG 1,15X,5HANG 2,15X,5HRANGE,
     414X,7HDOPPLER/(I3,5E20.8))
C***  END BINARY OUTPUT
61    CONTINUE
      DO 60 I=1,5
      DO 60 J=1,13
60    ISEM(I,J)=0
      NSTA=0
      DO 62 J=1,12
      L=ISC(J)
      IF(L.EQ.0) GO TO 63
      NSTA=NSTA+1
      STNM(J)=S(1,L)
      ISEM(1,L)=IS(1,L)
      DO 62 I=2,5
62    IF(IS(I,L).NE.0) ISEM(I,L)=1
63    NBEA=0
      IF(IBC.EQ.0) GO TO 80
      DO 64 I=1,10
      IF(IBC(I).EQ.0) GO TO 65
64    NBEA=NBEA+1
65    DO 66 I=1,4
66    IF(IB(I,11).NE.0) ISEM(I,13)=1
      ISEM(5,13)=IB(3,12)
80    CONTINUE
      KEY=2
      IF(NCH.LT.0) KEY=-2
      IF(INITS.EQ.0) GO TO 22
      TSTDP=(TI(1)+TI(2))*SPMSD+TWT(1)-TSECO
      TSPDP=TSTDP+FLTIM-TWT(1)
      DO 81 I=1,20
   81 STIMNX(I)=TSTDP
      WRITE(12) (HEAD(I),I=13,23),NSTA,STNM,TSTDP,TSPDP,STIMNX
C***  CHECK BINARY OUTPUT
      WRITE(6,603) (HEAD(I),I=13,23),TSTDP
  603 FORMAT(1H1,45X,29H***  SIMULATED DATA TAPE  ***/8H HEADER,11A6/
     119H0FIRST DATA ON TAPE,D23.16,20H SECONDS FROM 1950.0)
      TEMP=(TSTDP-.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,604) TSPDP
  604 FORMAT(19H0 LAST DATA ON TAPE,D23.16,20H SECONDS FROM 1950.0)
      TEMP=(TSPDP-.5364576D9)*RSPMSD
      CALL TFRAC(6209.,TEMP,XO1,XO2)
      CALL DATOUT(XO1,XO2,YO1,YO2,0)
      WRITE(6,605) (STNM(I),STIMNX(I),I=1,NSTA)
  605 FORMAT(1H0,51X,16HSTATIONS ON TAPE/49X,4HNAME,5X,13HFIRST ON TIME/
     1/(48X,A6,E18.8))
C***  END BINARY OUTPUT
```

```
      INITS = 0                                                         SKDL0150
22    CONTINUE                                                          SKDL0151
      TSTART = TWT(1)                                                   SKDL0152
230   RETURN                                                            SKDL0153
      END                                                               SKDL0154
```

Subroutine: SMEAS

Purpose: Makes a set of station measurements, RMEAS(2-5), at time RMEAS(1).

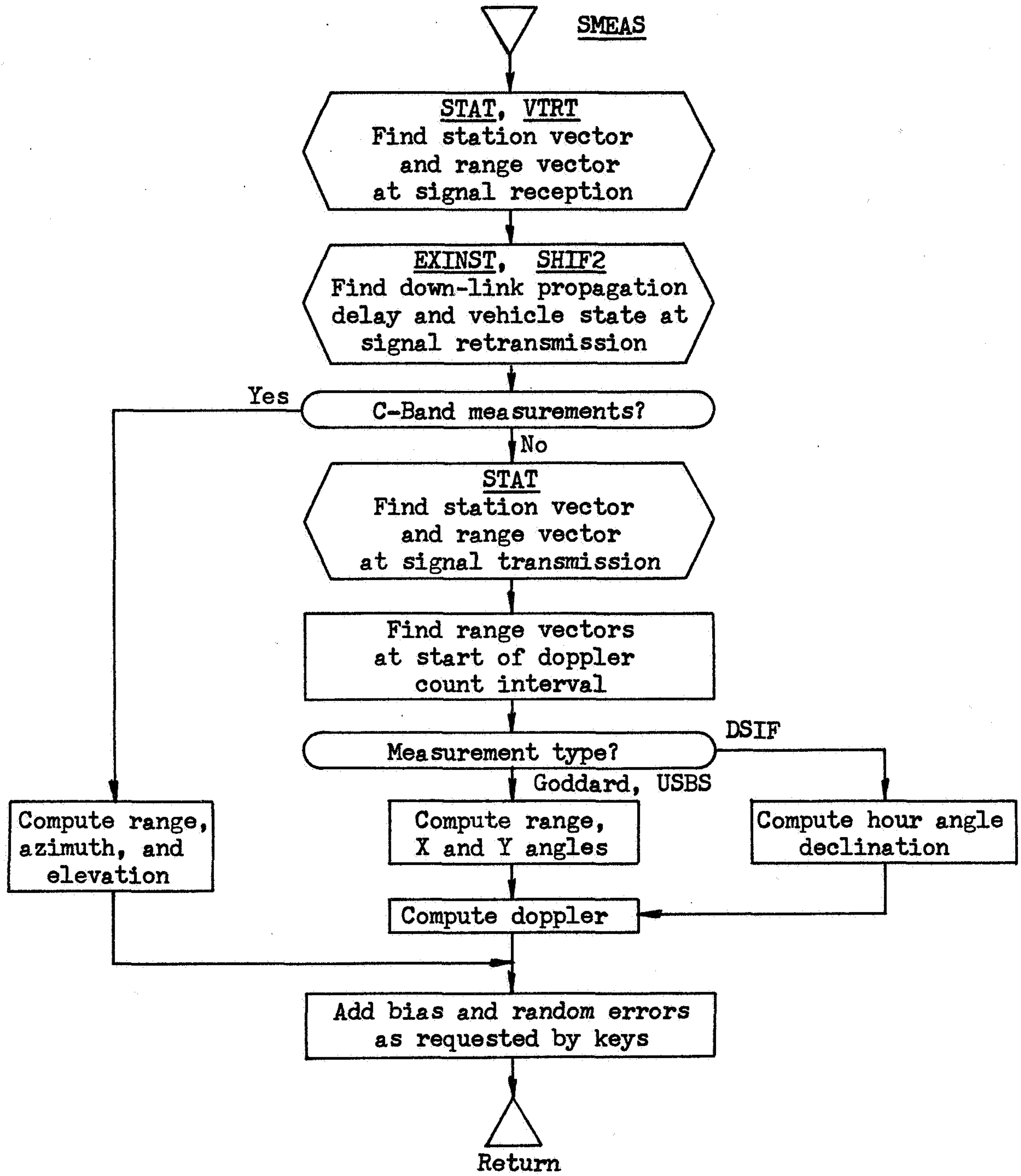
Calling Sequence: CALL SMEAS (SA,BRAD,XØUT,RMEAS,ISA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	SA	(23)			Appropriate column of station data array, S.
I	BRAD	(3)			1 - Earth semi-major axis 2 - Earth semi-minor axis 3 - Earth spin rate.
I	XØUT	(6,10)			Vehicle position, velocity with respect to all bodies.
Ø	RMEAS	(5)			1 - Time 2 - Angle 1 3 - Angle 2 4 - Range 5 - Doppler
I	ISA	(11)			Station measurement treatment keys.

Common storages used: /INPCØM/,/WCØM/,/EXIC/

Subroutines required: ARKTNS,CRØSS,ERRØUT,EXINST,FNØRM,SHIF2,STAT, TFRAC,VTRN,VTRT


SMEAS
STAT, VTRT
Find station vector and range vector at signal reception
EXINST, SHIF2
Find down-link propagation delay and vehicle state at signal retransmission
C-Band measurements?
Yes
No
STAT
Find station vector and range vector at signal transmission
Find range vectors at start of doppler count interval
Measurement type?
DSIF
Goddard, USBS
Compute range, azimuth, and elevation
Compute range, X and Y angles
Compute hour angle declination
Compute doppler
Add bias and random errors as requested by keys
Return

SMEAS-2

```
$IBFTC MC13SM  NOREF,M94,NODD,XR3
CMC13SM   COMPUTES STATION MEASUREMENTS, RMEAS(2-5), AT T=RMEAS(1)
      SUBROUTINE SMEAS(SA,BRAD,XOUT,RMEAS,ISA)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)/EXIC/W(36),CRX(9,2)
      DIMENSION    AN(3,3),        BRAD(3),        CROB(3),        DCXSOB(3)
     1,            DNOB(3),        DNTR(3),        DUM(3),         EAOB(3)
     2,            EATR(3),        ENOB(3),        ENTR(3),        EK(3)
     3,            ISA(11),        KRX(9,2),       OMG(3),         R(4)
     4,            RMEAS(5),       RTOB(3),        RTTR(3),        SA(23)
     5,            SCEK(3),        SCOB(5),                        SXOB(3)
     6,            SXTR(3),        TI(2),          TIX(2),         TMOB(9)
     7,            TMTR(9),        X(6),           XIN(6),         XOUT(6,10)
     8,            XSOB(6),        XSOBX(3),       XSTR(6),        XSTRX(3)
      DIMENSION    TMOBX(9),       DUM1(3),        DUM2(3),        XX(3)
     1,            RTMG(2),        EDOT(2),        RM(2),          E(2)
     2,            DELTE(2),       DR(2),          DRX(2),         Z1(6)
     3,            DCXSTR(3)
      EQUIVALENCE  (C(116),CFRAC1),  (C(117),CFRAC2)
      EQUIVALENCE  (C(7),RSPMSD),    (C(9),SPDLT),       (IW(382),NOR)
     1,            (IW(489),ITRIG),  (IW(490),KOUNT)
     2,                              (CW(1140),TSEC),    (CW(1160),TI)
     3,            (CW(1163),AN),    (CW(1172),TSECO),   (CW(1173),GHAR)
     4,            (CRX,KRX),        (R(1),ANG1),        (R(2),ANG2)
     5,            (SXOB(1),XTNOB),  (SXOB(2),XTEOB),    (SXOB(3),XTDOB)
     6,            (SXTR(1),XTNTR),  (SXTR(2),XTETR),    (SXTR(3),XTDTR)
     7,            (TMOB(1),ENOB),   (TMOB(4),EAOB),     (TMOB(7),DNOB)
     8,            (TMTR(1),ENTR),   (TMTR(4),EATR),     (TMTR(7),DNTR)
      EQUIVALENCE  (RM(1),RATOB),    (RM(2),RATTR)
      DOUBLE PRECISION  TWR
      DATA     OMG,EK/5*0.,1./
      MTYPE=ISA(1)
      NALIGN=ISA(10)
      NFRAC=ISA(11)
      TIX(1)=TI(1)
      TIX(2)=TI(2)
      RMEAS(1)=TSEC
      OMG(3)=BRAD(3)
C     RECEIVING STATION POSITION AT END OF COUNT INTERVAL
      GHAN=GHAR+BRAD(3)*(TSEC-TSECO)
      CALL STAT(SA(3),GHAN,TMOB,RTOB,SCOB,BRAD)
      SPHI=SCOB(1)
      CPHI=SCOB(2)
      RTMG(1)=FNORM(RTOB)
C     DOWN-LINK PROPAGATION DELAY
      CALL VTRN(AN,XOUT,X)
      DO 10 I=1,3
   10 XSOB(I)=X(I)-RTOB(I)
      RATOB=FNORM(XSOB)
      ITERAT=0
      TSET=TSEC
   20 ITERAT=ITERAT+1
      TV=TSEC-RATOB/SPDLT
      TWR=TV
      M=1
      N=3
   30 CONTINUE
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
      IF(M.LE.0) CALL ERROUT(1,10H(4H0M1=I2),M)
      IF(N) 50,40,60
   40 CONTINUE
      CALL ERROUT(1,28H(23H0ERROR ON BIN TAPE READ))
   50 CONTINUE
      N=-N
      NOR=KRX(7,N)
      GO TO 30
   60 CONTINUE
      CALL TFRAC(TIX(1),TIX(2)+(TV-TSET)*RSPMSD,TIX(1),TIX(2))
      TSET=TV
      CALL SHIF2(2,NOR,TIX,XIN,1,XOUT)
      CALL VTRN(AN,XOUT,X)
      DO 70 I=1,3
   70 XSOB(I)=X(I)-RTOB(I)
      R2OB=DOT(XSOB,XSOB)
      RATOB=SQRT(R2OB)
      IF(ITERAT.EQ.1) GO TO 20
```

```
      CALL CROSS(OMG,RTOB,DUM)
      CALL VTRN(AN,XOUT(4,1),X(4))
      DO 80 I=4,6
   80 XSOB(I)=X(I)-DUM(I-3)
      CALL VTRT(TMOB,XSOB,SXOB)
C     WE NOW HAVE DOWN-LINK RANGE AND ASSOCIATED VECTORS, END OF COUNT
      IF(MTYPE.EQ.1) GO TO 131
C     NOW WE GET DOWN-LINK VECTORS AT BEGINNING OF COUNT INTERVAL
      GHAX=GHAR+BRAD(3)*(TSEC-TSECO-SA(23))
      CALL STAT(SA(3),GHAX,TMOBX,RTOB,SCOB,BRAD)
      TWR=TSEC-SA(23)
      M=1
      N=3
   81 CONTINUE
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
      IF(M.LE.0) CALL ERROUT(1,10H(4HOM1=I2),M)
      IF(N) 83,82,84
   82 CONTINUE
      CALL ERROUT(1,28H(13H0ERROR ON BIN TAPE READ))
   83 CONTINUE
      N=-N
      NOR=KRX(7,N)
      GO TO 81
   84 CONTINUE
      CALL TFRAC(TI(1),TI(2)-SA(23)*RSPMSD,TIX(1),TIX(2))
      CALL SHIF2(2,NOR,TIX,XIN,1,XOUT)
      CALL VTRN(AN,XOUT,XX)
      DO 85 I=1,3
   85 XSOBX(I)=XX(I)-RTOB (I)
      RATOBX=FNORM(XSOBX)
C     WE NOW HAVE RV(TV), RV(TV-TAU), AND ASSOCIATED VECTORS
C     NOW WE GET RT(TTR), RT(TTR-TAU),AND ASSOCIATED VECTORS
C     FIRST, UP-LINK VECTORS AT END OF COUNT
C     UP-LINK PROPAGATION DELAY
      TTR=TV-RATOB/SPDLT
      GHAX=GHAR+BRAD(3)*(TTR-TSECO)
      CALL STAT(SA(3),GHAX,TMTR,RTTR,SCOB,BRAD)
      DO 90 I=1,3
   90 XSTR(I)=X(I)-RTTR(I)
      RATTR=FNORM(XSTR)
      ITERAT=0
  100 ITERAT=ITERAT+1
      TTR=TV-RATTR/SPDLT
      GHAX=GHAR+BRAD(3)*(TTR-TSECO)
      CALL STAT(SA(3),GHAX,TMTR,RTTR,SCOB,BRAD)
      DO 110 I=1,3
  110 XSTR(I)=X(I)-RTTR(I)
      R2TR=DOT(XSTR,XSTR)
      RATTR=SQRT(R2TR)
      IF(ITERAT.EQ.1) GO TO 100
      RTMG(2)=FNORM(RTTR)
      CALL CROSS(OMG,RTTR,DUM)
      DO 120 I=4,6
  120 XSTR(I)=X(I)-DUM(I-3)
      CALL VTRT(TMTR,XSTR,SXTR)
C     NOW WE GET UP-LINK VECTORS AT BEGINNING OF COUNT INTERVAL
      GHAX=GHAX-BRAD(3)*SA(23)
      CALL STAT(SA(3),GHAX,TMOBX,RTOB,SCOB,BRAD)
      DO 130 I=1,3
  130 XSTRX(I)=XX(I)-RTOB (I)
      RATTRX=FNORM(XSTRX)
C     COMPUTE AZIMUTH AND ELEVATION FOR C-BAND OR REFRACTION
  131 E(1)=ATAN(-XTDOB/SQRT(XTNOB*XTNOB+XTEOB*XTEOB))
      AZ=ARKTNS(360,XTNOB,XTEOB)
      IF(NFRAC.EQ.2) GO TO 139
      JJ=1
      IF(MTYPE.EQ.1) GO TO 132
      JJ=2
      E(2)=ATAN(-XTDTR/SQRT(XTNTR*XTNTR+XTETR*XTETR))
C     HERE WE COMPUTE RANGE AND ELEVATION REFRACTION CORRECTION
  132 DO 138 I=1,JJ
      DELTE(I)=0.
      IF(E(I).LT..01) GO TO 138
      SE=SIN(E(I))
      CE=COS(E(I))
```

```
      IF(E(I).GT..17453293) GO TO 137                                   SMES0150
      T=(1.03585796-(.01072014-(.1279119E-7-.1227363E-7/E(I))/E(I))/    SMES0151
     1E(I))*CFRAC1 *CE/SE                                               SMES0152
      F=RTMG(I)/RM(I)                                                   SMES0153
      DELTE(I)=T-F*((CFRAC1+T*T/2.)*CE-T*SE)                            SMES0154
      GO TO 138                                                         SMES0155
  137 DELTE(I)=CFRAC1*CE/SE                                             SMES0156
  138 DR(I)=CFRAC1/(CFRAC2*SIN(E(I)+DELTE(I)))*1.E-3                    SMES0157
  139 GO TO(140,150,150,180),MTYPE                                      SMES0158
C     C-BAND MEASUREMENTS                                               SMES0159
  140 CONTINUE                                                          SMES0160
      ANG1=AZ                                                           SMES0161
      ANG2=E(1)                                                         SMES0162
      R(3)=RATOB                                                        SMES0163
      R(4)=-.12345678E20                                                SMES0164
      GO TO (141,190),NFRAC                                             SMES0165
C     C-BAND REFRACTION                                                 SMES0166
C     AZIMUTH IS ASSUMED UNAFFECTED BY REFRACTION                       SMES0167
  141 ANG2=ANG2+DELTE(1)                                                SMES0168
      R(3)=R(3)+DR(1)                                                   SMES0169
      GO TO 190                                                         SMES0170
C     GODDARD OR USBS MEASUREMENTS                                      SMES0171
  150 CONTINUE                                                          SMES0172
      IF(MTYPE.EQ.2) NALIGN=1                                           SMES0173
      GO TO (160,170),NALIGN                                            SMES0174
C     30-FOOT DISH                                                      SMES0175
  160 CALL CROSS(ENOB,XSOB,CROB)                                        SMES0176
      ANG1=ATAN2(XTEOB,-XTDOB)                                          SMES0177
      ANG2=ATAN(XTNOB/FNORM(CROB))                                      SMES0178
      GO TO 175                                                         SMES0179
C     85-FOOT DISH                                                      SMES0180
  170 CALL CROSS(EAOB,XSOB,CROB)                                        SMES0181
      ANG1=ATAN2(-XTNOB,-XTDOB)                                         SMES0182
      ANG2=ATAN(XTEOB/FNORM(CROB))                                      SMES0183
  175 R(3)=RATOB+RATTR                                                  SMES0184
      GO TO(176,177),NFRAC                                              SMES0185
C     USBS ANGLE REFRACTION                                             SMES0186
  176 SX=SIN(ANG1)                                                      SMES0187
      CX=COS(ANG1)                                                      SMES0188
      SY=SIN(ANG2)                                                      SMES0189
      CY=COS(ANG2)                                                      SMES0190
      ANG1=ANG1-SX/(CY*COS(E(1)))*DELTE(1)                              SMES0191
      ANG2=ANG2-CX*SY/COS(E(1))*DELTE(1)                                SMES0192
      R(3)=R(3)+DR(1)+DR(2)                                             SMES0193
  177 IF(MTYPE .EQ.2) R(3)=R(3)/SPDLT                                   SMES0194
  180 CONTINUE                                                          SMES0195
C     DOPPLER OBSERVABLE FOR GODDARD, USBS, OR DSIF                     SMES0196
      IF(MTYPE.EQ.2) SA(21)=1.                                          SMES0197
      CONST=SA(21)*SA(22)/SPDLT                                         SMES0198
      R(4)=SA(20)*SA(23)+CONST*(RATOB+RATTR-RATOBX-RATTRX)              SMES0199
      GO TO(181,185),NFRAC                                              SMES0200
C     DOPPLER REFRACTION CORRECTION                                     SMES0201
  181 CALL CROSS(XSOB,OMG,Z1)                                           SMES0202
      CALL CROSS(XSTR,OMG,Z1(4))                                        SMES0203
      DO 1810 I=1,3                                                     SMES0204
      Z1(I)=Z1(I)+XSOB(I+3)                                             SMES0205
 1810 Z1(I+3)=Z1(I+3)+XSTR(I+3)                                         SMES0206
      CALL CROSS(DNOB,XSOB,DCXSOB)                                      SMES0207
      CALL CROSS(DNTR,XSTR,DCXSTR)                                      SMES0208
      CALL CROSS(DCXSOB,XSOB,DUM1)                                      SMES0209
      CALL CROSS(DCXSTR,XSTR,DUM2)                                      SMES0210
      DEN1=R2OB*FNORM(DCXSOB)                                           SMES0211
      DEN2=R2TR*FNORM(DCXSTR)                                           SMES0212
      DO 182 I=1,3                                                      SMES0213
      DUM1(I)=DUM1(I)/DEN1                                              SMES0214
  182 DUM2(I)=DUM2(I)/DEN2                                              SMES0215
      EDOT(1)=DOT(DUM1,Z1)                                              SMES0216
      EDOT(2)=DOT(DUM2,Z1(4))                                           SMES0217
      E1X=DELTE(1)-SA(23)*EDOT(1)                                       SMES0218
      E2X=DELTE(2)-SA(23)*EDOT(2)                                       SMES0219
      DRX(1)=CFRAC1/(CFRAC2*SIN(E1X))*1.E-3                             SMES0220
      DRX(2)=CFRAC1/(CFRAC2*SIN(E2X))*1.E-3                             SMES0221
      R(4)=R(4)+CONST*(DR(1)+DR(2)-DRX(1)-DRX(2))                       SMES0222
  185 IF(MTYPE.EQ.2) R(4)=2.*SA(20)*SA(23)-R(4)                         SMES0223
      IF(MTYPE.NE.4) GO TO 190                                          SMES0224
```

```
C     DSIF ANGLES                                                       SMES0225
      R(3)=-.12345678E20                                                SMES0226
      ANG1=GHAN+SA(4)  -ARKTNS(360,XSOB(1),XSOB(2))                     SMES0227
      CALL CROSS(XSOB,EK,SCEK)                                          SMES0228
      ANG2=ATAN(XSOB(3)/FNORM(SCEK))                                    SMES0229
      GO TO (189,190),NFRAC                                             SMES0230
C     DSIF ANGLE REFRACTION                                             SMES0231
  189 ANG1=ANG1+SIN(ANG1)**2*CPHI/(SIN(AZ)*COS(E(1))**2)*DELTE(1)       SMES0232
      ANG2=ANG2+(COS(E(1))*SPHI-COS(AZ)*SIN(E(1))*CPHI)/COS(ANG2)       SMES0233
     1*DELTE(1)                                                         SMES0234
  190 CONTINUE                                                          SMES0235
C     ADD RANDOM AND BIAS ERRORS                                        SMES0236
      DO 210 I=1,4                                                      SMES0237
      J=MOD(I+1,4)+2                                                    SMES0238
      IF(ISA(J).EQ.0) GO TO 209                                         SMES0239
      IF(ISA(J).EQ.2) GO TO 200                                         SMES0240
      BIAS=SA(J+10)                                                     SMES0241
      IF(I.EQ.3.AND.ISA(1).EQ.2) BIAS=BIAS/SPDLT                        SMES0242
      IF(I.EQ.4) BIAS=BIAS*SA(23)                                       SMES0243
      R(I)=R(I)+BIAS                                                    SMES0244
  200 RANDOM=SA(J+6)                                                    SMES0245
      IF(I.EQ.3.AND.ISA(1).EQ.2) RANDOM=RANDOM/SPDLT                    SMES0246
      IF(I.EQ.4) RANDOM=RANDOM*SA(23)                                   SMES0247
      R(I)=R(I)+RANDOM*BARN(-1)                                         SMES0248
      RMEAS(I+1)=R(I)                                                   SMES0249
      GO TO 210                                                         SMES0250
  209 RMEAS(I+1)=-.12345678E20                                          SMES0251
  210 CONTINUE                                                          SMES0252
C     INSURE ANGLES TO BE BETWEEN ZERO AND TWO PI                       SMES0253
      IF(RMEAS(2).LT.0..AND.RMEAS(2).NE.-.12345678E20) RMEAS(2)=RMEAS(2)SMES0254
     1+6.28318531                                                       SMES0255
      IF(RMEAS(3).LT.0..AND.RMEAS(3).NE.-.12345678E20) RMEAS(3)=RMEAS(3)SMES0256
     1+6.28318531                                                       SMES0257
      IF(RMEAS(2).GT.6.28318531) RMEAS(2)=RMEAS(2)-6.28318531           SMES0258
      IF(RMEAS(3).GT.6.28318531) RMEAS(3)=RMEAS(3)-6.28318531           SMES0259
      IF(ISA(9).NE.0) RMEAS(1)=RMEAS(1)+SA(19)                          SMES0260
      RETURN                                                            SMES0261
      END                                                               SMES0262
```

Subroutine: SØLRD

Purpose: To compute and sum the acceleration due to solar radiation pressure (see also DPSØLR).

Calling Sequence: CALL SØLRD (RC,RS,RP,C,A)

Input and Output

I/Ø	Symbolic Name of Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	RC	d(3)	R	km	Vehicle position relative to the central body.
I	RS	d(3)	R_s	km	Sun's position relative to the central body.
I	RP	d	R_o	km	Central body radius.
I	C	d	C	km^3/sec^2	Solar pressure coefficient.
IØ	A	d(3)	A	km/sec^2	Acceleration.

Common storages used: 16 cells

Subroutines required: DDØT, DVNØRM

```
$IBFTC MC132R  XR3,M94,NODD,LIST
      SUBROUTINE SOLRD (RC,RS,RP,C,A)
C     SOLAR RADIATION PRESSURE ACCELERATION
      DOUBLE PRECISION  A(3),C,RC(3),RS(3),RP
     1                 ,D,DDOT,DVNORM
      COMMON            D(8)
C
    1 D(1) = DDOT(RS,RS)
      D(2) = DDOT(RC,RS)/D(1)
      DO 2 I=1,3
      D(I+2) = RC(I)-RS(I)
    2 D(I+5) = RC(I)-D(2)*RS(I)
      IF(D(2).GE.0.D0) GO TO 3
      D(1) = DDOT(D(6),D(6))-RP*RP
      IF (D(1).LT.0.D0) GO TO 999
    3 D(1) = DVNORM(D(3),D(3))
      D(1) = C/D(1)/D(1)
      DO 4 I=1,3
    4 A(I) = A(I)+D(1)*D(I+2)
  999 RETURN
      END
```

Subroutine: SØLRDP

Purpose: To compute: the acceleration due to solar radiation pressure, the gradient of the acceleration, and the partial derivative of the acceleration with respect to the coefficient.

Calling Sequence: CALL SØLRDP (RC,RS,RP,C,A,AP,G,N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	RC	d(3)	R_c	km	Vehicle position relative to the central body.
I	RS	d(3)	R_s	km	Sun's position relative to the central body.
I	RP	d	R_o	km	Central body radius.
I	C	d	C	km^3/sec^2	Solar pressure coefficient.
IØ	A	d(3)	A	km/sec^2	Acceleration.
Ø	AP	d(3)	$\frac{1}{c}A$	$1/km^2$	∂(Acceleration)/∂C
IØ	G	d(3,3)	∇A	$1/sec^2$	Gradient of the acceleration.
I	N				Option key: =o: acceleration only. ≠o: ∂A/∂C and gradient.

Common storages used: 18 cells

Subroutines required: DDØT, DVNØRM

Method:

The subroutine computes acceleration due to solar radiation pressure from

$$A = C \frac{R}{r^3}$$

and sums this acceleration with the input acceleration. Here R is the position of the vehicle relative to the sun,

$$R = R_C - R_s$$

$$r = |R|$$

The gradient

$$\nabla_R A = \frac{C}{r^3} \left(I - \frac{3RR^T}{r^3}\right)$$

is summed with the input gradient, and if $N<0$, the partial derivative

$$\frac{\partial A}{\partial C} = \frac{R}{r^3}$$

is returned in AP.

Note that C is the value (assumed constant) of the solar radiation pressure constant ($.97 \times 10^{-7}$ lb/ft^2) x the reflective surface x the surface reflectivity - the vehicle mass, and that the component of acceleration normal to R is neglected.

The component of R normal to R_s is

$$R_n = R_c - d\, R_s$$

$$d = R_c \cdot R_s / R_s^2$$

The subroutine assumes that the space vehicle is shaded by the central body if both

$$d<0$$

$$|R_n| < R_o$$

