TR-DA1509
December 1967

SUBROUTINE DESCRIPTIONS AND LISTINGS FOR THE ORBIT DETERMINATION PROGRAM VOLUME I

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 I

Contract Number NAS5-9939

Prepared by

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

for

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

PHILCO-FORD CORPORATION

Space & Re-entry Systems Division

FOREWORD

Contract NAS5-9939 covered development of the Orbit Determination Program system described in this report, and several modifications to the Mark II Error Propagation Program developed under an earlier contract. Documentation of the modifications to the Mark II program was completed as additions to the Mark II manuals, and was delivered as each modification was completed. Final documentation of the Orbit Determination Program system is contained in

1.	TR-DA1508	Program Description and Theoretical Basis for the Orbit Determination Program.
2.	TR-DA1509	Subroutine Descriptions and Listings for the Orbit Determination Program.
3.	TR-DA1510	Input - Output Summary for the Orbit Determination Program.

The program development was done in the Systems Development and Mission Analysis Department under the technical direction of R. E. Brown, Engineering Section Supervisor. Major technical contributors were:

W. S. Bjorkman	Senior Engineering Specialist
W. F. Colescott	Project Engineer
D. E. Ekman	Project Engineer
M. J. Brooks	Senior Programmer

ACKNOWLEDGEMENT

The contributions of Dr. Stanley F. Schmidt of Astro Consultants, Inc. are gratefully acknowledged. Dr. Schmidt is principally responsible for the filter design. He was also a major contributor in the definition of program capabilities and in the program conceptional design.

TABLE OF CONTENTS

LIST OF TABLES

SECTION 1

LABELLED COMMONS

1.0 INTRODUCTION

Labelled commons are used by the Orbit Determination Program system for all communication of data between subroutines where common usage is indicated. Blank common is used only for temporary storage within a given subroutine, and never for communication.

The data are assigned to the various commons according to the following rules:

1. Data required only by a set of functionally related subroutines are grouped into a functional common, referenced only by that set of subroutines.

2. Data required only by a given link are preferrably stored in that link.

3. Data involving program dimensions which should be changeable are grouped according to the dimension involved.

This section contains the descriptions and block data listings of all labelled commons used by the ODP, arranged in alphabetical order. The descriptions contain general classifications of the data and detailed descriptions of the individual quantities. Where appropriate, references to paragraph numbers or equation numbers in the referenced reports are given in parenthesis.

/BUFCOM/BUFF1(40), KEY

No block data.

BUFCOM is used as a storage array from which to write the key record of each record pair on the output data tape. It also contains an output tape controlling switch KEY. KEY is not written on the tape.

/BUFCOM/BUFF1(40)

LOC	Symbol	Dimension	Definition
1	NPAIR		Record pair counter
2	NEOT		Beginning and end of tape indicator -1: First record after summary +1: Last record on tape 0: Other
3	NAMSTA		Name of receiving station
4	NMSTA 1		Name of transmitting station. In simulator NAMSTA = NMSTA1
5	NRCD		Arc record counter
6	NPTS		Number of time points in following record
7	KONT		Arc continuation key 1: Record pair is a continuation of previous record pair 0: Record pair initiates a new arc
8	MTYPE		Measurement system type indicator 1: C-Band 2: Goddard Range and Range Rate 3: Unified S-Band 4: DSIF (JPL)
9	NALIGN		Antenna alignment indicator for USBS 1: 30-foot dish, principal axis north 2: 85-foot dish, principal axis east
10	MODE		Doppler mode indicator 2: Two way 3: Three-way

/BUFCOM/BUFF1 (40) (Continued)

LOC	Symbol	Dimension	Definition
11	DELT		Observation period, seconds
12	KTAU		Doppler n-count/t-count indicator 1: Count interval is fixed and found in TAU; data contains doppler counts 0: Doppler count is fixed and is found in TAU; data contains doppler count interval In simulator, KTAU = 1 always
13	ONTIME	d	Arc start time, seconds from 1950
15	TFIRST	d	Record start time, seconds from 1950
17	TLAST	d	Record stop time, seconds from 1950
19	TAU	d	Fixed doppler count interval or fixed count. See KTAU
21	FTR	d	Transmitter carrier frequency
23	C1	d	Refraction constants
25	C2	d	
27	DR	d	Range ambiguity in units of range data
29	BIAS	d	Doppler offset bias
31	RETR	d	Transponder transmit/receive ratio
33	------	d (4)	Not used by simulator

/CROCOM/CCRO (86)

No block data.

CROCOM is used for communication between the polynomial fitting subroutine POLYFT and the simultaneous equation solver CROSIM. It contains the augmented matrix, a tag indicating the number of polynomial coefficients to be solved for, and a singularity flag. Subroutine CROSIM solves the equation AX=C. The augmented B matrix must be supplied as $\left[A \mid C\right]$. The matrix A is destroyed in computation and the vector X is developed in place of C.

/CROCOM/CCRO (86)

LOC	Symbol	Dimension	Description
1	B	d(6, 7)	Augmented matrix. Matrix A must be stored in the first KK rows and columns of B. The vector C must be stored in the (KK+1)th column. The solution X is developed in column KK+1.
85	KK		Number of rows of B to use; number of elements of x to solve for.
86	ISING		Singularity flag, set by CROSIM. Set = 1 for singular A; set = 0 otherwise.

/DATCOM/CDAT (2400)

No block data.

DATCOM is used for communication between the four raw data decoding subroutines (CBTEST, GRTEST, SBTEST, DSTEST) and the polynomial fitting subroutine POLYFT of the TDEP. It contains two time tags, a quality indicator, and the values or four observables for up to 300 time points.

The size of the arrays in DATCOM arbitrarily causes any station pass containing more than 300 time points to be split into two or more station passes. If space permits, these arrays may be made larger by altering the DIMENSION and EQUIVALENCE statements in subroutines CBTEST, GRTEST, SBTEST, DSTEST, and POLYFT, and also altering the value of the variable INPMAX set by a DATA statement in the subroutines CBTEST, GRTEST, SBTEST, and DSTEST.

/DATCOM/CDAT (2400)
Edited Data Array

LOC	Symbol	Dimension	Description
1	TIMTAG	d(300)	Data time tag, seconds from 1950.
601	TXTRA	(300)	Adjusted data time tag. TXTRA(I) = TIMTAG (I) - TIMTAG (1)
901	IQ	(300)	Quality indicator 0: Good data 1: Angles bad 2: Range bad 4: Doppler bad: > 4: Combination of above
1201	XANG	(300)	X-angle, azimuth, or hour-angle; radians.
1501	YANG	(300)	Y-angle, elevation, or declination; radians
1801	RANGE	(300)	Range; km or seconds
2101	DOP	(300)	Doppler; counts or seconds

/DATCOM/CDAT(299)

DATCOM is used for communication between the subroutines which compute the observables and the data processing routines of the DCP. It contains all the data which must be supplied for computation of the observables as well as the computed observables and their partial derivatives.

/DATCOM/CDAT(299)

Table 1

Measurement Data

Loc	Symbol	Dimension	Definition
1	BIAS	d(2)	Doppler bias frequency Spacecraft retransmission ratio
5	OBS	d(64)	See Table 2
133	FTR	d	Doppler transmission frequency
135	OMEGA	d	Earth rotation rate
137	SPDLT	d	Speed of light
139	STA	d(10)	See Table 3
159	TAU	d	Doppler count interval
161	TB2CO	d(18)	Transformations from B-frame to C-frame at signal reception by the receiving station, at end of doppler interval and at beginning of doppler interval
197	TB2CT	d(18)	Transformations from B-frame to C-frame at signal transmission by the transmitting station, at end of doppler interval and at beginning of doppler interval.
233	TT2BO	d(9)	Unit North, East, Down vectors at receiving station in the B-frame
251	TT2BT	d(9)	Unit vectors at the transmitting station
269	XV	d(12)	C-frame components of spacecraft position and velocity at end of doppler interval and at beginning of doppler interval

Table 1 - Continued

Loc	Symbol	Dimension	Definition
293	MLT		If = 1, speed of light partials are computed; if = 2, not computed.
294	MODE		Set = 2 or 3 for 2 or 3-way doppler, respectively
295	MSTA		If = 1, station location partials are computed; if = 2, not computed
296	MTIM		If = 1, time bias partials are computed; if = 2, not computed
297	NALIGN		X-Y mount alignment key: =1 for principal axis north, =2 for principal axis east.
298	NANG		If = 1, angles and their partials are computed; if = 2, not computed
299	NFRAC		If = 1, refraction effects are included; if = 2, not included

OBS d(64)

Table 2

Observables and Partial Derivatives

α_1 Azimuth X-Angle or Hour Angle	α_2 Elevation Y-Angle or Declination	ρ Ranging Observable	$\dot{\rho}$ Doppler Observable	Comments
1 α_1	17 α_2	33 ρ	49 $\dot{\rho}$	Measurements
$\frac{\partial \alpha_1}{\partial x_V}$	$\frac{\partial \alpha_2}{\partial x_V}$	$\frac{\partial \rho}{\partial x_V}$	$\frac{\partial \dot{\rho}}{\partial x_V}$	Vehicle Position Partials (3)
$\frac{\partial \alpha_1}{\partial \dot{x}_V}$	$\frac{\partial \alpha_2}{\partial \dot{x}_V}$	$\frac{\partial \rho}{\partial \dot{x}_V}$	$\frac{\partial \dot{\rho}}{\partial \dot{x}_V}$	Vehicle Velocity Partials (3)
$\frac{\partial \alpha_1}{\partial \alpha_1}$	$\frac{\partial \alpha_2}{\partial \alpha_2}$	$\frac{\partial \rho}{\partial \dot{\rho}}$	$\frac{\partial \dot{\rho}}{\partial \omega_3}$	Measurement Bias Partials
$\frac{\partial \alpha_1}{\partial t}$	$\frac{\partial \alpha_2}{\partial t}$	$\frac{\partial \rho}{\partial t}$	$\frac{\partial \dot{\rho}}{\partial t}$	Observing Station Clock Bias Partials (total time derivative)
$\frac{\partial \alpha_1}{\partial c}$	$\frac{\partial \alpha_2}{\partial c}$	$\frac{\partial \rho}{\partial c}$	$\frac{\partial \dot{\rho}}{\partial c}$	Speed of Light Partials
$\frac{\partial \alpha_1}{\partial x_R}$	$\frac{\partial \alpha_2}{\partial x_R}$	$\frac{\partial \rho}{\partial x_R}$	$\frac{\partial \dot{\rho}}{\partial x_R}$	Receiving Station Position Partials (3)
0 0 0	0 0 0	$\frac{\partial \rho}{\partial x_T}$	$\frac{\partial \rho}{\partial x_T}$	Transmitting Station Position Partials (when operating in three-way doppler mode) (3)

STA d(10)

TABLE 3

STATION DATA

Loc	Quantity
1-3	Receiving station position in the B-frame
4-5	Receiving station refraction coefficients.
6-8	Transmitting station position.
9-10	Transmitting station refraction coefficients.

/DCPCOM/CDCP(900)

Block data: Deck MC1378

DCPCOM is used for communication between the major links of the DCP. It contains those quantities which are required by two or more of the links, and those which are to be saved regardless of program overlay during a run.

/DCPCOM/CDCP(900)
Table 1
DCP Main Common

LOC	Symbol	Dimension	Definition
1	CON	d(8)	Universal constants (Table 2)
17	CBODY	(8, 11)	Central body constants (Table 3)
105	MAXZ		Maximum length of state vector (program dimension)
106	MAXQ		Maximum number of equation of motion parameters in the state
107	MAXD		Program dimension of equation of motion arrays (DQDCOM)
108	Spares	(2)	----------------------------------
110	YTEST		Value used for missing measurements on the data tape (-0. 12345678E20)
111	IERR		Error identifcation key
112	ETAPE	(4)	Parameters defining the planetary ephemeris tape position (see DPFMRS)
116	NRSEND		Number of record pairs on the residual tape
117	NESEND		Number of record pairs on the estimation tape
118	NEST		Estimate tape record number used for constructing a priori estimate
119	TBEGIN	d	Time of first data on the data tape, seconds from 1950 Jan 0. 0 ST
121	TEND	d	Time of last data on tape
123	STIMNX	(20)	Array of times of last data used in estimation, each station

Table 1 - (Continued)

LOC	Symbol	Dimension	Definition
143	SN	d(13, 20)	Station names, locations, orientations with respect to the Earth-fixed reference frame
663	IFPLNT	(11)	Yes/No options for inclusion of planetary attractions in the equations of motions
676	IFOUTP	(11)	Output options (Table 4)
687	IFEMPS	(8)	Equation of motion options (Table 5)
695	NEMP	(8)	Variational equation dimensions (Table 6)
703	IFHARM	(24)	Yes/No options for inclusion of harmonic coefficient sensitivities in the variational equations
727	Spares	(4)	----------------------------------
731	NPROC	(22)	Process control options (Table 7)
753	Spares	(3)	----------------------------------
756	IPROC	(5)	Internal process indices (Table 8)
761	KEYOUT	(10)	Output control keys for trajectory integration (see OUTXPD)
771	NCB	(3, 2)	Harmonic summation limits for Earth, Moon, and body NBX
777	ICB		Current central body numbers
778	JCB		Central body type (1 = Earth, 2 = Moon, 3 = Other)
779	KCB		Next probable central body
780	NTBL		Number of integration points skipped in filling SBFCOM
781	STIMR	d(2)	Process start and stop times, seconds from 1950 January 0.0 ST
785	ETIMV1	d	Desired anchor point time, seconds from 1950 January 0.0 ET
787	NSCTRL	(14)	Data start control indices (Table 9)
801	DATEV1	d(3)	Date of desired anchor point (See DATINP)
807	RSTOP	d	Radius from body NTARGT for propagation stop
809	DRECT	d	Rectification test magnitude.

Table 1 - (Continued)

LOC	Symbol	Dimension	Definition
811	EPSDC	d	Non-optional filter gain, ϵ
813	SCALE	d(6)	Scale factors for data start.
825	CONTROL	d(20)	Data start control parameters (Table 10).
865	Spares	(16)	-------------
881	MSKIP	(4, 4)	Array of missing measurement keys.
897	RSTP2	d	$(RSTOP)^2$
899	CBRAN2	d	$(\text{Slant range from central body})^2$ at the anchor point.

CON d(8)

Table 2

Universal Constants

Locations		Symbol	Math Symbol	Units	Block data value
CDCP	CON				
1	1	HPI	$\pi/2$		1.5707963267948966
3	2	PI	π		3.1415926535897932
5	3	TPI	2π		6.2831853071795864
7	4	RTD	$180/\pi$	deg/rad	57.295779513082329
9	5	DTR	$\pi/180$	rad/deg	0.017453292519943295
11	6	SPMSD		sec/day	86400.
13	7	RSPMSD		day/sec	$1.1574074074074074 \times 10^{-5}$
15	8	SPDLT	c	km/sec	299792.5 (speed of light)

CBODY (8, 11)

Table 3

Central Body Constants

Loc	Units	Definition
1, i		i th body name
2, i	km	Radius of sphere of influence
3, i	km	Equatorial radius
4, i	km	Polar radius
5, i	rad/sec	Angular velocity
6, i	rad	Right ascension of the prime meridian, from the ascending node of the Earth's ecliptic on the body's equator
7, i	rad	Inclination of Earth's ecliptic with body's equator
8, i	rad	Longitude of the node, measured in the Earth's ecliptic from the Earths mean vernal equinox of 1950.0

IFOUTP (11)

Table 4

Output Options

Location		Symbol	Affected Subroutines*	Definition
CDCP	IFOUTP			
676	1	NPROUT	OUTTRJ	Integration output key for the estimation/propagation link.
677	2	NRSOUT	OUTTR3	Integration output key for the residual link.
678	3	NCVOUT	COVOUT	Covariance matrix output key.
679	4	NESOUT	ESTMAT	Estimation process output key.
680	5	NSTTD	STTDAT	Output key for data start data selection.
681	6	NSTTF	STTFIT	Ouput key for data start polynomial smoothing.
682	7	NSTTG	STTBGN	Output key for data start initialization.
683	8	NSTTM	MXLEST	Output key for output during maximum likelihood estimation
684	9	NSTTR	STTINE	Output key for state output during data start.
685	10	NSTTS	STTEND	Output key for data start completion.
686	11	NDPØUT	ØUTTRJ ØUTTR3	Double precision output key.

* The values of IFOUTP(i) and their effects on output are described in the description of the named subroutines and in References 1 and 2.

IFEMP (8)

Table 5

Equation of Motion Options

Location CDCP	Location IFEMP	Symbol	Input Option*	Definition and values**
687	1	IFGRAV	X	Extra body harmonics included?
688	2	IFSOLR	X	Solar pressure included? 1 C_1 not in state 0 not in equations of motion -1 C_1 in state
689	3	IFATMD	X	Drag model included? 1 C_2, C_3 not in state 0 not in equations of motion - 1 C_2 in state - 2 C_3 in state - 3 C_2, C_3 in state
690	4	IFVENT	X	Venting included? 1 C_4 not in state 0 not in equations of motion -1 C_4 in state
691	5	IFTIME		ET-UT parameters 0 γ_3, γ_4 not in state -1 γ_3 in state -2 γ_4 in state -3 γ_3, γ_4 in state
692	6	IFHARM		Harmonic key*** 0 no harmonics for current central body. ±4 zonal harmonics only ±5 tesseral harmonics
693	7	IFDRAG		Drag key*** 0 if central body is not NBD IFATMD if central body is NBD
694	8	IFBODY		IFATMD + IFHARM 0 only if acceleration is independent of central body orientation

* Yes/No options for inclusion in equations of motion (=1: yes).
** Values are computed by SETCAS
*** Values are computed by TRAJD, TRAJDP

NEMP (8) *

Table 6

Variational Equation Dimensions

Location CDCP	Location NEMP	Symbol	Description
695	1	NDRAG	Location in RDEQ of drag parameters sensitivities
696	2	NHARM	Location in RDEQ of harmonic coefficient sensitivties
697	3	NSOLR	Location in RDEQ of solar pressure sensitivities
698	4	NVENT	Location in RDEQ of venting sensitivities
699	5	NTIME	Location in RDEQ of sensitivities to ET-UT
700	6	NPEND	First raw of RDEQ not used for differential equations
701	7	NEMP3	3 x NEMPS
702	8	NEMP5	Number of equation of motion parameters

*The NEMP array is loaded by SETCAS during interpretation of the IFEMP array.

NPROC (22)

Table 7

Process Control Options

Card Cols**	Field	NPROC LOC	Symbol*	Definition
1	I1	1	NPROCS	Process type specification
2	I1	2	NPROVR*	Does data follow the process card?
3-8	A6	3	NPRSTA	Name of receiving station
9-32	2(I3, I7, I2)	4	NDATE	Dates for process start and stop (dimensioned 3, 2)
33-36	I4	10	NPREST	Record number of a priori estimate
37-38	I2	11	NREWND*	Rewind data tape before search?
			NPSTOP	Propagation stop event identification
39-40	I2	12	NPRSTO*	Store new estimate on tape?
			NTARGT	Target body number (propagation only)
41-42	I2	13	NPSKIP	Number of time points to be skipped between processed points
43-50	4I2	14	IFSKIP*	Is Ith measurement to be skipped? (dimensioned 4)
51-52	I2	18	IFBADD*	Is flagged "bad" data to be processed?
53-54	I2	19	IFOUTL*	Are "outliers" to be processed?
55-56	I2	20	NRTAPE*	Is residual tape to be written?
57-58	I2	21	NRPLOT*	Is residual plot desired?
59-60	I2	22	NRLIST	Residual output level option

*The options marked by an asterisk are yes/no options.
In each case, "no" is designated by a zero or blank field and "yes" by any non-zero integer.

** The card locations of the options are given here for completeness. For a more detailed description of the card and options, see Reference 1.

IPROC (5)

Table 8

Internal Process Indices

Location CDCP	Location IPROC	Symbol	Description
756	1	IPROCS	NPROCS Mod 4 Process type, if any, to follow data start, if any.
757	2	ITRSTA	Location in NAMSTA of transmitting station.
758	3	INDSTA	Location in NAMSTA of receiving station.
759	4	IFTSTT	1 if process start time is specified, 2 if not.
760	5	IFTSTP	1 if process stop time is specified, 2 if not.

NSCTRL (14)

Table 9

Data Start Control Indices

Location		Symbol	Description
CDCP	NSCTRL		
787	1	NPOINT	Number of points to be used for the maximum liklihood estimation.
788	2	NFPTS	Number of points to be used for polynomial fit.
789	3	NFSKIP	Number of points skipped between used points in the polynomial fit.
790	4	NFPOLY	Degree of polynomial +1.
791	5	NFERR	Number of points to shift for retry of polynomial fit.
792	6	MFTRY	Maximum number of trials allowed for polynomial fit.
793	7	MTRY	Maximum number of trials allowed for maximum likelihood estimation.
794	8	KSCALE	Scaling option < 0: scale factors are input in SCALE = 0: no scaling > 0: normalize XM
795	9	MAXPTS	Maximum number of points to be loaded into working array DATA.
796	10	MINPTS	Minimum number of points to be loaded.
797	11	MAXAMB	Maximum number of range ambiguity trials for m. l. e.
798	12	KAUTO	Logical yes/no (T/F) option for allowing retrial with new data if m. l. e. is not completed with original data.
799	13	Spares	(2)

CONTRL d(20)

Table 10

Data Start Control Parameters

Location CDCP	Location CONTRL	Symbol	Description
825	1	TFIT	Time from ONTIME to start polynomial fit.
827	2	EPSBAR	Convergence level for mean wgt sq of m. l. e. residuals.
829	3	EPSDEL	Fractional change in wgt sq of residuals.
831	4	EPSEUD	Test threshold for pseudo inverse.
833	5	FITLEV	Test factor for polynomial fit acceptance.
835	6	RCNST	Position constraint factor.
837	7	VCNST	Velocity constraint factor.
839	8	CNST	(1): Constraint vector control = 0: input G(I) in CNST (I+1) > 0: set G(I) on each trial < 0: set G(I) on first (2-7): Input constraint vector
853	15	CNTL	Convergence rate factor (β) control = 0: $\beta = \beta_o$ < 0: β variable downward only > 0: β variable
855	16	BO	β_o
857	17	BUP	β increase factor
859	18	BDOWN	β decrease factor
861	19	AUPPR	Maximum fractional correction for increasing β.
863	20	ALOWR	Minimum fractional correction.

```
$IBFTC MC1378  XR3,M94,NODD,LIST
      BLOCK DATA
C      DCP MAIN COMMON
C
      COMMON              /DCPCOM/CDCP(900)
C
C
      DOUBLE PRECISION    CON     (8)
     1                   ,CONTRL(20) ,RSTOP   (2)
     2                   ,EPSDC       ,SCALE   (6)
C
      DIMENSION           CBODY(8,11) ,IFEMPS (8) ,KEYOUT(10)
     1                   ,NCB    (3,2) ,IFOUTP(11) ,MSKIP(4,4)
     2                                ,IFPLNT(11) ,NSCTRL(14)
C
      EQUIVALENCE                  (CDCP(663),IFPLNT) ,(CDCP(787),NSCTRL)
     1        ,(CDCP( 17),CBODY ) ,(CDCP(761),KEYOUT) ,(CDCP(780),NTBL   )
     2        ,(CDCP(   1),CON    ) ,(CDCP(107),MAXD    ) ,(CDCP(674),ITARGT)
     3        ,(CDCP(825),CONTRL) ,(CDCP(106),MAXQ    ) ,(CDCP(807),RSTOP  )
     4        ,(CDCP(811),EPSDC  ) ,(CDCP(105),MAXZ    ) ,(CDCP(813),SCALE  )
     5        ,(CDCP(687),IFEMPS) ,(CDCP(881),MSKIP  ) ,(CDCP(110),YTEST  )
     6        ,(CDCP(676),IFOUTP) ,(CDCP(771),NCB     )
C
C     UNIVERSAL CONSTANTS
C                                  CON(1)=PI/2             CON(2)=PI
      DATA  CON    /            1.5707963267948966      ,3.1415926535897932
C           CON(3)=2*PI            CON(4)=RTD              CON(5)=DTR
     1   ,6.2831853071795864 ,57.295779513082329      ,.017453292519943295
C           CON(6)=SPMSD           CON(7)=RSPMSD           CON(8)=SPDLT
     2   ,86400.D0              ,1.1574074074074074D-5 ,299792.5D0         /
C
C     CENTRAL BODY CONSTANTS
C     BODY NAME
      DATA (CBODY(1,I),I=1,11)/ 6HMERCRY ,6HVENUS   ,6HEARTH   ,6HMARS
     1                         ,6HJUPTER ,6HSATURN ,6HURANUS ,6HNEPTUN
     2                         ,6HPLUTO   ,6HSUN     ,6HMOON             /
C     SPHERE OF INFLUENCE (KM)
      DATA (CBODY(2,I),I=1,11)/ 111000.   ,616000.   ,925000.   ,565000.
     1                         ,480.E5    ,546.E5    ,520.E5    ,875.E5
     2                         ,135.E5    ,1.E10     ,6600.              /
C     EQUATORIAL RADIUS (KM)
      DATA (CBODY(3,I),I=1,11)/ 2500.     ,6200.     ,6378.165 ,3400.
     1                         ,71350.    ,60400.    ,25550.    ,25000.
     2                         ,2882.     ,695550.   ,1738.              /
C     POLAR RADIUS (KM)
      DATA (CBODY(4,I),I=1,11)/ 2500.     ,6200.     ,6356.5838,3400.
     1                         ,66600.    ,54040.    ,23725.    ,24445.
     2                         ,2882.     ,695500.   ,1738.              /
C     ANGULAR VELOCITY (RAD/SEC)
      DATA (CBODY(5,I),I=1,11)/ 11*0./
C     RT ASC OF PRIME MERIDIAN (RAD)
      DATA (CBODY(6,I),I=1,11)/ 11*0./
C     OBLIQUITY (RAD)
      DATA (CBODY(7,I),I=1,11)/ 11*0./
C     LONGITUDE OF NODE (RAD)
      DATA (CBODY(8,I),I=1,11)/ 11*0./
C
C     PROGRAM DIMENSIONS AND CONSTANTS
      DATA  MAXZ,MAXQ,MAXD/30,6,44/
      DATA  YTEST          /-.12345678E20/
C
C     OPTIONS
C
      DATA  IFPLNT/0, 1, 1, 1, 5*0, 1, 1/
      DATA  IFOUTP/2, 2, 1, 0, 6*10, 1/
      DATA  IFEMPS/0, 1, 1, 5*0/
      DATA  KEYOUT/0, 1, 3,-1, 6*0/
      DATA   NCB    /4, 2, 0, 0, 2, 0/
C
C     PROGRAM CONTROL PARAMETERS
C
      DATA  NSCTRL/ 30,  7,  1,  4,  4,  5, 10,  1,100,  0
     1             ,  4,  0,  1,  0/
      DATA  RSTOP / 0.D0, 1.D-1/
      DATA  EPSDC / 0.D0/
```

```
 DATA   SCALE / 6*1.D0/                                                 DCPC0075
 DATA   CONTRL/ 0.D+0, 1.D+1, 1.D-5, 1.D-8, 1.D+4                       DCPC0076
1             , 1.D-1, 1.D-1,-1.D+0,       6*0.D+0                       DCPC0077
2             , 1.D+0, 1.D+0, 2.D+0, 1.D-1, 1.3D+0, 1.D-8 /             DCPC0078
 DATA   MSKIP /0,0,0,1, 0,0,0,0, 0,0,0,0, 0,0,1,0/                      DCPC0079
 DATA   ITARGT/4/                                                       DCPC0080
 DATA   NTBL  /4/                                                       DCPC0081
 END
```

/DCRCOM/ZBAR(30), ZHAT(30), ZTOT(30), DCSAVE(74)

No block data.

DCRCOM is used for storing differential corrections and other quantities to be saved during linking in an estimation process. It is located in the DIFCOR link of the DCP. The differential corrections are:

ZBAR	The deviation of the "nominal" state at the anchor point from the a priori estimate.
ZHAT	The deviation of the current best estimate of state at the anchor point from the a priori estimate.
ZTOT	The deviation of the current a priori estimate from the original a priori estimate.

The vectors ZBAR, ZHAT, ZTOT have dimensions MAXZ, and must be re-dimensioned when the maximum length of the state vector is changed. The affected subroutines are DIFCOR, ESTOUT and ESTMAT.

The array DCSAVE contains quantities set during initialization of ESTMAT to be saved during any linking. The quantities are described in the table below.

DCSAVE (74)

ESTMAT Initialization Data

Loc	Symbol	Dimension	Description
1	OBIAS	d(4)	Measurement biases
9	YMOD	d(4)	Measurement moduli
17	EPSK	d(4)	Non-optimal filter gains (=EPSDC* measurement standard deviation)
25	Q2	d(4)	Measurement variances
33	DPREV	d	Square of position deviation from the nominal trajectory at last failure of the linearity test.
35	TFIRST	d	First station on-time, seconds from 1950 Jan 0.0
37	STPTIM		Process stop time, seconds from TFIRST
38	TREF		Start time for current data arc, seconds from TFIRST
39	LABEL	(12)	BCD measurement names, units
51	SYMBOL	(4)	Abbreviated measurement names
55	LSKIP	ℓ (4)	Measurement skip options
59	LSTA	ℓ	Internal switches
60	KOUTL	ℓ	Internal switches
61	NDIM	(5)	Internally computed dimensions
66	NASGN	(9)	Permanent transfer addresses (for assigned "GO TO's")

/DFMCOM/CDFM (2232)

Block data: Deck MC1324 or MC1344

DFMCOM is used for communication between DPFMRS and its using subroutines. The ephemeris tape record last used by DPFMRS is retained in DFMCOM to minimize tape searching.

DFMCOM (MC1324) and DPFMRS (MC1322) may be interchanged with DFMCOM (MC1344) and DPFMRS (MC1342) for space savings, with some reduction of ephemeris capabilities (see DPFMRS). Dimensions for the arrays in DFMCOM for both deck pairs are given in the table below. Other decks referencing DFMCOM need not be compiled provided DFMCOM is loaded before those decks.

/DFMCOM/CDFM(2232)
*CDFM(1524)

Planetary Ephemeris Storage

Loc	Symbol	Dimension	Description
1	IFM	(14)	Ephemeris output options 0: no output for the ith body 1: output position only 2: output position and velocity
15	RFM	d(6, 12)	Planetary positions and velocities
159	DFM	d(4)	Nutations in longitude and obliquity and nutation rates
167	BFM	d(829) d(577)	Position-velocity buffer
1825	SNT	(2, 204) (2, 102)	Nutation buffer

*Where two dimensions are given, the first refers to decks MC1322, MC1324, and the second refers to MC1342, MC1344.

```
$IBFTC MC1344  XR3,M94,NODD,LIST
      BLOCK DATA                                                        FMBD0001
      DOUBLE PRECISION   BFM,DFM,RFM                                    FMBD0002
      COMMON            /DFMCOM/IFM(14),RFM(6,12)                       FMBD0003
     1                         ,DFM (4),BFM (577),SNT(2,102)            FMBD0004
      DATA  IFM/0,0,2,7*0,2,2,1,0/                                      FMBD0005
      DATA  SNT/204*0./                                                 FMBD0006
      END
```

```
$IBFTC MC1324  XR3,M94,NODD,LIST
      BLOCK DATA                                                        FMBD0001
      DOUBLE PRECISION   BFM,DFM,RFM                                    FMBD0002
      COMMON            /DFMCOM/IFM(14),RFM(6,12)                       FMBD0003
     1                         ,DFM (4),BFM (829),SNT(2,204)            FMBD0004
      DATA  IFM/0,0,2,7*0,2,2,1,0/                                      FMBD0005
      DATA  SNT/408*0./                                                 FMBD0006
      END
```

/DQDCOM/CDQD (26N + 28)

No block data.

DQDCOM is used for communication between DEQD and the remainder of the propagation link of the DCP. Its size depends on the maximum number of variational equations to be allowed. If

M_Q = Maximum number of equation of motion parameters permitted in the state vector, then

$N = 3M_Q + 26$

For example, if up to six parameters are to be permitted simultaneously, N = 44, and DQDCOM has length 1172. The values of M_Q, N used must be compiled into MAXQ, MAXD of DCPCOM. Changes in these values require recompilation of

Subroutine	Deck
DEQD	MC133D
TRAJDP	MC133F
ACCTRJ	MC133A
OUTTRJ	MC1327

See also SBFCOM.

Table 1

Principal Arrays

Location	Name	Dimension	Definition
1	NDEQ	10	Array of fixed-point integration control parameters.
11	CDEQ	10	Array of floating-point control parameters.
21	XDEQ	d(4)	Independent variable values.
29	ADEQ	d(N)	Second derivatives and stopping functions.
2N+29	RDEQ	d(N)	Dependent variables.
4N+29	VDEQ	d(N)	First derivatives
6N+29	FDEQ	d(N, 10)	Storage array used by DEQD .

CDEQ

Table 2

Floating-Point Parameters

Component	Symbol	Definition
1	H	Current integration interval.
2	HP	Previous integration interval.
3	HO	Starting interval size.
4	HU	Upper limit for doubling interval size.
5	HL	Lower limit for halving interval size.
6	HE	Interval for even-interval output.
7	EU	Truncation error upper limit.
8	EL	Truncation error lower limit.
9-10		Not used.

NDEQ
Table 3
Fixed-Point Parameters

Component	Symbol	Definition and Values
1	NI	Integration mode option 0: Kutta Mode 1: Adams Mode
2	NO	Output option 0: Output special points only. ≠ 0: Output every NOth integration point.
3	NS	Number of stopping functions (≤ 5).
4	NA	Interval control option 0: fixed interval size ≠ 0: interval size controlled by accuracy testing
5	NC	Predictor-corrector option (Adams integration) 0: open formulas 1: predictor-corrector without recomputation of corrected second derivatives. 2: predictor-corrector
6	NE	Number of equations to be integrated.
7	NT	Number of equations + number of stopping functions.
8	IP	Point identification key 0: regular integration point 1: even-interval output point 3: stop point
9	IT	Row number of stopping function causing stop.

Note: Components 1-6 are supplied prior to the call of DEQD, and components 7-9 are supplied by DEQD.

XDEQ (4)

Table 4

Independent Variable

Component	Symbol	Definition
1	X	Running value of the independent variable.
2	XF	Final value of the independent variable.
3	XE	Next even-interval output point.
4	XO	Value of X at start of current integration interval.

/EDTCOM/INDDAT(40), BUFDAT(85, 6)

No block data.

EDTCOM is used for the storage of the identification (INDDAT) and data (BUFDAT) records from the edited data tape in the DCP. The contents of these records are described below.

INDDAT(40)
Data Identification Array

Location	Symbol	Dimension	Decription
1	NEDT		Record pair count
2	NEOT		End-of-tape indicator -1 if NEOT = 1 +1 if tape-end record 0 otherwise
3	NRSTA		Name of observing station
4	NTSTA		Name of transmitting station
5	NRCD		Arc record count, 1 on first pair for each data arc
6	NPTS		Number of time points in the block
7	KONT		Continuation key 0 if this block begins the arc 1 if another block preceeds
8	MTYPE		Measurement type
9	NALIGN		X-Y mount orientation key
10	MODE		Doppler mode, 2- or 3-way
11	DELT		Measurement Time interval

INDDAT(40) (continued)

Location	Symbol	Dimension	Description
12	KTAU		Count mode 1 fixed interval (in TAU) 0 fixed count (in TAU)
13	ONTIME	d	Arc start time, seconds from 1950
15	TFIRST	d	Block start time
17	TLAST	d	Block end time
19	TAU	d	Fixed count or interval
21	FTR	d	Transmitter frequency
23	CREFR	d(2)	Refraction coefficients
27	DRNGE	d	Range ambiguity
29	OMG3	d	Bias frequency
31	OMG4	d	Retransmission ratio
33	NBAD	(4)	Number of bad points, each measurement
37	SSDEV	(4)	Sample standard deviations

The BUFDAT array is a single precision array of dimension (85, 6) containing the edited data as read from the data tape. Each row contains the following data in the six columns:

BUFDAT (I, 1) = Quality indicator
(I, 2) = Time, seconds from ONTIME
(I, 3) = Angle 1
(I, 4) = Angle 2
(I, 5) = Range
(I, 6) = Doppler

/DQ3COM/CDQ3(236)

No block data.

DQ3COM is used for communication between DEQ3 and the remainder of the residual link of the DCP. It is referenced by

Subroutine	Deck
DEQ3	MC133E
DTRAJ	MC133G
TR3ACC	MC133B
TR3OUT	MC1328

Principal Arrays

Location	Name	Dimension	Definition
1	NDEQ	10	Array of fixed-point integration control parameters.
11	CDEQ	10	Array of floating-point control parameters.
21	XDEQ	d(4)	Independent variable values.
29	ADEQ	d(8)	Second derivatives and stopping functions.
45	RDEQ	d(8)	Dependent variables.
61	VDEQ	d(8)	First derivatives.
77	FDEQ	d(8, 10)	Storage array used by DEQ3.

The components of the arrays are the same as those described in DQDCOM. DQ3COM is limited to three differential equations, however, and five stopping functions.

Missing measurements at a given time are filled with the constant -0.12345678E20. Measurements which are rejected by the TDEP as outliers are flagged by a minus sign.

The quality indicator is the sum of three indicators, $I_1 + I_2 + I_4$, where each indicator is zero if the corresponding measurement values are valid, and non-zero if the measurements are missing or flagged "bad" by the tracking station operator or are garbled in transmission. The indicators are;

I_1 = 1 if angle data is bad or missing,

I_2 = 2 if range data is bad or missing,

I_4 = 4 if doppler data is bad or missing.

/EQMCOM/CEQM(2214)

Block data: Deck MC1376

EQMCOM is used for the storage of equation of motion parameters, their standard deviations and correlation coefficients. It is used by SETCAS in constructing the equation of motion portions of the a priori estimate and covariance matrix.

EQMCOM may be modified by overlay input at the case level (see References 1 and 2).

/EQMCOM/CEQM(2214)

Location	Symbol	Dimension	Description
1	SPCDAT	d(31)	Spacecraft position and velocity, values, standard deviations and normalized correlations.
63	EFEDAT	d(83)	Planetary gravitational constants, a.u., R_E.
229	PREDAT	d(15)	Solar pressure, drag (2), venting.
259	DELDAT	d(5)	ET-UT parameters (2).
269	EHADAT	d(324)	Earth harmonic coefficients.
917	MHADAT	d(324)	Moon harmonic coefficients.
1565	XHADAT	d(325)	Extra body harmonic coefficients.

```
$IBFTC MC1376  XR3,M94,NODD,LIST
      BLOCK DATA                                                         EQMC0001
C     EQUATION OF MOTION BLOCK DATA                                      EQMC0002
      COMMON /EQMCOM/EQMC(2214)                                          EQMC0003
      DOUBLE PRECISION  SPCDAT(31),    EFEDAT(83),    PREDAT(15)         EQMC0004
     1,  DELDAT(5),     EHADAT(324),   MHADAT(324),   XHADAT(325)        EQMC0005
      EQUIVALENCE             (EQMC(1),SPCDAT),      (EQMC(63),EFEDAT)   EQMC0006
     1,  (EQMC(229),PREDAT),  (EQMC(259),DELDAT),    (EQMC(269),EHADAT)  EQMC0007
     2,  (EQMC(917),MHADAT),  (EQMC(1565),XHADAT)                        EQMC0008
C                                                                        EQMC0009
C     SPACECRAFT POSITION (KM)                                           EQMC0010
      DATA (SPCDAT(I),I=1,31)/.20947126D4,-.62193760D4,-.32542986D1      EQMC0011
C                                                                        EQMC0012
C     SPACE CRAFT VELOCITY (KM/SEC)                                      EQMC0013
     1,.65028256D1,.21881185D1,.36959843D1                               EQMC0014
C                                                                        EQMC0015
C     POSITION AND VELOCITY STANDARD DEVIATIONS (KM, KM/SEC)             EQMC0016
     2,3*1.D-2,3*.33333333333333333D-2                                   EQMC0017
C                                                                        EQMC0018
C     POSITION AND VELOCITY CORRELATIONS                                 EQMC0019
     3,15*0.D0                                                           EQMC0020
C                                                                        EQMC0021
C     CENTRAL BODY AND TIME OF ESTIMATE (YM.  DHM.  SEC.)                EQMC0022
     4, 3.D0,  6701.D0,  100000.D0,  0.D0/                               EQMC0023
C                                                                        EQMC0024
C     PLANETARY MU-S (KM**3/SEC**2)                                      EQMC0025
C                              MERCURY         VENUS          EARTH      EQMC0026
      DATA (EFEDAT(I),I=1, 83)/21685.53D0,   324853.4D0,   398603.2D0    EQMC0027
C                                                                        EQMC0028
C        MARS         JUPITER        SATURN        URANUS        NEPTUNE EQMC0029
     1,42977.8D0,   .1267106D9,   .379187D8,    .5803292D7,   .7026072D7 EQMC0030
C                                                                        EQMC0031
C        PLUTO           SUN             MOON           BARYCENTER       EQMC0032
     2,.3317886D6,   .132715445D12,   4902.779D0,    .40350579D6         EQMC0033
C                                                                        EQMC0034
C     AU(KM) AND MEAN EARTH RADIUS(KM)                                   EQMC0035
     3,.149599D9,        6378.3255D0                                     EQMC0036
C                                                                        EQMC0037
C     PLANETARY MU STANDARD DEVIATIONS (KM**3/SEC**2)                    EQMC0038
C       MERC     VENUS      EARTH     MARS     JUPITER   SATURN    URANUS EQMC0039
     4,130.D0,   160.D0,   150.D0,   150.D0,   381.D1,   228.D1,   87.D3EQMC0040
C                                                                        EQMC0041
C       NEPTUNE   PLUTO      SUN      MOON                               EQMC0042
     5,1.05D5,   2475.D1,   4.D7,   25.D0                                EQMC0043
C                                                                        EQMC0044
C     THREE SPARE LOCATIONS SAVED HERE                                   EQMC0045
     6,3*0.D0                                                            EQMC0046
C                                                                        EQMC0047
C     CORRELATIONS                                                       EQMC0048
     7, 55*0.D0/                                                         EQMC0049
C                                                                        EQMC0050
C     VALUES OF               SOLAR PRESS  DRAG1      DRAG2      VENT    EQMC0051
      DATA (PREDAT(I),I=1,15)/.94D7,     .109D0,    .14294D0,    0.D0    EQMC0052
C                                                                        EQMC0053
C     STANDARD DEVIATIONS                                                EQMC0054
     1,                         .94D6,     .109D-1,  .1429D-1,  1.D-6    EQMC0055
C                                                                        EQMC0056
C     CORRELATIONS         CENTRAL BODY FOR DRAG                         EQMC0057
     2,  6*0.D0,                 3.D0/                                   EQMC0058
C                                                                        EQMC0059
C     VALUES OF               ET-UT      RATE,   STANDARD DEV-S   CORRS  EQMC0060
      DATA (DELDAT(I),I=1,5)/35.D0,    0.D0,    2.D0,    1.D-5,    0.D0/ EQMC0061
C                                                                        EQMC0062
C     EARTH HARMONICS              J20           J30        J40      J50 EQMC0063
      DATA (EHADAT(I),I=1,324)/.1082D-2,    -.23D-5,    -.18D-5,    0.D0 EQMC0064
C                                                                        EQMC0065
C       J60      J70      J21      L21        J22        L22        J31  EQMC0066
     1,0.D0,    0.D0,    0.D0,    0.D0,    -.12D-5,    -.46D0,    -.19D-5 EQMC0067
C                                                                        EQMC0068
C        L31       J32        L32        J33        L33      J41-L44     EQMC0069
     2,.08D0,   -.14D-6,   -.293D0,   -.1D-6,    .743D0,    8*0.D0       EQMC0070
C                                                                        EQMC0071
C     STANDARD DEVIATIONS J20         J30        J40     J50-J70         EQMC0072
     3,                  .108D-3,   .23D-6,   .18D-6,   3*.1D-6          EQMC0073
C                                                                        EQMC0074
```

```
C         J21        L21        J22        L22        J31        L31        J32       EQMC0075
     4,.1D-6,   .5D-1,   .12D-6,   .5D-1,   .19D-6,   .5D-1,   .14D-7               EQMC0076
C                                                                                    EQMC0077
C        L32        J33        L33        J41        L41        J42        L42        EQMC0078
     5,.5D-1,   .1D-7,   .5D-1,   .1D-7,   .5D-1,   .1D-7,   .5D-1                   EQMC0079
C                                                                                    EQMC0080
C         J43        L43        J44        L44                                       EQMC0081
     6,.1D-7,   .5D-1,   .1D-7,   .5D-1                                              EQMC0082
C                                                                                    EQMC0083
C      CORRELATIONS                                                                  EQMC0084
     7,276*0.D0/                                                                     EQMC0085
C                                                                                    EQMC0086
C      MOON HARMONICS                  J20          J30-J70                          EQMC0087
       DATA (MHADAT(I),I=1,324)/.207107D-3,    5*0.D0                                EQMC0088
C                                                                                    EQMC0089
C        J21       L21          J22          L22       J31-L44                       EQMC0090
     1,0.D0,    0.D0,    -.20716D-4,    0.D0,    14*0.D0                             EQMC0091
C                                                                                    EQMC0092
C      STANDARD DEVIATIONS J20                      J30-J70                          EQMC0093
     2,                   .207107D-4,           5*1.D-6                              EQMC0094
C                                                                                    EQMC0095
C        J21        L21          J22         L22        J31        L31       J32     EQMC0096
     3,1.D-6,   .5D-1,   .20716D-5,   .5D-1,   .1D-6,   .5D-1,   .1D-6               EQMC0097
C                                                                                    EQMC0098
C         L32        J33        L33        J41        L41        J42        L42      EQMC0099
     4,.5D-1,   .1D-6,   .5D-1,   .1D-6,   .5D-1,   .1D-6,   .5D-1                   EQMC0100
C                                                                                    EQMC0101
C        J43        L43        J44        L44                                        EQMC0102
     5,.1D-6,   .5D-1,   .1D-6,   .5D-1                                              EQMC0103
C                                                                                    EQMC0104
C      CORRELATIONS                                                                  EQMC0105
     6,276*0.D0/                                                                     EQMC0106
C                                                                                    EQMC0107
C      EXTRA BODY HARMONICS      J20-J70    J21-L44                                  EQMC0108
       DATA (XHADAT(I),I=1,325)/6*0.D0,    18*0.D0                                   EQMC0109
C                                                                                    EQMC0110
C      STANDARD DEVIATIONS J20-J70     J21        L21        J22        L22          EQMC0111
     1,                  6*.1D-6,   .1D-6,   .5D-1,   .1D-6,   .5D-1                 EQMC0112
C                                                                                    EQMC0113
C         J31        L31        J32        L32        J33        L33        J41      EQMC0114
     2,.1D-6,   .5D-1,   .1D-6,   .5D-1,   .1D-6,   .5D-1,   .1D-6                   EQMC0115
C                                                                                    EQMC0116
C         L41        J42        L42        J43        L43        J44        L44      EQMC0117
     3,.5D-1,   .1D-6,   .5D-1,   .1D-6,   .5D-1,   .1D-6,   .5D-1                   EQMC0118
C                                                                                    EQMC0119
C      CORRELATIONS                                                                  EQMC0120
     4,276*0.D0                                                                      EQMC0121
C                                                                                    EQMC0122
C      CENTRAL BODY NUMBER                                                           EQMC0123
     5,4.D0/                                                                         EQMC0124
      END
```

/ESOCOM/CESO(804), DESO($2N^2$+4N+294)

No block data.

ESOCOM is used only by SETCAS for the temporary storage of the estimate requested from tape by the case card. The arrays CESO and DESO are images of the arrays CEST (ESTCOM) and DEST (ES1COM). DESO and its component arrays must be redimensioned when ES1COM is redimensioned to preserve the correspondence between DESO and DEST.

/ESRCOM/CESR (304)

No block data.

ESRCOM is used by the residual link of the DCP to permit overlay of state elements before computation of residuals. It contains an image of the first 244 elements of ESTCOM, as well as SER(14, 2) containing the columns of SE(ESTCOM) corresponding to the receiving station (INDSTA) and transmitting station (ITRSTA), and CRF(2) containing refraction coefficients to be used in lieu of those of the edited data tape.

/ESTCOM/CEST(804)

No block data.

ESTCOM is used for storing the current estimate of the quantities which may be included in the extended state vector and certain quantities required for their interpretation. It is loaded in the main (0) link of the DCP (see also EQMCOM, SETCAS).

/ESTCOM/CEST(804)

Table 1

Location	Symbol	Dimension	Definition
1	NESPOS		Estimate number; estimation tape record count.
2	NBY		Central body number at anchor point.
3	NBD		Body with drag parameters in PREDAN.
4	NBX		Body with harmonics stored in XHADAN.
5	NBH		Body with harmonics included in state.
6	NUMSTA		Number of stations on tape.
7	NAMSTA	(20)	Names of stations on tape.
27	KOUNTN		Number of state elements.
28	Spare		------------------------
29	ETIMVA	d	Vehicle ephemeris time at anchor point, seconds from 1950 January 0.0 ET.
31	TX2Z	d(3, 3)	Transformation, C-frame to coordinates of state.
49	SPCDAN	d(6)	Vehicle position and velocity relative to Earth at anchor point, C-frame.