```
$IBFTC MC132S  XR3,M94,NODD,LIST
      SUBROUTINE SOLRDP (RC,RS,RP,C,A,AP,G,N)
C     SOLAR RADIATION PRESSURE ACCELERATION
      DOUBLE PRECISION   A(3),AP(3),C,G(3,3),RC(3),RS(3),RP
     1                   ,D,DDOT,DVNORM
      COMMON             D(9)
C
    1 D(1) = C
      D(2) = DDOT(RS,RS)
      D(3) = DDOT(RC,RS)/D(2)
      DO 2 I=1,3
      D(I+3) = RC(I)-RS(I)
    2 D(I+6) = RC(I)-D(3)*RS(I)
      IF(D(3).GE.0.D0) GO TO 3
      D(2) = DDOT(D(7),D(7))-RP*RP
      IF (D(2).LT.0.D0) D(1) = 0.D0
    3 D(2) = DVNORM(D(4),D(4))
      D(2) = D(1)/D(2)/D(2)
      D(1) = D(2)/D(1)
      IF (N.GE.0) GO TO 5
      DO 4 I=1,3
    4 AP(I) = D(1)*D(I+3)
    5 IF (D(1).EQ.0.D0) GO TO 999
      DO 6 I=1,3
    6 A(I) = A(I)+D(2)*D(I+3)
      IF (N.EQ.0) GO TO 999
      D(3) =-3.D0*D(2)
      DO 7 I=1,3
      G(I,I) = G(I,I)+D(2)
      D(1) = D(3)*D(I+3)
      DO 7 J=1,3
    7 G(I,J) = G(I,J)+D(1)*D(J+3)
  999 RETURN
      END
```

Subroutine: SØRDR

Purpose: To sort an array X in ascending order, preserving the correspondence between X and another array, NX.

Calling Sequence: CALL SØRDR (X,NX,KK)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	X	KK			Array to be sorted.
IØ	NX	KK			The elements of NX are sorted in the same way as the elements of X.
I	KK				Number of elements of X (and NX).

Common storages used: None

Subroutines required: None

```
$IBFTC MC13SO   NOREF,M94,NODD,XR3
CMC13SO      SORDR
      SUBROUTINE SORDR(X,NX,KK)
      DIMENSION X(1),NX(1)
      IF(KK.EQ.1) GO TO 6
      JJ=1
      DO 5 J=2,KK
      L=J-1
      IF(X(J).GT.X(L)) GO TO 5
      XSAV=X(J)
      NSAV=NX(J)
      X(J)=X(L)
      NX(J)=NX(L)
      GO TO (1,2),JJ
1     JJ=2
      IF(J-2) 2,4,2
2     DO 3 I=3,J
      K=L-1
      IF(XSAV.GT.X(K)) GO TO 4
      X(L)=X(K)
      NX(L)=NX(K)
      L=L-1
3     CONTINUE
4     X(L)=XSAV
      NX(L)=NSAV
5     CONTINUE
6     RETURN
      END
```

Subroutine: SØRDR2

Purpose: A push-down sort routine that sorts in ascending order a floating point number array and retains the correspondence with a fixed-point array and, optionally, a second floating point array.

Calling Sequence: CALL SØRDR2 (X,NX,KK,M,T)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	X	as required			Floating point array sorted in ascending order.
IØ	NX	as required			A fixed point array whose correspondence with X is retained.
I	KK				Number of elements of X to be sorted.
I	M				Option key M=1, Y is left untouched M=2, Y is altered to retain correspondence with X.
IØ	Y	as required			Secondary array, the treatment of which depends on M

Common storages used: None

Subroutines required: None

```
$IBFTC MC1302  NOREF,M94,NODD,XR3
CMC1302  SORDR2   2 FLOAT 1 FIX ARRAY ORDERED AS X
      SUBROUTINE SORDR2(X,NX,KK,M,Y)
      DIMENSION  X(1),    Y(1),    NX(1)
      LOGICAL KOY
      IF(KK.EQ.1) GO TO 6
      KOY=M.EQ.2
      JJ=1
      DO 5 J=2,KK
      L=J-1
      IF(X(J).GT.X(L)) GO TO 5
      XSAV=X(J)
      NSAV=NX(J)
      IF(KOY) YSAV=Y(J)
      X(J)=X(L)
      NX(J)=NX(L)
      IF(KOY) Y(J)=Y(L)
      GO TO (1,2),JJ
1     JJ=2
      IF(J-2) 2,4,2
2     DO 3 I=3,J
      K=L-1
      IF(XSAV.GT.X(K)) GO TO 4
      X(L)=X(K)
      NX(L)=NX(K)
      IF(KOY) Y(L)=Y(K)
      L=L-1
3     CONTINUE
4     X(L)=XSAV
      NX(L)=NSAV
      IF(KOY) Y(L)=YSAV
5     CONTINUE
6     RETURN
      END
```

Subroutine: STARTB

Purpose: To load the next beacon on time from the BSTART array into the critical event array in position EVNT(1).

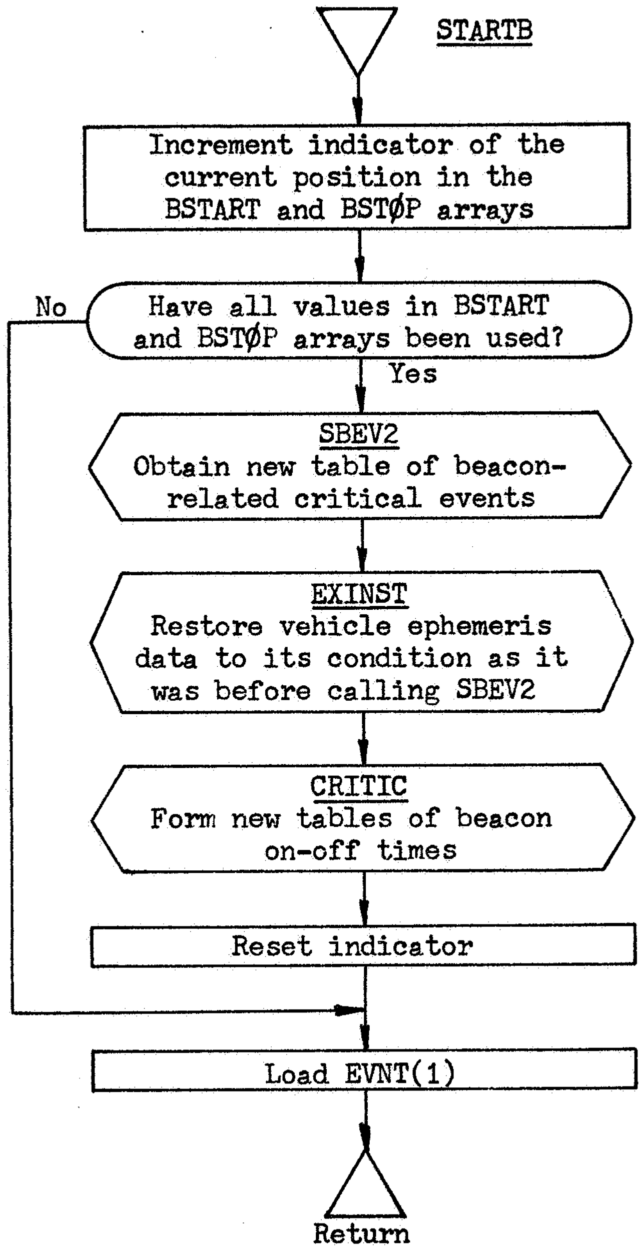
Calling Sequence: CALL STARTB (KMAXB,KBEA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	KMAXB				Number of on-off time pairs in the BSTART and BSTØP arrays.
IØ	KBEA	(75)			Beacon numbers corresponding to on-off time pairs in BSTART and BSTØP arrays.

Common storages used: /INPCØM/,/WCØM/,/EXIC/

Subroutines required: CRITIC, EXINST, SBEV2


STARTB
Increment indicator of the current position in the BSTART and BSTØP arrays
Have all values in BSTART and BSTØP arrays been used?
No
Yes
SBEV2
Obtain new table of beacon-related critical events
EXINST
Restore vehicle ephemeris data to its condition as it was before calling SBEV2
CRITIC
Form new tables of beacon on-off times
Reset indicator
Load EVNT(1)
Return

```
$IBFTC MC13AB  NOREF,M94,NODD,XR3
CMC13AB  STARTB  LOADS EVNT1 WITH NEXT BSTART
      SUBROUTINE STARTB(KMAXB,KBEA)
C  SUBROUTINES REQUIRED  SBEV2,   CRITIC
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)
      COMMON/EXIC/W(36),CRX(9,2)
      DIMENSION     IBEE(10),      IBC(10),           IBTIM(50)
     1,             B(91),         XIN(6),            TI(3)
     2,             BTIME(50),     BECR(10),          MBOD(4)
     3,             TIM(2),        IB(36)
      DIMENSION      BSTART(75),   BSTOP(75),      KBEA(75)
      EQUIVALENCE  (C(476),B)
      EQUIVALENCE  (IW(20),IBC),   (IW(367),IBEE),     (IW(382),NOR)
     1,            (IW(386),IK2),  (IW(387),KBMAX),    (IW(439),IBTIM)
     2,            (IW(312),IB),  (CW(1134),XIN),     (CW(1140),TSEC)
     3,            (CW(1160),TI), (CW(1180),EVNT1),   (CW(1189),TWT2)
     4,            (CW(1250),BTIME),                  (CW(1441),BECR)
      EQUIVALENCE   (CW(372),BSTART),  (CW(447),BSTOP)
      EQUIVALENCE  (CW(1179),FLTIM)
      EQUIVALENCE  (IW(489),ITRIG),  (IW(490),KOUNT)
      DOUBLE PRECISION TWR
      IK2 = IK2+1
      IF(IK2-KMAXB) 175,175,174
174   CONTINUE
      MBOD(1) = IB(36)
      MBOD(2) =0
      TIM=TSEC
C  SHOULD = BTIME(KBMAX)      HOPEFURLLY IT DOES FROM MEASUR
      TIM(2)=TWT2
      IF(TWT2.GT.FLTIM) TIM(2)=FLTIM
C   NEED XIN AND TI AT TIM  PERHAPS THERE FROM MEASUR(PROP)
      ITRIG=1
      CALL SBEV2(XIN,TI,NOR,TIM,B,8,10,IBC,BECR,IBEE,IBTIM,BTIME,MBOD)
      IF(KOUNT.LE.0) GO TO 1742
      DO 1741 I=1,KOUNT
 1741 BACKSPACE 10
      M=0
      N=18
      TWR=0.D0
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
 1742 KBMAX=MBOD(4)
      ITRIG=0
      KOUNT=0
      CALL CRITIC(10,IBC,BTIME,IBTIM,IBEE,KBMAX,TIM,BSTART,BSTOP,KBEA,
     1KMAXB)
      IK2 = 1
175   CONTINUE
      EVNT1=BSTART(IK2)
      RETURN
      END
```

Subroutine: STARTS

Purpose: To load the next station on time from the SSTART array into the critical event array in position EVNT(1).

Calling Sequence: CALL STARTS (KMAXS,KSTA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	KMAXS				Number of on-off time pairs in the SSTART and SSTØP arrays.
IØ	KSTA	(75)			Station numbers corresponding to on-off time pairs in SSTART and SSTØP arrays.

Common storages used: /INPCØM/,/WCØM/,/EXIC/

Subroutines required: CRITIC, EXINST, SBEV2

See also STARTB

```
$IBFTC MC13AS   NOREF,M94,NODD,XR3
CMC13AS  STARTS  LOADS EVNT1 WITH NEXT SSTART                           STAS0001
      SUBROUTINE STARTS(KMAXS,KSTA)                                     STAS0002
C  SUBROUTINES REQUIRED  SBEV2,    CRITIC                               STAS0003
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)                        STAS0004
      COMMON/EXIC/W(36),CRX(9,2)                                        STAS0005
      DIMENSION     ISEE(12),      ISC(12),       ISTIM(50)             STAS0006
     1,             S(23,12),      XIN(6),        TI(3)                 STAS0007
     2,             STIME(50),     SECR(12),      MBOD(4)               STAS0008
     3,             TIM(2)                                              STAS0009
      DIMENSION      SSTART(75),   SSTOP(75),      KSTA(75)             STAS0010
      EQUIVALENCE  (C(200),S)                                           STAS0011
      EQUIVALENCE   (IW(383),ITARG)                                     STAS0012
      EQUIVALENCE  (IW(8),ISC),    (IW(355),ISEE),     (IW(382),NOR)    STAS0013
     1,            (IW(384),IK1), (IW(385),KSMAX),    (IW(389),ISTIM)   STAS0014
     2,            (CW(1134),XIN),(CW(1140),TSEC)                       STAS0015
     3,            (CW(1160),TI), (CW(1179),FLTIM),   (CW(1180),EVNT1)  STAS0016
     4,            (CW(1189),TWT2),(CW(1200),STIME),  (CW(1360),SECR)   STAS0017
      EQUIVALENCE   (CW(222),SSTART),  (CW(297),SSTOP)                  STAS0018
      EQUIVALENCE  (IW(489),ITRIG),  (IW(490),KOUNT)                    STAS0019
      DOUBLE PRECISION TWR                                              STAS0020
      IK1 = IK1+1                                                       STAS0021
      IF(IK1-KMAXS) 169,169,168                                         STAS0022
168   CONTINUE                                                          STAS0023
      MBOD(2)=ITARG                                                     STAS0024
      IF(ITARG.EQ.1) MBOD(2)=0                                          STAS0025
      MBOD(3) = 0                                                       STAS0026
      TIM=TSEC                                                          STAS0027
C  SHOULD = STIME(KSMAX)      HOPEFURLLY IT DOES FROM MEASUR            STAS0028
      TIM(2)=TWT2                                                       STAS0029
      IF(TWT2.GT.FLTIM) TIM(2)=FLTIM                                    STAS0030
C   NEED XIN AND TI AT TIM  PERHAPS THERE FROM MEASUR(PROP)             STAS0031
      ITRIG=1                                                           STAS0032
      CALL SBEV2(XIN,TI,NOR,TIM,S,23,12,ISC,SECR,ISEE,ISTIM,STIME,MBOD) STAS0033
      IF(KOUNT.LE.0) GO TO 1682                                         STAS0034
      DO 1681 I=1,KOUNT                                                 STAS0035
 1681 BACKSPACE 10                                                      STAS0036
      M=0                                                               STAS0037
      N=18                                                              STAS0038
      TWR=0.D0                                                          STAS0039
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)              STAS0040
 1682 KSMAX=MBOD(4)                                                     STAS0041
      ITRIG=0                                                           STAS0042
      KOUNT=0                                                           STAS0043
      CALL CRITIC(12,ISC,STIME,ISTIM,ISEE,KSMAX,TIM,SSTART,SSTOP,KSTA,  STAS0044
     1KMAXS)                                                            STAS0045
      IK1 = 1                                                           STAS0046
169   CONTINUE                                                          STAS0047
      EVNT1=SSTART(IK1)                                                 STAS0048
      RETURN                                                            STAS0049
      END                                                               STAS0050
```

Subroutine: STAT

Purpose: To obtain the vector from a body center to a tracking station and an orthogonal transformation relating inertial Cartesian coordinates X, Y, Z to local tangent plane coordinates North, East, Down.

Calling Sequence: CALL STAT (STA,GHAR,TR,RT,S,BRAD)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	STA	(3)	θ	rad	STA(1) = Station geodetic latitude
			λ	rad	STA(2) = Station geographic longitude, positive East
			h	km	STA(3) = Station altitude
I	GHAR			rad	Greenwich hour angle
Ø	TR	(3,3)			TR(1,I) = North unit vector TR(2,I) = East unit vector TR(3,I) = Down unit vector
Ø	RT	(3)		km	Inertial radius vector to station
Ø	S	(5)			S(1) = $\sin\theta$ S(2) = $\cos\theta$ S(3) = $\sin(\lambda + GHAR)$ S(4) = $\cos(\lambda + GHAR)$
			c		S(5) = $(\cos^2\theta + b^2\sin^2\theta/a^2)^{\frac{1}{2}}$
I	BRAD	(2)	a		BRAD(1) = semi-major axis of central body
			b		BRAD(2) = semi-minor axis of central body

Common storages used: None

Subroutines required: None

Method

Let the inertial longitude by represented by v, where

$$v = \lambda + GHAR$$

Then the orthogonal transformation relating inertial Cartesian coordinates RT to local tangent plane coordinates is given by the TR matrix

$$TR = \begin{bmatrix} -\sin\theta\cos v & -\sin v & -\cos\theta\cos v \\ -\sin\theta\sin v & \cos v & -\cos\theta\sin v \\ \cos\theta & 0 & -\sin\theta \end{bmatrix}$$

The inertial position of the station, RT, is

$$RT = \begin{bmatrix} (\frac{a}{c} + \sin v)\cos\theta\cos v \\ (\frac{a}{c} + \sin v)\cos\theta\sin v \\ (\frac{b^2}{ac} + \sin v)\sin\theta \end{bmatrix}$$

```
$IBFTC MC13SA  NOREF,M94,NODD,XR3
CMC13SA    SUBR. STAT           7/1/65     INCREASED CALL LIST
      SUBROUTINE STAT(STA,GHAR,TR,RT,S,BRAD)
      DIMENSION  STA(3),   TR(3,3),   RT(3),   S(5)
     1,          BRAD(2)
C     STA(1)=LATITUDE
C     STA(2)=LONGITUDE
C     STA(3)=ALTITUDE
      A=BRAD(1)
      B=BRAD(2)
      S(1)=SIN(STA(1))
      S(2)=COS(STA(1))
      TA=STA(2)+GHAR
      S(3)=SIN(TA)
      S(4)=COS(TA)
      S(5)= SQRT(S(2)*S(2)+B*B*S(1)*S(1)/(A*A))
      TR(1,1) = -S(1)*S(4)
      TR(2,1) = -S(1)*S(3)
      TR(3,1) =   S(2)
      TR(1,2) = -S(3)
      TR(2,2) =   S(4)
      TR(3,2)=0.
      TR(1,3) = -S(2)*S(4)
      TR(2,3) = -S(2)*S(3)
      TR(3,3) = -S(1)
      RT(1)=-(A/S(5)+STA(3))*TR(1,3)
      RT(2)=-(A/S(5)+STA(3))*TR(2,3)
      RT(3)=-(B*B/(A*S(5))+STA(3))*TR(3,3)
      RETURN
      END
```

Subroutine: STATE

Purpose: To compute central body position and velocity and Earth orientation, and to compute vehicle state relative to Earth.

Calling Sequence: CALL STATE(ET,UT,XX,TR,KEY)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ET	d			Ephemeris time, seconds from 1950 Jan 0.0 ET.
I	UT	d			Universal time, second from 1950 Jan 0.0 UT.
Ø	XX	d(6)			Vehicle state with respect to Earth, computed along the conic defined by the last call of STEPDI.
Ø	TR	d(9)			Transformation from Earth-fixed (B-frame) coordinates at UT to C-frame coordinates
I	KEY				= 0, compute TR only. ≠ 0, also compute XX and central body state (loaded in DFMCØM).

Common storages used: //120 cells,/,/DCPCØM/,/DFMCØM/,/ESTCØM/,/MLECØM/

Subroutines required: DGTRN,DMVTRN,DPFMRS,STEPDT

```
$IBFTC MC13W5   XR3,M94,NODD,LIST
      SUBROUTINE STATF (ET,UT,XX,TR,KEY)
C     COMPUTES EARTH ORIENTATION
C     COMPUTES VEHICLE AND CENTRAL BODY STATE
C
      DOUBLE PRECISION   ET ,UT ,XX(6) ,TR(9)
C
      COMMON             /DCPCOM/CDCP(900)
      EQUIVALFNCE        (CDCP(112),ETAPE ) ,(CDCP(111),IERR  )
     1                   ,(CDCP(777),ICB   )
C
      COMMON             /ESTCOM/CEST(804)
      EQUIVALENCE        (CEST( 61),FFEDAN)
C
      COMMON             /DATCOM/CDAT(299)
      DOUBLE PRECISION   OMEGA
      EQUIVALENCE        (CDAT(135),OMEGA )
C
      COMMON             /MLECOM/CMLE(1070)
      DOUBLE PRECISION   TB12C(3,3) ,UTIMR1
      EQUIVALENCE        (CMLE( 81),TB12C ) ,(CMLE(   7),UTIMR1)
C
      COMMON             /DFMCOM/IFM(14),RFM(6,12)
     1                            ,DFM (4),BFM (577),SNT(2,102)
      DOUBLE PRECISION   BFM,DFM,RFM
C
      COMMON             SAVE(90) ,TB2B1(9) ,XA(6)
      DOUBLE PRECISION   TB2B1 ,XA
C
    1 IF (IERR.NE.0) GO TO 999
      CALL DGTRN  (TB2B1,0,(UTIMR1-UT)*OMEGA,TB12C(1,3))
      CALL DMVTRN (TB12C,TB2B1,TR,1,3)
      IF (KEY.EQ.0) GO TO 999
      CALL DPFMRS (ET,EFEDAN,ICB,IERR,ETAPE)
      CALL STEPDT (ET,XA)
      DO 2 I=1,6
    2 XX(I) = XA(I)-RFM(I,3)
  999 RETURN
      END
```

Subroutine: STEPDI

Purpose: Computes initialization quantities for surboutine STEPDT(q.v.).

Calling Sequence: CALL STEPDI(T,X,U)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T	d	t_o	time	Time at initialization
I	X	d(6)	X_o	length, length/time	Cartesian state at initialization
I	U	d	μ	$\frac{\text{Length}^3}{\text{time}^2}$	Gravitational constant

Common storages used: /STPCØM/

Subroutines required: DDØT,DNØRM

```
$IBFTC MC13SE  XR3,M94,NODD,LIST
      SUBROUTINE STEPDI (T,X,U)                                        STPI0001
C     INITIALIZATION SUBROUTINE FOR STEPDT                             STPI0002
C                                                                      STPI0003
C     T = INITIAL TIME, SECONDS                                        STPI0004
C     X = INITIAL CARTESIAN STATE                                      STPI0005
C     U = GRAVITATIONAL CONSTANT                                       STPI0006
C                                                                      STPI0007
      DOUBLE PRECISION  T     ,X(6) ,U     ,DDOT ,DNORM,DSQRT,TPI      STPI0008
C                                                                      STPI0009
      COMMON      /STPCOM/TI    ,XI(6),SU   ,RI   ,AI   ,SA   ,DI   ,SR STPI0010
     1                  ,PR    ,HP   ,OM   ,EP   ,R    ,G(4) ,B    ,FB STPI0011
     2                  ,D     ,F(4)                                   STPI0012
      DOUBLE PRECISION  TI     ,XI   ,SU   ,RI   ,AI   ,SA   ,DI   ,SR STPI0013
     1                  ,PR    ,HP   ,OM   ,EP   ,R    ,G    ,B    ,FB STPI0014
     2                  ,D     ,F                                      STPI0015
      DATA  TPI/6.2831853071795866/                                    STPI0016
C                                                                      STPI0017
      DO 1 I=1,6                                                       STPI0018
    1 XI(I) = X(I)                                                     STPI0019
      TI = T                                                           STPI0020
      RI = DNORM(XI)                                                   STPI0021
      SU = DSQRT(U)                                                    STPI0022
      AI = 2.D0/RI-DDOT(XI(4),XI(4))/U                                 STPI0023
      PR = DABS(AI)                                                    STPI0024
      SA = DSQRT(PR)                                                   STPI0025
      DI = DDOT(XI,XI(4))/SU                                           STPI0026
      PR = TPI/SU/SA/PR                                                STPI0027
      HP = PR/2.D0                                                     STPI0028
      OM = 1.D0-RI*AI                                                  STPI0029
      SR = SU/RI                                                       STPI0030
      EP = SU*5.D-12                                                   STPI0031
      G(1) = 1.D0                                                      STPI0032
      DO 2 I=2,6                                                       STPI0033
    2 G(I) = 0.D0                                                      STPI0034
      D  = DI                                                          STPI0035
      R  = RI                                                          STPI0036
      RETURN                                                           STPI0037
      END
```

Subroutine: STEPDP

Purpose: Computes the state transition matrix, from time t_o to time t, on a conic section. Time t_o is the time of the last call of STEPDI and time t is the time of the last call of STEPDT(q.v.).

Calling Sequence: CALL STEPDP (PHI)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	PHI	d(6,6)	$\phi(t;t_o)$		State transition matrix

Common storages used: //42 cells,/STPCØM/

Subroutines required: None.