Table 1 (Continued)

Location	Symbol	Dimension	Definition
61	EFEDAN	d(14)	Planetary gravitational constants and ephemeris scale factors.
89	PREDAN	d(4)	Solar pressure, drag (2), and venting coefficients.
97	DELDAN	d(2)	ET-UT, d(ET-UT)/dt.
101	EHADAN	d(24)	Earth harmonic coefficients.
149	MHADAN	d(24)	Moon harmonic coefficients.
197	XHADAN	d(24)	Body NBX harmonic coefficients.
245	SE	d(14, 20)	Station errors.

SE d(14, 20*)

Table 2

Row	Units	Definition
1	km	Station location error, northing
2	km	easting
3	km	down
4	sec	Station clock bias
5	sec/sec	Station clock bias rate
6	km/sec	Speed of light correction
7	rad	Measurement bias, angle 1
8	rad	angle 2
9	km	range
10	counts	doppler
11	rad	Measurement standard deviation, angle 1
12	rad	angle 2
13	km	range
14	counts	doppler

*The column number of a given station's errors corresponds to the location of that station's name in NAMSTA.

/ES1COM/DEST($2N^2$+4N+294)

No block data.

ES1COM is used for the storage of the covariance matrix and certain associated indices and treatment codes. It is loaded in the SETCAS link, and is dumped on the scratch tape, unit 09. It is read into ES2COM in the DIFCOR link of the DCP.

The dimensions of the arrays in ES1COM depend on N, the maximum permissible length of the state vector. The dimensions of the arrays and their locations in DEST are given as functions of N in the table below. Changes in N require recompilation of the following subroutines which reference the commons. The value used for N must be compiled into MAXZ in DCPCOM (See also DCRCOM).

	Subroutine	Deckname
ES1COM	SETCAS	MC13SW
	BLDCOV	MC13SX
	COVOUT	MC13CV
ES2COM	ESTMAT	MC133J
	ESTOUT	MC1330
	UPDATP	MC138A
	COVOUP	MC138B
ES3COM	STTEND	MC13S5

/ES1COM/DEST($2N^2$+4N+294)

General		N = 30			
Location	Dimension	Loc	Dimension	Array	Definition
1	d(N, N)	1	d(30, 30)	PNEW	Covariance matrix
$2N^2+1$	(N)	1801	(30)	STNAMN	Output names of state elements
$2N^2+N+1$	(N)	1831	(30)	TRAKER	Output names of stations for each state element
$2N^2+2N+1$	(N)	1861	(30)	ITRETN	State element treatment codes 1: element to be estimated 0: element to be included only
$2N^2+3N+1$	(N)	1891	(30)	ILOCN	State element identification keys
$2N^2+4N+1$	(54)	1921	(54)	KLOCN	State group identification keys
$2N^2+4N+55$	(12, 20)	1975	(12, 20)	NSN	Station error treatment codes

/EXIC/W(36), CRX(9, 2)

No block data.

EXIC is used as an input buffer for the reading of the trajectory tape by the TDS. The array W contains interpolation coefficients for the computation of vehicle state at a given time (see EXINST). The array CRX contains the "critical event" record from the trajectory tape, defining data relevant to initialization, patching, etc.

/INPCOM/C(700)

Block Data: Deck MC13IP

INPCOM is used for storage and communication of physical and mathematical constants by the TDS. It is also used in part as additional working common. The values compiled in the block data subroutine may be altered by use of the overlay feature of the program. Normally, values beyond C(117) should not be overlayed, since they are used as working common and the required values are generated by the program.

/INPCOM/C(700)

Table 1

LOC	Symbol	Dimension	Description
1	HPI		One-half pi
2	PI		Pi
3	TWPI		Two pi
4	RTD		Radian to degree conversion factor
5	DTR		Degree to radian conversion factor
6	SPMSD		Seconds per mean solar day
7	RSPMSD		Reciprocal of seconds per mean solar day
8	AU		Kilometers per astronomical unit
9	SPDLT		Speed of light in km/sec
10	STPSZ		Step size constant used by SBEV2
11	BODC	(10, 8)	Planetary constants. See Table 2
91	Spares	(21)	--------------------------
112	XMIN		Minimum acceptable length of data arcs (station passes) in seconds. Any arc shorter than XMIN will be eliminated

Table 1 (Continued)

LOC	Symbol	Dimension	Description
113	XMAX		Maximum acceptable length of data arcs in seconds. Any arc longer than XMAX will be divided into two or more arcs.
114	RDELAY		When the spacecraft appears above the horizon, the first measurement will be delayed by a random interval computed as the absolute value of a uniform distribution with mean zero, standard deviation of RDELAY. May be set to zero if desired.
115	XTROUT		Extra output key. If 0., only summary data will be written on the system output tape. If 1., all measurement values will be written out.
116	CFRAC1		Refraction constants
117	CFRAC2		
118	Spares	(82)	----------------------------------
200	S	(23, 12)	Array of tracking station parameters. Each column corresponds to a station number. Rows are assigned as follows: 1 Station name, BCD 2 Observation interval, seconds 3 Geodetic latitude, radians 4 Longitude, radians 5 Altitude, kilometers 6 Artificial horizon, radians 7 Artificial zenith, radians 8 Random range error standard deviation, kilometers 9 Random doppler error std. dev., counts/sec 10 Random angle 1 error std. dev., radians 11 Random angle 2 error std. dev., 12 Range bias, km 13 Doppler bias, counts/sec 14 Angle 1 bias, radians 15 Angle 2 bias, radians 16 Northing error, km 17 Easting error, km 18 Down error, km 19 Clock bias, seconds

Table 1 (Continued)

LOC	Symbol	Dimension	Description
(200 Cont.)			20 Doppler offset bias, Hz 21 Transponder retransmission ratio, (transmit/receive) 22 Transmitter carrier frequency, Hz 23 Doppler count interval, seconds
476	B	(91)	Array of beacon parameters. The first 80 elements may be considered as an 8 x 10 array where each column corresponds to a beacon number. Rows are as follows: 1 Name, BCD 2 Latitude, radians 3 Longitude, radians 4 Altitude, kilometers 5 Artificial horizon, radians 6 Latitude error, km 7 Longitude error, km 8 Altitude error, km The remaining 11 locations are: 81 Observation period, seconds 82 Spare 83 Random range error std. dev., km 84 Random range rate error, std. dev., km/sec 85 Random angle 1 error std. dev., radians 86 Random angle 2 error std. dev., radians 87 Range bias, km 88 Range rate bias, km/sec 89 Angle 1 bias, radians 90 Angle 2 bias, radians 91 Time or clock bias, seconds
556	DELBEC		DELBEC = B(81)
567	OB	(18)	Onboard measurement data as follows: 1 Random height error std. dev., km 2 Random height rate error std. dev., km/sec 3 Height bias, km 4 Height rate bias, km 5 Time bias, seconds

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
			6 Altitude to cease radar observations, km
			7 Period of radar measurements
			8 Angle 1 error k, where: $\sigma_e^2 = k_1^2 + k_2^2 + k_2^2 (2 \sin^{-1} \frac{r_p}{r})^2$ k_1 in radians
			9 Angle 1 k_2, dimensionless
			10 Angle 2 k_1
			11 Angle 2 k_2
			12 Subtended angle k_1
			13 Subtended angle k_2
			14 Period of angular measurements, seconds.
			15 Right ascension of reference star, radians.
			16 Declination of reference star, radians.
	Spare		17 ------------------
	Spare		18 ------------------
573	DELRAD		DELRAD = OB(7)
580	DELOPT		DELOPT = OB(14)
585	EMP	(24)	Equation of motion parameters, not now used
609	Spares	(64)	---------------------------
673	IKAS		Case counter
674	Spares		---------------------------
675	HEAD	(24)	HEAD(1-12) is a 12 word alphanumeric header used to identify the trajectory tape on tape 10. HEAD (13-23) is an 11-word alphanumeric header written on the output data tape HEAD (24) is not used

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
699	ITAPE		Key for data tape. If ITAPE ≠ 0, then no output tape will be written
700	IOCAS		Not used

BOCD (8, 10)

Table 2

The first 7 columns correspond to celestial bodies as follows:

1	Earth
2	Moon
3	Sun
4	Venus
5	Mars
6	Saturn
7	Jupiter

The remaining 3 columns are reserved for extra bodies. The row assignments are as follows:

1	Name, BCD
2	Gravity constants, km^3/sec^2
3	Semi-major axis, km
4	Semi-minor axis, km
5	Sphere of influence, km
6	Rotation rate, rad/sec
7	STEDT step size, seconds, not used
8	Interpolation interval, days, not used

```
$IBFTC MC13IP   NOREF,M94,NODD,XR3
CMC13IP   INPCOM BLOCK DATA FOR MESERP                                  BLOK0001
      BLOCK DATA                                                        BLOK0002
      COMMON/INPCOM/C(700)                                              BLOK0003
      DIMENSION    BODC(10,8),      CON(10)                             BLOK0004
     3,  S(23,12),    B(91),    OB(18),    EMP(24),     HEAD(24)        BLOK0005
      EQUIVALENCE (C(1),CON),    (C(11),BODC)                           BLOK0006
     1,      (C(112),XMIN),    (C(113),XMAX),    (C(114),RDELAY)        BLOK0007
     4,              (C(675),HEAD)                                      BLOK0008
     3, (C(200),S),   (C(476),B),   (C(567),OB),   (C(585),EMP)         BLOK0009
C (C(673),IKAS)                                                         BLOK0010
C (C(699),ITAPE)                                                        BLOK0011
C (C(700),IOCAS)                                                        BLOK0012
C                    C(1)= PI/2     C(2)= PI      C(3)= 2*PI            BLOK0013
      DATA(CON(I),I=1,10)/1.57079633,3.14159265,6.2831853               BLOK0014
C     C(4)= RTD    C(5)= DTR     C(6)= SPMSD    C(7)= RSPMSD  C(8)= KM/AU BLOK0015
     1,57.2957796,.0174532926,86400.,1.15740741E-5,.149599E9            BLOK0016
C     C(9)= SPD OF LIGHT     C(10)= SBEV  CONSTANT FOR STEPSIZE FORMULA BLOK0017
     2,299774., .16/                                                    BLOK0018
C     NAMES OF BODIES                                                   BLOK0019
      DATA (BODC(I,1),I=1,10)/5HEARTH,4HMOON,3HSUN,5HVENUS,4HMARS,      BLOK0020
     16HSATURN,6HJUPITR,6HBODY 8,6HBODY 9,6HBOD 10/                     BLOK0021
C     GRAVITY CONSTANTS                                                 BLOK0022
      DATA(BODC(I,2),I=1,7)/398603.2,4900.7588,.13271545E12,324769.5,   BLOK0023
     142977.8,3.791870E7,1.267106E8/                                    BLOK0024
C     SEMI-MAJOR AXIS                                                   BLOK0025
      DATA(BODC(I,3),I=1,7)/6378.165,1738.,695500.,6200.,3400.,60400.   BLOK0026
     1,71350./                                                          BLOK0027
C     SEMI-MINOR AXIS                                                   BLOK0028
      DATA(BODC(I,4),I=1,7)/6356.5838,1738.,695500.,6200.,3400.,54050.  BLOK0029
     1,66600./                                                          BLOK0030
C     SPHERE OF INFLUENCE (KM)                                          BLOK0031
      DATA (BODC(I,5),I=1,7)/925000.,66000.,1.E10,616000.,565000.       BLOK0032
     1,.546E8,.48E8/                                                    BLOK0033
C     ROTATION RATES (RAD/SEC)                                          BLOK0034
      DATA (BODC(I,7),I=1,7)/.72921152E-4,  .266169952E-5,  0.0         BLOK0035
     1, 0.0   ,.70882177E-4   ,.170553347E-3   ,.177491110E-3/          BLOK0036
C     MAX STEP SIZE   (SEC)                                             BLOK0037
      DATA (BODC(I,6),I=1,7)/43200.      ,21600.    ,864000.   ,43200.  BLOK0038
     1,43200.   ,86400.    ,172800./                                    BLOK0039
C     INTERPOLATION INTERVALS                                           BLOK0040
      DATA (BODC(I,8),I=1,7)/7*40./                                     BLOK0041
      DATAXMIN,XMAX,RDELAY/ 5., 43200., 10. /                           BLOK0042
      DATA (C(I),I=91,111)/21*0./                                       BLOK0043
C     EXTRA OUTPUT KEY                                                  BLOK0044
C     REFRACTION CONSTANTS                                              BLOK0045
      DATA (C(I),I=115,117)/0.,.34E-3,.138771E-3/                       BLOK0046
      DATA (C(I),I=118,700)/583*0./                                     BLOK0047
      END                                                               BLOK0048
```

/MESCOM/CMES(510)

No block data.

MESCOM is used for the transmission of output edited data to tape. It is written as the second record of each record pair on the edited data tapes. Subroutine POLYFT lifts data from DATCOM and transfers it to MESCOM for output in records of 85 time points. The units of the observables are unchanged from those of DATCOM. MESCOM is also used by subroutines MIXIT, EGRESS, and SCANIT.

/MESCOM/CMES(510)

Data Output Buffer

LOC	Symbol	Dimension	Description
1	XMES	(85, 6)	XMES (I, 1) Quality indicator 0: good data 1: Angles bad 2: Range bad 4: Doppler bad > 4: Combination of above XMES (I, 2) Time tag, seconds from start of arc XMES (I, 3) X, azimuth, or hour angle, radians XMES (I, 4) Y, elevation, or declination, radians XMES (I, 5) Range, km or seconds XMES (I, 6) Doppler, km or seconds

/MIXCOM/IMIX(4)

No block data.

MIXCOM is used for communication of tape handling information between the MAIN program and the merging subroutine MIXIT and the output subroutine EGRESS. It contains the tape unit numbers for the output tape and the previously edited data input tape as well as keys to indicate whether edited data are present on the edited data input tape NIN and the scratch tape 12.

/MIXCOM/IMIX(4)

LOC	Symbol	Dimension	Description
1	LONNIN	ℓ	Edited data present on tape NIN T: Yes F: No
2	LON12	ℓ	Edited data present on tape 12 T: Yes F: No
3	NIN		Edited data tape input tape unit number. Initially = 11
4	NOUT		Edited data tape output tape unit number. Initially and terminally = 10

/MLECOM/CMLE (1070)

Block data: Deck MC13B5

MLECOM is used for communication between the principal subroutines of the data start link of the DCP. It contains quantities describing the data to be used for the maximum likelihood estimation, option switches set during initialization, and temporary storage of the state and associated data.

/MLECOM/CMLE(1070)

Table 1

LOC	Symbol	Dimension	Definition
1	TSTART		Time from ONTIME to start of data start interval.
2	TFINAL		Time from ONTIME to end of data start interval.
3	TMEAN	d	Data start epoch, seconds from ONTIME (ST). Set to the mean time of the data to be used.
5	STIMR1	d	Seconds from 1950 Jan. 0.0 ST at TMEAN.
7	UTIMR1	d	UT at ST = STIMR1
9	ETIMR1	d	ET at ST = STIMR1
11	ETIMV1	d	ET at vehicle retransmission of the signal received at STIMR1
13	STIMR	d	ST of data point being processed.
15	UTIMR	d	UT at STIMR
17	UTIMR	d	ET at STIMR
19	ETIMV	d	ET at vehicle retransmission of signal received at STIMR.
21	UTIMT	d	UT at transmission by the transmitting station.
23	ETIMT	d	ET at UTIMT
25	DELAY	d	Down-leg transmission delay, ETIMR-ETIMV

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
27	STBIAS	d(2)	Receiving station clock bias and bias rate, STIMR-UTIMR at UTIMR = 0 and $\frac{d}{dt}$ (STIMR-UTIMR).
31	OBIAS	d(4)	Measurement biases.
39	DATAT	d(4)	Measurement values at STIMR.
47	Q	d(4)	Variances of measurement random errors.
55	QI	d(4)	QI(I) = 1/Q(I)
63	TD2C	d(3, 3)	Transform from true equator, equinox of date at ETIMR1 to C-frame.
81	TB12C	d(3, 3)	Transform from B- to C-frame at UTIMR1.
99	X1	d(6)	Vehicle state with respect to Earth, C-frame, at ETIMV1.
111	XA1	d(6)	Vehicle state with respect to the central body, C-frame, at ETIMV1.
123	XM	d(6)	Vehicle state in measurement coordinates at ETIMV1.
135	PX12M	d(6, 6)	$\partial X1/\partial XM$
207	YMOD	d(4)	Modulus of measurement values.
215	YR	d(4)	Measurement residuals, $y-\hat{y}$.
223	PI	d(6, 6)	Information matrix, $H^TQ^{-1}H$.
295	HY	d(6)	Projected residuals, $H^TQ^{-1}(y-\hat{y})$.
307	TEST	d(20)	Vector of residual sums (see Table 2).
347	COEF	d(2, 4)	Coefficients of measurement polynomials.
363	AMU	d	Central body gravitational constant.
365	CTAU	d	Constant doppler interval or count, from the edited data tape.
367	DRA	d	Nutation in right ascension at ETIMR1.
369	Spares	(12)	------------------------------------
381	ILMEAS	ℓ(4, 4)	Measurement inclusion options versus measurement number, type (T if measurement is included).
397	LMEAS	ℓ(4)	Measurement options for this type, this process.

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
401	LDATA	ℓ(4)	Measurement options for this time point.
405	LAUTO	ℓ	Internal automatic retry option (see KAUTO).
406	LMLE	ℓ	Maximum likelihood convergence indicator, T if converged.
407	LRESDL	ℓ	Residual switch, T output only, F include in information matrix and HY.
408	Spares	(3)	----------------------------------
411	INROT	(3,4)	Rotation indices for each measurement type (see MEAS2X).
423	NROT	(4)	Rotation indices for current measurement type.
427	NYR	(4)	Number of measurements included in statistics.
431	NSTATE		Number of state elements being estimated.
432	NPTSTT		Number of time points loaded in DATA.
433	IQUAL		Quality indicator value for full set of measurements.
434	ITIME		Column counter of current time in DATA.
435	ILBM	(12,4)	BCI measurement labels.
483	ILBS	(12,4)	BCI measurement coordinate labels.
531	LBM	(12)	} labels for this measurement type.
543	LBS	(12)	} labels for this measurement type.
555	LBALG	(2)	BCI antenna alignment label.
557	LBDPL		BCI label for 3-way doppler.
558	LBTAU	(2)	BCI label for doppler mode.
560	LBTYP	(4)	BCI measurement type labels.
564	Spares	(7)	----------------------------------
571	DATA	(5,100)	Interval data buffer, DATA (1,I) = time tag for Ith data "point." DATA (M+1,I) = Mth measurement value.

TEST d(20)

Table 2

Component	Symbol	Definition
1-4	TEST(M)	Sum of the squares of the residual for each measurement.
5	E	Weighted sum of the squares of the residuals ΣQI(M)*TEST(M).
6	EBAR	Mean weighted sum of the squares, ΣQI(M)*TEST(M)/NYR(M).
7	EDEL	Fractional change in E, (E-ELAST)/E.
8	EHAT	Value of E predicted for next trial.
9	EDHAT	Value of EDEL predicted for next trial.
10	Spare	--
11-16	TEST(I+10)	Values of TEST(I+4) (E, etc.) on previous trial.
17	EHATX	Value of EHAT without constraint on the differential correction.
18	EDHATX	Value of EDHAT without constraint.
19-20	Spares	--

```
$IBFTC MC13B5  M94,NODD,LIST
      BLOCK DATA                                                        MLEC0001
C     MAIN COMMON FOR DATA START LINK                                   MLEC0002
C                                                                       MLEC0003
      COMMON             /MLECOM/CMLE(1070)                             MLEC0004
C                                                                       MLEC0005
      DOUBLE PRECISION   YMOD    (4)                                    MLEC0006
      LOGICAL            ILMEAS(4,4)                                    MLEC0007
      DIMENSION          INROT(3,4)                                     MLEC0008
     1                  ,ILBM (12,4) ,LBALG (2) ,LBTYP (4)              MLEC0009
     2                  ,ILBS (12,4) ,LBTAU (2)                         MLEC0010
C                                                                       MLEC0011
      EQUIVALENCE               (CMLE(435),ILBM   ) ,(CMLE(557),LBDPL )  MLEC0012
     1                         ,(CMLE(483),ILBS   ) ,(CMLE(558),LBTAU )  MLEC0013
     2                         ,(CMLE(411),INROT  ) ,(CMLE(560),LBTYP )  MLEC0014
     3     ,(CMLE(381),ILMEAS) ,(CMLE(555),LBALG  ) ,(CMLE(207),YMOD  )  MLEC0015
C                                                                       MLEC0016
C     MEASUREMENT AMBIGUITIES                                           MLEC0017
      DATA    YMOD  / 2*6.2831853071795864 ,2*1.D30 /                   MLEC0018
C                                                                       MLEC0019
C     MEASUREMENT INCLUSION KEYS                                        MLEC0020
      DATA    ILMEAS/ T,T,T,F  ,T,T,T,T  ,T,T,T,T  ,T,T,F,T /           MLEC0021
C                                                                       MLEC0022
C     ROTATION INDICES (FOR MEAS2X)                                     MLEC0023
      DATA    INROT / 1,2,3  ,-3,-2,1  ,-3,1,2  ,1,-2,3 /               MLEC0024
C                                                                       MLEC0025
C     OUTPUT LABELS                                                     MLEC0026
      DATA ILBM /6HAZIMUT,6HELEVAT,6HRANGE ,6H      ,6HH     ,6HION     MLEC0027
     1          ,6H      ,6H      ,6HRAD   ,6HRAD   ,6HKM    ,6H        MLEC0028
     2          ,6HX     ,6HY     ,6H2-WAY ,6HDOPPLE,6H      ,6H        MLEC0029
     2          ,6HRANGE ,6HR     ,6HRAD   ,6HRAD   ,6HSEC   ,6HCYCLES  MLEC0030
     3          ,6HX     ,6HY     ,6H2-WAY ,6HDOPPLE,6H      ,6H        MLEC0031
     3          ,6HRANGE ,6HR     ,6HRAD   ,6HRAD   ,6HKM    ,6HCYCLES  MLEC0032
     4          ,6HHOUR A,6HDECLIN,6H      ,6HDOPPLE,6HNGLE  ,6HATION   MLEC0033
     4          ,6H      ,6HR     ,6HRAD   ,6HRAD   ,6H      ,6HCYCLES /MLEC0034
      DATA ILBS /6HRANGE ,6HAZIMUT,6HELEVAT,6HR-RATE,6HAZ-RAT,6HEL-RAT  MLEC0035
     1          ,6H      ,6HH     ,6HION   ,6H      ,6HE     ,6HE       MLEC0036
     2          ,6HRANGE ,6HX     ,6HY     ,6HR-RATE,6HX-RATE,6HY-RATE  MLEC0037
     3          ,6HRANGE ,6HX     ,6HY     ,6HR-RATE,6HX-RATE,6HY-RATE  MLEC0038
     2          ,6H      ,6H      ,6H      ,6H      ,6H      ,6H        MLEC0039
     3          ,6H      ,6H      ,6H      ,6H      ,6H      ,6H        MLEC0040
     4          ,6HRANGE ,6HR.A.  ,6HDECLIN,6HR-RATE,6HR.A.-R,6HDEC-RA  MLEC0041
     4          ,6H      ,6H      ,6HATION ,6H      ,6HATE   ,6HTE     /MLEC0042
      DATA LBALG/6HN-S    ,6HE-W    /                                   MLEC0043
      DATA LBDPL/6HR-3WAY/                                              MLEC0044
      DATA LBTAU/6HINTRVL,6HCOUNT /                                     MLEC0045
      DATA LBTYP/6HC-BAND,6HG-RR    ,6HUSBS   ,6HDSIF   /               MLEC0046
C                                                                       MLEC0047
      END
```

/OUTCOM/COUT (40)

No block data.

OUTCOM is used by nearly every subroutine in the Tracking Data Editor. It is used for the transmission of output edited data to tape. It is written as the first record of each record pair on the edited data tapes. It contains key information about the associated data record written from MESCOM. It is also written without an associated data record as the last record on an edited data tape, in which case the end of tape condition is signalled by a flag in one of the words of the record.