```
$IBFTC MC13S1  XR3,M94,NODD,LIST
      SUBROUTINE STEPDP (PHI)
C     CONIC TRANSITION MATRIX
C
      DOUBLE PRECISION   PHI(6,6)
      COMMON      /STPCOM/TI     ,XI(6),SU    ,RI    ,AI    ,SA    ,DI    ,SR
     1                    ,PR    ,HP    ,OM    ,EP    ,R     ,G(4)  ,B     ,FB
     2                    ,D     ,F(4)
      DOUBLE PRECISION    TI     ,XI    ,SU    ,RI    ,AI    ,SA    ,DI    ,SR
     1                    ,PR    ,HP    ,OM    ,EP    ,R     ,G     ,B     ,FB
     2                    ,D     ,F
      COMMON              C
      DOUBLE PRECISION    C(18)
C
    1 CONTINUE
      C( 1) = SU*SU
      C( 6) = G(3)/R
      C( 2) =      -B*G(2)
      C( 3) = G(4)-B*G(3)
      C( 4) =-(C(2)+G(3)+G(3))/AI
      C( 5) =-(C(3)+G(4)+G(4))/AI
      C(18) = (C(4)+DI*C(3)+RI*C(2))/C(1)/R
      C( 2) = (C(5)+DI*C(4)+RI*C(3))/C(1)/SU
      C(12) = C(18)*F( 3)
      C(18) =-C(18)*C( 6)
      C( 9) = C( 2)*F( 3)+C( 4)/C(1)/RI
      C(15) =-C( 2)*C( 6)+C( 5)/C(1)/SU
      C(16) = C( 1)/R/R/R
      C(13) = C(16)*F(1)
      C(16) = C(16)*F(2)
      C(12) = C(12)-C( 2)*C(13)+C( 3)/R/RI/SU
      C(18) = C(18)-C( 2)*C(16)+C( 4)/R/C(1)
      C( 5) = G( 3)/C( 1)
      C( 8) =-C( 5)*F( 3)
      C(14) = C( 5)*C( 6)
      C(17) = G( 2)/SU/R
      C(11) = C( 5)*C(13)-C(17)*F( 3)
      C(17) = C( 5)*C(16)+C(17)*C( 6)
      C( 1) = C( 1)/RI/RI/RI
      C( 2) = G( 2)/RI/SU
      C( 3) = G( 1)/RI/R
      C( 7) = C( 9)+C( 5)
      C(10) = C(12)+G( 2)/R/SU
      C( 7) = C( 1)*C( 7)-C( 2)*F( 3)
      C(10) = C( 1)*C(10)+C( 2)*C(13)-C( 3)*F( 3)
      C(13) = C( 1)*C(15)+C( 2)*C( 6)
      C(16) = C( 1)*C(18)+C( 2)*C(16)+C( 3)*C( 6)
      DO 3 I=1,3
      DO 2 J=1,6
      C(J) = C(J+6)*XI(I)+C(J+12)*XI(I+3)
    2 CONTINUE
      DO 3 J=1,3
      PHI(I  ,J  ) = C(1)*XI(J)+C(2)*XI(J+3)
      PHI(I  ,J+3) = C(2)*XI(J)+C(3)*XI(J+3)
      PHI(I+3,J  ) = C(4)*XI(J)+C(5)*XI(J+3)
      PHI(I+3,J+3) = C(5)*XI(J)+C(6)*XI(J+3)
    3 CONTINUE
      DO 4 I=1,15,7
      PHI(I   ,1) = PHI(I   ,1) + F(1)
      PHI(I+18,1) = PHI(I+18,1) + F(2)
      PHI(I+ 3,1) = PHI(I+ 3,1) + F(3)
      PHI(I+21,1) = PHI(I+21,1) + F(4)
    4 CONTINUE
      RETURN
      END
```

Subroutine: STEPDT

Purpose: Provides the Cartesian state on a conic trajectory at a time T. Subroutine STEPDI provides the initialization of necessary quantities for subsequent calls of STEPDT

Calling Sequence: CALL STEPDT (T,X)

Input and **Output**

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T	d	t	time	Time, referenced to same epoch as t_o in call of STEPDI.
Ø	X	d(6)	X(t)	length, length/ time	Cartesian state at time T.

Common storages used: //12 cells,/STPCØM/

Subroutines required: None.

References:

1. R.H. Battin, "Astronautical Guidance," McGraw-Hill, New York, 1964, pp. 50-52.

2. S. Pines, "Mean Conic State Transition Matrix," Astro Consultants, Inc., Huntington, L.I., N.Y.

Usage:

Subroutine STEPDI,STEPDT,STEPDP use the universal functions for the solution of Kepler's equation, and for computing the state transition matrix. Subroutine STEPDI computes the constants used by STEPDT and STEPDP, and initializes the universal functions. STEPDT solves Kepler's equation using a second-order iteration and outputs position and velocity at the desired time. STEPDP outputs the state transition matrix. The labelled common STPCØM is used for storage of the orbit constants and current values of the universal function.

Once STPCØM is loaded by STEPDI, STEPDT may be called at any sequence of times. Each call of STEPDP, however, must be preceded by a call of STEPDT at the same time.

Theory:

Let R(t),V(t) denote the conic position and velocity of time t, and let

$$R_o = R(t_o)$$

$$V_o = V(t_o)$$

From R_o,V_o, we may compute

$$r_o = |R_o|$$

$$\frac{1}{a} = \frac{2}{r_o} - \frac{V_o^2}{\mu}$$

$$d_o = R_o \cdot V_o / \sqrt{\mu}$$

The regularized incremental eccentric anomaly, β, is defined as

$$\beta = \sqrt{|a|}\,(E - E_o)$$

where E is the eccentric anomaly at time t. Then Kepler's equation may be written

$$\sqrt{\mu}(t - t_o) = a[\beta - (d - d_o)] . \qquad (1)$$

The transcendental functions F_i,G_i are defined as

$$F_i(\alpha) = \sum_{k=0}^{\infty} (-1)^k \frac{\alpha^{2k}}{(i + 2k)!}$$

$$G_i(\beta) = \beta^i F_i(\alpha)$$

$$\alpha^2 = \beta^2/a$$

The G_i may be written as the equivalent functions:

$$\left.\begin{aligned} G_o &= \cos(E - E_o) \\ G_1 &= \sqrt{a}\,\sin(E - E_o) \\ G_2 &= a[1 - \cos(E - E_o)] \\ G_3 &= a^{3/2}[(E - E_o) - \sin(E - E_o)] \end{aligned}\right\} \text{for ellipses}$$

$$\left.\begin{aligned} G_o &= \cosh(E - E_o) \\ G_1 &= \sqrt{|a|}\,\sinh(E - E_o) \\ G_2 &= |a|\,[\cosh(E - E_o) - 1] \\ G_3 &= |a|^{3/2}[\sinh(E - E_o) - (E - E_o)] \end{aligned}\right\} \text{for hyperbolas}$$

Kepler's equation (1) may now be written in the form

$$\begin{aligned} \sqrt{\mu}\,\tau &= F(\beta) \\ &= G_3 + d_o\,G_2 + r_o G_1 \end{aligned} \tag{2}$$

where τ is the increment of time, $t - t_o$. Then

$$\begin{aligned} \sqrt{\mu}\,\frac{d\tau}{d\beta} &= F'(\beta) = G_2 + d_o\,G_1 + r_o\,G_o \\ \sqrt{\mu}\,\frac{d^2\tau}{d\beta^2} &= F''(\beta) = d_o\,G_o + \left(1 - \frac{r_o}{a}\right)G_1 \end{aligned} \tag{3}$$

To determine β for a given time increment, τ, we write

$$F(\beta + \Delta\beta) = F(\beta) + F'(\beta)\Delta\beta + F''(\beta)\frac{\Delta\beta^2}{2!} + \ldots \qquad (4)$$

and solve for $\Delta\beta$ so that

$$F(\beta + \Delta\beta) - \sqrt{\mu}\,\tau = 0$$

Let

$$F(\beta) - \sqrt{\mu}\,\tau = a_o = (G_3 + d_o\, G_2 + r_o\, G_1) - \sqrt{\mu}\,\tau$$

$$F'(\beta) \qquad = a_1 = G_2 + d_o\, G_1 + r_o\, G_o \qquad (5)$$

$$F''(\beta) \qquad = a_2 = d_o\, G_o + \left(1 - \frac{r_o}{a}\right) G_1$$

Then

$$a_o + a_1\, \Delta\beta + a_2\, \frac{\Delta\beta^2}{2} = 0 \qquad (6)$$

The solution of equation (6) is

$$\Delta\beta = \frac{-\,2a_o}{a_1 \pm \sqrt{a_1^2 - 2a_oa_2}} \qquad (7)$$

In equation (7), the positive sign is chosen so that the sign of $\Delta\beta$ is the same as the sign of error/slope.

The square root is approximated by

$$\sqrt{a_1^2 - 2a_oa_2} = a_1 - \frac{a_oa_2}{a_1} \qquad (8)$$

The program uses the equation

$$\Delta\beta = \text{sgn}(a_o) \left| \frac{a_o}{a_1 - \frac{1}{2}\frac{a_o a_2}{a_1}} \right| \qquad (9)$$

where sgn() means the sign of the argument. This insures that the direction is proper for reducing the error in β since $F'(\beta)$ is always positive.

The magnitude of $\Delta\beta$ is limited so that

$$\frac{|\Delta\beta|}{\sqrt{|a|}} \leq 2 \qquad (10)$$

For ellipses, the restriction implied by equation (10) prevents incremental anomaly steps greater than two radians.

The subroutines are designed so that starting values of β, $F'(\beta)$, and $F''(\beta)$ are computed by STEPDI. Thereafter, STEPDT retains the last computed values of these quantities for subsequent calls. This reduces the number of iterations required when STEPDT is called repeatedly at a series of time points on the same conic.

The iteration is terminated when either

$$|\Delta\beta| \leq |\beta| \cdot 3 \times 10^{-16} \qquad (11)$$

or

$$|\Delta\beta| \leq \left(5 \times 10^{-12}\sqrt{\mu}\right) \frac{|\tau|}{r} \qquad (12)$$

Equation (11) requires the $\Delta\beta$ to change the 16^{th} figure of β.

Equation (12) may be understood by recognizing that

$$\frac{d\tau}{d\beta} = \frac{r}{\sqrt{\mu}}$$

hence, the $\Delta\beta$ change must be large enough to be equivalent to a change of 5 places in the 12th significant figure of τ, where τ is the time increment from the last call of STEPDI.

Having found values of β and the G_i series that satisfy the tolerances, the values of R(t) and V(t) are given by

$$R(t) = fR_o + gV_o \qquad (13)$$
$$V(t) = \dot{f}R_o + \dot{g}V_o$$

where

$$f = 1 - \frac{G_2}{r_o}$$

$$g = \tau - \frac{G_3}{\sqrt{\mu}}$$

$$\dot{f} = - \frac{\sqrt{\mu}\, G_1}{rr_o} \qquad (14)$$

$$\dot{g} = 1 - \frac{G_2}{r}$$

STEPDT has a special initialization test for:

(a) Preventing time steps greater than one-half period for ellipses.

(b) Setting the signs of the last values of β, G_1, and G_3 properly to correspond to the sign of the incremental time step.

Constraint (a) is accomplished by setting

$$DT = t - \left[\text{integer part} \left(\frac{t}{\text{period}} \right) \right] \text{period}.$$

If $| DT | > \frac{1}{2}$ period, then DT is assigned the new value DT_n where

$$DT_n = - \text{sgn}(DT)(\text{period} - | DT |).$$

Constraint (b) is accomplished by testing whether β and DT have the same sign. If not, the signs of β, G_1, and G_3 are reversed.

The 6 X 6 transition matrix

$$\Phi = \begin{bmatrix} \frac{\partial R}{\partial R_o} & \frac{\partial R}{\partial V_o} \\ \frac{\partial V}{\partial R_o} & \frac{\partial V}{\partial V_o} \end{bmatrix}$$

is computed by STEPDP, using the equations

$$\frac{\partial R}{\partial R_o} = fI + \delta VE_1^T + \frac{\mu}{r_o^3} (E_3 + \frac{G_2}{\mu} R_o)R_o^T$$

$$\frac{\partial R}{\partial V_o} = gI + \delta VE_2^T + E_3V_o^T \qquad (16)$$

$$\frac{\partial V}{\partial R_o} = \dot{f}I - \frac{\mu R}{r^3}E_1^T + \frac{\mu}{r_o^3} (E_4 + \frac{G_1}{r\sqrt{\mu}} R_o)R_o^T + \delta VE_5^T$$

$$\frac{\partial V}{\partial V_o} = \dot{g}I - \frac{\mu R}{r^3} E_2^T + E_4V_o^T + \delta VE_6^T$$

where

$$\delta V = V - V_o = \dot{f}R_o + (\dot{g} - 1)V_o$$

$$R = fR_o + gV_o$$

$$E_1 = \frac{P_1}{r_o^3} R_o + \frac{r}{\mu} \delta V$$

$$E_2 = \frac{P_1}{\mu} V_o - \frac{G_2}{\mu} R_o$$

$$E_3 = \frac{2G_4 - \beta G_3}{r_o \mu} R_o + \frac{3G_5 - \beta G_4}{\mu \sqrt{\mu}} V_o \qquad (17)$$

$$E_4 = \frac{G_3 - \beta G_2}{r r_o \sqrt{\mu}} R_o + \frac{2G_4 - \beta G_3}{r\mu} V_o$$

$$E_5 = \frac{P_2 - r_o^2 G_o}{r r_o^3} R_o - \frac{G_1}{r\sqrt{\mu}} V_o$$

$$E_6 = \frac{P_2}{r\mu} V_o - \frac{G_1}{r\sqrt{\mu}} R_o$$

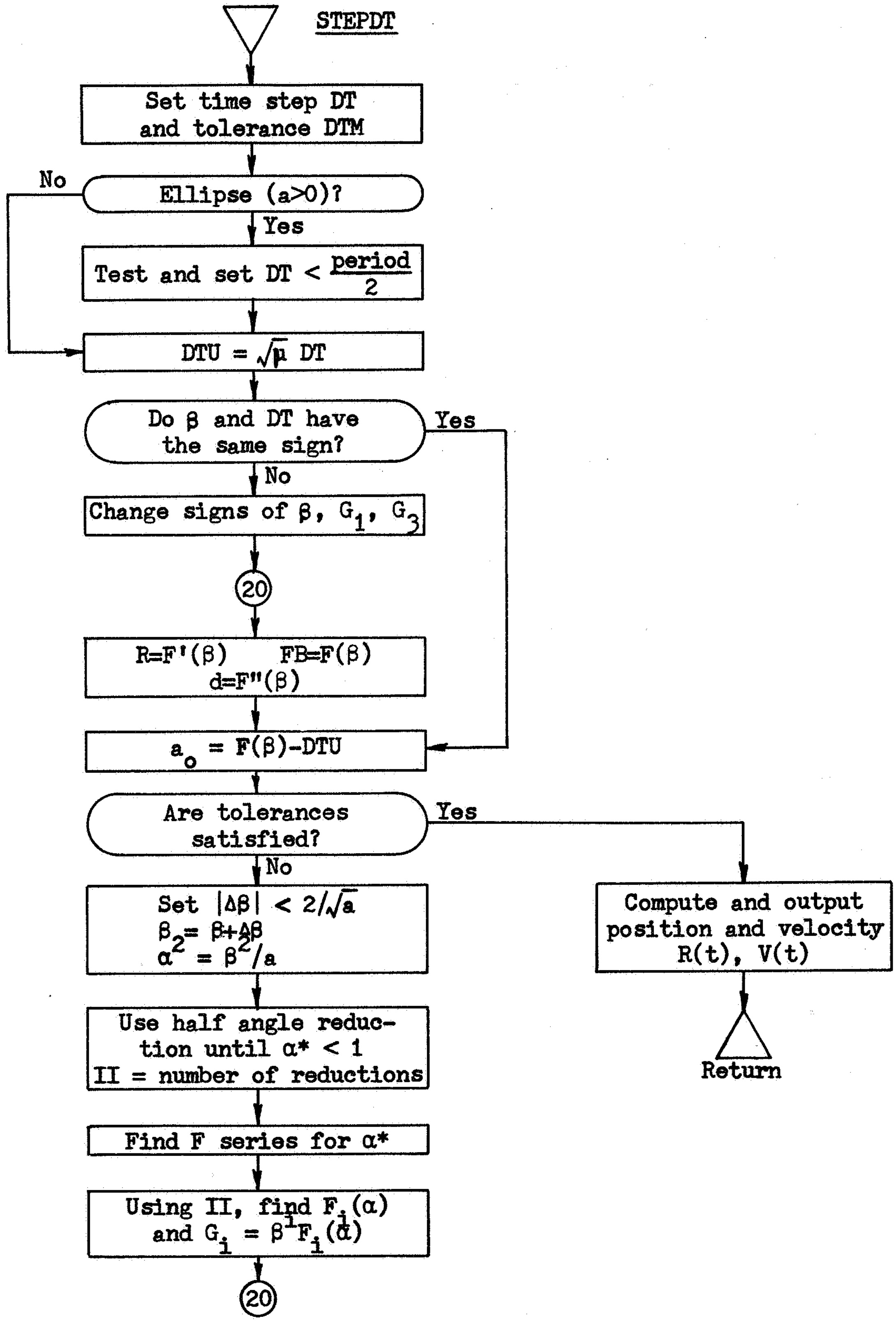
And

$$P_1 = \frac{(3G_5 - \beta G_4) + d_o(2G_4 - \beta G_3) + r_o(G_3 - \beta G_2)}{\sqrt{\mu}}$$

$$(18)$$

$$P_2 = (2G_4 - \beta G_3) + d_o(G_3 - \beta G_2) - r_o \beta G_1$$

The constants and functions of β are obtained from STPCØM, and must have been stored by appropriate calls of STEPDI and STEPDT.


STEPDT
Set time step DT and tolerance DTM
Ellipse (a>0)?
No
Yes
Test and set DT < period/2
DTU = √μ DT
Do β and DT have the same sign?
Yes
No
Change signs of β, G1, G3
20
R=F'(β) FB=F(β) d=F"(β)
a0 = F(β)-DTU
Are tolerances satisfied?
Yes
No
Set |Δβ| < 2/√a
β2= β+Δβ
α2 = β2/a
Use half angle reduction until α* < 1
II = number of reductions
Find F series for α*
Using II, find Fi(α) and Gi = βiFi(α)
20
Compute and output position and velocity R(t), V(t)
Return

```
$IBFTC MC13SJ  XR3,M94,NODD,LIST
      SUBROUTINE STEPDT (T,X)
C     CONIC POSITION, VELOCITY AT TIME T
C
C     T = TIME, SECONDS
C     X = CARTESIAN POSITION,VELOCITY
C
      DOUBLE PRECISION  T      ,X(6) ,DMOD
      COMMON      /STPCOM/TI    ,XI(6),SU    ,RI    ,AI    ,SA    ,DI    ,SR
     1                   ,PR    ,HP    ,OM    ,EP    ,R     ,G(4) ,B     ,FB
     2                   ,D     ,F(4)
      DOUBLE PRECISION  TI     ,XI    ,SU    ,RI    ,AI    ,SA    ,DI    ,SR
     1                   ,PR    ,HP    ,OM    ,EP    ,R     ,G     ,B     ,FB
     2                   ,D     ,F
      COMMON            C(6)
      DOUBLE PRECISION  C
      DOUBLE PRECISION  C4(10)       ,C3(10)
      DATA  C4/420.,342.,272.,210.,156.,110.,72.,42.,20.,6./
      DATA  C3/380.,306.,240.,182.,132.,90.,56.,30.,12.,2./
C
    1 C(1) = T-TI
      C(2) = DABS(C(1))*EP
      IF (AI.LE.0.) GO TO 2
      C(1) = DMOD(C(1),PR)
      C(3) = DABS(C(1))
      IF (C(3).LE.HP) GO TO 2
      C(1) =-DSIGN(PR-C(3),C(1))
    2 C(4) = C(1)*SU
      IF (B*C(1).GT.0.) GO TO 21
    3 CONTINUE
      G(2) =-G(2)
      G(4) =-G(4)
      B    =-B
      GO TO 20
C
   10 IF (C(3)*SA.GT.2.D0) C(5) = DSIGN(2.D0/SA,C(5))
      B = B+C(5)
      C(6) = AI*B*B
      II = 0
   11 IF (DABS(C(6)).LE.1.D0) GO TO 12
      II = II+1
      C(6) = C(6)/4.D0
      GO TO 11
   12 CONTINUE
      G(3) = 1.D0/C3(1)
      G(4) = 1.D0/C4(1)
      DO 13 I=2,10
      G(3) = (1.D0-C(6)*G(3))/C3(I)
   13 G(4) = (1.D0-C(6)*G(4))/C4(I)
   14 G(2) = 1.D0-C(6)*G(4)
      G(1) = 1.D0-C(6)*G(3)
      IF (II.LE.0) GO TO 15
      C(6) = C(6)*4.D0
      II = II-1
      G(4) = (G(3)+G(1)*G(4))/4.D0
      G(3) = G(2)*G(2)/2.D0
      GO TO 14
   15 CONTINUE
      G(2) = B*G(2)
      C(5) = B*B
      G(3) = C(5)*G(3)
      G(4) = B*C(5)*G(4)
C
   20 R  = G(3)+RI*G(1)+DI*G(2)
      FB = G(4)+RI*G(2)+DI*G(3)
      D  = DI*G(1)+OM*G(2)
   21 CONTINUE
      C(6) = C(4)-FB
      C(3) = C(6)*D/R/2.D0+R
      C(5) = DSIGN(C(6)/C(3),C(6))
      C(3) = DABS(C(5))
      IF (C(3).LE.DABS(B)*3.D-16) GO TO 30
      IF (C(3).GT.C(2)/R) GO TO 10
C
   30 F(1) = 1.D0-G(3)/RI
```

```
      F(2) = C(1)-G(4)/SU                                        STPT0075
      F(3) =-SR*G(2)/R                                           STPT0076
      F(4) = 1.D0-G(3)/R                                         STPT0077
      DO 31 I=1,3                                                STPT0078
      X(I)   = F(1)*XI(I)+F(2)*XI(I+3)                           STPT0079
   31 X(I+3) = F(3)*XI(I)+F(4)*XI(I+3)                           STPT0080
  999 RETURN                                                     STPT0081
      END
```

Subroutine: STEPI

Purpose: To compute an array, CB, of constants of the conic determined by the position and velocity R, V, for use by subroutine STEPT. (q.v.)

Calling Sequence: CALL STEPI (R,V,U,CB)

Input and Output

I/Ø	Symbolic Name or Location	Probram Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	R	(3)	R_o	L	Cartesian position at the initial time, t_o.
I	V	(3)	V_o	L/T	Cartesian velocity at t_o.
I	U		u	L^3/T^2	Gravitational constant.
Ø	CB	(6)			
	CB(1)		r	L	Magnitude of R_o.
	CB(2)		a	L	Absolute value of the semi-major axis of the conic.
	CB(3)		n	1/T	Mean motion $(=\sqrt{u/a^3})$
	CB(4)				r/a
	CB(5)				$R_o \cdot V_o/\sqrt{ua}$
	CB(6)				=1. if the conic is an ellipse =2. otherwise.

Common storages used: None

Subroutines required: FNØRM, DØT

```
$IBFTC MC13SI  NOREF,M94,NODD,XR3
CMC13SI  STEPI PERFORMS INITIALIZATION FUNCTION FOR STEPT
      SUBROUTINE STEPI(R,V,U,CB)
C  STEPI COMPUTES CB, A STORAGE VECTOR WHOSE ELEMENTS ARE
C      (1) RADIUS MAGNITUDE  R
C      (2) ABSOLUTE VALUE OF SEMI-MAJOR AXIS  ABA
C      (3) MEAN MOTION
C      (4) TIME COEFFICIENT  R/ABA
C      (5) TIME COEFFICIENT  R DOT V /SQRT(U*A)
C      (6) SWITCH 1 ELLIPSE, 2 HYPERBOLA
C      SUBROUTINES REQUIRED   DOT,FNORM
      DIMENSION   R(3),   V(3),   CB(6)
      CB(1)=FNORM(R)
      CB(2)=CB(1)/(2.-CB(1)*DOT(V,V)/U)
      IF(CB(2)) 1,1,2
1     CB(2)=-CB(2)
      CB(6)=2.
      GO TO 3
2     CB(6)=1.
3     CONTINUE
      CB(4)=CB(1)/CB(2)
      CB(3)=SQRT(U/CB(2))/CB(2)
      CB(5)=DOT(R,V)/SQRT(U*CB(2))
      RETURN
      END
```

Subroutine: STEPT

Purpose: To obtain the position and velocity of a body which is separated by a time increment from a specified state described by parameters in an array CB computed by STEPI (q.v.).

Calling Sequence: CALL STEPT (T,R,V,RR,VV,CB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T		δt	T	Incremental time.
I	R	(3)	R_o	L	Cartesian position at time t_o.
I	V	(3)	V_o	L/T	Cartesian velocity at time t_o.
Ø	RR	(3)	R	L	Cartesian position at $t_o+\delta t$.
Ø	VV	(3)	V	L/T	Cartesian velocity at $t_o+\delta t$.
I	CB	(6)			Array of constants computed by subroutine STEPI.

Common storages used: None

Subroutines required: GØTØR

Discussion

The inverse-square central force law is characterized by the equations

$$\dot{R} = \frac{dR}{dt} = V$$
$$\dot{V} = \frac{dV}{dt} = -\frac{u}{r^3} R \qquad (1)$$

where r is defined to be $\sqrt{R \cdot R}$. The angular momentum vector, H, defined by the vector cross-product

$$H = R \times V \qquad (2)$$

is a constant with respect to time, since

$$\dot{H} = \dot{R} \times V + R \times \dot{V} = V \times V + R \times (-\frac{u}{r^3} R) = 0$$

Then

$$H = H_o = R_o \times V_o$$

Because the magnitude, h_o, of H_o is generally non-zero, the plane equations

$$R \cdot H = R \cdot H_o = 0$$
$$V \cdot H = V \cdot H_o = 0 \qquad (3)$$

which follow from the definition of H, lead to the conclusion that all motion occurs in the plane normal to H_o. That is, if R_o and V_o are non-zero and non-colinear vectors, any other vectors R and V must lie in the p plane formed by R_o and V_o. Algebraically,

$$R = f\,R_o + g\,V_o$$
$$V = \dot{f}\,R_o + \dot{g}\,V_o \qquad (4)$$

The second of equations (4) follows from (1) and from the fact that R_o and V_o are not functions of the time increment, t. The scalars f, $\dot{f}$, g, and $\dot{g}$, are functions of R_o, V_o and δt. The determination of these scalars is described below.

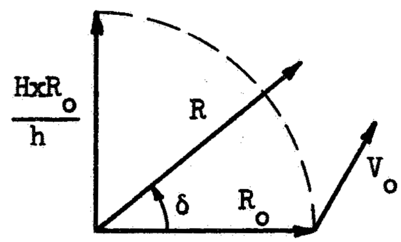


Figure 1

The vector $(H \times R_o)/h$ is perpendicular to R_o and has the same magnitude, r_o. From Figure 1, we note that R may be written as the combination

$$R = \frac{r}{r_o}\left[R_o \cos\delta + \frac{H \times R_o}{h}\sin\delta\right] \tag{5}$$

Equation (5) may be rewritten by expanding the triple cross product

$$H \times R_o = (R_o \times V_o) \times R_o = V_o(R_o \cdot R_o) - R_o(R_o \cdot V_o)$$

$$R = \frac{r}{r_o}\left[R_o\left(\cos\delta - \frac{R_o \cdot V_o}{h}\sin\delta\right) + V_o\,\frac{r_o^{\,2}}{h}\sin\delta\right] \tag{6}$$

Comparing equation (6) with equation (4),

$$f = \frac{r}{r_o}\left(\cos\delta - \frac{R_o \cdot V_o}{h}\sin\delta\right)$$
$$g = \frac{r r_o}{h}\sin\delta \tag{7}$$

At this point, the reader should note that while r_o, $R_o \cdot V_o$, and h are computable directly from R_o and V_o, the quantities r and δ have not been specifically defined in terms of R_o, V_o, and δt. In order to do this, it is necessary to examine the conic solution of the inverse-square law equations of motion.

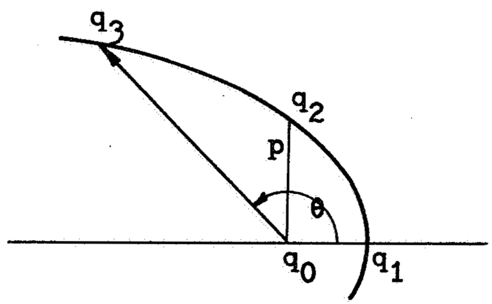


Figure 2

Figure 2 shows a conic section with focus at q_0 and pericenter at q_1. The point q_3 on the conic is specified by the polar coordinates originating at q_0, r, the radius, and θ, the true anomaly. The equation relating r, θ on a conic section is

$$r = \frac{p}{1 + e \cos \theta} \qquad (8)$$

where p is the length $\overline{q_0 q_2}$, or the semi-latus rectum, and the eccentricity, e, which may be calculated from

$$e = \frac{\overline{q_0 q_2}}{q_0 q_1} - 1$$

is a parameter specifying the shape of the conic. Conic sections are characterized by eccentricity as follows:

circular	if $e = 0$
elliptic	if $0 < e < 1$
parabolic	if $e = 1$
hyperbolic	if $e > 1$

The two cases of interest here are the elliptic and hyperpolic cases. Figure 3 below shows an ellipse. The semi-major axis, a, is the distance

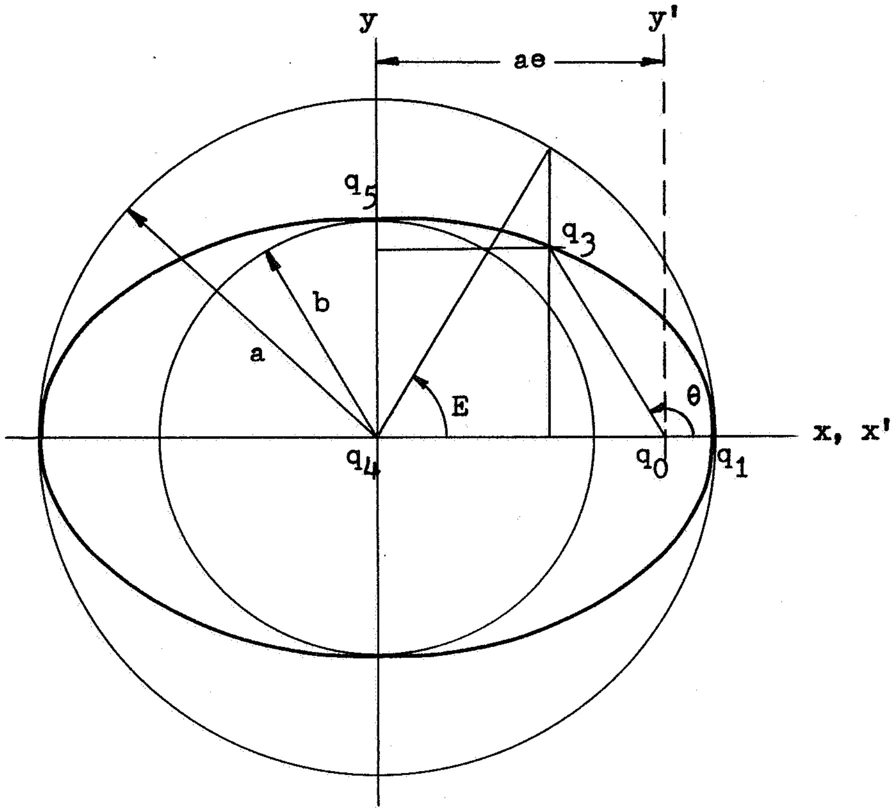


Figure 3

from the point of symmetry, q_4, to the pericenter, q_1. The semi-minor axis, b, is the length $\overline{q_4 q_5}$. The eccentricity is equal to the ratio

$$e = \sqrt{a^2 - b^2} \; / \; a$$

The eccentric anomaly, E, is the argument of the projection of the point, q_3, from the line $q_4 q_1$ onto a circle of radius a concentric with the ellipse. That is, the cartesian coordinates of q_3 relative to q_4 are

$$\begin{aligned} x &= a \cos E \\ y &= b \sin E \end{aligned} \qquad (9)$$

The coordinates of q_3 relative to q_0 are

$$\begin{aligned} x' &= r \cos \theta = a(\cos E - e) \\ y' &= y = r \sin \theta = b \sin E = a \sqrt{1-e^2} \sin E \\ r &= a(1 - e \cos E) \end{aligned} \qquad (10)$$

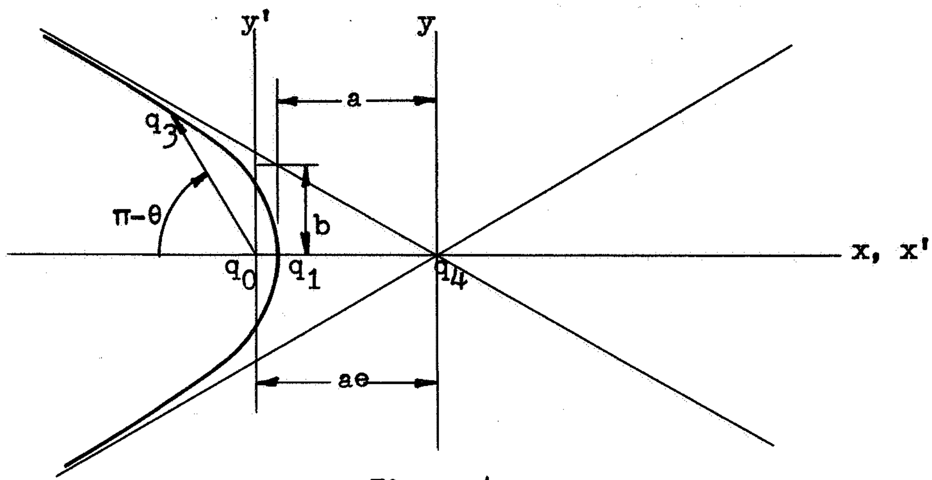


Figure 4

Figure 4 shows a hyperbola. The asymptotes are the lines $\frac{x}{a} - \frac{y}{b} = 0$ and $\frac{x}{a} + \frac{y}{b} = 0$. The semi-major axis, a, is again the distance from the point of symmetry, q_4, to the pericenter, q_1. The semi-minor axis, b, is the intercept of the asymptotes on the lines $x = \pm a$. The eccentricity is

$$e = \sqrt{a^2 + b^2} \, / \, a$$

The eccentric anomaly, E (sometimes denoted F), is so defined that the coordinates of q_3 relative to q_4 are

$$\begin{aligned} x &= -a \cosh E \\ y &= b \sinh E \end{aligned} \qquad (11)$$

Then the coordinates of q_3 relative to q_0 are

$$\begin{aligned} x' &= r \cos \theta = a(e - \cosh E) \\ y' &= r \sin \theta = a\sqrt{e^2 - 1} \sinh E \\ r &= a(e \cosh E - 1) \end{aligned} \qquad (12)$$

Solution of the differential equations (1) yields p, a, and e as constants of the motion. It is easily verified that

$$h = \sqrt{r_o^2 v_o^2 - (R_o \cdot V_o)^2}$$

$$p = h^2/u$$

$$a = 1/\left(\frac{2}{r_o} - \frac{v_o^2}{u} \right) \qquad (13)$$

$$e^2 = 1 - p/a$$

$$n = \sqrt{u/|a^3|}$$

where we have adopted the convention that a is negative for the hyperbolic case. Some useful identities which follow from equations (10) and (12) are listed below. In equations (14) and those which follow, we use the ambiguous sign with the convention that the upper sign applies to the hyperbolic case and the lower to the elliptic case. The symbols sE and cE mean sin E and cos E for the elliptic case, but sinh E and cosh E for the hyperbolic case.

$$\frac{R \cdot V}{r} = \frac{d}{dt}(R \cdot R)^{\frac{1}{2}} = \dot{r} = \frac{ep\dot{\theta}\,\sin\theta}{(1+e\cos\theta)^2} = \frac{ue}{h}\sin\theta$$

$$\cos\theta = \frac{a}{r}(cE - e)$$

$$\sin\theta = \frac{|a|}{r}\sqrt{1 - e^2}\;\; sE = \sqrt{\frac{|a|}{u}}\,\frac{h}{r}\, sE \qquad (14)$$

$$e\;\, sE = \frac{R \cdot V}{\sqrt{u|a|}}$$

$$e\;\, cE = 1 - r/a$$

Then, if we set

$$\alpha = E - E_o$$

it follows that

$$sE = s\alpha \; cE_o + c\alpha \; sE_o$$
$$cE = c\alpha \; cE_o \pm s\alpha \; sE_o \tag{15}$$

The angle δ of equation (5) is the incremental true anomaly on the conic section containing R_o and R. Then

$$\sin\delta = \sin(\theta - \theta_o) = s\theta \; c\theta_o - c\theta \; s\theta_o$$
$$\cos\delta = \cos(\theta - \theta_o) = c\theta \; c\theta_o + s\theta \; s\theta_o \tag{16}$$

The incremental true anomaly may now be related to R_o and V_o through substitution of equations (14) and (15) into (16).

$$\sin\delta = \mp \frac{h}{nrr_o}\left(sEcE_o - cEsE_o - esE + esE_o\right)$$
$$= \mp \frac{h}{nrr_o}\left(s\alpha - es\alpha cE_o - ec\alpha sE_o + esE_o\right)$$
$$= \mp \frac{h}{nrr_o}\left[\frac{r_o}{a} s\alpha - \frac{R_o \cdot V_o}{\sqrt{u|a|}}(c\alpha - 1)\right]$$

$$\cos\delta = \frac{a^2}{rr_o}\left(cEcE_o - ecE - ecE_o + e^2 \pm (e^2-1)sEsE_o\right)$$
$$= \frac{a^2}{rr_o}\left(c\alpha(1-ecE_o \pm e^2 s^2 E_o) \mp s\alpha esE_o(1-ecE_o) + e^2 - ecE_o\right)$$
$$= \frac{a^2}{rr_o}\left[c\alpha\left[\frac{r_o}{a} \pm \frac{(R_o \cdot V_o)^2}{u|a|}\right] \mp s\alpha \frac{r_o}{a}\frac{R_o \cdot V_o}{\sqrt{u|a|}} - \frac{p}{a} + \frac{r_o}{a}\right]$$
$$= \frac{a^2}{rr_o}\left[\left[\frac{r_o}{a} - \frac{(R_o \cdot V_o)^2}{ua}\right](c\alpha - 1) \mp \frac{R_o \cdot V_o}{\sqrt{u|a|}}\frac{r_o}{a} s\alpha + \frac{r_o^2}{a^2}\right]$$

The coefficients, f and g of equations (7), are finally found to be:

$$f = \frac{r}{r_o}\left(\cos\delta - \frac{R_o \cdot V_o}{h} \sin\delta \right)$$

$$= 1 + \frac{a}{r_o}(c\alpha - 1) \qquad (17)$$

$$g = \frac{rr_o}{h} \sin\delta$$

$$= \mp \frac{1}{n}\left[\frac{r_o}{a} s\alpha - \frac{R_o \cdot V_o}{\sqrt{u|a|}} (c\alpha - 1) \right] \qquad (18)$$

The velocity coefficients, $\dot{f}$ and $\dot{g}$, are found by noting that

$$\dot{\alpha} = \dot{E} = \sqrt{\frac{u}{|a|}} \frac{1}{r}$$

and therefore

$$\dot{f} = -\frac{\sqrt{u|a|}}{rr_o} s\alpha$$

$$\dot{g} = \frac{a}{r}\left[\frac{r_o}{a} c\alpha \pm \frac{R_o \cdot V_o}{\sqrt{u|a|}} s\alpha \right] \qquad (19)$$

where

$$\frac{r}{a} = 1 - ecE$$

$$= (1 - c\alpha) + \frac{r_o}{a} c\alpha \mp \frac{R_o \cdot V_o}{\sqrt{u|a|}} s\alpha \qquad (20)$$

Kepler's equation relates the incremental time, δt, to the incremental anomaly, α.

$$n\delta t = \mp (E - esE) \pm (E_o - esE_o)$$

$$= \mp (\alpha - es\alpha cE_o - ec\alpha sE_o + esE_o)$$

$$= \mp (\alpha - s\alpha) \mp \frac{r_o}{a} s\alpha \pm \frac{R_o \cdot V_o}{\sqrt{u|a|}}(c\alpha - 1) \qquad (21)$$

Subroutine GØTØR is called to obtain α as a function of δt by an iterative process. GØTØR returns the vector

$$F(1) = \mp (\alpha - s\alpha)$$
$$F(2) = \mp (1 - c\alpha)$$
$$F(3) = s\alpha$$
$$F(4) = c\alpha$$

The coefficients are then calculated from

$$f = 1 - \frac{a}{r_o} F(2)$$

$$g = \delta t - \frac{1}{n} F(1)$$

$$\frac{r}{a} = F(2) + \frac{r_o}{a} F(4) + \frac{R_o \cdot V_o}{\sqrt{u\,|a|}} F(3)$$

$$\dot{f} = - an\, F(3)/r_o(\frac{r}{a})$$

$$\dot{g} = 1 \pm \frac{a}{r} F(2)$$

The output position and velocity are then computed from equations (4).

```
$IBFTC MC13SG  NOREF,M94,NODD,XR3
CMC13SG  STEPT (STEPD FOR TIME ONLY,NO INIT, CB(6) COMPUTED IN STEPI)
      SUBROUTINE STEPT(T,R,V,RR,VV,CB)
C     SUBROUTINES REQUIRED   GOTOR
C               CB IS A STORAGE VECTOR WHOSE ELEMENTS ARE
C     (1) RADIUS MAGNITUDE  R
C     (2) ABSOLUTE VALUE OF SEMI-MAJOR AXIS  ABA
C     (3) MEAN MOTION
C     (4) TIME COEFFICIENT  R/ABA
C     (5) TIME COEFFICIENT  R DOT V /SQRT(U*A)
C     (6) SWITCH 1 ELLIPSE, 2 HYPERBOLA
      DIMENSION R(3),   V(3),   RR(3),   VV(3),   F(4),   CB(6)
      EMDT=CB(3)*T
      K=CB(6)
      GO TO(1110,1111),K
1110  E1=EMDT
      GO TO 1112
1111  E1=0.0
1112  CONTINUE
      CALL GOTOR(K,EMDT,CB(4),F,E1)
      EF=-F(2)/CB(4)+1.
      GE=-F(1)/CB(3)+T
      ROA=F(2)+CB(4)*F(4)+CB(5)*F(3)
      EFD=-CB(2)*CB(3)*F(3)/(CB(1)*ROA)
      GED=-F(2)/ROA+1.
      DO 13 I=1,3
      RR(I)=EF*R(I)+GE*V(I)
      VV(I)=EFD*R(I)+GED*V(I)
13    CONTINUE
      RETURN
      END
```

Subroutine: STTBGN

Purpose: To initialize the data start link. Loads constants into /DATCØM/ and /MLECØM/. Optionally outputs data type identification. Reads overlay data.

Calling Sequence: CALL STTBGN

Common storages used: /DATCØM/,/DCPCØM/,/EDTCØM/,/ESTCØM/,/MLECØM/

Subroutines required: DMVTRN,ØVRLYD

Output Control:

NSTTG = 0		No output
	≥ 1	Output receiving station and central body names
	≥ 2	Output measurement system identification
	≥ 3	Output measurement names, units, biases, variances, and doppler parameters.

```
$IBFTC MC13G5  XR3,M94,NODD,LIST
      SUBROUTINE STTBGN
C     INITIALIZES DATA START LINK
C
      COMMON             /DCPCOM/CDCP(900)
      DOUBLE PRECISION    DCDCP(450) ,SN(13,20) ,STIMR(2)
      DIMENSION           CBODY(8,11),NPROC(22)
      EQUIVALENCE               (CDCP(758),INDSTA) ,(CDCP(732),NPROVR)
     1                         ,(CDCP(757),ITRSTA) ,(CDCP(682),NSTTG )
     2     ,(CDCP( 17),CBODY ) ,(CDCP(798),KAUTO ) ,(CDCP(143),SN     )
     3     ,(CDCP(  1),DCDCP ) ,(CDCP(787),NPOINT) ,(CDCP(781),STIMR )
     4     ,(CDCP(777),ICB    ) ,(CDCP(731),NPROC )
C
      COMMON             /ESTCOM/CEST(804)
      DOUBLE PRECISION    EFEDAN(14),SE(14,20)
      EQUIVALENCE        (CEST( 61),EFEDAN) ,(CEST(245),SE     )
     1                  ,(CEST(   2),NBY    )
C
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)
      DOUBLE PRECISION    DATIND(20) ,DRANGE ,ONTIME
      EQUIVALENCE        (INDDAT( 1),DATIND) ,(INDDAT( 3),NRSTA )
     1                  ,(INDDAT(27),DRANGE) ,(INDDAT( 4),NTSTA )
     2                  ,(INDDAT(12),KTAU   ) ,(INDDAT(13),ONTIME)
     3                  ,(INDDAT( 8),MTYPE  )
C
      COMMON             /DATCOM/BIAS (2) ,OBS (64) ,FTR      ,OMEGA
     1                          ,SPDLT    ,STA (10) ,TAU      ,TB2CO(18)
     2                          ,TB2CT(18),TT2BO(9) ,TT2BT(9) ,XV   (12)
     3                          ,MLT      ,MODE     ,MSTA     ,MTIM
     4                          ,NALIGN   ,NANG     ,NFRAC
      DOUBLE PRECISION    BIAS   ,OBS   ,FTR   ,OMEGA ,SPDLT ,STA
     1                   ,TAU    ,TB2CO ,TB2CT ,TT2BO ,TT2BT ,XV
C
      COMMON             /MLECOM/CMLE(1070)
      DOUBLE PRECISION    AMU   ,OBIAS(2) ,Q (4) ,STBIAS(2)
     1                   ,CTAU            ,QI(4) ,YMOD  (4)
      DIMENSION           ILBS(12,4) ,LBALG(2) ,LBTAU(2)
     1                   ,ILBM(12,4) ,LBM (12) ,LBTYP(5)
     2                   ,INROT(3,4) ,LBS (12) ,NROT (4)
      LOGICAL             ILMEAS(4,4),LMEAS(4)
      EQUIVALENCE                                  (CMLE(432),NPTSTT)
     1                        ,(CMLE(405),LAUTO ) ,(CMLE(423),NROT   )
     2     ,(CMLE(363),AMU    ) ,(CMLE(555),LBALG ) ,(CMLE( 31),OBIAS  )
     3     ,(CMLE(365),CTAU   ) ,(CMLE(557),LBDPL ) ,(CMLE( 47),Q      )
     4     ,(CMLE(435),ILBM   ) ,(CMLE(531),LBM   ) ,(CMLE( 55),QI     )
     5     ,(CMLE(483),ILBS   ) ,(CMLE(543),LBS   ) ,(CMLE( 27),STBIAS)
     6     ,(CMLE(381),ILMEAS) ,(CMLE(558),LBTAU ) ,(CMLE(  2),TFINAL)
     7     ,(CMLE(411),INROT  ) ,(CMLE(560),LBTYP ) ,(CMLE(  1),TSTART)
     8     ,(CMLE(433),IQUAL  ) ,(CMLE(397),LMEAS ) ,(CMLE(207),YMOD   )
C
  601 FORMAT(1H0,A6,23H IS USED FOR DATA START,60X,A6,16H IS CENTRAL BOD
     1Y)
  602 FORMAT(14HODATA TYPE IS ,A6)
  603 FORMAT(/16X,4HTYPE,19X,5HUNITS,11X,4HBIAS,26X,8HVARIANCE/1H )
  604 FORMAT(12X,I1,3H - ,2A6,11X,A6,2(E22.8,8X))
  605 FORMAT(12X,18HX-AXIS IS ALIGNED ,A3)
  606 FORMAT(12X,18HRANGE AMBIGUITY IS,D31.16)
  607 FORMAT(/9X,7HDOPPLER/12X,A6,9H FIXED AT,D30.16/12X,25HTRANS FREQ
     1  = ,D30.16/12X,15HREFRCT COEF    =,2D30.16/12X,15HBIAS FREQ     =,
     2D30.16/12X,15HRETRNS RATIO   =,D30.16)
  608 FORMAT(/9X,17HTRANS STATION IS ,A6)
C
C     READ OVERLAY DATA AND SET CONSTANTS
    1 IF (NPROVR.NE.0) CALL OVRLYD(CDCP)
      NPROVR = 0
C     SET UP DATCOM
      MLT      = 2
      MODE     = INDDAT(10)
      MSTA     = 2
      MTIM     = 2
      NALIGN   = INDDAT( 9)
      NFRAC    = 1
      NANG     = 1
      FTR      = DATIND(11)
      OMEGA    = CBODY(8,3)
```

```
      SPDLT    = DCDCP(8)+SE(8,INDSTA)                                  STBG0075
      TAU      = DATIND(10)                                             STBG0076
      BIAS(1)  = DATIND(15)+SE(10,INDSTA)/TAU                           STBG0077
      BIAS(2)  = DATIND(16)                                             STBG0078
      DO 2 I=1,9                                                        STBG0079
      TT2BO(I) = SN(I+4,INDSTA)                                         STBG0080
    2 TT2BT(I) = SN(I+4,ITRSTA)                                         STBG0081
      CALL DMVTRN (TT2BO,SE(1,INDSTA),STA(1),1,1)                       STBG0082
      CALL DMVTRN (TT2BT,SE(1,ITRSTA),STA(6),1,1)                       STBG0083
      DO 3 I=1,3                                                        STBG0084
      STA(I)    = SN(I+1,INDSTA)+STA(I)                                 STBG0085
    3 STA(I+5) = SN(I+1,ITRSTA)+STA(I+5)                                STBG0086
      STA( 4) = DATIND(12)                                              STBG0087
      STA( 5) = DATIND(13)                                              STBG0088
      STA( 9) = STA(4)                                                  STBG0089
      STA(10) = STA(5)                                                  STBG0090
      JJ = MTYPE                                                        STBG0091
      GO TO (4,7,6,5) ,MTYPE                                            STBG0092
    4 MODE = 0                                                          STBG0093
    5 NALIGN = 0                                                        STBG0094
      GO TO 7                                                           STBG0095
    6 IF (NALIGN.EQ.1) JJ = 2                                           STBG0096
    7 CONTINUE                                                          STBG0097
C     SET UP MLECOM                                                     STBG0098
   10 TSTART = 0.                                                       STBG0099
      TFINAL = 0.                                                       STBG0100
      IF (STIMR(1).GT.0.) TSTART = STIMR(1)-ONTIME                      STBG0101
      IF (STIMR(2).GT.0.) TFINAL = STIMR(2)-ONTIME                      STBG0102
      ICB = MOD(NBY,100)                                                STBG0103
      CTAU  = TAU                                                       STBG0104
      AMU   = EFEDAN(ICB)                                               STBG0105
      NPTSTT = NPOINT                                                   STBG0106
      DO 11 I=1,4                                                       STBG0107
      Q(4)   = SE(I+10,INDSTA)                                          STBG0108
      Q(I) = Q(4)*Q(4)                                                  STBG0109
      QI(I) = 1.D0/Q(I)                                                 STBG0110
      OBIAS(I) = SE(I+6,INDSTA)                                         STBG0111
      LMEAS(I) = ILMEAS(I,MTYPE)                                        STBG0112
      IF (NPROC(I+13).NE.0) LMEAS(I) = .FALSE.                          STBG0113
   11 CONTINUE                                                          STBG0114
      OBIAS(4) = 0.D0                                                   STBG0115
      STBIAS(1) = SE(4,INDSTA)                                          STBG0116
      STBIAS(2) = SE(5,INDSTA)                                          STBG0117
      YMOD(3) = DRANGE                                                  STBG0118
      IQUAL = 0                                                         STBG0119
      IF (.NOT.LMEAS(3)) IQUAL = 2                                      STBG0120
      IF (.NOT.LMEAS(4)) IQUAL = IQUAL+4                                STBG0121
      DO 12 I=1,3                                                       STBG0122
   12 NROT(I) = INROT(I,JJ)                                             STBG0123
      NROT(4) = JJ                                                      STBG0124
      LAUTO    = KAUTO                                                  STBG0125
      DO 13 I=1,12                                                      STBG0126
      LBS(I) = ILBS(I,MTYPE)                                            STBG0127
   13 LBM(I) = ILBM(I,MTYPE)                                            STBG0128
      IF (MODE.NE.3) GO TO 14                                           STBG0129
      LBM(8) = LBDPL                                                    STBG0130
      LBS(8) = LBDPL                                                    STBG0131
   14 CONTINUE                                                          STBG0132
C                                                                       STBG0133
C     OUTPUT DATA TYPE IDENTIFICATION                                   STBG0134
   20 IF (NSTTG.EQ.0) GO TO 999                                         STBG0135
      WRITE (6,601) NRSTA,CBODY(1,ICB)                                  STBG0136
      IF (NSTTG.EQ.1) GO TO 999                                         STBG0137
      WRITE (6,602) LBTYP(MTYPE)                                        STBG0138
      IF (NSTTG.EQ.2) GO TO 999                                         STBG0139
      WRITE (6,603)                                                     STBG0140
      DO 21 I=1,4                                                       STBG0141
      IF (.NOT.LMEAS(I)) GO TO 21                                       STBG0142
      WRITE (6,604) I,LBM(I),LBM(I+4),LBM(I+8),OBIAS(I),Q(I)            STBG0143
   21 CONTINUE                                                          STBG0144
      IF (NALIGN.NE.0) WRITE (6,605) LBALG(NALIGN)                      STBG0145
      IF (LMEAS(3))    WRITE (6,606) DRANGE                             STBG0146
      IF (MODE.EQ.0)   GO TO 999                                        STBG0147
      WRITE (6,607)    LBTAU(KTAU+1),TAU,FTR,STA(4),STA(5),BIAS         STBG0148
      IF (MODE.EQ.3)   WRITE (6,608) NTSTA                              STBG0149
C
  999 RETURN
      END
```

Subroutine: STTDAT

Purpose: To choose the data which is to be used for a data start. This data is selected from the points on the edited data tape according to a number of input options, and placed in the array DATA of MLEĆOM.

Calling Sequence: CALL STTDAT

Common storages used: //13 cells,/DCPCØM/,/EDTCØM/,/MLECØM/

Subroutines required: DATØUP

Discussion:

The following quantities are output from STTDAT into MLEØOM

DATA(5,100)	DATA(1,I) is a time T, in seconds from ØNTIME. DATA(M + 1, I) is measurement M at time T.
NPTS	Number of points loaded into DATA. Points are loaded from BUFDAT and all points with a "bad quality" indicator or with all negative measurements are skipped. In addition, if TFINAL has been set, only points with T < TFINAL are used.
LAUTØ	LAUTØ determines whether or not an automatic new data start is to be allowed if the fit using the present data fails. LAUTØ is set .FALSE. (no new start) if all available data is used, or if TFINAL > 0. (On entry to STTDAT, LAUTØ equals KAUTØ, an input switch.)

STTDAT requires that, on entry, the edited data tape must be positioned at the end of the record pair currently given in /EDTCØM/. The tape and /EDTCØM/ are reset to this condition before returning from STTDAT.

```
$IBFTC MC13D5  XR3,M94,NODD,LIST
      SUBROUTINE STTDAT
C     LOADS WORKING DATA ARRAY FROM DATA TAPE BUFFER
C
      COMMON           /DCPCOM/CDCP(900)
      EQUIVALFNCF      (CDCP(795),MAXPTS) ,(CDCP(680),NSTTD )
     1                 ,(CDCP(796),MINPTS)
C
      COMMON           /EDTCOM/INDDAT(40),BUFDAT(85,6)
      DOUBLE PRECISION   ONTIME
      EQUIVALENCE                (INDDAT( 8),MTYPE ) ,(INDDAT( 6),NPTS   )
     1     ,(INDDAT( 7),KONT   ) ,(INDDAT( 2),NEOT   ) ,(INDDAT(13),ONTIME)
C
      COMMON           /MLECOM/CMLE(1070)
      DIMENSION          DATA(5,100) ,LBM(12)
      LOGICAL            LAUTO
      EQUIVALENCE      (CMLE(570),DATA   ) ,(CMLE(433),QUAL   )
     1                 ,(CMLE(405),LAUTO ) ,(CMLE(   2),TFINAL)
     2                 ,(CMLE(531),LBM    ) ,(CMLE(   1),TSTART)
     3                 ,(CMLE(432),NPTSTT)
C
      COMMON             SAVE(10),TP(3)
      DATA  IBLK /6H
     1    /,NO   /6HNO     /
C
  601 FORMAT(19H1DATA SPECIFICATION,10X,A3,78HAUTOMATIC NEW DATA START A
     1LLOWED IF THIS DATA DOES NOT YIELD STARTING ESTIMATE)
  602 FORMAT(39H0OBSERVATION TIMES REFERRED TO ONTIME =,D24.16,4H SEC)
  603 FORMAT(/12X,13HSTARTING TIME,6X,1H=,E20.8,4H SEC)
  604 FORMAT(12X,10HFINAL TIME,9X,1H=,E20.8,4H SEC)
  605 FORMAT(12X,26HNUMBER OF POINTS SPECIFIED,8X,1H=,I5/12X,35HMINIMUM
     1NUMBER OF POINTS ALLOWED  =,I5/12X,35HMAXIMUM NUMBER OF POINTS ALL
     2OWED  =,I5)
  606 FORMAT(/5X,38HAUTOMATIC NEW DATA OPTION SWITCHED OFF/5X,28HLAST AL
     1LOWABLE POINT REACHED)
  607 FORMAT(20H0    * NO DATA LOADED)
  608 FORMAT(36H0     DATA (NEGATIVE VALUES NOT USED)//5X,2HPT,8X,4HTIME,
     18X,4(11X,2A6))
  609 FORMAT(4X,I3,E20.8,4E23.8)
C
C     SET SWITCHES AND OUTPUT DATA IDENTIFICATION
    1 CONTINUE
      ASSIGN 12 TO N1
      N = 2
      IF (NSTTD.EQ.0)     GO TO 10
      I = NO
      IF (LAUTO) I=IBLK
      WRITE (6,601) I
      WRITE (6,602) ONTIME
      CALL DATOUP (ONTIME,TP,0)
      WRITE (6,603) TSTART
      IF (TFINAL.EQ.0.)  GO TO 2
      WRITE (6,604) TFINAL
    2 WRITE (6,605) NPTSTT,MINPTS,MAXPTS
C
C     LOAD DATA FROM EDTCOM
   10 IF (NPTSTT.GT.MAXPTS) NPTSTT=MAXPTS
      IF (NPTSTT.LT.MINPTS) NPTSTT=MINPTS
      J = 0
   11 CONTINUE
      DO 19 K=1,NPTS
      TP = BUFDAT(K,2)
      GO TO N1, (12,13,14)
C     FIRST POINT TO BE USED
   12 IF (TP.LT.TSTART) GO TO 19
      ASSIGN 14 TO N1
      IF (TFINAL.EQ.0.) GO TO 14
C     FINAL TIME SPECIFIED
      ASSIGN 13 TO N1
   13 IF (TP.GT.TFINAL) GO TO 30
C     IS THIS SET TO BE USED
   14 IF (BUFDAT(K,1).EQ.0.)    GO TO 15
      IF (BUFDAT(K,1).NE.QUAL) GO TO 19
   15 DO 16 I=1,4
      IF (BUFDAT(K,I+2).GE.0.) GO TO 17
```

```
   16 CONTINUE
      GO TO 19
C     USE THIS POINT
   17 CONTINUE
      J = J+1
      DO 18 I=1,5
   18 DATA(I,J) = BUFDAT(K,I+1)
      IF (J.GE.NPTSTT)    GO TO 34
   19 CONTINUE
C
C     BUFDAT IS EXHAUSTED, RELOAD FROM TAPE
   20 N = N+1
      READ (10) INDDAT
      IF (NEOT.NE.0)      GO TO 30
      IF (KONT.EQ.0)      GO TO 30
      N = N+1
      READ (10) BUFDAT
      GO TO 11
C
C     LAST ALLOWABLE POINT REACHED
   30 CONTINUE
      IF (.NOT.LAUTO)     GO TO 32
      IF (NSTTD.NE.0)     WRITE (6,606)
      LAUTO = .FALSE.
   32 NPTSTT = J
C     REPOSITION DATA TAPE
   34 CONTINUE
      IF (N.EQ.2)    GO TO 40
      DO 35 I=1,N
   35 BACKSPACE 10
      READ (10) INDDAT
      READ (10) BUFDAT
C
C     OUTPUT DATA SUMMARY
   40 CONTINUE
      IF (NPTSTT.NE.0)    GO TO 41
      WRITE (6,607)
      GO TO 999
   41 IF (NSTTD.LT.2)     GO TO 999
      WRITE (6,608) (LBM(I),LBM(I+4),I=1,4)
      DO 42 J=1,NPTSTT
   42 WRITE (6,609) J,(DATA(I,J),I=1,5)
C
  999 RETURN
      END
```

Subroutine: STTEND

Purpose: To store maximum likelihood estimate as the current state estimate.

Calling Sequence: CALL STTEND

Common storages used: //13 cells,/DCPCØM/,/ESTCØM/,/ES1CØM/,/MLECØM/

Subroutines required: DATØUP,STEPDI,STEPDT

Usage:

STTEND accepts the maximum likelihood estimate of vehicle position and velocity, X1, at the epoch ETIMV1, shifts the estimate along a conic section to the desired spoch, ETIMVE, and stores the vehicle state as the current estimate,SPCDAN. It assumes X1 in C-frame components relative to the central body ICB.

If it is desired to store the new estimate on the estimate tape, the covariance matrix and state indices are taken from the scratch tape, unit 9. Some care is required in using the data start following estimation of the state, since correlations between vehicle state and other parameters will be retained, though meaningless.