/OUTCOM/COUT (40)
Data Identification Buffer

LOC	Symbol	Dimension	Description
1	NREC		Record pair counter
2	NEOT		End of tape key -1, First record pair +1, Last record 0, Other
3	OBSNAM		Receiving station name
4	TRANAM		Transmitting station name
5	NRCD		Record pair counter for current arc
6	NPTS		Number of time points in data record
7	KONT		0; First record pair for this arc 1; Continuation record pair
8	MTYPE		Measurement system type 1; C-Band 2; Goddard Range & Range Rate 3; Unified S-Band 4; DSIF (JPL)
9	NALIGN		Antenna size (for USBS only) 1: 30-foot, principal axis north 2: 85-foot, principal axis east
10	MODE		Doppler mode 2: Two-way 3: Three-way
11	DELT		Time interval between time points, seconds

Data Identification Buffer (continued)

LOC	Symbol	Dimension	Description
12	KTAU		Doppler type: 0: Doppler count is fixed in COUT (19), data contains time. 1: Doppler interval is fixed in COUT (19), data contains counts.
13	ONTIME	d	Start of data arc, seconds from 1950
15	TFIRST	d	Start of record, seconds from 1950
17	TLAST	d	End of record, seconds from 1950
19	TAU	d	Fixed doppler count or interval (sec) see KTAU, COUT (12)
21	FTR	d	Transmitter carrier freq. Hz
23	CRI	d	Refraction constants
25	CR2	d	
27	DR	d	Range ambiguity, units of range
29	BIAS	d	Fixed doppler offset bias, Hz
31	RETR	d	Transponder retransmission/reception ratio, e.g.: = 240/221 for USBS
33	NB	(4)	Outlier or missing data point count, four observables
37	SD	(4)	Sample standard deviations, four observables, units of observables, computed over entire arc.

/POTCOM/PLOTC(42)

No block data.

POTCOM is used to communicate data to the plot routine RSPLOT. It is also used by RSPLOT to store quantities which must be saved if linking within the residual link.

/POTCOM/PLOTC(42)

Loc	Symbol	Dimension	Description
1	TIME		Time (from initialization time) of present RSPLOT call, in seconds.
2	DATA	(4)	Up to 4 data values to be plotted at this time.
6	KDATA	(4)	If KDATA(I) = 1, plot DATA (I)
10	DT		Time scale increment, in seconds.
11	DATMAX	(4)	DATMAX (I) is the maximum value of DATA (I) to be plotted on scale.
15	LABEL	(12)	Each data type is identified on the plot by a one-character BCI symbol. The symbol is identified as to data type by preliminary output of a label which may be up to 12 characters in length. For data type I, I=1, 2, , 4 LABEL (I) is the 1st word of the BCI label. LABEL (I+4) is the 2nd word of the BCI label. LABEL(I+8) is the BCI label of units of DATA (I).
27	SYMBOL	(4)	and SYMBOL(I) is the BCI character used in the plot.
31	DORD	(4)	DORD (I) is used in placing the symbol for DATA(I) in the correct plot position.
35	TM	(3)	TM(1) = even time interval point of the last entry to RSPLOT. TM(2) = TM(1) - DT/2. TM(3) = TM(1) + DT/2.

38	LINE		Counter for the number of intervals (DT), to control output of a horizontal line every 10 intervals.
39	LINE2		Counter for the number of intervals, to control output of a time coordinate every 5 lines.
40	LINE3		Counter for the vertical axis label.
41	LTM		Temporary storage for a character of the vertical axis label.
42	N1		Used in assigned GO TO which controls output of vertical axis label.

/RSCOM/KPTIME(8), RLSAVE(56)

Block data: Deck MC135B

RSCOM contains the maximum values of the scales on which data is to be plotted. It is used by RESOUT in setting the data maximums for RSPLOT.

TABLE 1
/RSCOM/KPTIME(8)

Loc	Symbol	Dimension	Description
1	KPTIME		Logical variable if KPTIME is TRUE, the time interval for the plots is taken as PTIME. Otherwise, it is computed from BUFDAT.
2	PTIME		See KPTIME
3	KPMAX		If KPMAX is TRUE, the maximum value of data type M to be plotted on scale is taken as PMAX(M+1). Otherwise, it is computed to be PMAX(1) + SE(M+10, INDSTA).
4	PMAX	5	See KPMAX

TABLE 2

RLSAVE (56)

RESOUT Initialization Data

Loc	Symbol	Dimension	Definition
1	OBIAS	d(4)	Measurement biases
9	YMOD	d(4)	Measurement moduli
17	TFIRST	d	First station on-time, seconds from 1950 Jan 0.0
19	STPTIM		Process stop time, seconds from TFIRST
20	TREF		Start time for current data arc, seconds from TFIRST
21	Spare		------
22	KOUTL	ℓ	Logical variable for inclusion of outliers
23	LSKIP	ℓ(4)	Measurement skip options
27	NSUM	(4)	Number of measurements included in SUM, SUMSQ
31	SUM	(4)	Sum of residuals, each type
35	SUMSQ	(4)	Sum of squares of residuals, each type
39	NASGN	(18)	Permanent transfer addresses (for assigned "GO TO's")

```
$IBFTC MC135B  LIST,NODD
      BLOCK DATA                                                        RSCM0001
      COMMON            /RSCOM/KPTIM(8),RLSAVE(56)                      RSCM0002
      LOGICAL            KPMAX ,KPTIME                                  RSCM0003
      EQUIVALENCE       (KPTIM( 1),KPTIME) ,(KPTIM( 2),PTIME )          RSCM0004
     1                  ,(KPTIM( 3),KPMAX ) ,(KPTIM( 4),PMAX  )         RSCM0005
      DATA KPMAX,KPTIME/T,T/                                            RSCM0006
      DATA PTIME/60./                                                   RSCM0007
      DATA PMAX/1.,0.1,0.1,2*100./                                      RSCM0008
      END
```

/RSLCOM/INDRSL(16), BUFRSL(18, 14)

No block data.

RSLCOM is used to keep track of output quantities related to the data and residuals. This data must be saved until the entire residual process is completed since the plotting, if requested, is output line by line for each time point. All other printer output must therefore be deferred until the plots are finished.

TABLE 1

INDRSL(16)

Loc	Symbol	Dimension	Description
1	NRSPOS		Present position on the residual tape (11)
2	NREOT		Record-type descriptor for the residual tape.
3	NRKONT		Data arc descriptor (=0 for a record not the last record of the arc) for the residual tape.
4	NRBLK		Number of INDRSL, BUFRSL, pairs written on residual tape.
5	NRPTS		Number of time points used.
6	NRSSTA		Station name for these residuals.
7	NREST		Number of estimation tape record pair used to define state for this residual process.
8	MTYPE		Data type. MTYPE = 1 C-Band 2 Goddard R, R 3 United S-Band 4 DSIF
9	TRON	d	Time to start computing residuals seconds from 1950 Jan 0.0 ST
11	TROFF	d	Time to stop computing residuals seconds from 1950 Jan 0.0 ST. If TROFF=0, this time is the end of the data arc.

13	TRFRST	d	Time of the 1st residual in BUFRSL (in a block of 14 points on the tape), seconds from 1950 Jan 0.0 ST.
15	TRLAST	d	Time of the last residual of the block of 14, seconds from 1950 Jan 0.0 ST.

TABLE 2

BUFRSL(18, 14)

The residuals are collected in BUFRSL until 14 have been accumulated. They are put on tape (11) either as temporary storage before output or because the tape option was requested. For each point I, I=1, 14 the following is a table of BUFRSL(, I)

Loc	Dimension	Definition
1		Quality word. This is a fixed point number, $=\sum 10^m$ for all M for which a valid residual appears for data type M.
2		Time of the data, seconds from 1950 Jan 0.0 ST.
3		Residual, data type 1
4		Residual, data type 2
5		Residual, data type 3
6		Residual, data type 4
7	d	X
9	d	Y
11	d	Z
13	d	$\dot{X}$
15	d	$\dot{Y}$
17	d	$\dot{Z}$

State at time BUFRSL(2, I) (applies to Loc 7–17)

/SBFCOM/CSBF(18N+12)

No block data.

SBFCOM is used for communication between the propagation and estimation links. It contains the quantities computed during the trajectory integration which are required in the processing of data.

The dimension of the arrays in SBFCOM depend on M, the maximum permissible number of motion parameters in the state vector. The dimension, N, is

$$N = 27 + 3M$$

The dimensions of the arrays and their location in CSBF are given as functions of N in the table below. The value of M used must be compiled into MAXQ of DCPCOM (see also DQDCOM). Changes in M require the recompilation of the following subroutines which reference SBFCOM.

Subroutine	Deckname
DIFCOR	MC133H
TRAJDP	MC133F
OUTTRJ	MC1327
ESTMAT	MC133J
ESTOUT	MC1330
UPDATP	MC138A

SBFCOM is loaded in the DIFCOR link of the DCP. It is loaded in the RSIDUL link under the name SB3COM, with fixed dimensions RBF(6), VBF (6, 8).

/SBFCOM/CSBF (18N+12)

General		N=45 (M=6)			
Location	Dimension	Loc	Dimension	Array	Definition
1		1		KBF	Ordinate counter.
2		2		NTP	Process type index. = 1: propagate = 2: establish anchor point = 3: estimate/shift anchor point
3		3		NPT1	Data point counter value at iteration start.
4		4		NPT2	Data point counter value at iteration end.
5		5		LCB	Current central body.
6		6		ITER	Iteration counter.
7	d	7	d	HBF	Time increment for interpolation formulas.
9	d	9	d	TBF	Interpolation interval initial time.
11	d	11	d	TFF	Interpolation interval final time, TFF = TBF + 7 HBF.
13	d(N)	13	d(45)	RBF	Initial "position" vector.
2N+13	d(N, 8)	103	d(45, 8)	VBF	"Velocity" ordinates, interpolation coefficients.

/SB3COM/CSBF (120)

No block data.

SB3COM is used for communication between the sub-links of the RSIDUL link of the DCP. It contains the quantities computed during trajectory integration which are required for the computation of residuals. It is identical to SBFCOM except that the dimensions of RBF, VBF do not depend on the number of equation of motion parameters. For SB3COM, N=6 always.

/STNCOM/CSTN(7250)

Block Data: Deck MC1377

STNCOM is used for the storage of nominal station parameters, their standard deviations and correlation coefficients. It is used by SETSTA (only) in constructing the working arrays for the stations for which data is to be processed.

STNCOM may be changed only by recompilation of the block data. The working array, however, may be changed by overlay input at the case level (see References 1 and 2).

The dimensions of the arrays in STNCOM allow inclusion of up to 50 stations in the block data. If a larger station library is required and space is available, STNCOM may be redimensioned by replacing each "50" on card SSTA0041 by the new dimension.

/STNCOM/CSTN(7250)
Nominal Station Parameters

Loc	Symbol	Dimension	Description
1	STANAM	(50)	Station names
51	STALOC	d(3, 50)	Station locations Latitude (degrees) Longitude (degrees) Altitude (meters)
351	STACOR	d(69, 50)	Station location and measurement bias errors, standard deviations, and normalized correlations.

```
$IBFTC MC1377  XR3,M94,NODD,LIST
      BLOCK DATA
C     TRACKING STATION BLOCK DATA
C
      COMMON              /STNCOM/CSTN(7250)
      DIMENSION            STANAM  (50)
      DOUBLE PRECISION     STALOC(3,50) ,STACOR(69,50)
      EQUIVALENCE         (CSTN(  1),STANAM) ,(CSTN( 51),STALOC)
     1                    ,(CSTN(351),STACOR)
C
C     STATION NAMES
      DATA  STANAM/6HMERRIT ,6HBAHAMA ,6HANTIGU ,6HBERMUD ,6HCANARY
     1            ,6HASCENS ,6HCRNRVN ,6HGUAM   ,6HHAWAII ,6HGUAYMS
     2            ,6HCORPUS ,6HMADRID ,6HMADRIX ,6HCANBRA ,6HCANBRX
     3            ,6HGLDSTN ,6HGLDSTX ,6HVANGRD ,6HREDSTN ,6HMERCRY
     4            ,6HHNTSVL ,6HWTRTWN ,6HKENEDY ,6HPATRIK ,6HSALVDR
     5            ,6HGRTURK ,6HPRETOR ,6HARGUEL ,6HWSANDS ,6HELGIN
     6            ,6HJOBURG ,6HWOMERA ,6HROSRAN ,6HMADGAR ,6HCARVON
     7            ,6HSNTAGR ,6HULASKR
     8
     9         ,13*6HSPARES                                     /
C
C     STATION LOCATIONS
C                        LATITUDES (DEGREES)
      DATA  (STALOC(1,I),I=1,50)
     1/  28.42486D0,  26.63636D0,  17.14380D0,  32.34777D0,  27.74486D0
     2,  -7.97299D0, -24.90639D0,  13.58333D0,  22.12527D0,  27.95841D0
     3,  27.65539D0,  40.41667D0,  40.41667D0, -35.31153D0, -35.31153D0
     4,  35.38964D0,  35.38964D0,   0.      D0,   0.     D0,   0.     D0
     5,   0.     D0,   0.     D0,  28.48177D0,  28.22655D0,  24.11899D0
     6,  21.46291D0, -25.94372D0,  34.58290D0,  32.35822D0,  30.42177D0
     7, -25.88735D0, -31.38287D0,  35.20014D0, -19.02028D0, -24.87500D0
     8, -33.14948D0,  64.97683D0,
     9
     1   13*0.D0/
C                        LONGITUDES (DEGREES)
      DATA  (STALOC(2,I),I=1,50)
     1/ -80.66442D0, -78.26772D0,  17.14380D0, -64.65364D0, -15.60200D0
     2, -14.40169D0, 113.72369D0, 144.92500D0,-159.66762D0,-110.72079D0
     3, -95.36358D0,  -3.66667D0,  -3.66667D0, 149.13583D0, 149.13583D0
     4,-116.84878D0,-116.84878D0,   0.     D0,   0.     D0,   0.     D0
     5,   0.     D0,   0.     D0, -80.57651D0, -80.59929D0, -74.52074D0
     6, -71.13204D0,  28.35850D0,-120.56115D0,-106.36956D0, -86.79811D0
     7,  27.64878D0, 136.88502D0, 271.12790D0,  47.30278D0, 113.70833D0
     8, 289.33091D0, 212.48490D0,
     9
     1   13*0.D0/
C                        ALTITUDES (METERS)
      DATA  (STALOC(3,I),I=1,50)
     1/    9.  D0,    13.  D0,    26.  D0,     3.  D0,    36.  D0
     2,  143.  D0,    58.  D0,    20.  D0,  1142.  D0,    18.  D0
     3,   50.  D0,    50.  D0,    50.  D0,    50.  D0,    50.  D0
     4, 1031.  D0,  1031.  D0,     0.  D0,     0.  D0,     0.  D0
     5,    0.  D0,     0.  D0,    14.  D0,    15.  D0,     3.  D0
     6,   25.  D0,  1618.  D0,   646.  D0,  1232.  D0,    28.  D0
     7, 1381.92D0,   150.8 D0,   880.87D0,  1392.02D0,    75.01D0
     8,  694.94D0,   294.44D0,
     9
     1   13*0.D0/
C
C     STATION ERRORS
C                                  NORTHING (KM)
      DATA  (STACOR( 1,I),I=1,50)/                    50* 0.D0   /
C                                  EASTING  (KM)
      DATA  (STACOR( 2,I),I=1,50)/                    50* 0.D0   /
C                                  DOWN     (KM)
      DATA  (STACOR( 3,I),I=1,50)/                    50* 0.D0   /
C                                  CLOCK BIAS (SEC)
      DATA  (STACOR( 4,I),I=1,50)/                    50* 0.D0   /
C                                  CLOCK RATE (SEC/SEC)
      DATA  (STACOR( 5,I),I=1,50)/                    50* 0.D0   /
C                                  SPDLT (KM/SEC)
      DATA  (STACOR( 6,I),I=1,50)/                    50* 0.D0   /
C                                  ANGLE 1 (RAD)
      DATA  (STACOR( 7,I),I=1,50)/                    50* 0.D0   /
```

```
C                                   ANGLE 2 (RAD)
      DATA  (STACOR( 8,I),I=1,50)/                    50* 0.D0   /
C                                   RANGE    (KM)
      DATA  (STACOR( 9,I),I=1,50)/                    50* 0.D0   /
C                                   DOPPLER (COUNTS)
      DATA  (STACOR(10,I),I=1,50)/                    50* 0.D0   /
C
C     STANDARD DEVIATIONS
C                                   NORTHING (KM)
      DATA  (STACOR(11,I),I=1,50)/                    50*10.D-3 /
C                                   EASTING  (KM)
      DATA  (STACOR(12,I),I=1,50)/                    50*10.D-3 /
C                                   DOWN     (KM)
      DATA  (STACOR(13,I),I=1,50)/                    50*10.D-3 /
C                                   CLOCK BIAS (SEC)
      DATA  (STACOR(14,I),I=1,50)/                    50* 1.D-3 /
C                                   CLOCK RATE (SEC/SEC)
      DATA  (STACOR(15,I),I=1,50)/                    50* 1.D-12/
C                                   SPDLT (KM/SEC)
      DATA  (STACOR(16,I),I=1,50)/                    50*10.D0   /
C                                   ANGLE 1 (RAD)
      DATA  (STACOR(17,I),I=1,50)/                    50* 1.D-3 /
C                                   ANGLE 2 (RAD)
      DATA  (STACOR(18,I),I=1,50)/                    50* 1.D-3 /
C                                   RANGE    (KM)
      DATA  (STACOR(19,I),I=1,50)/                    50* 1.D-1 /
C                                   DOPPLER (COUNTS)
      DATA  (STACOR(20,I),I=1,50)/                    50* 1.D0   /
C
C     CORRELATIONS
      DATA ((STACOR(I,J),J=1,50),I=21,65)/          2250* 0.D0   /
C
C     STANDARD DEVIATIONS IN MEAS RANDOM ERRORS (RAD,RAD,KM,COUNTS)
      DATA ((STACOR(I,J),J=1,50),I=66,69)/           100* 2.D-3 ,
     1                                                50* .3D-1 ,
     2                                                50*20.D0   /
      END
```

/STPCOM/CSTP(58)

No Block Data.

STPCOM is used for the storage of constants computed by the initialization subroutine STEPDI and the functions of the regularized incremental eccentric anomaly used in the solution of Kepler's equation by STEPDT.

/STPCOM/CSTP(58)
Conic Step Functions

Loc	Symbol	Dimension	Definition
1	TI	d	Initial time, seconds ET
3	XI	d(6)	Initial position and velocity, R_o, V_o
15	SU	d	$\sqrt{\mu}$, μ = gravitational constant
17	RI	d	$r_o = \|R_o\|$
19	AI	d	1/a, $\|a\|$ = semi-major axis of the conic
21	SA	d	$\sqrt{1/\ \|a\|}$
23	DI	d	$d_o = R_o \cdot V_o/\sqrt{\mu}$
25	SR	d	$\sqrt{\mu}/r_o$
27	PR	d	Orbital period (for ellipses)
29	HP	d	PR/2
31	OM	d	$(1 - r_o/a)$
33	EP	d	Convergence limit for STEPDT
35	R	d	$r = \|R\|$
37	G	d(4)	Universal functions, G_o, G_1, G_2, G_3
45	B	d	Regularized anomaly, β
47	FB	d	$F(\beta) = G_3 + d_oG_2 + r_oG_1$
49	D	d	$d = d_oG_o + (1 - r_o/a)G_1$
51	F	d(4)	R, V coefficients, f, g, $\dot{f}$, $\dot{g}$

/SUMCOM/SUMARY (56)

No block data.

SUMCOM is used to contain summary information about the output edited data tape. It is initially loaded by the MAIN program and is kept current by the decoding subroutines CBTEST, GRTEST, SRTEST, and DSTEST. It is also used by subroutines EGRESS and SCANIT. It is written as the first record on the output edited data tape.

/SUMCOM/SUMARY (56)
Data Tape Header Record

Loc	Symbol	Dimension	Description
1	HEADER	(11)	BCD tape header
12	NUMSTA		Total number of stations on tape
13	NAMSTA	(20)	Names of all stations on tape
33	TSTART	d	Time of first data on tape or earlier, seconds from 1950
35	TSTOP	d	Time of last data on tape or later, seconds from 1950
37	STIMNX	(20)	Time of earliest appearance of each station on tape as a receiving station (or earlier), seconds from 1950

/TRJCOM/CTRJ(246)

No block data.

TRJCOM is used for the working storage of quantities required to be saved or communicated during trajectory computation. It is loaded in the DIFCOR link, and is relabelled TRSCOM for the MLESTT link and TR3COM for the RSIDUL link.

/TRJCOM/CTRJ(246)

Loc	Symbol	Dimension	Definition
1	ETFMS	d	ET of last call of DPFMRS
3	UTIMV	d	Current universal time, seconds from 1950 Jan 0.0 UT
5	TC2D0	d(3, 3)	Transformation, mean equator, equinox of 1950.0 (C), to mean equator, equinox of date (D0)
23	TC2D	d(3, 3)	Transformation, to true equator, equinox of date (D)
41	TB2D	d(3, 3)	Transformation, central body-fixed (B), to true equator, equinox of date
59	TB2C	d(3, 3)	Transformation, central body-fixed to mean equator, equinox of 1950.0
77	CBOD	d(37)	Central body working storage
151	RC	d(3)	Vehicle position relative to the central body, C-frame components
157	VC	d(3)	Vehicle velocity relative to the central body, C-frame components
163	RCONIC	d(3)	Encke reference position
169	VCONIC	d(3)	Encke reference velocity
175	PCONIC	d(6, 6)	Conic transition matrix

CBOD d (37)

Central Body Working Storage

Location		Symbol	Dimension	Definition
CTRJ	CBOD			
77	1	CBN		Central body name
79	2	CBP	d	Sphere of influence
81	3	CBA	d	Equatorial radius
83	4	CBB	d	Polar radius
85	5	CBU	d	Gravitational constant
87	6	CBO	d	Angular velocity
89	7	DRA	d	Nutation in right ascension (Earth) of 1950.0 right ascension of the prime meridian (planets)
91	8	HRA	d	Hour angle of the vernal equinox
93	9	OBL	d	Mean obliquity of the ecliptic (Earth) or inclination of Earth's ecliptic on the planet's equator
95	10	DLN	d	Nutation in longitude (Earth) or 1950.0 longitude of the ascending node of the ecliptic on the equator (planets)
97	11	OBT	d	True obliquity (Earth) or Earth's mean obliquity of 1950.0 (planets)
99	12	NZH		Highest order zonal harmonic
101	13	NTH		Highest degree tesseral harmonic
103	14	ZH	d(6)	Zonal harmonic coefficients
115	20	TH	d(18)	Tesseral harmonic coefficients

/TRKCOM/CTRK(700)

Block data: Deck MC134S

TRKCOM is used for storage of tracking station descriptive data. It is loaded by a Block Data subroutine, but may be altered by overlay input at the beginning of the program run.

/TRKCOM/CTRK (700)
Tracking Station Identification

Loc	Symbol	Dimension	Description
1	STANAM	(50)	Station names
51	KODSTA	(50)	Station ID codes
101	MOUNT	(50)	USBS antenna size 1, 30-foot, principal axis north 2, 85-foot, principal axis east
151	TRF	d(50)	Transmitter frequency Hz
251	C1	d(50)	Refraction constants
351	C2	d(50)	Refraction constants
451	BIAS	d(50)	Doppler fixed offset bias, Hz, used only for USBS
551	RETR	d(50)	Transponder retransmission/reception ratio. Used for USBS and for Goddard Range - Range Rate in S-Band mode.
651	PAIR	(50)	Associated transmitting station names.

Use of the various arrays by the TDEP

STANAM

Tracking stations are identified on the Edited Data Tape by six-letter names rather than by numbers. This is the array of possible names.

KODSTA

Tracking stations are identified on the Raw Data Tapes by two-digit decimal numbers. The TDEP reads the number from the raw data tape, searches for a match in KODSTA, picks up the corresponding name in STANAM, and puts this name on the edited data tape. Exception: Subroutine GRTEST (Goddard System Data) reads the number from the raw data tape, reassigns a new number, and then uses the reassigned value to search for a match in KODSTA. This is necessary to avoid conflicts due to duplication of station ID numbers. The reassignments are:

On Tape	Reassigned
26	30
22	31
52	32
27	33
28	34

Care must be exercised if subroutine GRTEST is to be altered to change either the list of acceptable numbers from the raw data or the reassigned values. Reassigned values less than 33 indicate to subroutine GRTEST that the tracker is a regular system, whereas numbers greater than or equal to 33 indicate a hybrid system tracker.

NALIGN

This array has meaning only for USBS data and is used only by subroutine SBTEST.

FTR

This array is meaningless for C-Band data and is ignored by CBTEST. It is also ignored by GRTEST, which contains its own list of possible carrier frequencies.

C1 and C2

These are used by all data subroutines.

BIAS

This array is meaningless for C-Band data and is ignored by CBTEST. It is also ignored by GRTEST, which contains its own list of bias frequencies.

RETR

This array is meaningless for C-Band data and is ignored by CBTEST. It is always used by SBTEST. It is used by GRTEST for S-Band frequencies (but not VHF) but the Differential Correction Program (Subroutines DSDAT and DSTATP) were altered to ignore this value for Goddard data because the Goddard System receivers shift the received signal back up by the inverse of the transponder ratio to give an effective value of unity. This fact became apparent after the TDEP was written.