```
$IBFTC MC13S5  XR3,M94,NODD,LIST
      SUBROUTINE STTEND
C     STORES AND OUTPUTS STATE ESTIMATE
C
      COMMON            /DCPCOM/CDCP(900)
      DOUBLE PRECISION   DATEV1(3) ,ETIMVE
      DIMENSION          CBODY(8,11)
      EQUIVALENCE       (CDCP( 17),CBODY ) ,(CDCP(740),NPREST)
     1                  ,(CDCP(801),DATEV1) ,(CDCP(742),NPRSTO)
     2                  ,(CDCP(785),ETIMVE) ,(CDCP(685),NSTTS )
     3                  ,(CDCP(777),ICB    )
C
      COMMON            /ESTCOM/CEST(804)
      DOUBLE PRECISION   ETIMVA ,SPCDAN(6)
      EQUIVALENCE       (CEST( 29),ETIMVA) ,(CEST(  1),NESPOS)
     1                  ,(CEST(  2),NBY    ) ,(CEST( 49),SPCDAN)
C
      COMMON            /MLECOM/CMLE(1070)
      DOUBLE PRECISION   AMU ,ETIMV1 ,XA1(6)
      EQUIVALENCE       (CMLE(363),AMU    ) ,(CMLE(111),XA1    )
     1                  ,(CMLE( 11),ETIMV1)
C
      COMMON            /ES1COM/DEST(2214)
C
      COMMON             SAVE(10),TP(3)
C
  600 FORMAT(58H1FINAL STATE ESTIMATE IN C-FRAME COORDINATES, RELATIVE T
     10 ,A6/12H0AT ETIMV = ,D24.16,4H SEC)
  601 FORMAT(1H )
  602 FORMAT(45X,2HX(,I1,3H) =,D24.16)
  603 FORMAT(28H0NOT STORED ON ESTIMATE TAPE)
  604 FORMAT(35H0STORED ON TAPE AS ESTIMATE NUMBER ,I4)
C
C     SHIFT ESTIMATE TO DESIRED EPOCH
    1 DO 2 I=1,3
    2 TP(I) = DATEV1(I)
      CALL DATINP (TP,ETIMVE)
      IF (ETIMV1.EQ.ETIMVE) GO TO 10
      CALL STEPDI (ETIMV1,XA1,AMU)
      CALL STEPDT (ETIMVE,XA1)
C
C     LOAD ESTIMATE INTO ESTCOM
   10 NBY = ICB
      ETIMVA = ETIMVE
      DO 11 I=1,6
   11 SPCDAN(I) = XA1(I)
C
C     OUTPUT ESTIMATE
   20 IF (NSTTS.EQ.0)    GO TO 30
      WRITE (6,600) CBODY(1,ICB),ETIMVA
      CALL DATOUP (ETIMVA,TP,0)
      WRITE (6,601)
      WRITE (6,602) (I,XA1(I),I=1,6)
C
C     STORE ESTIMATE ON TAPE
   30 IF (NPRSTO.NE.0)   GO TO 31
      WRITE (6,603)
      GO TO 999
   31 IF (NPREST.EQ.0)   GO TO 32
      READ  (12) SKIP
      NPREST = 0
   32 READ  ( 9) DEST
      BACKSPACE 9
      NN = NESEND-NESPOS
      IF (NN.LE.0)       GO TO 35
      DO 34 I=1,NN
      READ  (12) SKIP
   34 READ  (12) SKIP
   35 NESEND = NESEND+1
      NESPOS = NESEND
      WRITE (12) CEST
      WRITE (12) DEST
      WRITE (6,604) NESPOS
C
  999 RETURN
      END
```

Subroutine: STTFIT

Purpose: To compute smoothed values of measurements and their time derivatives by making a least square polynomial fit for each measurement. An error return (IERR = 15) will occur unless a fit can be found for each measurement.

Calling Sequence: CALL STTFIT

Common storages used: //256 cells,/DCPCØM/,/MLECØM/

Subroutines required: DINVRT,DMPLY

Method - Theoretical

Given a number of data values y_j at times t_j, $j = 1, 2, \ldots, k$, we wish to find the coefficients a_i of an N^{th} degree polynomial

$$\hat{y}(t) = a_0 + a_1(t - \bar{t}) + a_2(t - \bar{t})^2 + \ldots$$

which fits the data in the least square sense.

For a given reference time, $\bar{t}$, we set

$$\Delta t_j = t_j - \bar{t}$$

and define the matrices

$$A = \text{col}(a_0, a_1, \ldots, a_N)$$

$$Y = \text{col}(y_1, y_2, \ldots, y_k)$$

$$\hat{Y} = \text{col}(\hat{y}(t_1), \hat{y}(t_2) \ldots, \hat{y}(t_k))$$

$$TM = \begin{bmatrix} 1 & \Delta t_1 & \Delta t_1^2 & \ldots & \Delta t_1^N \\ 1 & \Delta t_2 & \Delta t_2^2 & \ldots & \Delta t_2^N \\ \vdots & \vdots & & & \\ 1 & \Delta t_k & \Delta t_k^2 & \ldots & \Delta t_k^N \end{bmatrix}$$

Using

$$Y = TM \cdot A$$

the A which minimizes $||Y - \hat{Y}||$ is

$$A = (TM^T \cdot TM)^{-1} TM^T \cdot Y$$

at $t = \bar{t}$,

$$\hat{Y}(\bar{t}) = a_0$$

$$\frac{d}{dt}\hat{Y}(t) = a_1$$

For each measurement type, we accept the solution A if the mean square of the residuals satisfies

$$||Y - \hat{Y}||^2 \leq (\text{FITLEV}) \cdot (\text{variance of the measurement})$$

where FITLEV is an appropriate constant. The coefficients a_0, a_1 for the M^{th} measurement type are returned in CØEF(1,M) and CØEF(2,M).

Method - Computational

The following quantities are computed:

TMEAN $= \bar{t}$ = (DATA(1,1) + DATA(1,NPTS))/2.

JM = index in DATA of first time point < TMEAN, unless TFIT ≠ 0. If TFIT ≠ 0, then fit starts with the first time point < TFIT(=DATA(1,JM)), and considers only points with time > TFIT.

NB = First row of TM to be computed.

If TFIT ≠ 0, NB = 1, and rows 1 through NFPTS (=k) of TM are computed. If TFIT = 0, NB = NFPTS/2 + 1 and rows NB through NFPTS are computed using data starting at DATA(1,JM). Then rows NB - 1 through 1 are computed, running backwards through DATA from DATA(1, JM - 1).

If at any time in the above computation the program runs out of time points, an error exit is taken. The process of forming TM, Y, and loading CØEF is performed separately for each measurement type M for which LMEAS(M) is .TRUE. Since measurements with negative values in DATA are skipped, an error exit (IERR = 15) will occur if any one of the measurement types fails to find enough data.

If a fit is found (that is, sufficient data to form TM was available) but does not pass the residual test, a retry may be made. If TFIT and TMEAN both = 0 (so that the reference point $\bar{t}$ is not fixed by the calling program), then MFTRY new tries are allowed. Each new try is made with the reference point $\bar{t}$ moved alternately backwards or forwards NERR time points (in DATA) from the last referenced point.

```
$IBFTC MC13F5  XR3,M94,NODD,LIST
      SUBROUTINE STTFIT
C     COMPUTES MEASUREMENT AND MEASUREMENT RATES
C     FROM POLYNOMIAL FIT
C
      COMMON            /DCPCOM/CDCP(900)
      DIMENSION          NSCTRL(14)
      DOUBLE PRECISION   TFIT
      EQUIVALENCE       (CDCP(833),FITLEV) ,(CDCP(787),NSCTRL)
     1                 ,(CDCP(111),IERR   ) ,(CDCP(681),NSTTF  )
     2                                      ,(CDCP(825),TFIT   )
      EQUIVALENCE       (NSCTRL( 6),MFTRY ) ,(NSCTRL( 2),NFPTS )
     1                 ,(NSCTRL( 5),NFERR ) ,(NSCTRL( 3),NFSKIP)
     2                 ,(NSCTRL( 4),NFPOLY)
C
      COMMON            /MLECOM/CMLE(1070)
      DOUBLE PRECISION   COEF(2,4) ,Q(4) ,TMEAN
      DIMENSION          DATA(5,100) ,LBM(12)
      LOGICAL            LMEAS(4)
      EQUIVALENCE                   (CMLE(531),LBM    ) ,(CMLE( 47),Q      )
     1     ,(CMLE(347),COEF   ) ,(CMLE(397),LMEAS ) ,(CMLE(  3),TMEAN )
     2     ,(CMLE(570),DATA   ) ,(CMLE(432),NPTSTT)
C
      COMMON   SAVE(20) ,RMS(4) ,TST(4) ,POLY(10) ,TMT(10,10) ,TPM(10)
      DOUBLE PRECISION   POLY ,RMS ,TMT ,TPM ,TST
C
C     MAXIMUM VALUE FOR NFPOLY = 10
C     MAXIMUM VALUE FOR MAXPTS = 30
C
      DOUBLE PRECISION   TM(30,10) ,FITYH(30) ,FITYR(30) ,TP
      DIMENSION          NP(30)     ,FITY (30) ,FITT (30) ,LBPOLY(10)
      DATA   LBPOLY/2HA0 ,2HA1 ,2HA2 ,2HA3 ,2HA4
     1              ,2HA5 ,2HA6 ,2HA7 ,2HA8 ,2HA9 /
C
  601 FORMAT(71H1POLYNOMIAL FITS USED TO OBTAIN SMOOTHED VALUES OF DATA
     1AND THEIR RATES/15H0    PTS USED = I3,11X,20HPOLYNOMIAL DEGREE = I2
     2,11X,14HPTS SKIPPED = I3,9X,18HFIT TEST FACTOR = E16.8)
  602 FORMAT(12H0    TMEAN = D31.16,4H SEC,12X,18HFIT TRIAL COUNT = I3,9X
     1,16HMAX NO TRIALS = I4)
  603 FORMAT(//9X,2A6,48H POLYNOMIAL = A0 + A1*T + A2*T**2 + A3*T**3 +..
     1.//(2X,3(7X,A5,1H=,D24.16)))
  604 FORMAT(/9X,2HPT,8X,4HTIME,19X,10HOBSERVABLE,13X,16HPOLYNOMIAL VALU
     1E,14X,8HRESIDUAL/(8X,I3,E20.8,E23.8,2D31.16))
  605 FORMAT(/9X,22HRESIDUAL MEAN SQUARE =,D24.16,14X,16HFIT TEST LEVEL
     1=,D24.16)
  606 FORMAT(///15H SUMMARY OF FIT)
  607 FORMAT(/22X,13HFITTED VALUES,13X,5HRATES,21X,20HRESIDUAL MEAN SQUA
     1RE,6X,10HTEST LEVEL/1H )
  608 FORMAT(4X,2A6,4D24.16)
  609 FORMAT(27H0 POLYNOMIAL FITS COMPLETED)
  610 FORMAT(6H0    * ,A6,32H POLYNOMIAL FAILED RESIDUAL TEST)
  611 FORMAT(9X,37HTRY NEW FIT WITH TMEAN MOVED. NFERR =,I3///)
  612 FORMAT(61H0    * ATTEMPTED TO FIT POLYNOMIAL BEYOND LIMITS OF DATA
     1ARRAY)
C
C     OUTPUT HEADING
    1 CONTINUE
      I = NFPOLY-1
      IF (NSTTF.EQ.0) GO TO 2
      WRITE (6,601) NFPTS,I,NFSKIP,FITLEV
C     INITIALIZE FIT COUNTERS
    2 CONTINUE
      NB = NFPTS/2+1
      TMEAN = (DATA(1,1)+DATA(1,NPTSTT))/2.
    3 STP = TMEAN
      IF (TFIT.EQ.0.D0)  GO TO 4
      STP = TFIT
      NB = 1
    4 CONTINUE
      DO 6 JM=1,NPTSTT
      IF (STP.LE.DATA(1,JM))  GO TO 8
    6 CONTINUE
    8 CONTINUE
      NFTRY = 0
      JL = JM
```

```
      NFRR = NFERR
C
C     NEW TRIAL
   10 NFTRY = NFTRY+1
      IF (NSTTF.GT.4) WRITE (6,602) TMEAN,NFTRY,MFTRY
      MEAS = 0
C
C     MEASUREMENT LOOP
   20 MEAS = MEAS+1
      IF (.NOT.LMEAS(MEAS)) GO TO 70
      ASSIGN 21 TO N1
      II = 1
      IP = JM-1
      K  = NB
      JJ = NFSKIP
   21 IF (K .GT.NFPTS)       GO TO 22
      IF (IP.GE.NPTSTT)      GO TO 90
      GO TO 24
   22 ASSIGN 23 TO N1
      II =-1
      IP = JM
      K  = NB-1
      JJ = 0
   23 IF (K .LT.1)           GO TO 30
      IF (IP.LE.1)           GO TO 90
   24 IP = IP+II
      STP = DATA(MEAS+1,IP)
      IF (STP.LT.0.)         GO TO 29
      JJ = JJ+1
      IF (JJ.LE.NFSKIP)      GO TO 29
C     LOAD TIME
   25 JJ = 0
      NP(K)   = IP
      FITY(K) = STP
      TM(K,1) = 1.D0
      FITT(K) = DATA(1,IP)
      TP = FITT(K)-TMEAN
      DO 26 J=2,NFPOLY
   26 TM(K,J) = TM(K,J-1)*TP
      K = K+II
   29 GO TO N1, (21,23)
C
C     TM IS LOADED
   30 CONTINUE
      DO 31 I=1,110
   31 TMT(I,1) = 0.D0
      DO 33 K=1,NFPTS
      STP = FITY(K)
      DO 32 I=1,NFPOLY
      TPM(I) = TPM(I)+TM(K,I)*STP
      DO 32 J=1,NFPOLY
   32 TMT(I,J) = TMT(I,J)+TM(K,I)*TM(K,J)
   33 CONTINUE
C
C     SOLVE FOR COEFFICIENTS
   40 CONTINUE
      CALL DINVRT (TMT,NFPOLY,10)
      CALL DMPLY  (TMT,TPM,POLY,NFPOLY,NFPOLY,1,10,10,10,0)
      COEF(1,MEAS) = POLY(1)
      COEF(2,MEAS) = POLY(2)
C
C     COMPUTE RESIDUALS AND CHECK FIT
   50 CONTINUE
      CALL DMPLY (TM,POLY,FITYH,NFPTS,NFPOLY,1,30,10,30,0)
      TP   = 0.D0
      STP = NFPTS
      DO 52 I=1,NFPTS
      FITYR(I) = FITY(I)-FITYH(I)
      TP   = TP+FITYR(I)**2
   52 CONTINUE
      TP   = TP/STP
      TPM = FITLEV*Q(MEAS)
      RMS(MEAS) = TP
      TST(MEAS) = TPM
C
```

```
C     OUTPUT FIT DATA                                                   STFT0150
   60 IF (NSTTF.LT.4) GO TO 66                                          STFT0151
      WRITE (6,603) LBM(MEAS),LBM(MEAS+4),(LBPOLY(I),POLY(I),I=1,NFPOLY)STFT0152
   62 IF (NSTTF.LT.5) GO TO 64                                          STFT0153
      WRITE (6,604) (NP(I),FITT(I),FITY(I),FITYH(I),FITYR(I),I=1,NFPTS) STFT0154
   64 WRITE (6,605) TP,TPM(1)                                           STFT0155
   66 CONTINUE                                                          STFT0156
      IF (TP.GT.TPM) GO TO 80                                           STFT0157
C                                                                       STFT0158
C     END MEASUREMENT LOOP                                              STFT0159
   70 IF (MEAS.LT.4) GO TO 20                                           STFT0160
C     FIT COMPLETED AND ACCEPTED                                        STFT0161
      IERR = 0                                                          STFT0162
      IF (NSTTF.LT.2) GO TO 999                                         STFT0163
      WRITE (6,606)                                                     STFT0164
      WRITE (6,602)    TMEAN,NFTRY,MFTRY                                STFT0165
      WRITE (6,607)                                                     STFT0166
      DO 72 I=1,4                                                       STFT0167
      IF (.NOT.LMEAS(I)) GO TO 72                                       STFT0168
      WRITE (6,608) LBM(I),LBM(I+4),COEF(1,I),COEF(2,I),RMS(I),TST(I)   STFT0169
   72 CONTINUE                                                          STFT0170
      IF (NSTTF.GT.0) WRITE (6,609)                                     STFT0171
      GO TO 999                                                         STFT0172
C                                                                       STFT0173
C     NO FIT OBTAINED                                                   STFT0174
   80 IF (NSTTF.GE.3) WRITE (6,610) LBM(MEAS)                           STFT0175
      IF (NFTRY.GE.MFTRY) GO TO 95                                      STFT0176
      IF (NSTTF.GE.3) WRITE (6,611) NFERR                               STFT0177
      I  = JL                                                           STFT0178
      JL = JM                                                           STFT0179
      JM = I+NERR                                                       STFT0180
      TMEAN = DATA(1,JM)                                                STFT0181
      NERR = -NERR                                                      STFT0182
      GO TO 10                                                          STFT0183
C                                                                       STFT0184
C     EXIT                                                              STFT0185
   90 IF (NSTTF.GE.3) WRITE (6,612)                                     STFT0186
   95 IERR = 15                                                         STFT0187
C                                                                       STFT0188
  999 RETURN                                                            STFT0189
      END
```

Subroutine: STTIME

Purpose: Computes residuals and accumulates the information matrix (P^{-1}) and projected residuals $H^T Q^{-1}(y - \hat{y})$ at a time point for use in maximum likelihood estimate. Optionally computes and outputs residuals.

Common storages used: // 144 cells,/DATCØM/,/DCPCØM/,/EDTCØM/,/ESTCØM/, /MLECØM/

Subroutines required: CBDATP,DMPLY,DMVTRN,DSDATP,GRDATP,SBDATP, STATE,STEPDP

Method:

STTIME accepts the universal time corresponding to the time tag for a data point, and the conic section corresponding to the estimate of state at the epoch, and either output residuals or accumulates the information matrix and projected residuals.

The down-leg transmission delay is approximated from

$$\Delta t = \frac{1}{c} | R_v(t_r) - R_s(t_{ur}) |$$

where

R_v = vehicle position

R_s = observing station position

t_r = E.T. of signal reception

t_{ur} = U.T. of signal reception

c = speed of light

This delay is used for both up-leg and down-leg delays for the computation of vehicle time, t_v, and transmission time, t_t. The transformations T_{B2C} at the transmission and reception times, and the vehicle state at vehicle time, are computed for both ends of the doppler count interval by calls of subroutine STATE.

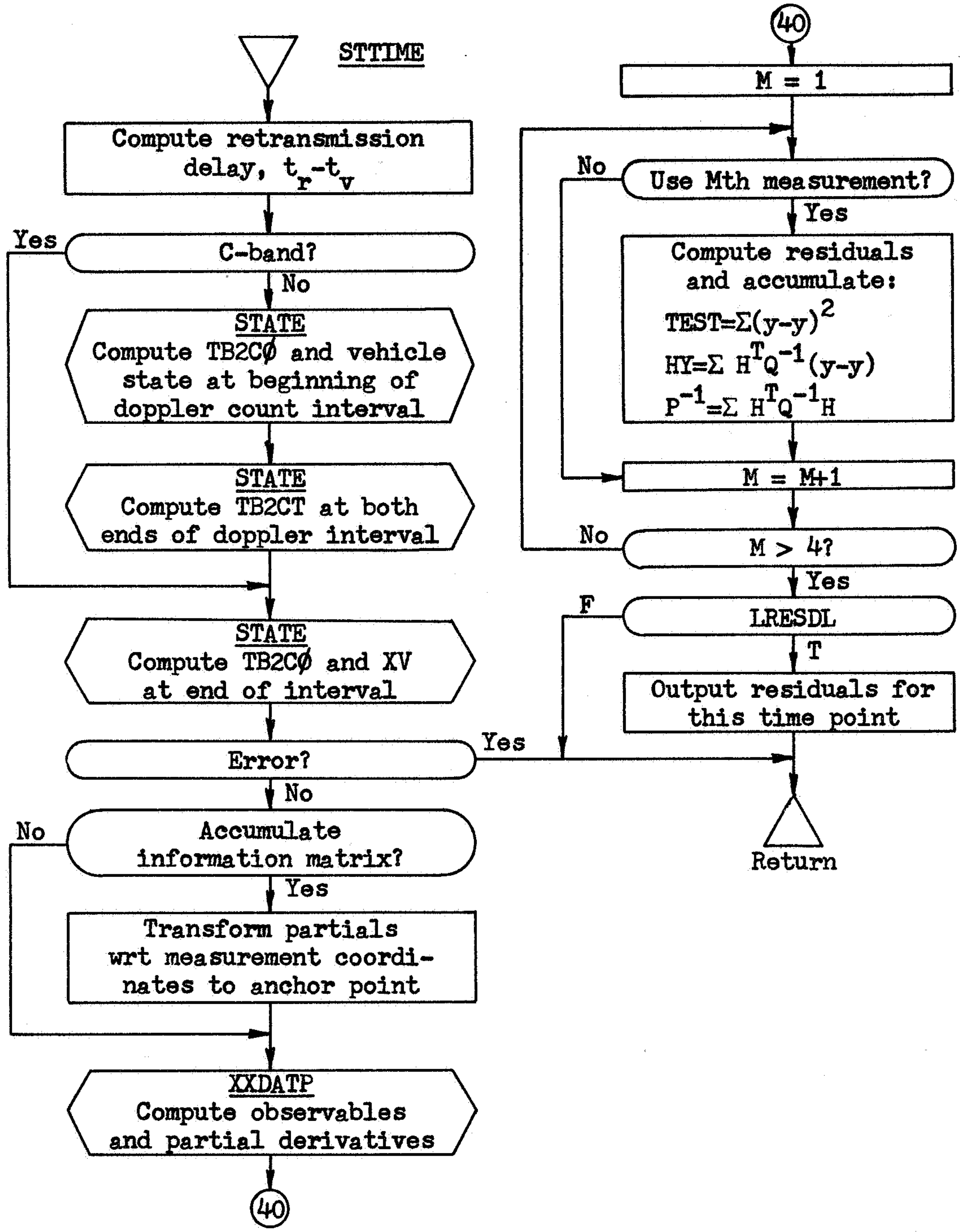
The conic transition matrix $\varphi(t_v, t_{v1})$, where t_{v1} is the vehicle E.T. at epoch, is used to compute the partial derivatives

$$\frac{\partial x(t_v)}{\partial x_m} = \varphi(t_v \; ; \; t_{v1}) \frac{\partial x(t_{v1})}{\partial x_m}$$

where $x(t)$ is the cartesian state and x_m are the measurement coordinates of t_{v1} (see subroutine MEAS2X). The information matrix, P^{-1}, and projected residuals, $H^T Q^{-1}(y - \hat{y})$ are accumulated using

$$P^{-1} = H^T Q^{-1} H$$

$$H = \frac{\partial y}{\partial x_m} = \frac{\partial x(t_v)}{\partial x_m}$$


STTIME
Compute retransmission delay, $t_r - t_v$
C-band?
Yes
No
STATE
Compute TB2CØ and vehicle state at beginning of doppler count interval
STATE
Compute TB2CT at both ends of doppler interval
STATE
Compute TB2CØ and XV at end of interval
Error?
Yes
No
Accumulate information matrix?
No
Yes
Transform partials wrt measurement coordinates to anchor point
XXDATP
Compute observables and partial derivatives
40
40
M = 1
Use Mth measurement?
No
Yes
Compute residuals and accumulate:
TEST=Σ(y-y)^2
HY=Σ H^{T}Q^{-1}(y-y)
P^{-1}=Σ H^{T}Q^{-1}H
M = M+1
M > 4?
No
Yes
LRESDL
F
T
Output residuals for this time point
Return

```
$IBFTC MC13T5  XR3,M94,NODD,LIST
      SUBROUTINE STTIME
C     COMPUTES AND OUTPUTS RESIDUALS
C     ACCUMULATES INFORMATION MATRIX
C
      DOUBLE PRECISION   DSQRT
C
      COMMON             /DCPCOM/CDCP(900)
      EQUIVALENCE        (CDCP(111),IERR   ) ,(CDCP(684),NSTIR )
C
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)
      EQUIVALENCE        (INDDAT(8),MTYPE )
C
      COMMON             /ESTCOM/CEST(804)
      DOUBLE PRECISION    DELDAN(2)
      EQUIVALENCE        (CEST( 97),DELDAN)
C
      COMMON             /DATCOM/CDAT(299)
      DOUBLE PRECISION    OBS(16,4) ,TB2CO(3,6) ,XV(12)
     1                   ,SPDLT      ,TB2CI(3,6) ,TAU
      EQUIVALENCE              (CDAT(137),SPDLT ) ,(CDAT(161),TB2CO )
     1                        ,(CDAT(139),SIA   ) ,(CDAT(197),TB2CI )
     2     ,(CDAT(   5),OBS    ) ,(CDAT(159),TAU   ) ,(CDAT(269),XV    )
C
      COMMON             /MLECOM/CMLE(1070)
      DOUBLE PRECISION    DATAT(4) ,ETIMR ,HY   (6) ,QI   (4) ,UTIMT
     1                   ,DELAY    ,ETIMT ,OBIAS(4) ,TEST(4) ,YMOD(4)
     2                   ,DRA      ,ETIMV ,PI (6,6) ,UTIMR    ,YR   (4)
      DIMENSION           DATA(5,100) ,NYR(4)
      LOGICAL             LDATA    (4) ,LRESDL
      EQUIVALENCE              (CMLE( 19),ETIMV ) ,(CMLE(223),PI     )
     1                        ,(CMLE(295),HY     ) ,(CMLE(135),PX12M )
     2     ,(CMLE(570),DATA   ) ,(CMLE(434),ITIME ) ,(CMLE( 55),QI     )
     3     ,(CMLE( 39),DATAT ) ,(CMLE(401),LDATA ) ,(CMLE(307),TEST   )
     4     ,(CMLE( 25),DELAY ) ,(CMLE(407),LRESDL) ,(CMLE( 15),UTIMR )
     5     ,(CMLE(367),DRA    ) ,(CMLE(431),NSTATE) ,(CMLE( 21),UTIMT )
     6     ,(CMLE( 17),ETIMR ) ,(CMLE(427),NYR    ) ,(CMLE(207),YMOD   )
     7     ,(CMLE( 23),ETIMT ) ,(CMLE( 31),OBIAS ) ,(CMLE(215),YR     )
C
      COMMON              PX2M(6,6) ,TP(36)
      DOUBLE PRECISION    PX2M,TP
C
  601 FORMAT(8X,5E17.8)
C
C     RETRANSMISSION DELAY
C
    1 CONTINUE
      ETIMR = UTIMR+DELDAN(1)+DELDAN(2)*UTIMR
      CALL STATE  (ETIMR,UTIMR,TP(31),TB2CO,1)
      CALL DMVTRN (TB2CO,STA,TP,1,1)
      TP(4) = 0.D0
      DO 2 I=1,3
      TP(5) = TP(I+30)-TP(I)
    2 TP(4) = TP(4)+TP(5)*TP(5)
      DELAY = DSQRT(TP(4))/SPDLT
      ETIMV = ETIMR-DELAY
C
C     VEHICLE STATE AND EARTH ORIENTATION
C
   10 IF (MTYPE.EQ.1)    GO TO 11
C     AT RECEPTION, BEGINNING OF COUNT
      CALL STATE  (ETIMV-TAU,UTIMR-TAU,XV(7),TB2CO(1,4),1)
C     AT TRANSMISSION
      ETIMT = ETIMV-DELAY
      UTIMT = ETIMT-DELDAN(1)-DELDAN(2)*ETIMT
      CALL STATE  (TP,UTIMT     ,TP,TB2CT      ,0)
      CALL STATE  (TP,UTIMT-TAU,TP,TB2CT(1,4),0)
C     AT RECEPTION, END OF COUNT
   11 CALL STATE  (ETIMV,UTIMR,XV,TB2CO,1)
      IF (IERR.EQ.0)     GO TO 20
      IERR = IERR+15
      GO TO 999
C
C     UPDATE ANCHOR POINT PARTIALS
C
```

PHILCO
PHILCO-FORD CORPORATION

Space & Re-entry
Systems Division

```
   20 IF (LRESDL)          GO TO 30
      CALL STEPDP (TP)
      CALL DMPLY  (TP,PX12M,PX2M,6,6,6,6,6,6,0)
C
C     COMPUTE MEASUREMENT ESTIMATES AND PARTIALS
C
   30 CONTINUE
      GO TO (31,32,33,34) ,MTYPE
C     C-BAND
   31 CONTINUE
      CALL CBDATP (TP)
      GO TO 40
C     STADAN/VHF
   32 CONTINUE
      CALL GRDATP (TP)
      GO TO 40
C     USBS
   33 CONTINUE
      CALL SBDATP (TP)
      GO TO 40
C     DSIF
   34 CONTINUE
      CALL DEHA   (UTIMR,DRA,TP(2),TP)
      CALL DSDATP (TP(2),TP)
C
C     COMPUTE RESIDUALS AND ACCUMULATE INFORMATION MATRIX
C
   40 CONTINUE
      DO 49 M=1,4
      YR(M) =-0.12345678E20
      IF (.NOT.LDATA(M)) GO TO 49
      YR(M) = DATAT(M)-OBS(1,M)-OBIAS(M)
      TP = YR(M)/YMOD(M)+0.5D0
      IF (TP.LT.0.D0) TP = TP-1.D0
      TP = AINT(TP)
      YR(M) = YR(M)-TP*YMOD(M)
      TEST(M) = TEST(M)+YR(M)**2
      NYR(M) = NYR(M)+1
      IF (LRESDL)          GO TO 49
      DO 44 J=1,NSTATE
      TP(J+30) = 0.D0
      DO 42 I=1,6
   42 TP(J+30) = TP(J+30)+OBS(I+1,M)*PX2M(I,J)
   44 CONTINUE
      DO 46 J=1,NSTATE
      TP = QI(M)*TP(J+30)
      HY(J) = HY(J)+TP*YR(M)
      DO 46 I=1,NSTATE
   46 PI(I,J) = PI(I,J)+TP*TP(I+30)
   49 CONTINUE
C
C     OUTPUT RESIDUALS
C
   50 IF (.NOT.LRESDL.OR.NSTTR.EQ.0) GO TO 999
      WRITE (6,601) DATA(1,ITIME),(YR(I),I=1,4)
  999 RETURN
      END
```

Subroutine: TCØNIC

Purpose: To compute time from periapsis corresponding to a given true anomaly on a specified conic section.

Calling Sequence: CALL TCØNIC (U,EC,A,SLR,TA2,T,FAC)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	U		GM	km^3/sec^2	Gravitational constant
I	EC		e		Eccentricity
I	A		a	km	Semi-major axis
I	SLR		p	km	Semi-latus rectum
I	TA2		θ	rad	True anomaly
Ø	T		t	sec	Time from periapsis
Ø	FAC		1/n	sec/rad	Reciprocal mean motion

Common storages required: None

Subroutines required: None

Method

Time from periapsis is computed by the following equations and procedures.

$$T = \frac{1}{n} = a\sqrt{a/GM}$$

$$R = \frac{1 - e}{1 + e}$$

$$F = \sqrt{|R|}$$

$$E = 2 \tan^{-1}(F \tan \frac{\theta}{2})$$

If the conic is elliptical, the time is calculated by

$$t = T \cdot (E - e \sin E)$$

If the conic is hyperbolic, the time is calculated by

$$t = T \cdot (e \tanh E - \ln \tan (\frac{E}{2} + \frac{\pi}{4}))$$

If the conic is very nearly parabolic, the time is calculated from the truncated series

$$T = \frac{2\sqrt{p^3/GM}}{(1 + e)^2}$$

$$G = R \tan^2 (\frac{\theta}{2})$$

$$t = T \cdot (\tan \frac{\theta}{2} + \frac{(1-2R)}{3}\tan^3 \frac{\theta}{2} - \frac{(2-3R)}{5}G^3 + \frac{(3-4R)}{7}G^5 - \frac{(4-5R)}{9}G^7)$$

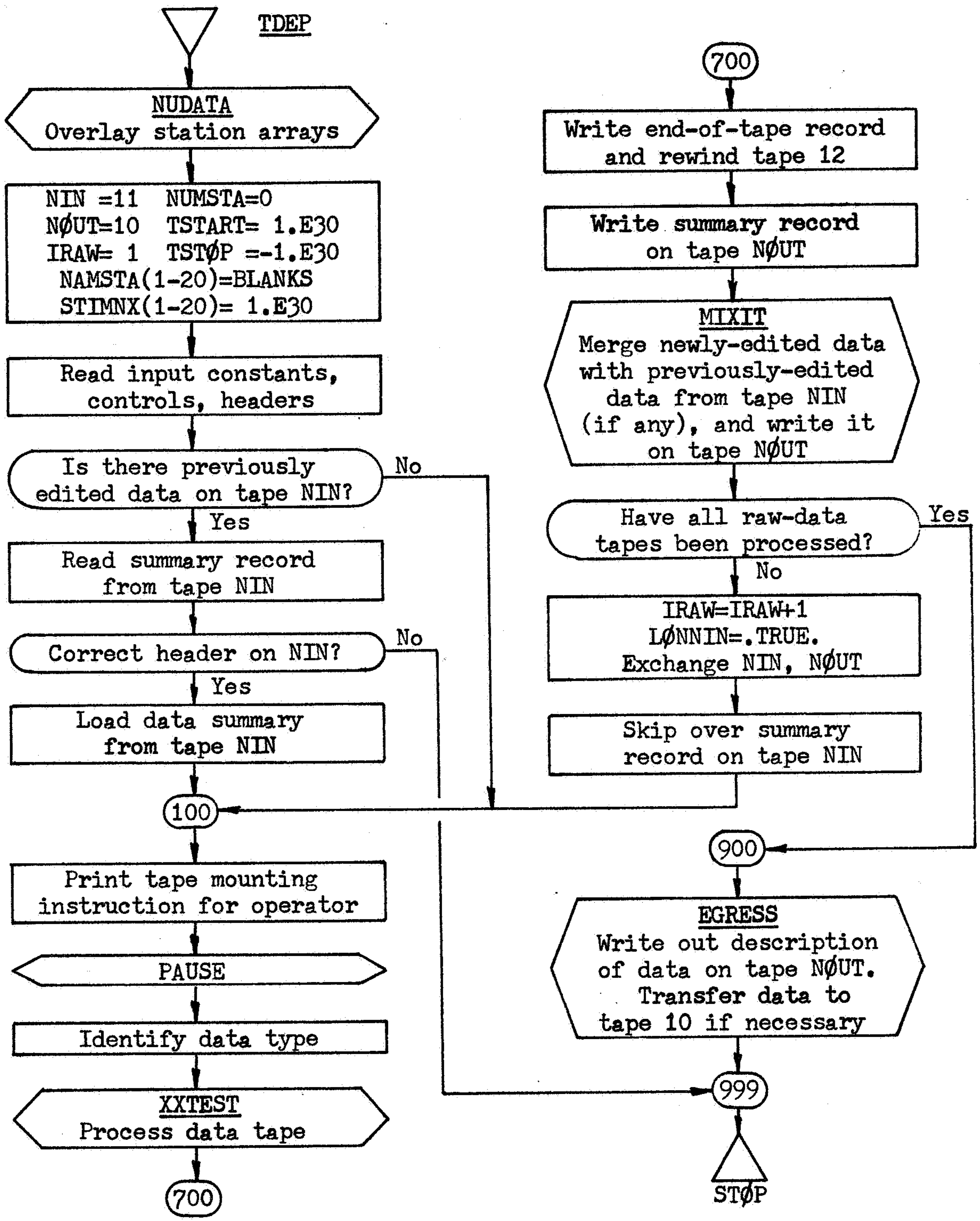
```
$IBFTC MC13TC  XR3,M94,NOREF,NODD
      SUBROUTINE TCONIC(U,EC,A,SLR,TA2,T,FAC)                              TCON0010
      TANGF(X)=SIN (X)/COS (X)                                             TCON0020
      AB=ABS (A)                                                           TCON0030
      FAC=AB*SQRT (AB/U)                                                   TCON0040
      ECA=(1.-EC)/(1.+EC)                                                  TCON0050
      ABE=SQRT (ABS (ECA))                                                 TCON0060
      THE=TANGF(.5*TA2)                                                    TCON0070
      IF(ABE-.00005)11,11,12                                               TCON0080
   12 CONTINUE                                                             TCON0090
      ECA=2.*ATAN (ABE*THE)                                                TCON0100
      IF(A)14,11,13                                                        TCON0110
   13 T=FAC*(ECA-EC*SIN (ECA))                                             TCON0120
      GO TO 16                                                             TCON0130
   14 ANG=ABE*THE                                                          TCON0140
      ANG=1.+2.*ANG/(1.-ANG)                                               TCON0150
      T=FAC*(EC*TANGF(ECA)-ALOG(ANG))                                      TCON0160
      GO TO 16                                                             TCON0170
   11 FAC=SQRT (SLR**3/U)*2./((1.+EC)**2)                                  TCON0180
      EC1=ECA*THE**2                                                       TCON0190
      T=FAC*(THE+THE**3*((1.-2.*ECA)/3.-(2.-3.*ECA)*EC1/5.+(3.-4.*ECA)*ETCON0200
     1C1**2/7.-(4.-5.*ECA)*EC1**3/9.))                                     TCON0210
   16 CONTINUE                                                             TCON0220
      RETURN                                                               TCON0230
      END                                                                  TCON0240
```

Subroutine: TDEP

Purpose: Executive driver to control the logic flow in the Tracking Data Editing Program.

Common storages used: /MIXCØM/,/ØUTCØM/,/SUMCØM/,/TSTCØM/

Subroutines required: CBTEST, DATINP, DATØUP, DSTEST, EGRESS, GRTEST, MIXIT, NUDATA, SBTEST, SCANIT


TDEP
NUDATA
Overlay station arrays
NIN =11 NUMSTA=0
NØUT=10 TSTART= 1.E30
IRAW= 1 TSTØP =-1.E30
NAMSTA(1-20)=BLANKS
STIMNX(1-20)= 1.E30
Read input constants, controls, headers
Is there previously edited data on tape NIN?
No
Yes
Read summary record from tape NIN
Correct header on NIN?
No
Yes
Load data summary from tape NIN
100
Print tape mounting instruction for operator
PAUSE
Identify data type
XXTEST
Process data tape
700
700
Write end-of-tape record and rewind tape 12
Write summary record on tape NØUT
MIXIT
Merge newly-edited data with previously-edited data from tape NIN (if any), and write it on tape NØUT
Have all raw-data tapes been processed?
Yes
No
IRAW=IRAW+1
LØNNIN=.TRUE.
Exchange NIN, NØUT
Skip over summary record on tape NIN
900
EGRESS
Write out description of data on tape NØUT. Transfer data to tape 10 if necessary
999
STØP

```
$IBFTC MC134Z  XR3,M94,NODD,LIST
C     TDEP - TRACKING DATA EDITING PROGRAM                              TDEP0001
C                                                                       TDEP0002
C     DEVELOPED BY  PHILCO-FORD CORPORATION                             TDEP0003
C               FOR GODDARD SPACE FLIGHT CENTER                         TDEP0004
C               CONTRACT NAS5-9939                                      TDEP0005
C                                                                       TDEP0006
C     REFERENCE TR-DA1508, PROGRAM DESCRIPTION AND THEORETICAL BASIS,   TDEP0007
C                          ORBIT DETERMINATION PROGRAM                  TDEP0008
C               TR-DA1509, SUBROUTINE DESCRIPTIONS AND LISTINGS,        TDEP0009
C                          ORBIT DETERMINATION PROGRAM                  TDEP0010
C               TR-DA1510, INPUT-OUTPUT SUMMARY,                        TDEP0011
C                          ORBIT DETERMINATION PROGRAM                  TDEP0012
C                                                                       TDEP0013
C     DECEMBER 1967                                                     TDEP0014
C                                                                       TDEP0015
      COMMON  /MIXCOM/IMIX(4)                                           TDEP0016
      LOGICAL LONNIN,LON12                                              TDEP0017
      EQUIVALENCE              (IMIX(1),LONNIN),    (IMIX(2),LON12)     TDEP0018
     1,  (IMIX(3),NIN),        (IMIX(4),NOUT)                           TDEP0019
C                                                                       TDEP0020
      COMMON  /TSTCOM/CTEST(400)                                        TDEP0021
      DOUBLE PRECISION  FINISH(10),    TLO(10),       THI(10)           TDEP0022
      DIMENSION    NYR(10),       NPTS(10),      NSTEP(10),    NDEG(10) TDEP0023
     1,            CSD(4,10),     IFOMIT(4,10), ISTAOK(20,10),KMAX(10)  TDEP0024
      EQUIVALENCE              (CTEST(1),NYR),      (CTEST(11),NPTS)    TDEP0025
     1,  (CTEST(21),NSTEP),    (CTEST(31),NDEG),    (CTEST(41),CSD)     TDEP0026
     2,  (CTEST(81),IRAW),     (CTEST(87),FINISH)                       TDEP0027
     3,  (CTEST(107),IFOMIT), (CTEST(147),KMAX),    (CTEST(157),ISTAOK) TDEP0028
     4,  (CTEST(357),TLO),     (CTEST(377),THI)                         TDEP0029
C                                                                       TDEP0030
      COMMON  /SUMCOM/SUMARY(56)                                        TDEP0031
      DOUBLE PRECISION TSTART,TSTOP                                     TDEP0032
      DIMENSION    HEADER(11),    STIMNX(20)                            TDEP0033
      REAL NAMSTA(20)                                                   TDEP0034
      EQUIVALENCE              (SUMARY(1),HEADER),  (SUMARY(12),NUMSTA) TDEP0035
     1,  (SUMARY(13),NAMSTA), (SUMARY(33),TSTART), (SUMARY(35),TSTOP)   TDEP0036
     2,  (SUMARY(37),STIMNX)                                            TDEP0037
C                                                                       TDEP0038
      COMMON  /OUTCOM/COUT(40)                                          TDEP0039
      EQUIVALENCE  (COUT(2),NEOT)                                       TDEP0040
C                                                                       TDEP0041
      DIMENSION    SUMNIN(56)                                           TDEP0042
      DOUBLE PRECISION TFNIN,TLNIN                                      TDEP0043
      DIMENSION    HEDNIN(11),    STININ(20)                            TDEP0044
      REAL NAMNIN(20)                                                   TDEP0045
      EQUIVALENCE              (SUMNIN(1),HEDNIN),  (SUMNIN(12),NUMNIN) TDEP0046
     1,  (SUMNIN(13),NAMNIN), (SUMNIN(33),TFNIN),  (SUMNIN(35),TLNIN)   TDEP0047
     2,  (SUMNIN(37),STININ)                                            TDEP0048
C                                                                       TDEP0049
      DIMENSION    MESSGE(11,10),MTYPE(10),    HDRNIN(11),    DI(9)     TDEP0050
      LOGICAL XTROUT                                                    TDEP0051
C                                                                       TDEP0052
  501 FORMAT(I6,11A6)                                                   TDEP0053
  502 FORMAT(6X,11A6/5I3)                                               TDEP0054
  503 FORMAT(4F10.0)                                                    TDEP0055
  504 FORMAT(4I1)                                                       TDEP0056
  505 FORMAT(I3,20I2)                                                   TDEP0057
  506 FORMAT(9F8.2)                                                     TDEP0058
  601 FORMAT(1H1,45X,28H*** TRACKING DATA EDITOR ***/44H0NUMBER OF RAW DTDEP0059
     1ATA TAPES TO BE PROCESSED = ,I2/20H0OUTPUT TAPE HEADER,11A6/48H0METDEP0060
     2RGE (INDICATOR FOR PREVIOUSLY EDITED DATA) = ,I2/36H0PREVIOUSLY EDTDEP0061
     3ITED DATA TAPE HEADER,11A6//52X,16H****************)              TDEP0062
  602 FORMAT(////21H RAW DATA TAPE NUMBER,I3/54H0***  MOUNT THE FOLLOWINTDEP0063
     1G TAPE ON UNIT 9 AND HIT START/1X,11A6/27H0*** CONTROLS FOR THIS TTDEP0064
     2APE/20H0MEASUREMENT TYPE = ,I2,2X,45H(1 = C-BAND,  2 = GRRD,  3 = TDEP0065
     3USBS,  4 = DSIF)/15H0YEAR NUMBER = ,I3/38H0NUMBER OF POINTS IN POLTDEP0066
     4YNOMIAL FIT = ,I3/29H0POLYNOMIAL WALK STEP SIZE = ,I3/24H0DEGREE OTDEP0067
     5F POLYNOMIAL = ,I2/46H0NUMBER OF STANDARD DEVIATIONS ALLOWED IN DATDEP0068
     6TA/2X,35HANGLE 1   ANGLE 2   RANGE   DOPPLER/F8.2,F10.2,2F9.2/67H0TDEP0069
     7PROCESSING OF THIS TAPE WILL STOP IF THE FOLLOWING DATE IS REACHEDTDEP0070
     8)                                                                 TDEP0071
  603 FORMAT(36H0***  EDIT ASSIST DATA FOR THIS TAPE//6X,26H0BSERVABLE STDEP0072
     1UPRESSION KEYS/36H ANGLE 1   ANGLE 2   RANGE   DOPPLER/I5,I10,I9,ITDEP0073
     29/27H0ACCEPTABLE STATION CODES (,I2,1H)/I3,19(1H,,I3))            TDEP0074
```

```
  604 FORMAT(39HODATA LINES WITH TIME TAGS EARLIER THAN)
  605 FORMAT(14H OR LATER THAN)
  606 FORMAT(17H WILL BE REJECTED)
  908 FORMAT(60HO***  PROGRAM CANNOT PROCESS MORE THAN 10 TAPES SEQUENTI
     1ALLY/6X,24HPROGRAM RUN DISCONTINUED)
 1001 FORMAT(43HO*** EDITED DATA TAPE HEADER DOES NOT AGREE/1X,11A6/
     16X,22HPROGRAM RUN TERMINATED)
 1002 FORMAT(54HO***  MOUNT THE FOLLOWING TAPE ON UNIT 9 AND HIT START/
     11X,11A6)
      DATA BLANKS/6H      /
C     OVERLAY STATION DATA
      CALL NUDATA
C     INITIALIZE CONTROLS
      NIN=11
      NOUT=10
      IRAW=1
C     INITIALIZE SUMMARY DATA
      TSTART=1.E30
      TSTOP=-1.E30
      DO 1 I=1,20
      STIMNX(I)=1.E30
    1 NAMSTA(I)=BLANKS
      NUMSTA=0
C     READ INPUT CONSTANTS, CONTROLS, HEADERS
      READ(5,501) IMAX,HEADER
      XTROUT=IMAX.LT.0
      IMAX=IABS(IMAX)
      READ(5,501) MERGE,HDRNIN
      IF(IMAX.GT.10) GO TO 998
      DO 5 I=1,IMAX
      READ(5,502) (MESSGE(J,I),J=1,11),MTYPE(I),NYR(I),NPTS(I),NSTEP(I)
     1,NDEG(I)
      NPTS(I)=MIN0(NPTS(I),20)
      NSTEP(I)=MAX0(NSTEP(I),1)
      NDEG(I)=MIN0(MAX0(NDEG(I),1),5)
      READ(5,503) (CSD(J,I),J=1,4)
      DO 2 J=1,4
    2 IF(CSD(J,I).EQ.0.) CSD(J,I)=10.
      READ(5,504) (IFOMIT(J,I),J=1,4)
      READ(5,505) KK,(ISTAOK(J,I),J=1,KK)
      KMAX(I)=KK
      READ(5,506) DI
      CALL DATINP(DI,TLO(I))
      CALL DATINP(DI(4),THI(I))
      IF(DI(7).GT.0.) GO TO 5
      DI(7)=10001.
    5 CALL DATINP(DI(7),FINISH(I))
      WRITE(6,601) IMAX,HEADER,MERGE,HDRNIN
      LONNIN=MERGE.NE.0
      IF(.NOT.LONNIN) GO TO 100
C     READ SUMMARY FROM PREVIOUSLY EDITED DATA
      READ(NIN) SUMNIN
C     TEST FOR PROPER HEADER
      DO 10 I=1,11
      IF(HEDNIN(I).EQ.HDRNIN(I)) GO TO 10
      WRITE(6,1001) HEDNIN
      GO TO 999
   10 CONTINUE
C     STORE SUMMARY DATA FOR OUTPUT
      NUMSTA=NUMNIN
      DO 20 I=1,NUMSTA
      NAMSTA(I)=NAMNIN(I)
   20 STIMNX(I)=STININ(I)
      TSTART=TFNIN
      TSTOP=TLNIN
  100 CONTINUE
C     PRINT MOUNT INSTRUCTION TO OPERATOR
      PRINT 1002, (MESSGE(I,IRAW),I=1,11)
      WRITE(6,602) IRAW,(MESSGE(I,IRAW),I=1,11),MTYPE(IRAW),NYR(IRAW)
     1,NPTS(IRAW),NSTEP(IRAW),NDEG(IRAW),(CSD(I,IRAW),I=1,4)
      CALL DATOUP(FINISH(IRAW),DI,0)
      KK=KMAX(IRAW)
      WRITE(6,603) (IFOMIT(I,IRAW),I=1,4),KMAX(IRAW),(ISTAOK(I,IRAW),I=1
     1,KK)
      WRITE(6,604)
```

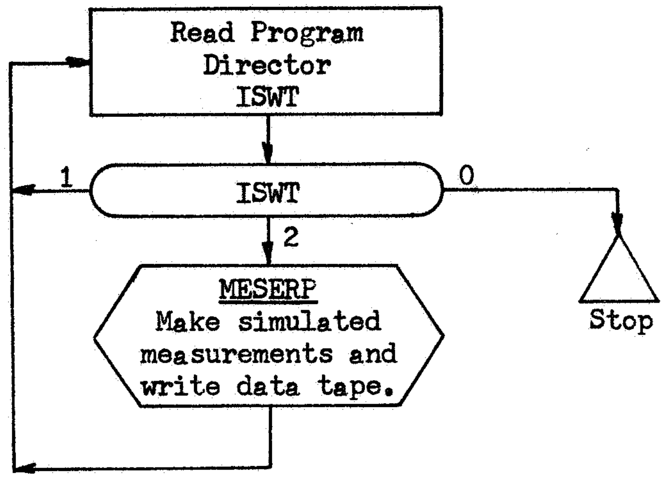
```
      CALL DATOUP(TLO(IRAW),DI,0)
      WRITE(6,605)
      CALL DATOUP(THI(IRAW),DI,0)
      WRITE(6,606)
C
C     FOR SEQUENTIAL PROCESSING OF MORE THAN ONE TAPE,
C     REMOVE C FROM COLUMN 1 OF NEXT CARD
C     PAUSE
C     ARE YOU SURE YOUR SYSTEM ALLOWS A PAUSE
C     ON SOME SYSTEMS A PAUSE CAUSES A TRANSFER TO THE EXIT ROUTINE
C
  200 CONTINUE
      KKK=MTYPE(IRAW)
      GO TO (300,400,500,600),KKK
C     PROCESS A TAPE OF C-BAND DATA
  300 CALL CBTEST(IERR)
      GO TO 700
C     PROCESS A TAPE OF GODDARD RANGE-RANGE RATE DATA
  400 CALL GRTEST(IERR)
      GO TO 700
C     PROCESS A TAPE OF USBS DATA
  500 CALL SBTEST(IERR)
      GO TO 700
C     PROCESS A TAPE OF DSIF (JPL) DATA
  600 CALL DSTEST(IERR)
  700 CONTINUE
C     REWIND RAW DATA TAPE
      REWIND 9
C     PUT END OF TAPE RECORD ON SCRATCH TAPE (UNIT12)
      NEOT=1
      WRITE(12) COUT
      REWIND 42
      LON12=.TRUE.
C     WRITE SUMMARY RECORD ON NOUT
      WRITE(NOUT) SUMARY
C     MERGE DATA ON TAPE 12 WITH DATA ON TAPE NIN
C     PUT MERGED DATA ON TAPE NOUT
      CALL MIXIT
      IF(IERR.EQ.2) GO TO 900
C     THIS WAY TO THE GRAND EGRESS
      IF(IRAW.GE.IMAX) GO TO 900
C     PREPARE FOR NEXT RAW DATA TAPE
      LONNIN=.TRUE.
      NNN=NIN
      NIN=NOUT
      NOUT=NNN
      IRAW=IRAW+1
C     SKIP OVER SUMMARY RECORD ON NIN
C     YOU STILL HAVE IT IN SUMCOM, HAVING JUST WRITTEN IT
      READ(NIN) DUMMY
      GO TO 100
  900 CONTINUE
C     WRITE SHORT DESCRIPTION OF FINAL OUTPUT
C     INSURE FINAL OUTPUT TO BE ON TAPE 10
      CALL EGRESS
C     WRITE OUT ALL DATA IF DESIRED
C     YOU CAN USE SUBROUTINE SCANIT AS A READING PROGRAM
C     FOR THESE EDITED DATA TAPES.
C     JUST REMOVE SUBROUTINE AND RETURN CARDS
C     BE SURE TO PROVIDE SUBROUTINE DATOUP
      IF(XTROUT) CALL SCANIT
      GO TO 999
  998 WRITE(6,908)
  999 STOP
      END
```

Subroutine: TDS

Purpose: Executive driver to direct the flow of the tracking data simulator program.

Common storages used: /INPCØM/

Subroutines required: MESERP


Read Program
Director
ISWT
ISWT
1
0
2
MESERP
Make simulated
measurements and
write data tape.
Stop