PAIR

This array is used only for USBS data in the three-way doppler mode in case the transmitting and receiving stations are not the same.

```
$IBFTC MC134S  M94,NODD,XR3
CMC134S    STATION BLOCK DATA FOR TRACKING DATA EDITOR                      CTRK0001
      BLOCK DATA                                                            CTRK0002
      COMMON  /TRKCOM/CTRK(700)                                             CTRK0003
      DOUBLE PRECISION  FTR(50),  C1(50),   C2(50),   BIAS(50), RETR(50)CTRK0004
      DIMENSION  STANAM(50),   KODSTA(50),   NALIGN(50),   PAIR(50)         CTRK0005
      EQUIVALENCE           (CTRK(1),STANAM),    (CTRK(51),KODSTA)          CTRK0006
     1,  (CTRK(101),NALIGN),  (CTRK(151),FTR),     (CTRK(251),C1)           CTRK0007
     2,  (CTRK(351),C2),      (CTRK(451),BIAS),    (CTRK(551),RETR)         CTRK0008
     3,  (CTRK(651),PAIR)                                                   CTRK0009
C                                                                           CTRK0010
      DATA (STANAM(I),KODSTA(I),NALIGN(I),PAIR(I),FTR(I),C1(I),C2(I)        CTRK0011
     1,BIAS(I),RETR(I),I=1,9)/                                              CTRK0012
C**TRACKING SHIPS                                                            CTRK0013
C  VANGUARD                                                                  CTRK0014
     16HVANGRD,   62,   1,    6HVANGRD,   2045.69D6,   .34D-3,   .138771D-3CTRK0015
     2,           1.D6,       1.085972859728597286,                         CTRK0016
C  REDSTONE                                                                  CTRK0017
     16HREDSTN,   63,   1,    6HREDSTN,   2045.69D6,   .34D-3,   .138771D-3CTRK0018
     2,           1.D6,       1.085972859728597286,                         CTRK0019
C  MERCURY                                                                   CTRK0020
     16HMERCRY,   64,   1,    6HMERCRY,   2045.69D6,   .34D-3,   .138771D-3CTRK0021
     2,           1.D6,       1.085972859728597286,                         CTRK0022
C  HUNTSVILLE                                                                CTRK0023
     16HHNTSVL,   65,   1,    6HHNTSVL,   2045.69D6,   .34D-3,   .138771D-3CTRK0024
     2,           1.D6,       1.085972859728597286,                         CTRK0025
C  WATERTOWN                                                                 CTRK0026
     16HWTRTWN,   66,   1,    6HWTRTWN,   2045.69D6,   .34D-3,   .138771D-3CTRK0027
     2,           1.D6,       1.085972859728597286,                         CTRK0028
C**LAND STATIONS                                                             CTRK0029
C  MERRITT ISLAND                                                            CTRK0030
     16HMERRIT,   71,   1,    6HMERRIT,   2045.69D6,   .34D-3,   .138771D-3CTRK0031
     2,           1.D6,       1.085972859728597286,                         CTRK0032
C  GRAND BAHAMA ISLAND                                                       CTRK0033
     16HBAHAMA,   41,   1,    6HBAHAMA,   2045.69D6,   .34D-3,   .138771D-3CTRK0034
     2,           1.D6,       1.085972859728597286,                         CTRK0035
C  ANTIGUA                                                                   CTRK0036
     16HANTIGU,   91,   1,    6HANTIGU,   2045.69D6,   .34D-3,   .138771D-3CTRK0037
     2,           1.D6,       1.085972859728597286,                         CTRK0038
C  BERMUDA                                                                   CTRK0039
     16HBERMUD,   02,   1,    6HBERMUD,   2045.69D6,   .34D-3,   .138771D-3CTRK0040
     2,           1.D6,       1.085972859728597286/                         CTRK0041
      DATA (STANAM(I),KODSTA(I),NALIGN(I),PAIR(I),FTR(I),C1(I),C2(I)        CTRK0042
     1,BIAS(I),RETR(I),I=10,18)/                                            CTRK0043
C  CANARY ISLANDS                                                            CTRK0044
     16HCANARY,   04,   1,    6HCANARY,   2045.69D6,   .34D-3,   .138771D-3CTRK0045
     2,           1.D6,       1.085972859728597286,                         CTRK0046
C  ASCENSION                                                                 CTRK0047
     16HASCENS,   75,   1,    6HASCENS,   2045.69D6,   .34D-3,   .138771D-3CTRK0048
     2,           1.D6,       1.085972859728597286,                         CTRK0049
C  CARNARVON                                                                 CTRK0050
     16HCRNRVN,   08,   1,    6HCRNRVN,   2045.69D6,   .34D-3,   .138771D-3CTRK0051
     2,           1.D6,       1.085972859728597286,                         CTRK0052
C  GUAM                                                                      CTRK0053
     16HGUAM  ,   24,   1,    6HGUAM  ,   2045.69D6,   .34D-3,   .138771D-3CTRK0054
     2,           1.D6,       1.085972859728597286,                         CTRK0055
C  HAWAII                                                                    CTRK0056
     16HHAWAII,   12,   1,    6HHAWAII,   2045.69D6,   .34D-3,   .138771D-3CTRK0057
     2,           1.D6,       1.085972859728597286,                         CTRK0058
C  GUAYMAS                                                                   CTRK0059
     16HGUAYMS,   14,   1,    6HGUAYMS,   2045.69D6,   .34D-3,   .138771D-3CTRK0060
     2,           1.D6,       1.085972859728597286,                         CTRK0061
C  CORPUS CHRISTI                                                            CTRK0062
     16HCORPUS,   16,   1,    6HCORPUS,   2045.69D6,   .34D-3,   .138771D-3CTRK0063
     2,           1.D6,       1.085972859728597286,                         CTRK0064
C  MADRID                                                                    CTRK0065
     16HMADRID,   23,   2,    6HMADRID,   2045.69D6,   .34D-3,   .138771D-3CTRK0066
     2,           1.D6,       1.085972859728597286,                         CTRK0067
C  MADRID (BACKUP)                                                           CTRK0068
     16HMADRIX,   94,   2,    6HMADRIX,   2045.69D6,   .34D-3,   .138771D-3CTRK0069
     2,           1.D6,       1.085972859728597286/                         CTRK0070
      DATA (STANAM(I),KODSTA(I),NALIGN(I),PAIR(I),FTR(I),C1(I),C2(I)        CTRK0071
     1,BIAS(I),RETR(I),I=19,22)/                                            CTRK0072
C  CANBERRA                                                                  CTRK0073
     16HCANBRA,   25,   2,    6HCANBRA,   2045.69D6,   .34D-3,   .138771D-3CTRK0074
```

```
      2,           1.D6,        1.085972859728597286,                             CTRK0075
C   CANBERRA (BACKUP)                                                              CTRK0076
      16HCANBRX,   93,    2,    6HCANBRX,   2045.69D6,   .34D-3,   .138771D-3CTRK0077
      2,           1.D6,        1.085972859728597286,                             CTRK0078
C   GOLDSTONE                                                                      CTRK0079
      16HGLDSTN,   28,    2,    6HGLDSTN,   2045.69D6,   .34D-3,   .138771D-3CTRK0080
      2,           1.D6,        1.085972859728597286,                             CTRK0081
C   GOLDSTONE (BACKUP)                                                             CTRK0082
      16HGLDSTX,   92,    2,    6HGLDSTX,   2045.69D6,   .34D-3,   .138771D-3CTRK0083
      2,           1.D6,        1.085972859728597286/                             CTRK0084
       DATA (STANAM(I),KODSTA(I),NALIGN(I),PAIR(I),FTR(I),C1(I),C2(I)           CTRK0085
      1,BIAS(I),RETR(I),I=23,27)/                                                  CTRK0086
C**GODDARD RANGE-RANGE RATE SYSTEM STATIONS                                         CTRK0087
C**REGULAR                                                                          CTRK0088
C   ROSMAN                                                                          CTRK0089
      16HROSRAN,   30,    1,    6HROSRAN,   2271.9328D6,.34D-3,   .138771D-3CTRK0090
      2,           .5D6,        .75D0,                                            CTRK0091
C   MADAGASCAR                                                                     CTRK0092
      16HMADGAR,   31,    1,    6HMADGAR,   2271.9328D6,.34D-3,   .138771D-3CTRK0093
      2,           .5D6,        .75D0,                                            CTRK0094
C   CARNARVON                                                                      CTRK0095
      16HCARVON,   32,    1,    6HCARVON,   2271.9328D6,.34D-3,   .138771D-3CTRK0096
      2,           .5D6,        .75D0,                                            CTRK0097
C**HYBRID                                                                           CTRK0098
C   SANTIAGO                                                                       CTRK0099
      16HSNTAGR,   33,    1,    6HSNTAGR,   2271.9328D6,.34D-3,   .138771D-3CTRK0100
      2,           .5D6,        .75D0,                                            CTRK0101
C   ALASKA                                                                         CTRK0102
      16HULASKR,   34,    1,    6HULASKR,   2271.9328D6,.34D-3,   .138771D-3CTRK0103
      2,           .5D6,        .75D0/                                            CTRK0104
       END                                                                         CTRK0105
```

/TSTCOM/CTEST(400)

No block data.

TSTCOM is used for storage of a variety of program controls and constants. Some are obtained from input data; others are computed.

/TSTCOM/CTEST(400)

LOC	Symbol	Dimension	Description
1	NYR	(10)	Year numbers for raw data tapes
11	NPTS	(10)	Number of points to include in polynomial fit, each tape
21	NSTEP	(10)	Polynomial walk step size in points, each tape
31	NDEG	(10)	Degree of polynomial to fit, each tape
41	CSD	(4, 10)	Number of standard deviations allowable for outlier tagging. Each of 4 observables in following order: 1. Azimuth, X, or hour angle 2. Elevation, Y, or declination 3. Range 4. Doppler
81	IRAW		Current location in above arrays; sequence number of raw data tape in process
82	KNTROL	(4)	Use indicator for each observable, standard order: 0: no data, this observable 1: good data, this observable 4: Goddard system angles, valid data every fourth time point
86	(Spare)		-------------------------
87	FINISH	d(10)	Time of last data to be processed, seconds from 1950, each tape.
107	IFOMIT	(4,10)	Data supression keys, each observable in standard order, each tape. +1: suppress 0: do not suppress

/TSTCOM/CTEST(400) (continued)

LOC	Symbol	Dimension	Description
147	KMAX	(10)	Number of valid station ID codes, each tape.
157	ISTAOK	(20, 10)	List of valid station codes, each tape uses one column.
357	TLO	d(10)	Bounds for valid data time tags, seconds from 1950, each tape.
377	THI	d(10)	
397 400	Spares		----------------------------------

/WCOM/IW(550), CW(1450)

No block data.

WCOM is used for storage of a large number of working common arrays that must be made available to a number of subroutines. All the values are generated by the program and may not be altered by any data overlay process.

/WCOM/IW(550)

Table 1

LOC	Symbol	Dimension	Definition
1	NSTX	(3)	Extra body keys once used by SHIFT. Not now used by SHIF2.
4	NOX	(3)	Array of central body numbers for up to three extra bodies; e.g., if the first extra body is earth centered, NOX(1) = 1.
7	IXT		Total number of extra bodies
8	ISC	(12)	Upper-loaded array of numbers of stations to be considered. For example, ISC = (1, 5, 12, 0, . . . , 0) indicates that only stations 1, 5, and 12 are to be considered.
20	IBC	(10)	Upper-loaded array of numbers of beacons to be considered.
30	IEMP	(24)	Not used
54	IOBR	(4)	Reserved for indicators for on-board measurements. Not presently in use.
58	IOBA	(18)	(IOBR and IOBA: reserved for indicators for on-board measurements. Not presently in use.)
59	Spares	(122)	------------------------------

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
180	IS	(11, 12)	The second index corresponds to the station identification number. The first row of this array IS(1, N) contains a 1, 2, 3, or 4 to indicate the type of measurements to be made as follows: 1: C-Band 2: Goddard Range and Range Rate 3: Unified S-Band 4: DSIF (JPL) The next nine rows are error source treatment indicators. Row order is: 2 Range treatment 3 Doppler treatment 4 Angle 1 treatment 5 Angle 2 treatment 6 Northing error treatment 7 Easting error treatment 8 Down error treatment 9 Clock bias treatment For rows 2-5 the following code is used: 0: No measurement 1: Measure with random and bias errors 2: Measure with random errors only Rows 6-9 use the following code: 0: No error 1: Include bias error Row 10 is a key to indicate antenna alignment (dish size) for USBS systems 1: 30-foot dish, principal axis north 2: 85-foot dish, principal axis east Row 11 is a key for refraction effects. 1: Include 2: Omit

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
312	IB	(3, 12)	Each of the first ten columns corresponds to a beacon number; the rows contain indicators for beacon location errors north, east, and down in that order. The code is identical to that used for rows 6-9 of the IS array. Columns 11 and 12 contain: IB(1, 11) Range treatment indicator IB(2, 11) Range-rate treatment indicator IB(3, 11) Angle 1 treatment indicator IB(1, 12) Angle 2 treatment indicator IB(2, 12) Time bias treatment indicator IB(3, 12) Body number on which all beacons are located The treatment indicators are like those used for station error treatments.
347	NBEAC		NBEAC = IB(3, 12), the number of the body on which all beacons are located.
348	Spares	(7)	----------------------------------
355	ISEE	(12)	Spacecraft in-view indicator for stations. Upper-loaded to correspond to ISC array -1: Cannot see spacecraft 0: Can see spacecraft
367	IBEE	(10)	In-view indicator for beacons. Upper-loaded to correspond to IBC array. -1: Cannot be seen by spacecraft 0: Can be seen by spacecraft
377	ICAS		Case number used to search for desired trajectory on tape 10.
378	Spares	(2)	----------------------------------
380	NMAX		Number of columns in measurement buffer XMES (NMAX = 85)
381	Spare		----------------------------------
382	NOR		Current central body number.
383	ITARG		Target body number. Only the target body can occult the spacecraft as viewed from the earth.

Table 1 (Continued)

LOC	Symbol	Dimension	Definition
384	IK1		Indicator showing current location in SSTART and SSTOP arrays.
385	KSMAX		Total number of station critical events found by SBEV2.
386	IK2		Indicator showing current location in BSTART and BSTOP arrays.
387	KBMAX		Total number of beacon critical events found by SBEV2.
388	Spare		------------------------------
389	ISTIM	(50)	Event type indicator in conjunction with STIME array times: +K: Station K into view -K: Station K out of view 13: Occulting starts 14: Occulting stops
439	IBTIM	(50)	Event type indicator in conjunction with BTIME array times: +K: Beacon K into view -K: Beacon K out of view
489	ITRIG		Trigger to start record counter KOUNT in EXINST: 0: No count 1: Count
490	KOUNT		Record counter incremented in EXINST when tape 10 is read and ITRIG = 1. Used for rapid repositioning of tape after search for critical events.
491	Spares	(60)	------------------------------

/WCOM/CW(1450)

Table 2

LOC	Symbol	Dimensions	Definition
1	RX	(3, 3)	Extra bodies' initial position vectors, km, equator and equinox of 1950. RX(I, J) is the Ith coordinate of extra body J with respect to body NOX(J).
10	VX	(3, 3)	Extra bodies' initial velocity vectors, km/sec, arranged as above.
19	TX	(2, 3)	Initial dates for extra bodies Row 1 Whole days from 1950.0 Row 2 Fractional days from 1950.0
25	Spares	(131)	------------------------------
156	BIASL	(3, 22)	The second index refers to station or beacon number; 1-12 for stations 1-12, 13-22 for beacons 1-10. Contains location errors in km, north, east, down in that order.
222	SSTART	(75)	Station on-times, seconds from epoch, arranged chronologically. Corresponding station numbers are in KSTA array.
297	SSTOP	(75)	Station off-times, seconds from epoch, arranged to pair with SSTART times.
372	BSTART	(75)	Beacon on-off times. Analogous to the SSTART and SSTOP arrays. Beacon numbers in KBEA array.
447	BSTOP	(75)	
522	XMES	(5, 85)	Storage array for measurements to be written on tape 12. Each column contains a set of measurements and a time tag. The rows are: 1 Time, seconds from epoch 2 Angle 1, radians 3 Angle 2, radians 4 Range, km or seconds 5 Doppler, counts This table is written on tape 12 by rows.
947	Spares	(187)	------------------------------

TABLE 2 (Continued)

LOC	Symbol	Dimension	Definition
1134	XIN	(6)	Current spacecraft state vector body NOR, equator and equinox of 1950, km and km/sec.
1140	TSEC		Current time from epoch in seconds.
1141	Spares	(16)	------------------------------
1157	REFS	(3)	Not used - reserved for on-board measurement reference data.
1160	TI	(3)	Current date in days from 1950.0 1 Whole days 2 Fractional days 3 Not used
1163	AN	(3, 3)	Transformation from equator and equinox of 1950 to equator and equinox of date.
1172	TSECO		Initial time, seconds from epoch.
1173	GHAR		Greenwich hour angle at TSECO, radians.
1174	Spares	(4)	------------------------------
1178	TSTART		Time, in seconds from epoch, at which measurements are to start or, if in process, time to read new control cards.
1179	FLTIM		Time in seconds from epoch at which to end the case.
1180	EVNT	(6)	Chronologically ordered critical event array which contains the next to be considered of each of the following six types of events: Start station measurements Start beacon measurements Start on-board optical measurements Start on-board radar measurements TSTART FLTIM
1186	Spares	(2)	------------------------------

TABLE 2 (Continued)

LOC	Symbol	Dimension	Definition
1188	TWT	(2)	Control times in seconds from 1950.0 1. Time to start measurements 2. Time to stop measurements and read next control cards.
1190	Spares	(10)	-------------------------------
1200	STIME	(50)	Ordered array of times at which critical events ISTIM occur.
1250	BTIME	(50)	Ordered array of times at which critical events IBTIM occur.
1360	SECR	(12)	Time array in calling sequence of SBEV2. Not now used.
1372	SHFC	(18)	Orbital parameters for extra bodies, set up by STEPI
1390	Spares	(36)	-------------------------------
1426	TWO		Initial time in whole days from 1950.0
1427	TFO		Initial time in fractional days from 1950.0
1428	Spares	(13)	-------------------------------
1441	BECR	(10)	Time array in calling sequence of SBEV2. Not now used.

SECTION 2

SUBROUTINES

2.0 INTRODUCTION

This section contains the descriptions and listings of all subroutines and functions used by the Orbit Determination Program system, arranged in alphabetical order. Each subroutine description lists the functions of the subroutine and the rules for its usage, including required input and resulting output. All referenced commons and subroutines not contained the standard IBSYS library are listed. Equations describing the computations are given where required, and references are quoted for necessary derivations.

Where labelled commons are referenced by the subroutines, a minimum description of the common variables is given. For more detailed descriptions, reference should be made to Section 1 of this report.

Blank common is used throughout the program system as a "scratch pad;" that is, for strictly internal temporary storage. Some care is required in any changes of subroutines which use blank common to avoid loss of stored variables. Normally, a call to a subroutine which uses N cells of blank common may be expected to destroy any data stored in these cells, and the scratch pad for the using subroutine should therefore begin at cell N+1. The length of blank common specified by the subroutine description is the length used by the subroutine and any referenced subroutines.

2.1 SUMMARY OF SUBROUTINE FUNCTIONS

The paragraphs below contain brief functional statements for each of the programs in the ODP system and their component subroutines. A tabular cross-reference summarizing all references to all commons and user-supplied subroutines is also given.

2.1.1 Differential Correction Program (DCP)

The Differential Correction Program accepts an *a priori* estimate of the space vehicle state and other parameters and a tape containing measurement values, and statistically combines these data into an improved estimate of the state and parameters.

DCP SUBROUTINES

Table 2.1

Subroutine Name	Function
ACCTRJ	Computes second derivatives for integration of the equations of motion and variational equations.
ACCTR3	Computes second derivatives for integration of the equations of motion.
BLDCOV	Builds portions of the covariance matrix.
CBDAT	Computes C-band observables as functions of vehicle state.
CBDATP	Computes C-band observables and partial derivatives as functions of vehicle state.
COVOUT	Writes normalized covariance matrix on the system output tape.
DATINP	Converts date input to seconds from 1950 January 0.0.
DATOUP	Converts seconds from 1950 January 0.0 to date format and writes date on the system output tape.
DCP	Main program for the DCP.
DCROSS	Computes the cross product of two vectors.
DDOT	Computes the dot product of two vectors.
DEHA	Computes hour angle of the vernal equinox and Earth's rotation rate.

Table 2.1 (Continued)

Subroutine Name	Function
DEQD	Integrates equations of motion and variational equations.
DEQ3	Integrates equations of motion.
DEQTR	Computes parameters defining the precession of Earth's equator.
DGTRN	Computes transformation matrix from a defined sequence of rotations.
DGTSN	Computes transformation matrix from sines and cosines of rotation angles.
DIFCOR	Main program for propagation and differential correction.
DINVRT	Computes the inverse of an NxN matrix.
DLUNE	Computes the transformation from Earth's equator, equinox of date to Moon-fixed coordinates.
DMPLY	Computes the product of two matrices.
DMVTRN	Computes the product of a 3 x 3 matrix and a 3 x N matrix.
DNORM	Computes the magnitude of a 3-vector.
DPRMRS	Computes planetary positions and velocities.
DRAGD	Computes atmospheric drag.
DRAGDP	Computes atmospheric drag and its partial derivatives.
DSDAT	Computes DSIF observables from vehicle state.
DSDATP	Computes DSIF observables and their partial derivatives from vehicle state.
DSTAT	Computes tracking station position vectors and transformation matrices.
DTRANP	Transforms covariance matrix from base coordinates to output coordinates.

Table 2.1 (Continued)

Subroutine Name	Function
DTRDB	Computes transformation matrix from base coordinates to Darboux coordinates.
DVNORM	Computes the magnitude of a 3-vector, and normalizes the vector.
ENCKED	Computes Encke acceleration.
ESTMAT	Computes and adds differential corrections to the state and covariance matrix.
ESTOUT	Writes state estimate and covariance matrix on the estimate tape and on the system output tape.
GRAVD	Computes accelerations due to gravitational harmonics.
GRAVDP	Computes accelerations and partial derivatives for gravitational harmonics.
GRDAT	Computes Goddard Range Range Rate observables from vehicle state.
GRDATP	Computes Goddard observables and partial derivatives from vehicle state.
GTR2BD	Computes central body transformations.
MEAS2X	Computes cartesian coordinates from measurement coordinates.
MLESTT	Main program for maximum likelihood data start.
MXLEST	Computes maximum likelihood estimate of vehicle state.
OUTTRJ	Writes integration output and forms state interpolation tables for differential correction.
OUTTR3	Forms state interpolation tables for residual computation.
OUTXPD	Writes vehicle state on system output tape.
OVRLYD	Double precision input subroutine.

TABLE 2.1 (Continued)

Subroutine Name	Function
PERTD	Computes planetary gravitational accelerations.
PERTDP	Computes planetary gravitational accelerations and their partial derivatives.
PSEUDO	Computes the pseudo inverse of an N x N matrix.
RESOUT	Controls output of residuals on residual tape and system output tape.
RSPLOT	Plots residuals on the system output tape.
SBDAT	Computes S-band observables from vehicle state.
SBDATP	Computes S-band observables and partial derivatives from vehicle state.
SETCAS	Performs case initialization. Builds a priori state and covariance.
SETSTA	Sets up working arrays of station data.
SETTAP	Checks or writes tape headers and initializes tape positions.
SOLRD	Computes solar pressure acceleration.
SOLRDP	Computes solar pressure acceleration and partial derivatives.
STATE	Computes vehicle state and transformation.
STEPDI	Initializes conic step functions.
STEPDP	Computes conic transition matrix.
STEPDT	Computes position and velocity on a conic.
STTBGN	Initializes maximum likelihood start.
STTDAT	Sets up data arrays for the m.ℓ.e. start.
STTEND	Stores completed maximum likelihood estimate.

Table 2.1 (Continued)

Subroutine Name	Function
STTFIT	Fits individual measurements to a time polynomial.
STTIME	Computes residuals for maximum likelihood start.
TRAJD	Initializes and controls trajectory integration for the residual computation.
TRAJDP	Initializes and controls trajectory integration for propagation and differential correction.
UPDATP	Updates the covariance matrix in time.
VENTD	Computes acceleration due to venting.
VENTDP	Computes acceleration due to venting and its partial derivatives.
X2ORBD	Computes orbital elements from cartesian coordinates.

2.1.2 Tracking Data Editing Program (TDEP)

The Tracking Data Editing Program accepts as input one or more BCD tapes of tracking data, each in one of four acceptable formats. The TDEP reads these data, performs necessary unit conversions, detects and tags outliers, and writes these data on a binary tape in a format suitable for input to the DCP. As an optional input, a tape of previously edited data may be combined with new data and written as the TDEP output tape.