```
$IBFTC MC13AA  XR3,M94,NODD,LIST
C     TDS - TRACKING DATA SIMULATOR                                     TDS 0001
C                                                                       TDS 0002
C     DEVELOPED BY  PHILCO-FORD CORPORATION                             TDS 0003
C               FOR GODDARD SPACE FLIGHT CENTER                         TDS 0004
C               CONTRACT NAS5-9939                                      TDS 0005
C                                                                       TDS 0006
C     REFERENCE TR-DA1508, PROGRAM DESCRIPTION AND THEORETICAL BASIS,  TDS 0007
C                          ORBIT DETERMINATION PROGRAM                  TDS 0008
C               TR-DA1509, SUBROUTINE DESCRIPTIONS AND LISTINGS,        TDS 0009
C                          ORBIT DETERMINATION PROGRAM                  TDS 0010
C               TR-DA1510, INPUT-OUTPUT SUMMARY,                        TDS 0011
C                          ORBIT DETERMINATION PROGRAM                  TDS 0012
C                                                                       TDS 0013
C     DECEMBER 1967                                                     TDS 0014
C                                                                       TDS 0015
      COMMON  /INPCOM/C(700)                                            TDS 0016
   10 READ(5,700) ISWT                                                  TDS 0017
  700 FORMAT(I5)                                                        TDS 0018
      IF(ISWT.EQ.0) GO TO 810                                           TDS 0019
      GO TO (100,200),ISWT                                              TDS 0020
  100 CONTINUE                                                          TDS 0021
C     THIS SPACE RESERVED FOR COMPUTING AND WRITING TRAJECTORY ON       TDS 0022
C     TAPE UNIT 10                                                      TDS 0023
      GO TO 10                                                          TDS 0024
  200 CONTINUE                                                          TDS 0025
C     COMPUTE AND WRITE MEASUREMENTS ON TAPE UNIT 12                    TDS 0026
      CALL MESERP                                                       TDS 0027
      GO TO 10                                                          TDS 0028
  810 CONTINUE                                                          TDS 0029
      REWIND 8                                                          TDS 0030
      REWIND 10                                                         TDS 0031
      REWIND 12                                                         TDS 0032
      STOP                                                              TDS 0033
      END
```

Subroutine: TFRAC

Purpose: Supplies integral and fractional parts of a sum. Used principally to update the internal date format.

Calling Sequence: CALL TFRAC (TWI,TFI,TWØ,TFØ)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TWI			days	Input whole days since 1950
I	TFI			days	Input fractional days (may be $\geq 1.$)
Ø	TWØ			days	Integer part of TWI+TFI
Ø	TFØ			days	Fractional part of TWI+TFI

Common storages used: None

Subroutines required: None

```
$IBFTC MC13U1  XR3,M94,NODD,LIST
      SUBROUTINE TFRAC (TWI,TFI,TWO,TFO)
      TSV = TWI
    1 TWO = AINT(TSV+TFI)
      TFO = (TSV-TWO)+TFI
      RETURN
      END
```

Subroutine: TIMED

Purpose: Converts time interval from time format to seconds.

Calling Sequence: CALL TIMED (TM,S)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TM				Input interval in time format (See TIMES)
Ø	S			seconds	Interval TM in seconds

Restrictions: If TM < 0, S is set equal to |TM| and no format conversion takes place. If conversion occurs, S is rounded to the nearest second.

Common storages used: None

Subroutines required: None

```
      SUBROUTINE TIMED (D,S)
    1 T1 = D
      IF (T1) 2,2,3
    2 S = -T1
      RETURN
    3 T2 = AINT(T1)
      T1 = 100.*(T1-T2)
      IF (T1.LT.60.) GO TO 4
      T2 = T2+1.
      T1 = 0.
    4 T3 = AINT(T1)
      T1 = 100.*(T1-T3)
      IF (T1.LT.60.) GO TO 5
      T3 = T3+1.
      T1 = 0.
    5 T2 = T2-76.*AINT(T2/100.)
      S = (T2*60.+T3)*60.+T1
      RETURN
      END
```

Subroutine: TIMES

Purpose: Converts time interval from seconds to time format, and prepares a six-word BCD array for simplified output of interval in days, hours, minutes, and seconds.

Calling Sequence: CALL TIMES (S,TM,A)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	S			seconds	Input time interval
Ø	TM				Interval S in time format: (Days)x100 + (Hours) + (Min)/100 + (Sec)/10000
Ø	A	(6)			BCD image of TM in the form bxxxxbDAYbxxbHRSbxxbMINbxx.xxxbSECbb where b = blank, x = numerical digit.

Common storages used: 8 cells

Subroutines required: BIBCD

```
$IBFTC MC13U4  XR3,M94,NODD,LIST
      SUBROUTINE TIMES (S,D,I)
      COMMON     C(2),N(6)
      DIMENSION  I(6),J(12),K(5)
      INTEGER    BIBCD
      DATA  K   /60,60,24,0010000000000,6H
     1    /,J(6)/42H SEC  DAY 00 HRS 00 MIN 001000  .0000 0000/
C
    1 C(1) = S
      N(1) = C(1)
      C(2) = C(1)-FLOAT(N(1))
      DO 2 L=1,3
      N(L+1) = N(L)/K(L)
    2 N(L) = N(L)-K(L)*N(L+1)
      N(5) = C(2)*1000.
      N(6) = N(4)
      DO 3 L=1,3
      M = 4-L
    3 N(6) = 100*N(6)+N(M)
      D = (C(2)+FLOAT(N(6)))/10000.
C
      N(6) = N(5)+10000*N(1)+2000
      J(5) = BIBCD(N(6))
      J(5) = J(5)+ISIGN(J(10),J(5))
      IF (J(5).LE.J(11)) J(5) = J(5)+J(12)
      N(6) = N(2)/10
      J(4) = J(9)+K(4)*(N(2)-10*N(6))
      IF (N(6).EQ.0) N(6) = 48
      J(3) = J(8)-N(6)
      J(2) = N(3)/10
      N(3) = N(3)-10*J(2)
      IF(J(2).EQ.0) J(2) = 48
      J(2) = 64*J(2)+N(3)+J(7)
      N(6) = 10*N(4)
      J(1) = BIBCD(N(6))-48
      IF (J(1).EQ.K(5)) J(1) = J(1)+3072
      DO 4 L=1,6
    4 I(L) = J(L)
      RETURN
      END
```

Subroutine: TRAJD

Purpose: Controls the integration of the equations of motion for the DCP residual output link.

Calling Sequence: CALL TRAJD

Common storages used: //24 cells, /DCPCØM/,/DFMCØM/,/DQ3CØM/,/EDTCØM/, /ESRCØM/,/SB3CØM/,/TRJCØM/

Subroutines required: ACCTR3, DDØT, DEQ3, DNØRM, DPFMRS, ØUTTR3, STEPDI, STEPDT

Description

TRAJD loads the necessary constants and controls for the integration of the equations of motion and the accumulation of interpolation tables for the DCP residual output link. It initializes the integration tables, shifts central bodies, and rectifies the Encke integration as required. The operation of TRAJD parallels that of TRAJDP (q.v.), but omits those operations concerned with the variational equations.

```
$IBFTC MC133G  XR3,M94,NODD,LIST
      SUBROUTINE TRAJD                                                  TRAJ0001
C     INTEGRATION CONTROL ROUTINE FOR DCP RESIDUAL LINK                 TRAJ0002
C                                                                       TRAJ0003
      EXTERNAL            ACCTR3,OUTTR3                                 TRAJ0004
      DOUBLE PRECISION    DNORM,DSQRT                                   TRAJ0005
C                                                                       TRAJ0006
      COMMON             /DCPCOM/CDCP(900)                              TRAJ0007
      DIMENSION           CBODY(8,11) ,IFEMP  (8) ,NCB  (3,2)           TRAJ0008
     1                   ,ETAPE    (4) ,IFPLNT(11) ,STIMNX(20)          TRAJ0009
      DOUBLE PRECISION    CBRAN2 ,SPDLT ,STIMR(2)                       TRAJ0010
      EQUIVALENCE                (CDCP(111),IERR  ) ,(CDCP(779),KCB    ) TRAJ0011
     1                          ,(CDCP(687),IFEMP ) ,(CDCP(771),NCB    ) TRAJ0012
     2     ,(CDCP( 17),CBODY ) ,(CDCP(663),IFPLNT) ,(CDCP(677),NRSOUT)  TRAJ0013
     3     ,(CDCP(899),CBRAN2) ,(CDCP(758),INDSTA) ,(CDCP( 15),SPDLT )  TRAJ0014
     4     ,(CDCP(112),ETAPE ) ,(CDCP(674),ITARGT) ,(CDCP(123),STIMNX)  TRAJ0015
     5     ,(CDCP(777),ICB    ) ,(CDCP(778),JCB    ) ,(CDCP(781),STIMR ) TRAJ0016
      EQUIVALENCE        (IFEMP( 3),IFATMD) ,(IFEMP( 1),IFGRAV)         TRAJ0017
     1                  ,(IFEMP( 8),IFBODY) ,(IFEMP( 6),IFHARM)         TRAJ0018
     2                  ,(IFEMP( 7),IFDRAG)                             TRAJ0019
C                                                                       TRAJ0020
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)                TRAJ0021
      DOUBLE PRECISION    ONTIME                                        TRAJ0022
      EQUIVALENCE        (INDDAT(13),ONTIME)                            TRAJ0023
C                                                                       TRAJ0024
      COMMON             /ESRCOM/CESR(304)                              TRAJ0025
      DOUBLE PRECISION    DELDAR (2) ,EHADAR(24,3) ,SER(14,2)           TRAJ0026
     1                   ,EFEDAR(14) ,ETIMVA        ,SPCDAR(6)          TRAJ0027
      EQUIVALENCE                (CESR(101),EHADAR) ,(CESR(  2),NBY    ) TRAJ0028
     1                          ,(CESR( 29),ETIMVA) ,(CESR(245),SER    ) TRAJ0029
     2     ,(CESR( 97),DELDAR) ,(CESR(  3),NBD    ) ,(CESR( 49),SPCDAR) TRAJ0030
     3     ,(CESR( 61),EFEDAR) ,(CESR(  4),NBX    )                     TRAJ0031
C                                                                       TRAJ0032
      COMMON             /SB3COM/CSBF(12),RBF(6),VBF(6,8)               TRAJ0033
      DOUBLE PRECISION    RBF ,VBF ,TBF                                 TRAJ0034
      EQUIVALENCE        (CSBF( 1),KBF    ) ,(CSBF( 2),NTP    )         TRAJ0035
     1                  ,(CSBF( 5),LCB    ) ,(CSBF( 9),TBF    )         TRAJ0036
C                                                                       TRAJ0037
      COMMON             /TRJCOM/CTRJ(246)                              TRAJ0038
      DOUBLE PRECISION    CBOD(37) ,RC(3) ,ZH(24)                       TRAJ0039
     1                   ,CBU        ,VC(3) ,ETFMS                      TRAJ0040
      EQUIVALENCE                (CTRJ( 99),NZH    ) ,(CTRJ(151),RC     ) TRAJ0041
     1     ,(CTRJ( 77),CBOD   ) ,(CTRJ(101),NTH    ) ,(CTRJ(157),VC     ) TRAJ0042
     2     ,(CTRJ( 85),CBU    ) ,(CTRJ(  1),ETFMS ) ,(CTRJ(103),ZH     ) TRAJ0043
C                                                                       TRAJ0044
      COMMON             /DQ3COM/NDEQ(10),CDEQ(10),ETIMV(4)             TRAJ0045
     1                          ,ADEQ (8),RDEQ (8),VDEQ (8),FDEQ(8,10)  TRAJ0046
      DOUBLE PRECISION    ETIMV,ADEQ,RDEQ,VDEQ,FDEQ                     TRAJ0047
C                                                                       TRAJ0048
      COMMON             /DFMCOM/IFM(14),RFM(6,12)                      TRAJ0049
     1                          ,DFM (4),BFM (577),SNT(2,102)           TRAJ0050
      DOUBLE PRECISION    BFM,DFM,RFM                                   TRAJ0051
C                                                                       TRAJ0052
      COMMON              SAVE(12),D(6)                                 TRAJ0053
      DOUBLE PRECISION    D                                             TRAJ0054
C                                                                       TRAJ0055
      DIMENSION           ND(5,2) ,NN(3)                                TRAJ0056
      DOUBLE PRECISION    ETIMR1 ,OBL50                                 TRAJ0057
      DATA                ND/1,1,3,1,2,0,1,3*0/                         TRAJ0058
      DATA                NN/11,3,0/                                    TRAJ0059
      DATA                OBL50/0.4092061925430292/                     TRAJ0060
C                                                                       TRAJ0061
  601 FORMAT(12HOPATCH FROM ,A6,4H TO ,A6)                              TRAJ0062
C                                                                       TRAJ0063
C     INITIALIZATION                                                    TRAJ0064
C                                                                       TRAJ0065
    1 ICB = NBY                                                         TRAJ0066
      LCB = NBY-100*(NBY/100)                                           TRAJ0067
      IF (ICB.NE.LCB)     ICB = 3                                       TRAJ0068
      JCB = 3                                                           TRAJ0069
      IF (LCB.EQ. 3)      JCB = 1                                       TRAJ0070
      IF (LCB.EQ.11)      JCB = 2                                       TRAJ0071
      NN(3) = ITARGT                                                    TRAJ0072
      KCB = NN(JCB)                                                     TRAJ0073
      DO 2 I=1,10                                                       TRAJ0074
```

```
      IFM(I) = 0                                                        TRAJ0075
    2 IF(IFPLNT(I).NE.0) IFM(I) = 1                                     TRAJ0076
      IFM( 3) = 2                                                       TRAJ0077
      IFM(11) = 2                                                       TRAJ0078
      IFM(12) = 2                                                       TRAJ0079
      IFM(13) = 1                                                       TRAJ0080
      NDEQ(6) = 3                                                       TRAJ0081
      CDEQ(6) = 1.E20                                                   TRAJ0082
      CDEQ(7) = 1.E-5                                                   TRAJ0083
      ETIMV(1) = ETIMVA                                                 TRAJ0084
      ETIMV(2) = 1.D20                                                  TRAJ0085
      ETIMV(3) = 1.D20                                                  TRAJ0086
      DO 3 I=1,6                                                        TRAJ0087
    3 RC(I) = SPCDAR(I)                                                 TRAJ0088
    6 IF (STIMR(1).EQ.0.D0)   STIMR(1) = STIMNX(INDSTA)                 TRAJ0089
      IF (STIMR(1).LT.ONTIME) STIMR(1) = ONTIME                         TRAJ0090
      D(1)   = STIMR(1)+SER(4,1)+SER(5,1)*STIMR(1)                      TRAJ0091
      ETIMR1 = D(1)+DELDAR(1)+DELDAR(2)*D(1)                            TRAJ0092
C                                                                       TRAJ0093
C     PATCH TO CENTRAL BODY                                             TRAJ0094
C                                                                       TRAJ0095
  100 IFM(LCB) = 2                                                      TRAJ0096
      CALL DPEMRS (ETIMV,EFEDAR,LCB,IERR,ETAPE)                         TRAJ0097
      IF (IERR.NE.0)      GO TO 998                                     TRAJ0098
      DO 101 I=1,3                                                      TRAJ0099
      RC(I) = RC(I)+REM(I,ICB)                                          TRAJ0100
  101 VC(I) = VC(I)+REM(I+3,ICB)                                        TRAJ0101
      IF (ICB.NE.LCB)     IFM(ICB) = 1                                  TRAJ0102
      IFM( 3) = 2                                                       TRAJ0103
      IFM(11) = 2                                                       TRAJ0104
      ICB = LCB                                                         TRAJ0105
C                                                                       TRAJ0106
C     LOAD CENTRAL BODY WORKING STORAGE                                 TRAJ0107
C                                                                       TRAJ0108
  110 DO 111 I=1,4                                                      TRAJ0109
  111 CBOD(I) = CBODY(I,ICB)                                            TRAJ0110
      DO 112 I=1,2                                                      TRAJ0111
      CBOD(I+5) = CBODY(I+4,ICB)                                        TRAJ0112
  112 CBOD(I+8) = CBODY(I+6,ICB)                                        TRAJ0113
      CBOD( 5) = EFEDAR(ICB)                                            TRAJ0114
      CBOD(11) = OBL50                                                  TRAJ0115
      NZH = NCB(JCB,1)                                                  TRAJ0116
      NTH = NCB(JCB,2)                                                  TRAJ0117
      DO 113 I=1,24                                                     TRAJ0118
  113 ZH(I) = FHADAR(I,JCB)                                             TRAJ0119
C                                                                       TRAJ0120
C     SET CENTRAL BODY SWITCHES                                         TRAJ0121
C                                                                       TRAJ0122
  120 IFHARM = 0                                                        TRAJ0123
      IF (IFGRAV.EQ.0)    GO TO 122                                     TRAJ0124
      IF (NBX.NE.ICB.AND.JCB.EQ.3) GO TO 122                            TRAJ0125
      IF (NZH.NE.0)       IFHARM = 4                                    TRAJ0126
      IF (NTH.NE.0)       IFHARM = 5                                    TRAJ0127
  122 CONTINUE                                                          TRAJ0128
      IFDRAG = 0                                                        TRAJ0129
      IF (IFATMD.EQ.0)    GO TO 123                                     TRAJ0130
      IF (NBD.NE.ICB)     GO TO 123                                     TRAJ0131
      IFDRAG = IFATMD                                                   TRAJ0132
  123 CONTINUE                                                          TRAJ0133
      IFBODY = IFHARM+IFDRAG                                            TRAJ0134
C                                                                       TRAJ0135
C     RECTIFY                                                           TRAJ0136
C                                                                       TRAJ0137
  200 CALL STEPDI (ETIMV,RC,CBU)                                        TRAJ0138
      DO 201 I=1,3                                                      TRAJ0139
      RDEQ(I) = 0.D0                                                    TRAJ0140
  201 VDEQ(I) = 0.D0                                                    TRAJ0141
C                                                                       TRAJ0142
C     ESTIMATE DOWN-LEG DELAY TIME                                      TRAJ0143
C                                                                       TRAJ0144
  210 CONTINUE                                                          TRAJ0145
      IF (KBF.NE.0)       GO TO 220                                     TRAJ0146
      CALL STEPDT (ETIMR1,D)                                            TRAJ0147
      IF (ICB.EQ.3)       GO TO 212                                     TRAJ0148
      DO 211 I=1,3                                                      TRAJ0149
```

```
  211 D(I) = D(I)-RFM(I,3)
  212 VBF(6,1) = DNORM(D)/SPDLT
      TBF = ETIMR1-VBF(6,1)
C
C     SET INTEGRATION INTERVAL
C
  220 CONTINUE
      CBRAN2 = DDOT(RC,RC)
      D(1) = DSQRT(CBRAN2)
      D(2) = DSQRT(D(1)/CBU)
      CDEQ(4) = AINT(D(1)*D(2))/2.
      CDEQ(3) = CDEQ(4)/4.
      CDEQ(5) = CDEQ(3)/8.
      CDEQ(1) = CDEQ(3)
      IF (CDEQ(1).GT.4000.) CDEQ(1) = 4000.
      L = 1
      IF (ABS(TBF-ETIMV)/CDEQ(1).LT.2.) NTP = 3
      IF (NTP.NE.3)       GO TO 222
  221 L = 2
      CDEQ(3) = CDEQ(1)
      CDEQ(4) = CDEQ(1)
  222 CONTINUE
      DO 223 I=1,5
  223 NDEQ(I) = ND(I,L)
C
C     INTEGRATE
C
  300 KBF = 0
      ETFMS = 0.D0
      CALL DEQ3 (ACCTR3,OUTTR3)
      IF (IERR.NE.0)      GO TO 998
      NST = NDEQ(9)
      IF (NST.EQ.0)       GO TO (999,430,999) ,NTP
      NST = NST-3
      GO TO (200,400,410) ,NST
C
C     SET UP STOP/RESTRAT
C
C     PATCH OUT
  400 IF (ICB.NE.11)      GO TO 401
      JCB =  1
      KCB = 11
      LCB =  3
      GO TO 420
  401 JCB =  3
      LCB = 10
      KCB = NN(3)
      GO TO 420
C
C     PATCH IN
  410 LCB = KCB
      JCB =  3
      KCB = 10
      IF (LCB.EQ.11)      JCB = 2
      IF (LCB.NE. 3)      GO TO 420
      JCB =  1
      KCB = 11
C
C     IDENTIFY STOP/RESTART POINT
  420 IF (NRSOUT.EQ.0)    GO TO 100
      WRITE (6,601)       CBODY(1,ICB) ,CBODY(1,LCB)
      GO TO 100
C
C     END-TIME
  430 NTP = 3
      GO TO 100
C
  998 IERR = IERR+15
  999 RETURN
      END
```

Subroutine: TRAJDP

Purpose: Controls the integration of the equations of motion and the variational equations for the DCP propagation/estimation link.

Calling Sequence: CALL TRAJDP

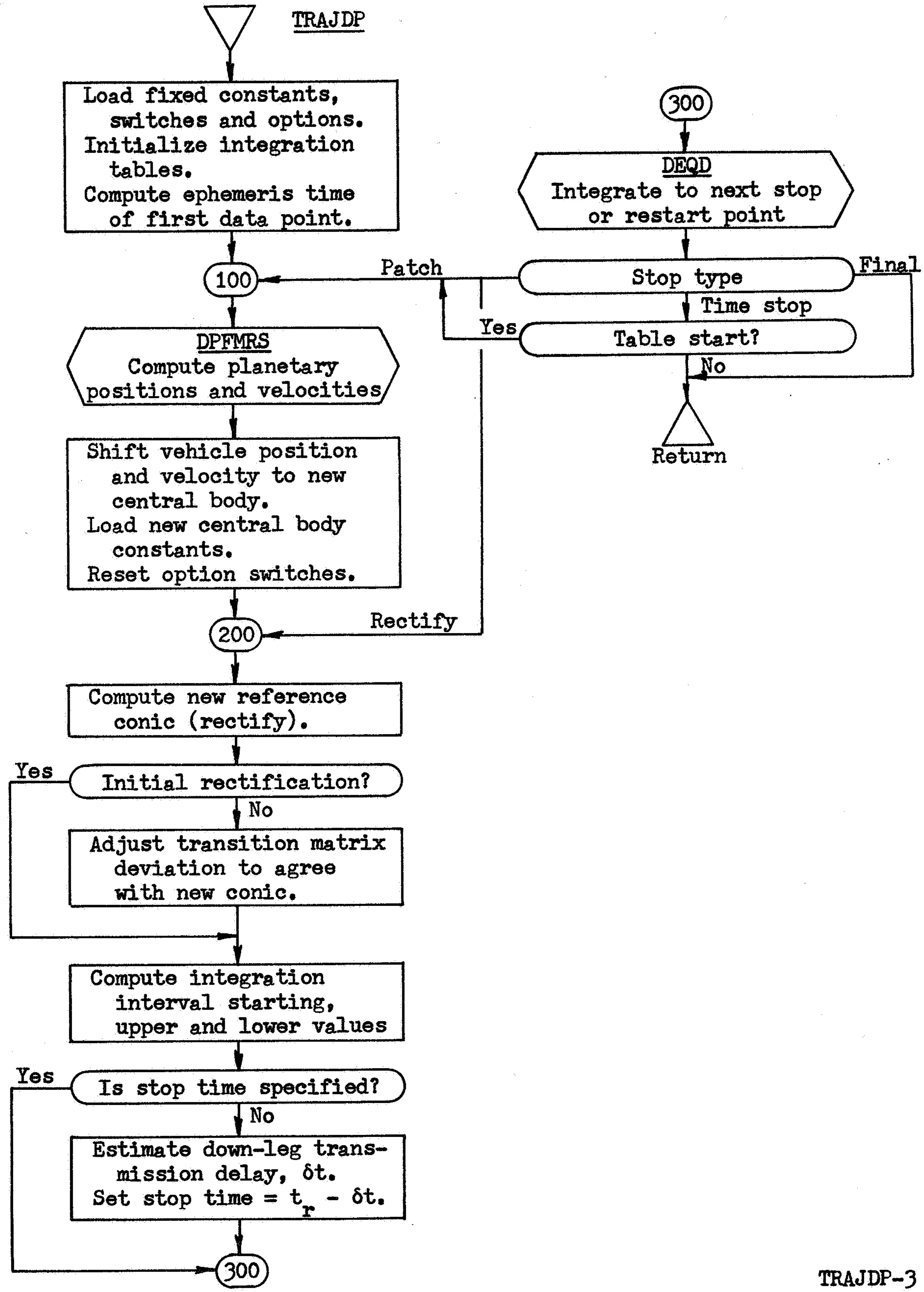
Common storages used: //18 cells, /DCPCØM/,/DFMCØM/,/DQDCØM/,/EDTCØM/, /ESTCØM/,/SBFCØM/,/TRJCØM/

Subroutines required: ACCTRJ, DATINP, DDØT, DEQD, DNØRM, DPFMRS, ØUTTRJ, STEPDI, STEPDP, STEPDT

Description

TRAJDP loads the necessary constants and controls for the integration of the equations of motion and the variational equations and for the accumulation of interpolation tables for the DCP propagation/estimation link. It initializes the integration tables, shifts central bodies, and rectifies the Encke integration as required.

The accelerations and the variational equation derivatives are computed by ACCTRJ using the dynamic model described in Appendix D of Reference 1. The integration and testing for stop and restart points (rectification, etc.) is accomplished by DEQD.


TRAJDP
Load fixed constants, switches and options.
Initialize integration tables.
Compute ephemeris time of first data point.
100
DPFMRS
Compute planetary positions and velocities
Shift vehicle position and velocity to new central body.
Load new central body constants.
Reset option switches.
200
Compute new reference conic (rectify).
Initial rectification?
Yes
No
Adjust transition matrix deviation to agree with new conic.
Compute integration interval starting, upper and lower values
Is stop time specified?
Yes
No
Estimate down-leg transmission delay, δt.
Set stop time = t_r - δt.
300
300
DEQD
Integrate to next stop or restart point
Stop type
Patch
Final
Time stop
Table start?
Yes
No
Return
Rectify

TRAJDP-3

```
$IBFTC MC133F  XR3,M94,NODD,LIST
      SUBROUTINE TRAJDP
C     INTEGRATION CONTROL ROUTINE FOR DCP ESTIMATION/PROPAGATION
C
      EXTERNAL            ACCTRJ,OUTTRJ
      DOUBLE PRECISION    DNORM,DSQRT
C
      COMMON             /DCPCOM/CDCP(900)
      DIMENSION           CBODY(8,11) ,IFEMP  (8) ,NCB  (3,2)
     1                   ,ETAPE   (4) ,IFPLNT(11) ,STIMNX(20)
      DOUBLE PRECISION    CBRAN2     ,ETIMV1 ,RSTOP ,SPDLT
     1                   ,DATEV1(3)          ,RSTP2 ,STIMR(2)
      EQUIVALENCE                (CDCP(687),IFEMP ) ,(CDCP(676),NPROUT)
     1                          ,(CDCP(663),IFPLNT) ,(CDCP(741),NPSTOP)
     2    ,(CDCP( 17),CBODY ) ,(CDCP(758),INDSTA) ,(CDCP(742),NTARGT)
     3    ,(CDCP(899),CBRAN2) ,(CDCP(756),IPROCS) ,(CDCP(897),RSTP2 )
     4    ,(CDCP(801),DATEV1) ,(CDCP(674),ITARGT) ,(CDCP(807),RSTOP )
     5    ,(CDCP(112),ETAPE ) ,(CDCP(778),JCB   ) ,(CDCP( 15),SPDLT )
     6    ,(CDCP(785),ETIMV1) ,(CDCP(779),KCB   ) ,(CDCP(123),STIMNX)
     7    ,(CDCP(777),ICB   ) ,(CDCP(771),NCB   ) ,(CDCP(781),STIMR )
     8    ,(CDCP(111),IERR  ) ,(CDCP(700),NPEND )
      EQUIVALENCE        (IFEMP( 3),IFATMD) ,(IFEMP( 1),IFGRAV)
     1                  ,(IFEMP( 8),IFBODY) ,(IFEMP( 6),IFHARM)
     2                  ,(IFEMP( 7),IFDRAG)
C
      COMMON             /EDTCOM/INDDAT(40),BUFDAT(85,6)
      DOUBLE PRECISION    ONTIME
      EQUIVALENCE        (INDDAT(13),ONTIME)
C
      COMMON             /ESTCOM/CEST(804)
      DOUBLE PRECISION    DELDAN (2) ,EHADAN(24,3) ,SE(14,20)
     1                   ,EFEDAN(14) ,ETIMVA       ,SPCDAN(6)
      EQUIVALENCE                (CEST(101),EHADAN) ,(CEST(  2),NBY    )
     1                          ,(CEST( 29),ETIMVA) ,(CEST(245),SE     )
     2    ,(CEST( 97),DELDAN) ,(CEST(  3),NBD    ) ,(CEST( 49),SPCDAN)
     3    ,(CEST( 61),EFEDAN) ,(CEST(  4),NBX    )
C
      COMMON             /SBFCOM/CSBF(12),RBF(45),VBF(45,8)
      DOUBLE PRECISION    RBF ,VBF ,TBF ,PHI(6,6)
      EQUIVALENCE        (VBF(1,2),PHI)
      EQUIVALENCE        (CSBF( 1),KBF    ) ,(CSBF( 2),NTP    )
     1                  ,(CSBF( 5),LCB    ) ,(CSBF( 9),TBF    )
C
      COMMON             /TRJCOM/CTRJ(246)
      DOUBLE PRECISION    CBOD(37) ,RC(3) ,ZH(24) ,PCONIC(6,6)
     1                   ,CBU       ,VC(3) ,ETFMS  ,RCONIC  (6)
      EQUIVALENCE                (CTRJ( 99),NZH    ) ,(CTRJ(151),RC     )
     1    ,(CTRJ( 77),CBOD   ) ,(CTRJ(101),NTH    ) ,(CTRJ(157),VC     )
     2    ,(CTRJ( 85),CBU    ) ,(CTRJ(175),PCONIC) ,(CTRJ(103),ZH     )
     3    ,(CTRJ(  1),ETFMS  ) ,(CTRJ(163),RCONIC)
C
      COMMON             /DQDCOM/NDEQ(10),CDEQ(10),ETIMV(4)
     1                           ,ADEQ(44),RDEQ(44),VDEQ(44),FDEQ(44,10)
      DOUBLE PRECISION    ETIMV,ADEQ,RDEQ,VDEQ,FDEQ
C
      COMMON             /DFMCOM/IFM(14),RFM(6,12)
     1                           ,DFM (4),BFM (577),SNT(2,102)
      DOUBLE PRECISION    BFM,DFM,RFM
C
      COMMON              SAVE(12),S(6)
      DOUBLE PRECISION    D(3)
      EQUIVALENCE        (S,D)
C
      DIMENSION           ND(5,2) ,NN(3)
      DOUBLE PRECISION    ETIMR1 ,OBL50
      DATA                ND/1,1,3,1,2,0,1,3*0/
      DATA                NN/11,3,0/
      DATA                OBL50/0.4092061925430292/
C
  601 FORMAT(12H0PATCH FROM ,A6,4H TO ,A6)
  602 FORMAT(21H0TRAJECTORY END POINT)
C
C     INITIALIZATION
C
    1 ICB = NBY
```

```
      ITP = NBY/100
      LCB = NBY-100*ITP
      IF (ICB.NE.LCB)     ICB = 3
      JCB = 3
      IF (LCB.EQ. 3)      JCB = 1
      IF (LCB.EQ.11)      JCB = 2
      NN(3) = ITARGT
      IF (IPROCS.EQ.1.AND.NTARGT.NE.0)  NN(3) = NTARGT
      KCB = NN(JCB)
      DO 2 I=1,10
      IFM(I) = 0
    2 IF(IFPLNT(I).NE.0) IFM(I) = 1
      IFM( 3) = 2
      IFM(11) = 2
      IFM(12) = 2
      IFM(13) = 1
      NDEQ(6) = NPEND-1
      NDEQ(8) = 0
      CDEQ(6) = 1.E20
      CDEQ(7) = 1.E-5
      ETIMV(1) = ETIMVA
      ETIMV(2) = 1.D20
      ETIMV(3) = 1.D20
      IF (IPROCS+NPSTOP.EQ.1) ETIMV(2) = STIMR(2)
      DO 3 I=1,6
      RBF(I+42)    = 0.D0
      VBF(I+42,1) = 0.D0
    3 RC(I) = SPCDAN(I)
      DO 4 I=5,NPEND
      RDEQ(I-1) = 0.D0
    4 VDEQ(I-1) = 0.D0
      IF (IPROCS.EQ.1)    GO TO 100
      IF (KBF.NE.0)       GO TO 6
      DO 5 I=1,3
    5 S(I) = DATEV1(I)
      CALL DATINP (S,ETIMV1)
      IF (ETIMV1.EQ.ETIMVA) GO TO 6
      NTP = 1
      ETIMV(2) = ETIMV1
      GO TO 100
    6 IF (STIMR(1).EQ.0.D0)    STIMR(1) = STIMNX(INDSTA)
      IF (STIMR(1).LT.ONTIME) STIMR(1) = ONTIME
      D(1)    = STIMR(1)+SE(4,INDSTA)+SE(5,INDSTA)*STIMR(1)
      ETIMR1 = D(1)+DELDAN(1)+DELDAN(2)*D(1)
C
C     PATCH TO CENTRAL BODY
C
  100 IFM(LCB) = 2
      CALL DPFMRS (ETIMV,EFEDAN,LCB,IERR,ETAPE)
      IF (IERR.NE.0)      GO TO 998
      DO 101 I=1,3
      RC(I) = RC(I)+RFM(I,ICB)
  101 VC(I) = VC(I)+RFM(I+3,ICB)
      IF (ICB.EQ.LCB)     GO TO 103
      IFM(ICB) = 1
      IF (ITP.EQ.0)       GO TO 103
      DO 102 I=1,3
      RBF(I+42)    = RFM(I   ,3)
  102 VBF(I+42,1) = RFM(I+3,3)
  103 IFM( 3) = 2
      IFM(11) = 2
      ICB = LCB
C
C     LOAD CENTRAL BODY WORKING STORAGE
C
  110 DO 111 I=1,4
  111 CBOD(I) = CBODY(I,ICB)
      DO 112 I=1,2
      CBOD(I+5) = CBODY(I+4,ICB)
  112 CBOD(I+8) = CBODY(I+6,ICB)
      CBOD( 5) = EFEDAN(ICB)
      CBOD(11) = DBL50
      NZH = NCB(JCB,1)
      NTH = NCB(JCB,2)
      DO 113 I=1,24
```

```
  113 ZH(I) = EHADAN(I,JCB)
C
C     SET CENTRAL BODY SWITCHES
C
  120 IFHARM = 0
      IF (IFGRAV.EQ.0)    GO TO 122
      IF (NBX.NE.ICB.AND.JCB.EQ.3) GO TO 122
      IF (NZH.NE.0)       IFHARM =-4
      IF (NTH.NE.0)       IFHARM =-5
  122 CONTINUE
      IFDRAG = 0
      IF (IFATMD.EQ.0)    GO TO 123
      IF (NBD.NE.ICB)     GO TO 123
      IFDRAG = IFATMD
  123 CONTINUE
      IFBODY = IFHARM+IFDRAG
      RSTP2 = 0.D0
      ND(3,1) = 3
      IF (IPROCS.NE.1)    GO TO 126
      IF (NPSTOP-2)       126,125,124
  124 ND(3,1) = 5
      GO TO 126
  125 ND(3,1) = 4
      RSTP2 = RSTOP*RSTOP
  126 CONTINUE
      IF (NPEND.LE.22)    GO TO 200
      DO 127 I=23,NPEND
  127 ADEQ(I-1) = 0.D0
C
C     RECTIFY
C
  200 CALL STEPDI (ETIMV,RC,CBU)
      DO 201 I=1,3
      RDEQ(I) = 0.D0
  201 VDEQ(I) = 0.D0
      IF (NDEQ(8).EQ.0)  GO TO 210
      CALL STEPDT (ETIMVA,RCONIC)
      CALL STEPDI (ETIMVA,RCONIC,CBU)
      CALL STEPDT (ETIMV,RCONIC)
      L = 4
      DO 202 J=1,6
      DO 202 I=1,3
      PHI(I  ,J) = PCONIC(I  ,J)+RDEQ(L)
      PHI(I+3,J) = PCONIC(I+3,J)+VDEQ(L)
  202 L = L+1
      CALL STEPDP (PCONIC)
      L = 4
      DO 203 J=1,6
      DO 203 I=1,3
      RDEQ(L) = PHI(I  ,J)-PCONIC(I  ,J)
      VDEQ(L) = PHI(I+3,J)-PCONIC(I+3,J)
  203 L = L+1
C
C     SET INTEGRATION INTERVAL
C
  210 CONTINUE
      CBRAN2 = DDOT(RC,RC)
      D(1) = DSQRT(CBRAN2)
      D(2) = DSQRT(D(1)/CBU)
      CDEQ(4) = AINT(D(1)*D(2))/2.
      CDEQ(3) = CDEQ(4)/4.
      CDEQ(5) = CDEQ(3)/8.
      CDEQ(1) = CDEQ(3)
      L = 1
C
C     ESTIMATE DOWN-LEG DELAY TIME
C
  220 CONTINUE
      IF (NTP.EQ.1)       GO TO 224
      IF (KBF.NE.0)       GO TO 223
      CALL STEPDT (ETIMR1,D)
      IF (ICB.EQ.3)       GO TO 222
      DO 221 I=1,3
  221 D(I) = D(I)-RFM(I,3)
  222 VBF(NPEND+2,1) = DNORM(D)/SPDLT
```

```
      TBF = ETIMR1-VBF(NPEND+2,1)
  223 IF (CDEQ(1).GT.4000.) CDEQ(1) = 4000.
      IF (ABS(TBF-ETIMV)/CDEQ(1).LT.2.) NTP = 3
      IF (NTP.NE.3)       GO TO 224
      L = 2
      CDEQ(3) = CDEQ(1)
      CDEQ(4) = CDEQ(1)
  224 DO 225 I=1,5
  225 NDEQ(I) = ND(I,L)
C
C     INTEGRATE
C
  300 KBF = 0
      ETFMS = 0.D0
      CALL DEQD (ACCTRJ,OUTTRJ)
      IF (IERR.NE.0)      GO TO 998
      NST = NDEQ(9)
      IF (NST.EQ.0)       GO TO (435,430,999) ,NTP
      NST = NST-NPEND+1
      GO TO (200,400,410,425,425) ,NST
C
C     SET UP STOP/RESTRAT
C
C     PATCH OUT
  400 IF (ICB.NE.11)      GO TO 401
      JCB =  1
      KCB = 11
      LCB =  3
      GO TO 420
  401 JCB =  3
      LCB = 10
      KCB = NN(3)
      GO TO 420
C
C     PATCH IN
  410 LCB = KCB
      JCB =  3
      KCB = 10
      IF (LCB.EQ.11)      JCB = 2
      IF (LCB.NE. 3)      GO TO 420
      JCB =  1
      KCB = 11
C
C     IDENTIFY STOP/RESTART POINT
  420 IF (NPROUT.EQ.0)    GO TO 421
      WRITE (6,601)       CBODY(1,ICB) ,CBODY(1,LCB)
  421 IF (IPROCS.NE.1)    GO TO 100
      IF (NPSTOP.NE.1)    GO TO 100
      IF (LCB.EQ.NN(3))   GO TO 425
      IF (ICB.NE.NN(3))   GO TO 100
  425 IF (NPROUT.NE.0)    WRITE (6,602)
      GO TO 999
C
C     END-TIME
  430 NTP = 3
      GO TO 100
  435 IF (IPROCS.EQ.1)    GO TO 425
      NTP = 2
      GO TO 999
C
  998 IERR = IERR+15
  999 RETURN
      END
```

Subroutine: UPDATP

Purpose: Updates the covariance matrix in time.

Calling Sequence: CALL UPDATP

Common storages used: //72 cells, /DCPCØM/,/ESTCØM/,/ES1CØM/,/SBFCØM/

Subroutines required: DMPLY

Method:

The transition matrix and sensitivities to equation of motion parameters are computed for C-frame components of the vehicle state in the form

$$\Phi(t;t_o) = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_u \\ \dot{\varphi}_1 & \dot{\varphi}_2 & \dot{\varphi}_u \end{bmatrix}$$

where

$$\begin{bmatrix} X_C(t) \\ \dot{X}_C(t) \end{bmatrix} = \Phi(t;t_o) \begin{bmatrix} X(t_o) \\ \dot{X}(t_o) \\ u \end{bmatrix}$$

X_C = deviation in vehicle position with respect to Earth, C-frame components,

$X(t_o)$ = deviation in position at the anchor point time, t_o, in the coordinates of SPCDAN.

Let T_{X2Z} be an orthogonal transformation

$$X = T_{X2Z} X_C$$

and let Z be the deviation in the extended state vector

$$
Z(t) = \begin{bmatrix} T_{X2Z} X_C \\ T_{X2Z} \dot{X}_C \\ u \\ v \end{bmatrix}
$$

The extended state vector obeys

$$
Z = \Phi(t;t_o)Z(t_o)
$$

$$
\Phi = \begin{bmatrix} \varphi'(t;t_o) & & \phi \\ \phi & I & \phi \\ \phi & \phi & I \end{bmatrix}
$$

and the covariance matrix

$$
P(t) = \Phi(t;t_o)P(t_o)\Phi^T(t;t_o)
$$

where

$$
\varphi'(t;t_o) = \begin{bmatrix} T_{X2Z} & \phi \\ \phi & T_{X2Z} \end{bmatrix} \varphi(t;t_0)
$$

Usage:

The subroutine UPDATP is called at two logical points, identified by NTP. In each case, the 3 × N matrices $[\varphi_1, \varphi_2, \varphi_u]$ and $[\dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3]$ are stored in RBF(4) to RBF(3N+3) and VBF(4,1) to VBF(3N+3,1), respectively.

The two points are:

NTP = 1 : End-point of progagation.
$P(t_o)$ is read **from** unit 09, and P(t) is required for output on unit 06, only.

NTP = 2 : Anchor-point for data processing has been **reached.**
$P(t_o)$ is read from unit 09, and P(t) is required for output on unit 06 and as the a priori covariance for data processing.

```
$IBFTC MC138A  XR3,M94,NODD,LIST
      SUBROUTINE UPDATP
C     UPDATES THE COVARIANCE MATRIX IN TIME
C
      COMMON            /DCPCOM/CDCP(900)
      EQUIVALENCE       (CDCP(105),MAXZ   ) ,(CDCP(740),NPREST)
     1                  ,(CDCP(702),NEMPS  )
C
      COMMON            /ESTCOM/CEST(804)
      EQUIVALENCE       (CEST( 27),KOUNTN) ,(CEST( 31),TX2Z   )
C
      COMMON            /ES1COM/PNEW(30,30),STNAMN(30),TRAKER(30)
     1                          ,ITRETN (30),ILOCN (30),KLOCN (54)
     2                          ,NSN (12,20)
      DOUBLE PRECISION   PNEW
      DIMENSION          DEST(2214)
      EQUIVALENCE       (PNEW,DEST)
C
      COMMON            /SBFCOM/CSBF(12),RBF(45),VBF(45,8)
      DOUBLE PRECISION   RBF,VBF,PHI(6,12)
      EQUIVALENCE       (VBF(1,2),PHI)
C
      COMMON             D(6,6)
      DOUBLE PRECISION   D
C
C     READ IN OLD COVARIANCE MATRIX
    1 IF (NPREST.EQ.0) GO TO  2
      READ (12) DEST
      IF (NTP.LE.1)     GO TO 10
      WRITE (9) DEST
      GO TO 3
    2 READ  (9) DEST
    3 BACKSPACE 9
C
C     TRANSFORM TRANSITION MATRIX
   10 IF (KLOCN(1).NE.1) GO TO 999
      CALL DMPLY (TX2Z,RBF( 4)   ,PHI(1,1),3,3.3,3,3,6,0)
      CALL DMPLY (TX2Z,RBF(13)   ,PHI(1,4),3,3,3,3,3,6,0)
      CALL DMPLY (TX2Z,VBF( 4,1),PHI(4,1),3,3,3,3,3,6,0)
      CALL DMPLY (TX2Z,VBF(13,1),PHI(4,4),3,3,3,3,3,6,0)
      IF (NEMPS.EQ.0)     GO TO 20
      CALL DMPLY (TX2Z,RBF(22)   ,PHI(1,7),3,3,NEMPS,3,3,6,0)
      CALL DMPLY (TX2Z,VBF(22,1),PHI(4,7),3,3,NEMPS,3,3,6,0)
C
C     UPDATE COVARIANCE MATRIX
   20 L = 6+NEMPS
      CALL DMPLY (PHI,PNEW,D,6,L,6,6,MAXZ,6,0)
      M = KOUNTN-6
      IF (M.GT.0) GO TO 21
      CALL DMPLY (D,PHI,PNEW,6,6,6,6,6,MAXZ,2)
      GO TO 999
   21 CALL DMPLY (PNEW(1,7),PHI,PNEW(7,1),M,L,6,MAXZ,6,MAXZ,3)
      DO 22 I=1,6
      DO 22 J=1,6
   22 PNEW(I,J) = D(I,J)
      DO 23 I=7,KOUNTN
      DO 23 J=1,6
   23 PNEW(J,I) = PNEW(I,J)
      CALL DMPLY (PNEW,PHI,D,6,L,6,MAXZ,6,6,2)
      DO 24 I=1,6
      DO 24 J=I,6
      PNEW(I,J) = D(I,J)
   24 PNEW(J,I) = D(I,J)
C
  999 RETURN
      END
```

Subroutine: VENTD

Purpose: To compute the acceleration due to venting.

Calling Sequences: CALL VENTD (V,T,A)

Input and **Output**

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definitions
I	V	d(3)	V	km/sec	Vehicle velocity.
I	T	d	T/m	km/sec^2	Thrust-to-mass ratio.
IØ	A	d(3)	A	km/sec^2	Acceleration.

Common storages used: None

Subroutines required: DDØT

```
$IBFTC MC132V  XR3,M94,NODD,LIST
      SUBROUTINE VENTD (V,T,A)                                          VENT0001
C     ACCELERATION DUE TO VENTING                                       VENT0002
      DOUBLE PRECISION  A(3),C,T,V(3),DDOT,DSQRT                        VENT0003
C                                                                       VENT0004
    1 C = DDOT(V,V)                                                     VENT0005
      C = DSQRT(C)                                                      VENT0006
      C = T/C                                                           VENT0007
      DO 2 I=1,3                                                        VENT0008
    2 A(I) = A(I)+C*V(I)                                                VENT0009
  999 RETURN                                                            VENT0010
      END
```

Subroutine: VENTDP

Purpose: To compute the acceleration due to venting, its gradient and its partial derivative with respect to the acceleration magnitude.

Calling Sequence: CALL VENTDP (V,T,A,AP,G,N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	V	d(3)	V	km/sec	Vehicle velocity.
I	T	d	T/m	km/sec^2	Thrust to mass ratio.
IØ	A	d(3)	A	km/sec^2	Acceleration.
Ø	AP	d(3)	$\frac{\partial A}{\partial (T/m)}$		
Ø	G	d(3,3)	$\nabla_V A$	1/sec	Velocity gradient of the acceleration.
I	N				Option =0: acceleration only ≠0: $\partial A/\partial (T/m)$ and $\nabla_V A$

Common storages used: 10 cells

Subroutines required: DDØT, DVNØRM

Method

The subroutine computes and adds to the input acceleration a thrust acceleration of constant magnitude in the direction of the vehicle velocity relative to the central body. That is,

$$A = \frac{T}{m} \frac{V}{|V|}$$

where (T/m) is the constant magnitude and V is the velocity vector. The gradient with respect to velocity of this acceleration is computed, if N≠0, from

$$\nabla_V A = \frac{T}{m} \frac{1}{|V|} \left(I - \frac{VV^T}{V^2} \right)$$

and the partial derivative is the unit vector

$$\frac{\partial A}{\partial (T/m)} = \frac{V}{|V|}$$

computed if $N < 0$.

```
$IBFTC MC132W  XR3,M94,NODD,LIST
      SUBROUTINE VENTDP (V,T,A,AP,G,N)
C     ACCELERATION DUE TO VENTING
      DOUBLE PRECISION   A(3),AP(3),G(3,3),T,V(3)
     1                  ,C,DVNORM
      COMMON             C(5)
C
    1 C(4) = DVNORM (V,C)
      IF (N.GE.0) GO TO 3
      DO 2 I=1,3
    2 AP(I) = C(I)
    3 DO 4 I=1,3
    4 A(I) = A(I)+T*C(I)
      IF (N.EQ.0) GO TO 999
      C(4) = T/C(4)
      DO 5 I=1,3
      C(5) =-C(4)*C(I)
      G(I,I) = G(I,I)+C(4)
      DO 5 J=1,3
    5 G(I,J) = G(I,J)+C(5)*C(J)
  999 RETURN
      END
```

Subroutine: VNØRM

Purpose: Computes the magnitude of a 3-vector and normalizes the vector.

Calling Sequence: Z = VNØRM (X,Y)

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	(3)			Input vector
Ø	Y	(3)			Unit vector along X Y = X/Z
Ø	Z				Z = |X|

Common storages used: None

Subroutines required: None

```
$IBFTC VNORM    XR3,M94,NODD
      FUNCTION VNORM(X,Y)
      DIMENSION  X(3) ,Y(3)
      SUM = 0.
      DO 1 I=1,3
    1 SUM = SUM+X(I)*X(I)
      SUM = SQRT(SUM)
      DO 2 I=1,3
    2 Y(I) = X(I)/SUM
      VNORM = SUM
      RETURN
      END
```

Subroutine: VTRN

Purpose: Computes the matrix product of a 3X3 and a 3X1.

Calling Sequence: CALL VTRN(A,B,C)

Input and Output

I/Ø	Symbolic Name or Location	Program Diemsions	Math Symbol	Data Dimensions or Units	Definition
I	A	(3,3)			Multiplier.
I	B	(3)			Input vector.
Ø	C	(3)			AB

Common storages used: None

Subroutines required: None

```
$IBFTC VTRN    XR3,M94,NODD
      SUBROUTINE VTRN(A,B,C)
      DIMENSION  A(3,3) ,B(3) ,C(3)
      DO 1 I=1,3
      C(I) = 0.
      DO 1 K=1,3
    1 C(I) = C(I)+A(I,K)*B(K)
      RETURN
      END
```

Subroutine: VTRT

Purpose: Computes the matrix product of a 3×3 and a 3×1.

Calling Sequence: CALL VTRT(A,B,C)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	(3,3)			Multiplier.
I	B	(3)			Input vector.
Ø	C	(3)			$A^T B$

Common storages used: None

Subroutines required: None

```
$IBFTC VTRT    XR3,M94,NODD
      SUBROUTINE VTRT(A,B,C)
      DIMENSION  A(3,3) ,B(3) ,C(3)
      DO 1 I=1,3
      C(I) = 0.
      DO 1 K=1,3
    1 C(I) = C(I)+A(K,I)*B(K)
      RETURN
      END
```

Subroutine: X2ØRBD

Purpose: To compute orbital elements from Cartesian position and velocity.

Calling Sequence: Call X2ØRDB (XI,XØ,GM)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	XI	d(6)	R,V	km,km/sec	Cartesian position (1-3) and velocity (4-6).
I	GM	d	μ	km^3/sec^2	Central body gravitational constant.
Ø	XØ	d(9)			Orbital elements.

Common storates used: 32 cells.
Subroutines required: DCRØSS,DDØT,DVNØRM

Discussion:

The nine components of the output vector, XØ, are the orbital elements

XØ(1) = a = semi-major axis (km)

XØ(2) = e = eccentricity

XØ(3) = i = inclination (rad), with the xy-plane

XØ(4) = Ω = longitude (rad) of the ascending node on the xy-plane, from the x-axis.

XØ(5) = ω = argument of periapsis (rad)

XØ(6) = θ = true anomaly (rad)

XØ(7) = p = semi latus-rectum (km)

XØ(8) = v_h = hyperbolic excess velocity (km/sec)

XØ(9) = r_p = radius of closest approach (km).

They are computed from the Cartesian position (R) and velocity (V), using the equations below. Note that $a < 0$ when $e > 1$ and $v_h < 0$ when $e < 1$.

$$H = R X V = \mathrm{col}(h_x, h_y, h_z)$$

$$d = R \cdot V/\mu$$

$$a = 1/\left(\frac{2}{r} - \frac{v^2}{\mu}\right)$$

$$e = \sqrt{1 - h^2/\mu a}$$

$$i = \tan^{-1}\left(\sqrt{h_x^2 + h_y^2}/h_z\right)$$

$$\Omega = \tan^{-1}(h_x/(-h_y))$$

$$\omega = \tan^{-1}(zh/(yh_x - xh_y)) - \theta$$

$$\theta = \tan^{-1}(dh/(p - r))$$

$$p = h^2/\mu$$

$$v_h = \sqrt{\mu/|a|}$$

$$r_p = p/(1 + e)$$

```
$IBFTC MC13ZR  XR3,M94,NODD,LIST
      SUBROUTINE X2ORBD (XI,XO,GM)
C     COMPUTES ORBITAL ELEMENTS FROM CARTESIAN STATE
C     XI = INPUT CARTESIAN STATE WRT CENTRAL BODY
C     XO = OUTPUT ORBITAL ELEMENTS
C          1 - SEMI-MAJOR AXIS
C          2 - ECCENTRICITY
C          3 - INCLINATION
C          4 - LONGITUDE OF ASCENDING NODE
C          5 - ARGUMENT OF PERIAPSIS
C          6 - TRUE ANOMALY
C          7 - SEMI-LATUS RECTUM
C          8 - HYPERBOLIC EXCESS VELOCITY
C          9 - RADIUS OF CLOSEST APPROACH
C     GM = GRAVITATIONAL CONSTANT OF CENTRAL BODY
C
      DOUBLE PRECISION   XI(6) ,XO(9) ,GM
     1                   ,TPI    ,EPS
     2                   ,DATAN2,DDOT   ,DMOD   ,DSQRT ,DVNORM
     3                   ,D(16)
      COMMON              D
      DATA TPI,EPS/6.2831853071795864,1.D-13/
C
    1 CONTINUE
      D(3) = GM
      CALL DCROSS (XI,XI(4),D(10))
      D(4) = DVNORM(XI,D(13))
      D(1) = DDOT(XI(4),XI(4))
      D(16) = DDOT(XI,XI(4))
      D(1) = 2.D0/D(4)-D(1)/D(3)
      D(7) = DDOT(D(10),D(10))/D(3)
      D(8) = DSQRT(D(3)*DABS(D(1)))
      IF (D(1).GT.0.D0) D(8) = -D(8)
      D(2) = DSQRT(1.D0-D(7)*D(1))
      D(9) = DSQRT(D(3)*D(7))
      D(5) = 0.D0
      D(6) = 0.D0
      D(6) = DATAN2(D(15)*D(9),(D(14)*D(10)-D(13)*D(11)))
      IF (D(2).LT.EPS) GO TO 5
C     NON-CIRCULAR
      D(5) = D(6)
      D(6) = DATAN2(D(16)*D(9),D(3)*(D(7)-D(4)))
      D(5) = DMOD(D(5)-D(6),TPI)
    5 CONTINUE
      D(3) = DATAN2(DSQRT(D(10)*D(10)+D(11)*D(11)),D(12))
      D(4) = 0.D0
      IF (D(3).LT.EPS) GO TO 6
      D(4) = DATAN2(D(10),-D(11))
    6 CONTINUE
      D(1) = 1.D0/D(1)
      D(9) = D(7)/(1.D0+D(2))
      DO 7 I=1,9
    7 XO(I) = D(I)
      RETURN
      END
```

REFERENCES

1. "Program Description and Theoretical Basis for the Orbit Determination Program," Philco-Ford Corporation, TR-DA1508, Palo Alto, California, December 1967.

2. "Input-Output Summary for the Orbit Determination Program," Philco-Ford Corporation, TR-DA1510, Palo Alto, California, December 1967.