TDEP SUBROUTINES

Table 2.2

Subroutine Name	Function
CBTEST	Reads and decodes a BCD tape of C-Band tracking data.
CROSIM	Solves a set of simultaneous, linear algebraic equations.
DATINP	Converts date input to seconds from 1950 January 0.0.
DATOUP	Converts seconds from 1950 January 0.0 to date format and writes date on the system output tape.
DSTEST	Reads and decodes a BCD tape of DSIF (JPL) tracking data.
EGRESS	Transfers the final edited data to tape 10 if necessary and writes a summary of the edited data on the system output tape.
GRTEST	Reads and decodes a BCD tape of Goddard Range and Range Rate System tracking data.
MIXIT	Reads edited data from tapes 12 and NIN; combines these data and writes them on tape NOUT.
NUDATA	Allows the nominal values in the tracking station arrays to be overlayed with different values.
• OPTW.	Allows an optional return from the IBSYS error subroutine FXEM when illegal characters are encountered on tracking data read-in.
POLYFT	Tests a data arc (station pass) of edited data with walking polynomials; tags outliers.
SBTEST	Reads and decodes a BCD tape of unified S-band tracking data.
SCANIT	Reads the final edited data tape and writes all the data on the system output tape.
TDEP	Main program for Tracking Data Editing Program; performs overall control.

2.1.3 Tracking Data Simulator (TDS)

The Tracking Data Simulator accepts as input a spacecraft trajectory tape, a planetary ephemeris tape, and one or more sets of input cards that specify a tracking station network and the time interval over which measurements are to be made. The TDS uses these inputs to write a tape of simulated measurement data in a format suitable for input to the Differential Correction Program.

TDS SUBROUTINES

Table 2.3

Subroutine Name	Function
ANTR1	Provides planetary positions and velocities by interpolation on values read from the planetary ephemeris tape.
ARKTNS	Four-quadrant arctangent subroutine.
BACK	Backspaces a specified tape (always tape 10 in the TDS) a specified number of records.
BARN	Random number generator. Will generate uniform or Gaussian random variables.
BECSTA	Computes pairs of on and off times for stations and beacons.
BIBCD	Converts a binary integer to BCD.
BMEAS	Makes a set of beacon measurements at a specified point in time.
CARDIN	Reads data cards that specify the tracking network or changes thereto.
CRITA	Outputs the time at which a body starts or stops occulting the vehicle.
CRITIC	Takes arrays of station- or beacon-related critical events and forms pairs of on-off times.

Table 2.3 (Continued)

Subroutine Name	Function
CRITO	Outputs the station or beacon critical events.
CROSS	Computes the cross product of two vectors.
DATOUT	Converts whole and fractional days from 1950 January 0.0 to date and writes date on the system output tape.
DELEV	Sorts the critical event array into chronological order and determines the required measurement interval.
DOT	Computes the scalar product of two vectors.
ERROUT	A FORTRAN dummy replacement for the MAP-coded error trace routine of the same name.
EXINST	Provides spacecraft position and velocity by interpolation on tables read from the spacecraft trajectory tape.
FIEF	Reads tapes forward or backspaces tapes over end-of-file marks.
FIFL	Finds a specified case on the spacecraft trajectory tape and reads the first record thereof.
FIST	Rewinds the spacecraft trajectory tape and checks the header. Also initializes FIFL.
FNORM	Computes the magnitude of a vector.
GHA	Computes the Greenwich hour angle of the first point of Aries at a given time.
GOTOR	Solves Kepler's equation for incremental eccentric anomaly given incremental mean anomaly.
INITAP	Initialization subroutine for the TDS.
LAYO	Reads station, (beacon and onboard) data cards and converts units as required.
LOCAT	Adds location errors to station and beacon locations.

Table 2.3 (Continued)

Subroutine Name	Function
MEASUR	Computes and stores an arc of appropriate type measurement data.
MESERP	Directs the overall logic flow for making simulated data tapes with the TDS.
MNA	Provides the rotation matrix that transforms coordinates in the earth's true equator and equinox to coordinates in the moon's true equator.
MTRN	Computes the 3 x 3 product of two 3 x 3 matrices.
NUTAIT	Computes the transformation from mean equator and equinox to true equator and equinox.
ORB	Computes orbital elements from a Cartesian state vector and writes the elements in the system output tape.
OYAL	Writes the data read by LAYO on the system output tape.
PARAB	Fits a parabola through three points.
QUARTC	Finds the zeros of a quadratic.
ROTEQ	Computes the transformation from mean equator and equinox of 1950 to mean equator and equinox of date.
ROVLEY	Reads fixed, floating, and alphanumeric data into labeled common.
SBEV2	Creates an ordered array of times at which the vehicle comes into or goes out of view of stations or beacons.
SHIF2	Given the state of a body or spacecraft with respect to a body, returns the state with respect to any other body or bodies.
SKDOUT	Writes out the measurement schedule on the system output tape.
SKDUL	Reads the control time cards and calls SKDOUT to output the measurement schedule.

Table 2.3 (Continued)

Subroutine Name	Function
SMEAS	Makes a set of station measurements of a specified type at a given time.
SORDR	Sorts an array X into ascending order, preserving the correspondence between X and another array NX.
SORDR2	Sorts an array X into ascending order, preserving the correspondence between X and another array NX and, optionally, a third array Y.
STARTB	Loads the next beacon on time from the BSTART array into position EVNT(1) of the critical event array.
STARTS	Loads the next station on time from the SSTART array into position EVNT(1) of the critical event array.
STAT	Obtains a vector from the body center to a station or beacon and computes an orthogonal transform relating inertial Cartesian quantities X, Y, Z to local tangent plane quantities North, East, Down.
STEPI	Performs initialization for STEPT
STEPT	Finds the state of a body on a conic at a time, that conic having been initialized by a call of STEPI.
TCONIC	Computes the time from periapsis corresponding to a given true anomaly on a given conic section.
TDS	Executive driver for the TDS.
TFRAC	Computes integral and fractional parts of a sum. Used to update the internal date format.
TIMED	Converts a time interval from time format to seconds.
TIMES	Converts a time interval in seconds to time format and prepares a six-word BCD array for simplified output of interval in days, hours, minutes, and seconds.
VNORM	Normalizes a vector.
VTRN	Given a 3 x 3 matrix A and a 3 x 1 vector X, computes the vector AX.
VTRT	Given a 3 x 3 matrix A and a 3 x 1 vector X, computes the vector $A^T X$.

2.1.4 Residual Output Program (ROP)

The Residual Output Program accepts as input the estimate and residual tapes written by the DCP for use in special purpose analysis of the residuals. The program is intended only as a guide for the use of the input tapes, and each analytical function required must be programmed as required. Only the main program is supplied.

2.2 SUBROUTINE DESCRIPTIONS

Descriptions and listings of each of the subroutines used by the ODP are given in the following pages. Each subroutine is written as a separate section to facilitate replacement as the subroutines are modified.

Subroutine: ACCTRJ

Purpose: To compute the perturbation acceleration for Encke integration of the equations of motion. Also, computes derivatives for variational equations.

Calling Sequence: CALL ACCTRJ

Common storages used: //144 cells,/DCPCØM/,/DFMCØM/,/DQDCØM/,/ESTCØM/, /TRJCØM/

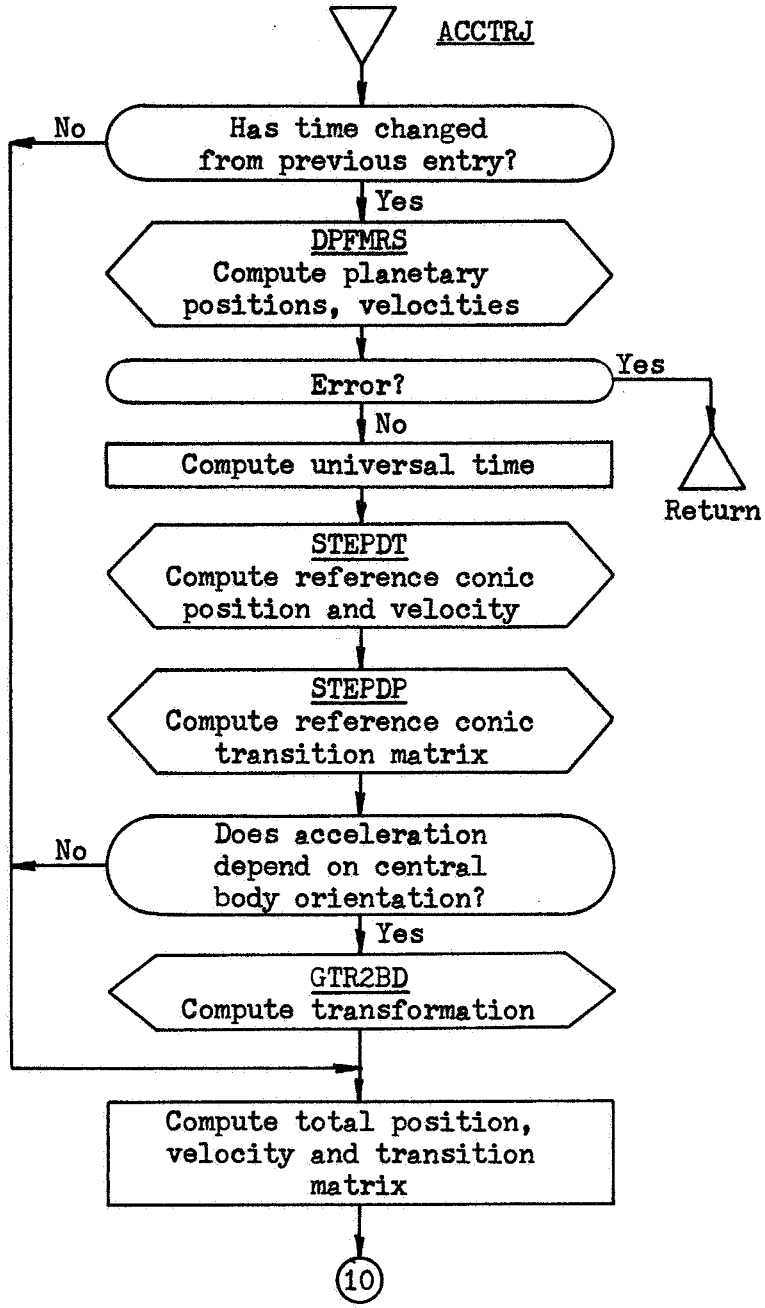
Subroutines required: DCRØSS,DDØT,DMPLY,DMVTRN,DPFMRS,DRAGDP,DVNØRM, ENCKED,GRAVDP,GTR2BD,PERTDP,STEPDP,STEPDT,VENTDP, SØLRDP

Usage:

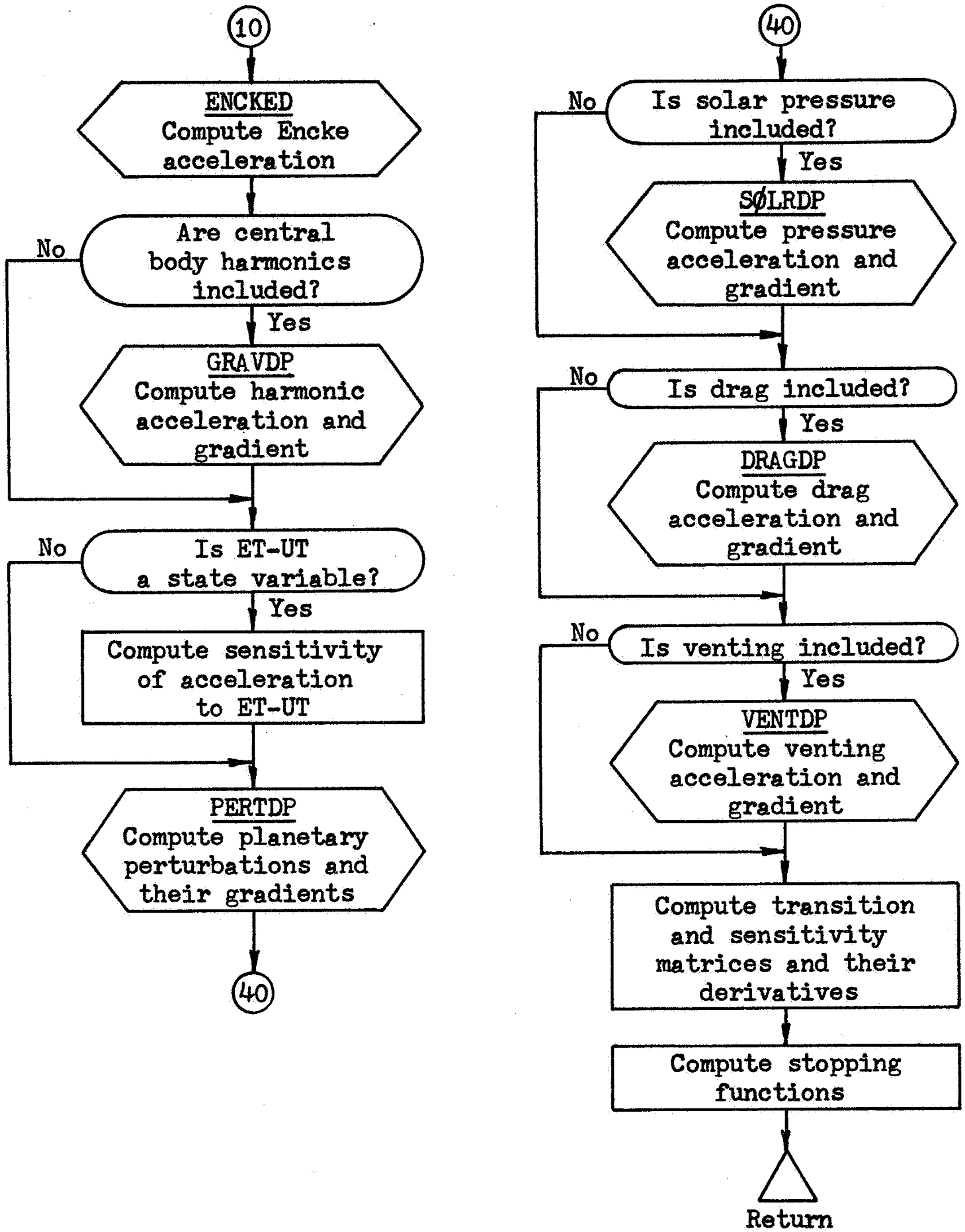
ACCTRJ accepts the integrated dependent variables RDEQ,VDEQ of /DQDCØM/, and computes the accelerations ADEQ. The components of RDEQ,VDEQ,ADEQ are

I = 1, 2, 3	R , $\dot{R}$, $\ddot{R}$, in C-frame coordinates
I = 4 - 21	φ , $\dot{\varphi}$, $\ddot{\varphi}$
I = 22 - (NPEND-1)	φ_u, $\dot{\varphi}_u$, $\ddot{\varphi}_u$

The equation of motion model is described in Appendix D of Reference 1.



Central Body Position, Velocity and Orientation
Vehicle Total Position, Velocity



Accumulation of
Perturbation Accelerations

```
$IBFTC MC133A  XR3,M94,NODD,LIST
      SUBROUTINE ACCTRJ
C     COMPUTES PERTURBATION ACCELERATION FOR ENCKE INTEGRATION
C     COMPUTES STATE TRANSITION MATRIX DERIVATIVE AND
C     EQUATION OF MOTION PARAMETER SENSITIVITIES
C     REFERENCE - TR-DA1508, APPENDIX D
C
      DOUBLE PRECISION   DDOT ,DSQRT ,DVNORM
C
      COMMON            /DCPCOM/CDCP(900)
      DOUBLE PRECISION   DRECT ,RSTP2
      DIMENSION          CBODY(8,11) ,ETAPE (4) ,IFEMP(8)
     1                  ,IPL      (11) ,IHM   (24) ,NEMP (8)
      EQUIVALENCE               (CDCP(112),ETAPE ) ,(CDCP(663),IPL    )
     1                         ,(CDCP(777),ICB    ) ,(CDCP(779),KCB    )
     2                         ,(CDCP(111),IERR   ) ,(CDCP(107),MAXD   )
     3       ,(CDCP( 17),CBODY ) ,(CDCP(687),IFEMP  ) ,(CDCP(695),NEMP   )
     4       ,(CDCP(809),DRECT ) ,(CDCP(703),IHM    ) ,(CDCP(897),RSTP2 )
      EQUIVALENCE               (IFEMP( 8),IFBODY) ,(NEMP ( 1),NDRAG )
     1                         ,(IFEMP( 7),IFDRAG) ,(NEMP ( 8),NEMPS )
     2                         ,(IFEMP( 6),IFHARM) ,(NEMP ( 7),NEMP3 )
     3                         ,(IFEMP( 3),IFSOLR) ,(NEMP ( 2),NHARM )
     4                         ,(IFEMP( 5),IFTIME) ,(NEMP ( 6),NPEND )
     5                         ,(IFEMP( 4),IFVENT) ,(NEMP ( 3),NSOLR )
     6                                             ,(NEMP ( 5),NTIME )
     7                                             ,(NEMP ( 4),NVENT )
C
      COMMON            /ESTCOM/CEST(804)
      DOUBLE PRECISION   DELDAN(2) ,EFEDAN(14) ,EHADAN(24,3) ,PREDAN(4)
     1                  ,ETIMVA
      EQUIVALENCE       (CEST( 97),DELDAN) ,(CEST(101),EHADAN)
     1                 ,(CEST( 61),EFEDAN) ,(CEST( 89),PREDAN)
     2                 ,(CEST( 29),ETIMVA)
C
      COMMON            /TRJCOM/CTRJ(246)
      DOUBLE PRECISION   CBOD(37) ,ETFMS ,PCONIC(6,6) ,RC(3) ,TB2C(3,3)
     1                  ,CBO ,CBU ,UTIMV ,RCONIC(6)    ,VC(3)
     2                  ,CBP              ,VCONIC(3)
      EQUIVALENCE               (CTRJ(  1),ETFMS ) ,(CTRJ( 59),TB2C   )
     1                         ,(CTRJ(175),PCONIC) ,(CTRJ(  3),UTIMV )
     2                         ,(CTRJ(151),RC     ) ,(CTRJ(157),VC     )
     3       ,(CTRJ( 77),CBOD   ) ,(CTRJ(163),RCONIC) ,(CTRJ(169),VCONIC)
      EQUIVALENCE               (CBOD ( 2),CBP    ) ,(CBPD (12),NZH    )
     1       ,(CBOD ( 3),CBA    ) ,(CBOD ( 5),CBU    ) ,(CBOD (20),TH     )
     2       ,(CBOD ( 6),CBO    ) ,(CBOD (13),NTH    ) ,(CBOD (14),ZH     )
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 ,DTP ,GRAD
      DOUBLE PRECISION   D(48) ,DTP(3,2) ,GRAD(3,6)
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 999
      ETFMS = ETIMV
      UTIMV = ETIMV-DELDAN(1)-DELDAN(2)*ETIMV
C     REFERENCE CONIC
      CALL STEPDT (ETIMV,RCONIC)
      CALL STEPDP (PCONIC)
      IF (IFBODY.EQ.0)     GO TO 2
C     CENTRAL BODY ORIENTATION
      CALL GTR2BD (ICB,DFM,0)
C     POSITION AND VELOCITY RELATIVE TO CENTRAL BODY
    2 CONTINUE
```

```
      DO 3 I=1,3                                                        ACCJ0075
      RC(I) = RDEQ(I)+RCONIC(I)                                         ACCJ0076
      VC(I) = VDEQ(I)+VCONIC(I)                                         ACCJ0077
      DO 3 J=1,6                                                        ACCJ0078
      GRAD(I,J) = 0.D0                                                  ACCJ0079
    3 CONTINUE                                                          ACCJ0080
C                                                                       ACCJ0081
C     ENCKE ACCELERATION                                                ACCJ0082
C                                                                       ACCJ0083
   10 CALL ENCKED (CBU,RCONIC,RDEQ,ADEQ)                                ACCJ0084
C                                                                       ACCJ0085
C     GRAVITATIONAL HARMONICS                                           ACCJ0086
C                                                                       ACCJ0087
   20 IF (IFHARM.EQ.0)     GO TO 30                                     ACCJ0088
      CALL GRAVDP (CBA,CBU,NZH,NTH,ZH,TH,RC,TB2C,ADEQ,IFHARM            ACCJ0089
     1            ,GRAD,ADEQ(NHARM),IHM,IHM(7))                         ACCJ0090
      IF (IFTIME.EQ.0)     GO TO 30                                     ACCJ0091
C     ET-UT SENSITIVITIES                                               ACCJ0092
      K =-1-IFTIME                                                      ACCJ0093
      CALL DCROSS (RC,TB2C(1,3),DTP)                                    ACCJ0094
      DO 21 I=1,3                                                       ACCJ0095
   21 DTP(I,1) = CBO*DTP(I,1)                                           ACCJ0096
      IF (K.EQ.0) GO TO 23                                              ACCJ0097
      DTP(3,2) = ETIMV-ETIMVA                                           ACCJ0098
      DO 22 I=1,3                                                       ACCJ0099
   22 DTP(I,K) = DTP(3,2)*DTP(I,1)                                      ACCJ0100
      GO TO 24                                                          ACCJ0101
   23 K = 1                                                             ACCJ0102
   24 CALL DMVTRN (GRAD,DTP,ADEQ(NTIME),1,K)                            ACCJ0103
C                                                                       ACCJ0104
C     PLANETARY PERTURBATIONS                                           ACCJ0105
C                                                                       ACCJ0106
   30 CALL PERTDP (ICB,IPL,RC,EFEDAN,RFM,ADEQ,ADEQ(22),GRAD,1)          ACCJ0107
C                                                                       ACCJ0108
C     SOLAR PRESSURE                                                    ACCJ0109
C                                                                       ACCJ0110
   40 IF (IFSOLR.EQ.0)     GO TO 50                                     ACCJ0111
      CALL SOLRDP (RC,RFM(1,10),CBA,PREDAN,ADEQ,ADEQ(NSOLR),GRAD,IFSOLR)ACCJ0112
C                                                                       ACCJ0113
C     ATMOSPHERIC DRAG                                                  ACCJ0114
C                                                                       ACCJ0115
   50 IF (IFDRAG.EQ.0)     GO TO 60                                     ACCJ0116
      CALL DRAGDP (RC,VC,TB2C,CBO,CBA,PREDAN(2),ADEQ                    ACCJ0117
     1            ,ADEQ(NDRAG),GRAD,IFDRAG)                             ACCJ0118
C                                                                       ACCJ0119
C     VENTING                                                           ACCJ0120
C                                                                       ACCJ0121
   60 IF (IFVENT.EQ.0)     GO TO 80                                     ACCJ0122
      CALL VENTDP (VC,PREDAN(4),ADEQ,ADEQ(NVENT),GRAD(1,4),IFVENT)      ACCJ0123
C                                                                       ACCJ0124
C     EMP SENSITIVITIES                                                 ACCJ0125
C                                                                       ACCJ0126
   80 IF (NEMPS.EQ.0)      GO TO 83                                     ACCJ0127
      K = 22                                                            ACCJ0128
      DO 82 I=1,4,3                                                     ACCJ0129
      CALL DMVTRN (GRAD(1,I),RDEQ(K),D,1,NEMPS)                         ACCJ0130
      DO 81 J=1,NEMPS                                                   ACCJ0131
   81 ADEQ(J+22) = ADEQ(J+22)+D(J)                                      ACCJ0132
   82 K = K+MAXD                                                        ACCJ0133
C                                                                       ACCJ0134
C     TRANSITION MATRIX                                                 ACCJ0135
   83 CALL DMVTRN (GRAD(1,1),RDEQ(4),D( 1),1,6)                         ACCJ0136
      CALL DMVTRN (GRAD(1,4),VDEQ(4),D(19),1,6)                         ACCJ0137
      D(40) = CBU/DVNORM(RCONIC,D(37))**3                               ACCJ0138
      D(41) = 3.D0*D(40)                                                ACCJ0139
      DO 84 I=1,3                                                       ACCJ0140
      D(42) = D(41)*D(I+36)                                             ACCJ0141
      GRAD(I,I) = GRAD(I,I)+D(40)                                       ACCJ0142
      DO 84 J=1,3                                                       ACCJ0143
   84 GRAD(I,J) = GRAD(I,J)-D(42)*D(J+36)                               ACCJ0144
      CALL DMPLY (GRAD,PCONIC,ADEQ(4),3,6,6,3,6,3,0)                    ACCJ0145
      DO 85 I=1,18                                                      ACCJ0146
   85 ADEQ(I+3) = ADEQ(I+3)+D(I)+D(I+18)                                ACCJ0147
C                                                                       ACCJ0148
C     STOPPING FUNCTIONS                                                ACCJ0149
```

```
C                                                                       ACCJ0150
   90 IF (NDEQ(3).EQ.0)   GO TO 999                                     ACCJ0151
      D(1) = DDOT(RC,RC)                                                ACCJ0152
      ADEQ(NPEND+1) = D(1)-CBP*CBP                                      ACCJ0153
      DO 91 I=1,3                                                       ACCJ0154
   91 D(I+1) = RC(I)-RFM(I,KCB)                                         ACCJ0155
      ADEQ(NPEND  ) = DDOT(RDEQ,RDEQ)/D(1)-DRECT*DRECT                  ACCJ0156
      ADEQ(NPEND+2) = DDOT(D(2),D(2))-CBODY(2,KCB)*CBODY(2,KCB)         ACCJ0157
      ADEQ(NPEND+3) = D(1)-RSTP2                                        ACCJ0158
      ADEQ(NPEND+4) = DDOT(RC,VC)                                       ACCJ0159
  999 RETURN                                                            ACCJ0160
      END
```

Subroutine: ACCTR3

Purpose: To compute the perturbation acceleration for Encke integration of the equations of motion. (See also ACCTRJ).

Calling Sequence: CALL ACCTR3

Common storages used: //90 cells,/DCPCØM/,/DFMCØM/,/DQ3CØM/,/ESRCØM/, /TRJCØM/

Subroutines required: DDØT,DPFMRS,DRAGD,ENCKED,GRAVD,GTR2BD,PERTD, STEPDT,VENTD, SØLRD

```
$IBFTC MC133B  XR3,M94,NODD,LIST
      SUBROUTINE ACCTR3                                                 ACC30001
C     COMPUTES PERTURBATION ACCELERATION FOR ENCKE INTEGRATION          ACC30002
C     REFERENCE - TR-DA1508, APPENDIX D                                 ACC30003
C                                                                       ACC30004
      DOUBLE PRECISION   DDOT                                           ACC30005
C                                                                       ACC30006
      COMMON            /DCPCOM/CDCP(900)                               ACC30007
      DOUBLE PRECISION   DRECT ,RSTP2                                   ACC30008
      DIMENSION          CBODY(8,11) ,ETAPE (4) ,IFEMP(8)               ACC30009
      EQUIVALENCE                 (CDCP(809),DRECT ) ,(CDCP(687),IFEMP ) ACC30010
     1                           ,(CDCP(112),ETAPE ) ,(CDCP(663),IPL   ) ACC30011
     2                           ,(CDCP(777),ICB   ) ,(CDCP(779),KCB   ) ACC30012
     3      ,(CDCP( 17),CBODY ) ,(CDCP(111),IERR  ) ,(CDCP(897),RSTP2 ) ACC30013
      EQUIVALENCE                 (IFEMP( 7),IFDRAG) ,(IFEMP( 3),IFSOLR) ACC30014
     1      ,(IFEMP( 8),IFBODY) ,(IFEMP( 6),IFHARM) ,(IFEMP( 4),IFVENT) ACC30015
C                                                                       ACC30016
      COMMON            /ESRCOM/CESR(304)                               ACC30017
      DOUBLE PRECISION   DELDAR(2) ,EFEDAR(14) ,EHADAR(24,3) ,PREDAR(4) ACC30018
     1                  ,ETIMVA                                         ACC30019
      EQUIVALENCE       (CESR( 97),DELDAR) ,(CESR(101),EHADAR)          ACC30020
     1                 ,(CESR( 61),EFEDAR) ,(CESR( 89),PREDAR)          ACC30021
     2                 ,(CESR( 29),ETIMVA)                              ACC30022
C                                                                       ACC30023
      COMMON            /TRJCOM/CTRJ(246)                               ACC30024
      DOUBLE PRECISION   CBOD(37) ,ETFMS ,RCONIC(3) ,RC(3)              ACC30025
     1                  ,CBO ,CBP ,UTIMV ,VCONIC(3) ,VC(3)              ACC30026
      EQUIVALENCE                 (CTRJ(151),RC     ) ,(CTRJ(  3),UTIMV ) ACC30027
     1      ,(CTRJ( 77),CBOD  ) ,(CTRJ(163),RCONIC) ,(CTRJ(157),VC    ) ACC30028
     2      ,(CTRJ(  1),ETFMS ) ,(CTRJ( 59),TB2C  ) ,(CTRJ(169),VCONIC) ACC30029
      EQUIVALENCE                 (CBOD ( 2),CBP   ) ,(CBPD (12),NZH   ) ACC30030
     1      ,(CBOD ( 3),CBA   ) ,(CBOD ( 5),CBU   ) ,(CBOD (20),TH    ) ACC30031
     2      ,(CBOD ( 6),CBO   ) ,(CBOD (13),NTH   ) ,(CBOD (14),ZH    ) ACC30032
C                                                                       ACC30033
      COMMON            /DFMCOM/IFM(14),RFM(6,12)                       ACC30034
     1                         ,DFM (4),BFM (577),SNT(2,102)            ACC30035
      DOUBLE PRECISION   BFM,DFM,RFM                                    ACC30036
C                                                                       ACC30037
      COMMON            /DQ3COM/NDEQ(10),CDEQ(10),XDEQ (4)              ACC30038
     1                         ,ADEQ (8),RDEQ (8),VDEQ (8),FDEQ (8,10)  ACC30039
      DOUBLE PRECISION   ADEQ,FDEQ,RDEQ,VDEQ,XDEQ                       ACC30040
      DOUBLE PRECISION   ETIMV                                          ACC30041
      EQUIVALENCE       (XDEQ,ETIMV)                                    ACC30042
C                                                                       ACC30043
      COMMON             D                                              ACC30044
      DOUBLE PRECISION   D(4)                                           ACC30045
C                                                                       ACC30046
C     POSITION AND VELOCITY                                             ACC30047
C                                                                       ACC30048
    1 IF (ETFMS.EQ.ETIMV) GO TO 2                                       ACC30049
C     PLANETARY POSITIONS AND VELOCITIES                                ACC30050
      CALL DPFMRS (ETIMV,EFEDAR,ICB,IERR,ETAPE)                         ACC30051
      IF (IERR.NE.0)      GO TO 999                                     ACC30052
      ETFMS = ETIMV                                                     ACC30053
      UTIMV = ETIMV-DELDAR(1)-DELDAR(2)*ETIMV                           ACC30054
C     REFERENCE CONIC                                                   ACC30055
      CALL STEPDT (ETIMV,RCONIC)                                        ACC30056
      IF (IFBODY.EQ.0)    GO TO 2                                       ACC30057
C     CENTRAL BODY ORIENTATION                                          ACC30058
      CALL GTR2BD (ICB,DFM,0)                                           ACC30059
C     POSITION AND VELOCITY RELATIVE TO CENTRAL BODY                    ACC30060
    2 CONTINUE                                                          ACC30061
      DO 3 I=1,3                                                        ACC30062
      RC(I) = RDEQ(I)+RCONIC(I)                                         ACC30063
      VC(I) = VDEQ(I)+VCONIC(I)                                         ACC30064
    3 CONTINUE                                                          ACC30065
C                                                                       ACC30066
C     ENCKE ACCELERATION                                                ACC30067
C                                                                       ACC30068
   10 CALL ENCKED (CBU,RCONIC,RDEQ,ADEQ)                                ACC30069
C                                                                       ACC30070
C     GRAVITATIONAL HARMONICS                                           ACC30071
C                                                                       ACC30072
   20 IF (IFHARM.EQ.0)    GO TO 30                                      ACC30073
      CALL GRAVD  (CBA,CBU,NZH,NTH,ZH,TH,RC,TB2C,ADEQ,IFHARM)           ACC30074
```

```
C
C     PLANETARY PERTURBATIONS
C
   30 CALL PERTD  (ICB,IPL,RC,EFEDAR,RFM,ADEQ)
C
C     SOLAR PRESSURE
C
   40 IF (IFSOLR.EQ.0)     GO TO 50
      CALL SOLRD  (RC,RFM(1,10),CBA,PREDAR,ADEQ)
C
C     ATMOSPHERIC DRAG
C
   50 IF (IFDRAG.EQ.0)     GO TO 60
      CALL DRAGD  (RC,VC,TB2C,CBO,CBA,PREDAR(2),ADEQ)
C
C     VENTING
C
   60 IF (IFVENT.EQ.0)     GO TO 90
      CALL VENTD  (VC,PREDAR(4),ADEQ)
C
C     STOPPING FUNCTIONS
C
   90 IF (NDEQ(3).EQ.0)    GO TO 999
      D(1) = DDOT(RC,RC)
      ADEQ(5) = D(1)-CBP*CBP
      DO 91 I=1,3
   91 D(I+1) = RC(I)-RFM(I,KCB)
      ADEQ(4) = DDOT(RDEQ,RDEQ)/D(1)-DRECT*DRECT
      ADEQ(6) = DDOT(D(2),D(2))-CBODY(2,KCB)*CBODY(2,KCB)
      ADEQ(7) = D(1)-RSTP2
      ADEQ(8) = DDOT(RC,VC)
  999 RETURN
      END
```

Subroutine: ANTR,ANTR1

Purpose: To interpolate as a function of time for the coordinates of celestial bodies relative to a given central body.

Calling Sequence: CALL ANTR,ANTR1(TW,TF,NB,PØ,NV,VE,DIS)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TW				Time in whole days from 1950 Jan 0.0 UT.
I	TF				Time in fractional days from TW.
I	NB				Central body numbers.
Ø	PØ	(3,7)		km	Array of position values.
I	NV				Dummy variable
Ø	VE	(3,7)		km/sec	Array of velocity values.
I	DIS			km	Dummy variable.

Common storages used: None

Subroutines required: None

__Usage:__

ANTR is a single-precision planetary ephemeris routine which uses a JPL tape of modified central differences. The bodies and their associated numbers are:

I	Body I
0	Earth
1	Moon
2	Sun
3	Venus
4	Mars
5	Saturn
6	Jupiter

The coordinates of all central bodies are computed relative to the body NB. The output arrays contain

PØ(1-3,I) = Position of body I relative to body NB, mean equator, equinox of 1950.0.

VE(1-3,I) = Velocity of body I relative to body NB.

ANTR computes only the positions, and ANTR1 computes both positions and velocities.

__Method__

On each entry, ANTR locates and reads the proper tape record, containing

$$R_o = R(T_j)$$

$$R_1 = R(T_j + h)$$

h = ephemeris interval

and the modified central differences $\delta^2 R_o$, $\delta^2 R_1$, $\delta^4 R_o$, $\delta^4 R_1$.

The ephemeris time,

$$T = TW + TF + \Delta T$$

where

$$T_j \leq T < T_j + h$$

is meaningful, using

$$t = (T - T_j)/h$$

$$u = 1 - t$$

and the positions and velocities at time T are obtained from Everett's formula and its derivative

$$y(t) = \left\{ uy_o + ty_1 \right\} + \left\{ \frac{u(u^2 - 1)}{3!} \delta^2 y_o + \frac{t(t^2 - 1)}{3!} \delta^2 y_1 \right\}$$

$$+ \left\{ \frac{u(u^2 - 1)(u^2 - 4)}{5!} \delta^4 y_o + \frac{t(t^2 - 1)(t^2 - 4)}{5!} \delta^4 y_1 \right\}$$

$$h\dot{y}(t) = \left\{ - y_o + y_1 \right\} + \left\{ - \frac{3u^2 - 1}{3!} \delta^2 y_o + \frac{3t^2 - 1}{3!} \delta^2 y_1 \right\}$$

$$+ \left\{ - \frac{5u^4 - 15u^2 + 4}{5!} \delta^4 y_o + \frac{5t^4 - 15t^2 + 4}{5!} \delta^4 y_1 \right\}$$

```
$IBMAP MC13FM  600,MFTC
*      ANTR
       ENTRY   EWORTO
       ENTRY   HWORTO
       ENTRY   INTR
       ENTRY   INTR1
       ENTRY   ANTR
       ENTRY   ANTR1
       ENTRY   GRAV
       ENTRY   SCALE1
   FDH OPSYN   FDP
EGM    EQU     GRAV
MGM    EQU     GRAV+1
ANTR   NULL
  INTR TXL *+3,,**             POSITION ENTRY
ANTR1  NULL
INTR1  NULL
       SXD VEL,4               POSITION-VELOCITY ENTRY
       TXL *+2,,**
       STZ VEL
       SXA     LDIR,4
       SXA     SYSLOC,4
       CLA     6,4
       ADD     =21
       STA     OUP
       CLA     8,4
       ADD     =21
       STA     XN.V
 TAR   CLA*    3,4
       STO     TARG
 FRA   CLA*    4,4
       STO     TARG+1
 CEN   CLA*    5,4
       STO     CENTER
 RADS  CLA*    9,4
       STO     R
       SXD     TRAP,4
       SXD     TRAP+1,2
       SXD     HELIO-1,1
       ZET     FILE
       TRA     NFIL
       CALL    .FVIO.(TAPNO,FILE)
       CLA     FILE
       STA     FLE
       STA     FEL
       STA     FLEA
       STA     FLPP
       TSX     .OPEN,4
FLPP   MON     *
       CALL    SETN(COM,COM+1)
       CALL    .FVIO.(COM+1,FLUT)
NFIL   CLA     VEL
       TNZ NEU                 MUST INTERPOLATE TO
       REM                     OBTAIN VELOCITY
       CLA CENTER
       SUB KERNO
       TNZ NEU                 MUST INTERPOLATE FOR
       REM                     NEW CENTRAL BODY
       CLA TARG
       SUB TARGO
       TNZ NEU                 TIME HAS CHANGED
       CLA TARG+1
       SUB TARGO+1
       TZE FLEE+6
       REM                     ANALYZE TABLE
       REM                     NEEDS AS A
       REM                     FUNCTION OF
       REM                     CENTRAL BODY
   NEU CLA TARG
       FSB TABLE
       TMI     LOOK            POSITION EPHEMERIS TAPE
       STO COM+19              T-TO
       SUB 20.
       TPL     LOOK
       CLA COM+19
```

```
         FDH 4.                                                  ANTR0750
         STQ COM+18                                              ANTR0760
         STQ     TEM               (T-T0)/4                      ANTR0770
         *IFIX   TEM                                             ANTR0780
         STO     TEM                                             ANTR0790
         *FLOAT  TEM                                             ANTR0800
         STO COM+17                                              ANTR0810
         CHS                                                     ANTR0820
         FAD COM+18                                              ANTR0830
         STO COM+1                                               ANTR0840
         CLA TARG+1                                              ANTR0850
         FDH 4.                                                  ANTR0860
         STQ COM                                                 ANTR0870
         CLA COM                                                 ANTR0880
         FAD COM+1                                               ANTR0890
         STO TARG+2                                              ANTR0900
         LDQ COM+19                                              ANTR0910
         FMP     =9.                                             ANTR0920
         STO     TEM                                             ANTR0930
         *IFIX   TEM                                             ANTR0940
         ADD KERN0+1                                             ANTR0950
         STA GG                                                  ANTR0960
         LDQ COM+17                                              ANTR0970
         FMP     =63.              HELIO SPACING                 ANTR0980
         STO     TEM                                             ANTR0990
         *IFIX   TEM                                             ANTR1000
         ADD KERN0+2                                             ANTR1010
         STA HH                                                  ANTR1020
         AXT     2*SEP,4           MUST INTERPOLATE              ANTR1030
         STZ     XN+2*SEP,4        SO CLEAR STORAGE              ANTR1040
         TIX     *-1,4,1                                         ANTR1050
         AXT     6,4                                             ANTR1060
         STZ     SATPOS+6,4                                      ANTR1070
         TIX     *-1,4,1                                         ANTR1080
         AXT     BSEP,4            SET GRAVITATIONAL             ANTR1090
         STZ     KB0+BSEP,4        COEFFICIENTS                  ANTR1100
         TIX     *-1,4,1           TO ZERO                       ANTR1110
         LXA CENTER,4                                            ANTR1120
         PXD ,4                                                  ANTR1130
         COM                                                     ANTR1140
         PDX ,2                                                  ANTR1150
         TRA     HELIO                                           ANTR1160
    GEO  AXT     3,1                                             ANTR1170
         CLA GRAV+3,1                                            ANTR1180
         STO KB0+3,1                                             ANTR1190
         TIX *-2,1,1                                             ANTR1200
         STZ KB0-1,2                                             ANTR1210
         CLA EWORT0                                              ANTR1220
         STO EWORT                                               ANTR1230
         CLA     HWORTE                                          ANTR1240
         STO     HWORT                                           ANTR1250
         TXH     GG12,4,0                                        ANTR1260
         CLA     RJ                TEST FOR                      ANTR1270
         SUB     R                 INCLUSION OF                  ANTR1280
         TPL     GG12              JUPITER AND SATURN            ANTR1290
         CLA     HWORTJ            IF EARTH IS                   ANTR1300
         STO     HWORT             CENTRAL BODY                  ANTR1310
         CLA     GRAV+5                                          ANTR1320
         STO     KB6-1                                           ANTR1330
         CLA GRAV+6                                              ANTR1340
         STO KB6                                                 ANTR1350
         TXL     GG12,,**                                        ANTR1360
  HELIO  AXT     BSEP,1                                          ANTR1370
         CLA     GRAV+BSEP,1                                     ANTR1380
         STO     KB0+BSEP,1                                      ANTR1390
         TIX *-2,1,1                                             ANTR1400
         STZ     KB2-3,2                                         ANTR1410
  ORTHO  CLA     EWORT0                                          ANTR1420
         STO     EWORT                                           ANTR1430
         CLA HWORT0                                              ANTR1440
         STO HWORT                                               ANTR1450
   GG12  CLA     VEL               CHECK                         ANTR1460
         TZE     GG6               FOR VELOCITY                  ANTR1470
         CLA     EWORT0            OPTIONS                       ANTR1480
         STO EWORT                                               ANTR1490
```

```
       CLA     HWORTV
       STO HWORT
   GG6 CLA     EWORT
       TZE     GG+2
       CLA EVEL
       STO VEL+1
       CLA TARG+1          GEOCENTRIC
       TSX TAB,4           INTERPOLATION
    GG PZE     **,,9
       PZE     XN,,EWORT+1
       CLA HWORT
       TZE     HH+2
       CLA HVEL
       STO VEL+1
       CLA TARG+2          HELIOCENTRIC
       TSX TAB,4           INTERPOLATION
    HH PZE     **,,63
       PZE     XN+3,,HWORT+1
 RESET AXT     3,4               PLACE
       CLA     XN+6,4            COORDINATES
       FDH     EVEL              IN OLD
       STQ     XN.+18,4          FORMAT
       CLA     XN+9,4
       FDH     EVEL
       STQ     XN.+12,4
       LDQ     XN+3,4
       STQ     XN+6,4
       FMP     MU
       FSB     XN+18,4
       STO     XN+9,4            POSITION OF SUN
       STZ     XN+3,4
       LDQ     XN.+3,4
       STQ     XN.+6,4
       FMP     MU
       FSB     XN.+18,4
       STO     XN.+9,4           VELOCITY OF SUN
       STZ     XN.+3,4
       CLA     SATPOS+3,4
       STO     XN+18,4
       CLA     SATVEL+3,4
       STO     XN.+18,4
       TIX     RESET+1,4,1
       LXA CENTER,4
       LXD VEL,2           IX2 NOT ZERO IF VELOCITY OPTION
       TXH RVPRT,2,0       GET COORDINATES FOR PRINTING
       TRA     RVPRT
       CAL     KB6
       ORA     KB6-1
       TZE     FLEE
       AXT     3,1
       CLA XN+21,1
       FAD XN+9,1
       STO XN+21,1
       CLA     XN+18,1
       FAD     XN+9,1
       STO     XN+18,1
       TIX     *-6,1,1
       TRA     FLEE
 TESTM TXH     RVPRT,4,1
       AXT     3,1               MOON-CENTERED
       CLS XN+6,1
       STO XN+3,1
       CLA XN+9,1
       FSB XN+6,1
       STO XN+9,1
       STZ XN+6,1
       TIX *-6,1,1
  FLEE CLA CENTER
       STO KERNO
       CLA TARG
       STO TARGO
       CLA TARG+1
       STO TARGO+1
       AXT     21,1
       CLA     XN+21,1
```

```
OUP      STO      **,1
         TIX      OUP-1,1,1
         CLA      VEL
         TZE      OUT
         AXT      21,1
         CLA      XN.+21,1
XN.V     STO      **,1
         TIX      XN.V-1,1,1
OUT      LXD      HELIO-1,1
         LXD      TRAP+1,2
         LXD      TRAP,4
         TRA      1,4
         REM
         REM TSX TAB,4              (AC)=INTERPOLATIVE ARGUMENT
         REM PZE B,,K
         REM PZE A,,C
         REM                        B=START OF DATA BLOCK
         REM                        K=WORDS PER SUB BLOCK
         REM                        A=START OF RESULT BLOCK
         REM                        C=SKIP CODE WORD LOCATION
         REM
    TAB  SXD COM+9,4
         SXD COM+8,2
         SXD COM+7,1
         STO ARG
         CLA 1,4
         STA      TAB18
         LRS 18
         ADD 1,4
         STA      TAB21
         CLA 2,4
         PAX      ,1
         TXI      *+1,1,SEP
         SXA      TAB29,1
         ARS 18
         STA VELOP
         AXT      2,1
         NZT      VEL
         TXI VELOP,1,-1
  VELOP  CLA      **,1                   PICK UP SKIP
         STO      VEL+2                  CODE WORD
         STZ      SATURN
         ANA      =03                    TEST FOR INCLUSION OF SATURN
         SUB      =02                    WHENEVER JUPITER IS PRESENT
         TNZ      *+3
         CLA      =2
         STO      SATURN
         TXL      POSOP,1,1
         REM                        VELOCITY OPTION
         CLA ARG
         TSX COEFF.,4               FORM THE
         AXT      3,4                    E1(2J).
         CLS COM+13,4
         STO COM+16,4
         TIX *-2,4,1
         CLA 1.                     FORM THE
         FSB ARG                    E0(2J).
         TSX COEFF.,4
         TRA      TAB1-1
         REM                        POSITION OPTION
  POSOP  CLA      ARG
         TSX COEFF,4                FORM THE
         AXT      3,4                    E1(2J)
         CLA COM+13,4
         STO COM+16,4
         TIX *-2,4,1
         CLA 1.                     FORM THE
         FSB ARG                    E0(2J)
         TSX COEFF,4
         LXD      COM+9,4                REFER TO
         CLA      2,4                    POSITION
         STA      TAB29                  STORAGE
         SXD COM+4,1
  TAB1   AXT      0,2
         AXT      0,1
```

```
         LDQ     VEL+2                                          ANTR3000
         ZAC                                                    ANTR3010
         LGL     2                                              ANTR3020
         PAX     ,4                                             ANTR3030
         STQ     VEL+2                                          ANTR3040
         CAL     =03                                            ANTR3050
         ORS     VEL+2                                          ANTR3060
         TXH     END,4,2                                        ANTR3070
         TXH     FORT,4,0                                       ANTR3080
         TXI     *+1,1,-3                                       ANTR3090
         TXI     TAB1+2,2,-1                                    ANTR3100
FORT     AXT     3,4                                            ANTR3110
TAB18    LDQ     **,1               X(0)                        ANTR3120
         FMP COM+13,4                                           ANTR3130
         STO COM                                                ANTR3140
TAB21    LDQ     **,1               X(1)                        ANTR3150
         FMP COM+16,4                                           ANTR3160
         FAD COM                                                ANTR3170
         STO COM+4,4                                            ANTR3180
         TXI *+1,1,-1                                           ANTR3190
         TIX     TAB18,4,1                                      ANTR3200
         FAD COM+2                                              ANTR3210
         FAD COM+1                                              ANTR3220
TAB29    STO     **,2               X(T)                        ANTR3230
         TXI     TAB1+2,2,-1                                    ANTR3240
END      NZT     SATURN                                         ANTR3250
         TRA     END1                                           ANTR3260
         TXH     END1,2,-4          COMPLETED SATURN            ANTR3270
         LXD     COM+4,4                                        ANTR3280
         CLA     TABSAT+2,4                                     ANTR3290
         STA     TAB29                                          ANTR3300
         AXT     0,2                                            ANTR3310
         CAL     =053700000000      TWO ADDITIONAL COORDINATES  ANTR3320
         SLW     VEL+2                                          ANTR3330
         TRA     FORT                                           ANTR3340
END1     LXD     COM+4,1                                        ANTR3350
         TIX     VELOP,1,1                                      ANTR3360
         LXD COM+9,4                                            ANTR3370
         LXD COM+8,2                                            ANTR3380
         LXD COM+7,1                                            ANTR3390
         TRA 3,4                                                ANTR3400
COEFF    STO COM+10          CALCULATE                          ANTR3410
         LDQ COM+10          POSITION                           ANTR3420
         FMP COM+10          COEFFICIENTS                       ANTR3430
         STO COM+12          FOR EVERETT,S                      ANTR3440
         FSB 1.              INTERPOLATION                      ANTR3450
         FDH 6.              FORMULA                            ANTR3460
         FMP COM+10                                             ANTR3470
         STO COM+11                                             ANTR3480
         CLA COM+12                                             ANTR3490
         FSB 4.                                                 ANTR3500
         FDH 20.                                                ANTR3510
         FMP COM+11                                             ANTR3520
         STO COM+12                                             ANTR3530
         TRA 1,4                                                ANTR3540
COEFF.   STO COM             CALCULATE                          ANTR3550
         CLS 1.              VELOCITY                           ANTR3560
         FDH VEL+1           COEFFICIENTS                       ANTR3570
         STQ COM+10          FOR EVERETT,S                      ANTR3580
         LDQ COM             INTERPOLATION                      ANTR3590
         FMP COM             FORMULA                            ANTR3600
         STO COM+12                                             ANTR3610
         XCA                                                    ANTR3620
         FMP 3.                                                 ANTR3630
         FSB 1.                                                 ANTR3640
         FDH 6.                                                 ANTR3650
         FMP COM+10                                             ANTR3660
         STO COM+11                                             ANTR3670
         CLA COM+12                                             ANTR3680
         FSB 3.                                                 ANTR3690
         XCA                                                    ANTR3700
         FMP 5.                                                 ANTR3710
         XCA                                                    ANTR3720
         FMP COM+12                                             ANTR3730
         FAD 4.                                                 ANTR3740
```

```
        FDH 120.
        FMP COM+10
        STO COM+12
        TRA 1,4
 RVPRT  AXT     0,2                EXPRESS ALL
        AXT     3,4             BODIES GEOCENTRICALLY
        AXT     3,1
        CLA XN+9,2
        FAD XN+9,4
        STO XN+9,2
        CLA     VEL
        TZE     *+4
        CLA XN.+9,2
        FAD XN.+9,4
        STO XN.+9,2
        TXI *+1,2,-1
        TIX     RVPRT+3,4,1
        TXH     RVPRT+1,2,9-SEP
        CLA CENTER              BUFFER
        ALS 1                   CENTRAL
        ADD CENTER              BODY
        PAC     ,4
        CLA     XN,4
        STO COM+3,1
        CLA     XN.,4
        STO COM+6,1
        TXI *+1,4,-1
        TIX *-5,1,1
RVPRT1  AXT     0,2                EXPRESS ALL
        AXT     3,4                BODIES IN TERMS
        CLA     XN,2               OF THE CENTRAL
        FSB     COM+3,4            BODY
        STO XN,2
        CLA     VEL
        TZE     *+4
        CLA XN.,2
        FSB COM+6,4
        STO XN.,2
        TXI *+1,2,-1
        TIX     RVPRT1+2,4,1
        TXH     RVPRT1+1,2,-SEP
        TRA     FLEE
LOOK    CLA     TLAST
        FSB TARG
        TNZ *+2
        TSX ERP,4               ARGUMENT TOO LARGE
        TMI *-1
        CLA TARG
        FSB TFIRST
        TPL *+2
        TSX ERP,4               ARGUMENT TOO SMALL
        FDH 20.
        STQ     TEM
        *IFIX   TEM              HELIO SPACING
        STO     TEM
        *FLOAT  TEM
        XCA
        FMP 20.
        FAD TFIRST
        STO COM+1               TIME ON RECORD
        CLA TABLE
        FSB COM+1
        TMI FINDIT
        FDH 20.
        STQ     TEM
        *IFIX   TEM
        ADD 1F                  RECORDS TO
        PAX     ,1                BE BACKED
        TXL     BRS,1,20          OVER
        TSX     .REW,4
FILE    PZE     0
        TRA     FINDIT
BRS     TSX     .BSR,4
FLE     PZE     0,0,ERR
        TIX     BRS,1,1
```

```
FINDIT AXT     10,2                                                     ANTR4500
       TSX     .READ,4                                                  ANTR4510
FEL    PZE     *                                                        ANTR4520
       PZE     ERR,0,ERR                                                ANTR4530
       IOCT    TABLE,,569                                               ANTR4540
       CLA     TABLE                                                    ANTR4550
       SUB     COM+1                                                    ANTR4560
       TMI     *+2                                                      ANTR4570
       TNZ     LOOK               TAPE NOT POSITIONED PROPERLY          ANTR4580
       TNZ     FINDIT                                                   ANTR4590
       AXT     567,4                                                    ANTR4600
       CAL     TABLE              CHECK                                 ANTR4610
       ACL     A+567,4            SUM                                   ANTR4620
       TIX     *-1,4,1                                                  ANTR4630
       LAS     C                                                        ANTR4640
       TRA     *+2                                                      ANTR4650
       TRA     SCALE                                                    ANTR4660
BSRA   TSX     .BSR,4                                                   ANTR4670
FLEA   PZE     0,0,ERR                                                  ANTR4680
       TIX     FINDIT+1,2,1                                             ANTR4690
ERR    CALL    .FWRD.(FLUT,FMT)                                         ANTR4700
ERRR   TSX     .FFIL.,4                                                 ANTR4710
       CALL    EXIT                                                     ANTR4720
ERP    CALL    .FWRD.(FLUT,FMT1)                                        ANTR4730
       CLA     TARG                                                     ANTR4740
       TSX     .FCNV.,4                                                 ANTR4750
       TRA     ERRR                                                     ANTR4760
 FMT   BCI     7,(33H0TAPE ERROR IN ANTR. EXIT CALLED.)                 ANTR4770
FLUT   PZE     0               SYSTEM OUTPUT FILE ADDRESS               ANTR4780
FMT1   BCI     7,(7H0TIME =,E20.8,21H OUT OF RANGE IN ANTR)             ANTR4790
 SCALE AXT     189,4                                                    ANTR4800
       LDQ     A+189,4            SCALE                                 ANTR4810
       FMP     SCALE1             GEOCENTRIC                            ANTR4820
       STO     A+189,4            EPHEMERIS                             ANTR4830
       TIX     *-3,4,1                                                  ANTR4840
       AXT     378,4                                                    ANTR4850
       LDQ     B+378,4            SCALE                                 ANTR4860
       FMP     SCALE2             HELIOCENTRIC                          ANTR4870
       STO     B+378,4            EPHEMERIS                             ANTR4880
       TIX     *-3,4,1                                                  ANTR4890
       CLA     EGM                                                      ANTR4900
       FAD     MGM                                                      ANTR4910
       STO     COM                OF MOON                               ANTR4920
       CLA     MGM                                                      ANTR4930
       FDH     COM                BARYCENTER                            ANTR4940
       STQ     MU                 FOR POSITION OF EARTH                 ANTR4950
       TRA     NEU                                                      ANTR4960
SCALE1 DEC     6378.165           EARTH RADIUS                          ANTR4970
SCALE2 DEC     149599000.         ASTRONOMICAL UNIT                     ANTR4980
  GRAV DEC     3.986032E5         EARTH                                 ANTR4990
       DEC     4.900759E3         MOON                                  ANTR5000
       DEC     1.32715445E11      SUN                                   ANTR5010
       DEC     3.247695E5         VENUS                                 ANTR5020
       DEC     4.297780E4         MARS                                  ANTR5030
       DEC     3.791870E7         SATURN                                ANTR5040
       DEC     1.267106E8         JUPITER                               ANTR5050
TEMPDT DEC     34.,0              E.T.-U.T.                             ANTR5060
TFIRST DEC     10.0               0 HR JAN 11,1950 JD=2433292.5         ANTR5070
 TLAST DEC     11530.0            0 HR JUL 27,1981 JD=2444812.5         ANTR5080
    RJ DEC 1E6               JUPITER TEST DISTANCE                      ANTR5090
EVEL   DEC     86400.                                                   ANTR5100
  HVEL DEC 345600.                                                      ANTR5110
    MU PZE                        MASS RATIO OF MOON TO BARYCENTER      ANTR5120
       OCT     000000520052       MARS, JUPITER VELOCITY                ANTR5130
 HWORT OCT     0                                                        ANTR5140
       OCT     527777777777       MOON VELOCITY                         ANTR5150
 EWORT OCT     0                                                        ANTR5160
HWORTJ OCT     000000005252       BARYCENTER, JUPITER                   ANTR5170
HWORTE OCT     000000005200       BARYCENTER                            ANTR5180
EWORTO OCT     527777777777                                             ANTR5190
HWORTO OCT     000052525252                                             ANTR5200
HWORTV OCT     525252525252                                             ANTR5210
 TARGO DEC 20.,0             FORMER TIME                                ANTR5220
 KERNO PZE                        FORMER CENTER                         ANTR5230
       PZE     A                  GEOCENTRIC REFERENCE                  ANTR5240
```

```
        PZE      B                 HELIOCENTRIC REFERENCE            ANTR5250
   ARG                                                               ANTR5260
SATURN  DEC      0                 0=NOT NEEDED,OTHERWISE COMPUTE SATURN ANTR5270
TABSAT  PZE      SATVEL                                              ANTR5280
        PZE      SATPOS                                              ANTR5290
    1F  DEC      1                                                   ANTR5300
    1.  DEC      1.                                                  ANTR5310
    3.  DEC      3.                                                  ANTR5320
    4.  DEC      4.                                                  ANTR5330
    5.  DEC      5.                                                  ANTR5340
    6.  DEC      6.                                                  ANTR5350
   20.  DEC 20.                                                      ANTR5360
  120.  DEC 120.                                                     ANTR5370
86400.  DEC      86400.                                              ANTR5380
   VEL  BSS 3                  VELOCITY OPTION,H,SKIP CODE           ANTR5390
SATPOS  BSS      3                                                   ANTR5400
SATVEL  BSS      3                                                   ANTR5410
TRAP    BSS      2                                                   ANTR5420
*                                                                    ANTR5430
 TABLE  DEC      0                 RESERVE                           ANTR5440
     A  BSS      189               FOR                               ANTR5450
     B  BSS      378               WORKING                           ANTR5460
     C  BSS      1                 EPHEMERIS                         ANTR5470
*                                                                    ANTR5480
EPHTAB  SYN      TABLE+568                                           ANTR5490
COM     BSS      21                                                  ANTR5500
TARG    PZE                                                          ANTR5510
        PZE                                                          ANTR5520
        PZE                                                          ANTR5530
CENTER  PZE                                                          ANTR5540
KB0     PZE                                                          ANTR5550
KB2     PZE                                                          ANTR5560
KB6     PZE                                                          ANTR5570
R       PZE                                                          ANTR5580
XN      BSS      21                                                  ANTR5590
XN.     BSS      21                                                  ANTR5600
 TEM    PZE                                                          ANTR5610
 LDIR   LDIR                                                         ANTR5620
TAPNO   PZE      8                                                   ANTR5630
T       PZE                                                          ANTR5640
   SEP  SYN      XN.-XN                                              ANTR5650
  BSEP  SYN      7                                                   ANTR5660
        END                                                          ANTR5670
```

Subroutine: ARKTNS

Purpose: To obtain the arctangent of Y/X. The solution is obtained between ±180 degrees if N is set equal to 180. If N≠180, the angle is computed between 0 and 360.

Calling Sequence: Z = ARKTNS(N,X,Y)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	N				Range key (see above).
I	X				
I	Y				
Ø	Z			radians	$\tan^{-1}(Y/X)$

Common storages used: None

Subroutines required: None

```
$IBFTC MC13AT  XR3,M94,NOREF,NODD
      FUNCTION ARKTNS(N,X,Y)
C     COMPUTES 4-QUADRANT ARCTANGENT OF Y/X IN RADIANS
C     N=360  ANGLE LIES IN RANGE (0,360) DEG
C     N=180  ANGLE LIES IN RANGE (-180,180) DEG
C     USES ATAN2, ALLOWS 0/0=0
      IF(X)100,10,100
 10   IF(Y) 20,30,40
 20   ARKTNS =-1.5707963
      GO TO 200
 30   ARKTNS= 0.0
 35   RETURN
 40   ARKTNS=1.5707963
      GO TO 35
 100  ARKTNS = ATAN2(Y,X)
 200  IF(N-180)205,35,205
 205  IF(ARKTNS)210,35,35
 210  ARKTNS=ARKTNS+6.28318531
      GO TO 35
       END
```

Subroutine: BACK

Purpose: To backspace a binary tape N logical records or a BCD tape N physical records.

Calling Sequence: CALL BACK(M,N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	M				Tape logical number.
I	N				Number of records to be backspaced.

Common storages used: None

Subroutines required: None

```
$IBFTC MC13J3  XR3,M94,NODD
      SUBROUTINE BACK(M,N)                                              BACK0001
C THIS SUBROUTINE BACKSPACES BINARY TAPE M,N LOGICAL RECORDS             BACK0002
C OR BCD TAPE M, N PHYSICAL RECORDS                                      BACK0003
      DO 100 I=1,N                                                      BACK0004
      BACKSPACE M                                                       BACK0005
  100 CONTINUE                                                          BACK0006
      RETURN                                                            BACK0007
      END                                                               BACK0008
```

Subroutine: BARN

Purpose: Random number generator. Provides either uniform random variable over the interval (0, 1) or a normal random number with zero mean and unit variance.

Calling Sequence: Y = BARN (I)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	I				+1, uniform distribution -1, normal distribution
Ø	Y				Random number

Common storages used: None

Subroutines required: None

```
$IBMAP MC13B4
 BARN   SAVE     (1,2)                                          BARN0000
        CLA      1,4                                            BARN0001
        PDX      0,2                                            BARN0002
        TXL      GG,2,1                                         BARN0003
        CLA*     4,4                                            BARN0004
        TMI      GG                                             BARN0005
        CLA*     5,4                                            BARN0006
        STO      HERE                                           BARN0007
 GG     CLA*     3,4                                            BARN0008
        TMI      RDNN                                           BARN0009
RDUN    LDQ      HERE                                           BARN0010
        MPY      HERE+1                                         BARN0011
        LLS      4,0                                            BARN0012
        ALS      4,0                                            BARN0013
        LRS      4,0                                            BARN0014
        STQ      HERE                                           BARN0015
        ADD      HERE                                           BARN0016
        STO      HERE                                           BARN0017
        ARS      4,0                                            BARN0018
        ORA      HERE+2                                         BARN0019
        FAD      HERE+2                                         BARN0020
SWITCH  NOP      NEXT                                           BARN0021
        TRA      RETURN                                         BARN0022
HERE    OCT      2312421637,1737,200000000000                   BARN0023
RDNN    LXA      L20,1                                          BARN0024
        CLA      HERE+2                                         BARN0025
        STO      C,0                                            BARN0026
        LDQ      TRA                                            BARN0027
        SLQ      SWITCH                                         BARN0028
        TRA      RDUN                                           BARN0029
NEXT    LDQ      NOP                                            BARN0030
        SLQ      SWITCH                                         BARN0031
        FAD      C,0                                            BARN0032
        STO      C,0                                            BARN0033
        TIX      NEXT-3,1,1                                     BARN0034
        FDP      L20+1,0                                        BARN0035
        CLA      L20+4,0                                        BARN0036
        LLS      35,0                                           BARN0037
        FSB      L20+2,0                                        BARN0038
        LRS      35,0                                           BARN0039
        FMP      L20+3,0                                        BARN0040
RETURN  TXL      RETRN,2,1                                      BARN0041
        LDQ      HERE                                           BARN0042
        STQ*     5,4                                            BARN0043
RETRN   RETURN   BARN                                           BARN0044
L20     HTR      20,0                                           BARN0045
        DEC      20.,.5,15.49193340,0                           BARN0046
TRA     OCT      +002000000000                                  BARN0047
NOP     OCT      +076100000000                                  BARN0048
C       HTR      0,0                                            BARN0049
        END                                                     BARN0050
```

Subroutine: BECSTA

Purpose: Computes start and stop time arrays for stations and beacons.

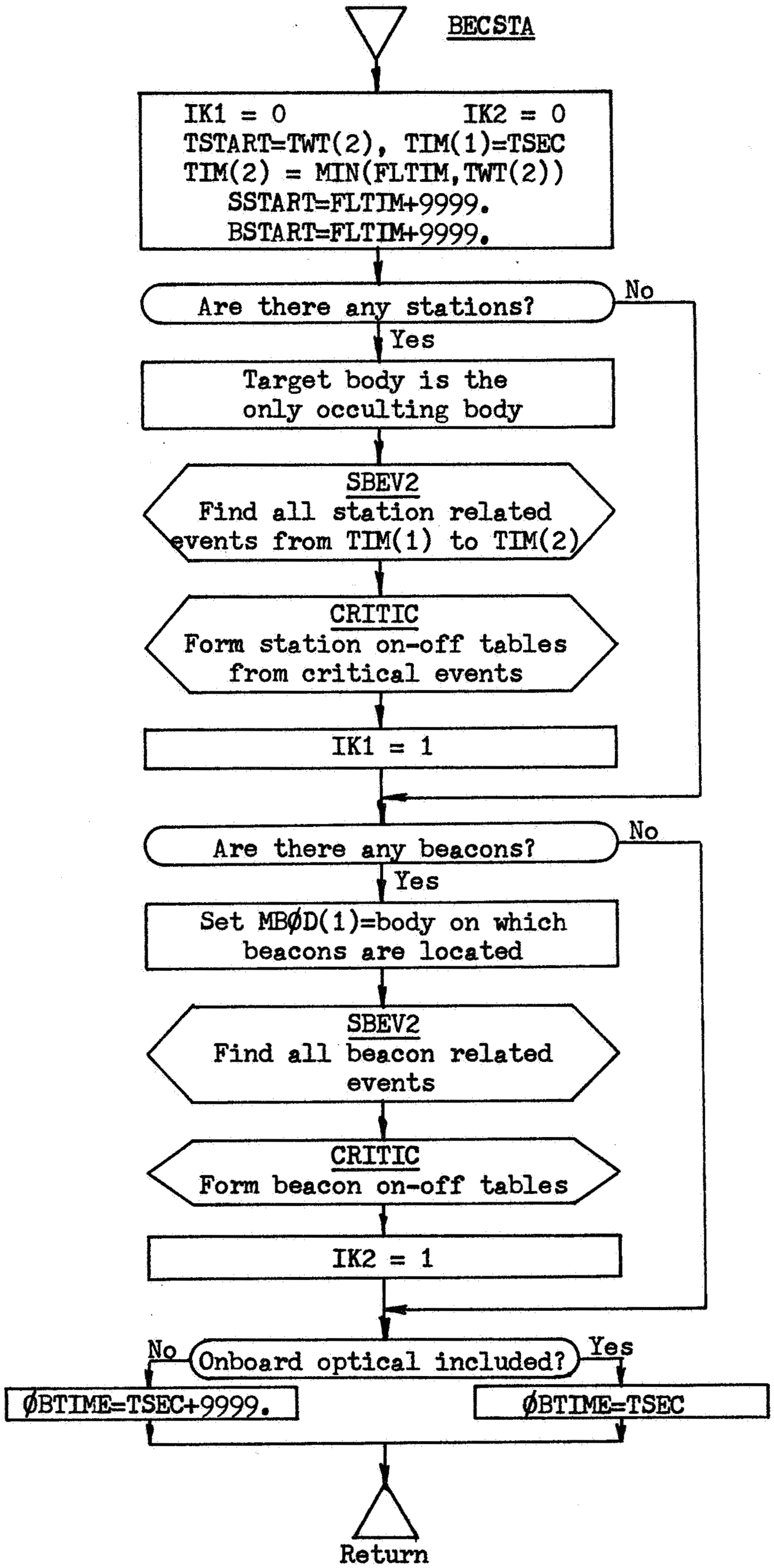
Calling Sequence: CALL BECSTA(KSTA,KBEA,KMAXS,KMAXB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	KSTA	75			Array of station numbers corresponding to times in SSTART,SSTØP arrays.
Ø	KBEA	75			Array of beacon numbers corresponding to times in BSTART,BSTØP arrays.
Ø	KMAXS				Number of station on-off time pairs found.
Ø	KMAXB				Number of beacon on-off time pairs found.

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

Subroutines required: CRITIC,EXINST,SBEV2


BECSTA
IK1 = 0
IK2 = 0
TSTART=TWT(2), TIM(1)=TSEC
TIM(2) = MIN(FLTIM,TWT(2))
SSTART=FLTIM+9999.
BSTART=FLTIM+9999.
Are there any stations?
No
Yes
Target body is the only occulting body
SBEV2
Find all station related events from TIM(1) to TIM(2)
CRITIC
Form station on-off tables from critical events
IK1 = 1
Are there any beacons?
No
Yes
Set MBØD(1)=body on which beacons are located
SBEV2
Find all beacon related events
CRITIC
Form beacon on-off tables
IK2 = 1
No
Onboard optical included?
Yes
ØBTIME=TSEC+9999.
ØBTIME=TSEC
Return

```
$IBFTC MC13BA  NOREF,M94,NODD,XR3
CMC13BA  BECSTA  COMPUTES START AND STOP ARRAYS FOR STATIONS AND BEACONS
      SUBROUTINE BECSTA(KSTA,KBEA,KMAXS,KMAXB)
C            SUBRS REQUIRED  SBEV2
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)
      COMMON/EXIC/W(36),CRX(9,2)
      DIMENSION     TWT(3),     TIM(2),     STIME(50),      ISC(12)
     1,             MBOD(4),  XIN(6),    TI(3),           S(23,12)
     2,             SECR(12), ISEE(12), ISTIM(50),      BTIME(50)
     3,             IBC(10),  IB(36),    B(91),          BECR(10)
     4,             IBEE(10), IBTIM(50),IOBA(3,6)
      DIMENSION     SSTART(75),     SSTOP(75),       KSTA(75)
     1,             BSTART(75),     BSTOP(75),       KBEA(75)
      EQUIVALENCE   (C(200),S),      (C(476),B)
      EQUIVALENCE   (IW(8),ISC),     (IW(20),IBC),       (IW(58),IOBA)
     1,             (IW(312),IB),    (IW(355),ISEE),     (IW(367),IBEE)
     2,             (IW(382),NOR),   (IW(383),ITARG)
     3,             (IW(384),IK1),   (IW(385),KSMAX),    (IW(386),IK2)
     4,             (IW(387),KBMAX),(IW(389),ISTIM),     (IW(439),IBTIM)
     5,             (IW(489),ITRIG),   (IW(490),KOUNT)
      EQUIVALENCE   (CW(1134),XIN),  (CW(1140),TSEC)
     1,             (CW(1160),TI),    (CW(1175),OBTIME),(CW(1178),TSTART)
     2,             (CW(1179),FLTIM),(CW(1188),TWT),    (CW(1441),BECR)
     3,             (CW(1200),STIME),(CW(1250),BTIME), (CW(1360),SECR)
     4,        (CW(222),SSTART),  (CW(297),SSTOP)
     5,        (CW(372),BSTART),  (CW(447),BSTOP)
      DOUBLE PRECISION TWR
      IK1=0
      IK2=0
      TSTART = TWT(2)
      TIM(1) = TSEC
      TIM(2) = TWT(2)
      IF(TIM(2).GT.FLTIM) TIM(2)=FLTIM
      SSTART=FLTIM+9999.
      IF(ISC .EQ. 0) GO TO 24
      MBOD(2) = ITARG
      IF(ITARG .EQ. 1) MBOD(2) =0
      MBOD(3) =0
      ITRIG=1
      CALL SBEV2(XIN,TI,NOR,TIM,S,23,12,ISC,SECR,ISEE,ISTIM,STIME,MBOD)
      IF(KOUNT.LE.0) GO TO 2
      DO 1 I=1,KOUNT
    1 BACKSPACE 10
      M=0
      N=18
      TWR=TSEC
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
    2 KSMAX=MBOD(4)
      ITRIG=0
      KOUNT=0
      CALL CRITIC(12,ISC,STIME,ISTIM,ISEE,KSMAX,TIM,SSTART,SSTOP,KSTA,
     1KMAXS)
      IK1 = 1
24    CONTINUE
      BSTART=FLTIM+9999.
      IF(IBC .EQ. 0) GO TO 25
      MBOD(1) = IB(36)
      MBOD(2) = 0
      ITRIG=1
      CALL SBEV2(XIN,TI,NOR,TIM,B,8,10,IBC,BECR,IBEE,IBTIM,BTIME,MBOD)
      IF(KOUNT.LE.0) GO TO 241
      DO 240 I=1,KOUNT
  240 BACKSPACE 10
      M=0
      N=18
      TWR=TSEC
      CALL EXINST(TWR,M,N,XIN,XIN(4),W,CRX,10,ITRIG,KOUNT)
  241 KBMAX=MBOD(4)
      ITRIG=0
      KOUNT=0
      CALL CRITIC(10,IBC,BTIME,IBTIM,IBEE,KBMAX,TIM,BSTART,BSTOP,KBEA,
     1KMAXB)
      IK2 = 1
25    CONTINUE
      OBTIME = TSEC
```

```
      IF(IOBA .EQ. 0) OBTIME = FLTIM +9999.                            BCST0075
      RETURN                                                           BCST0076
      END                                                              BCST0077
```

Subroutine: BIBCD

Purpose: To convert a binary integer to BCD.

Calling Sequence: BCD = BIBCD(BIN)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	BIN				Full word binary integer $0 \le BIN < 10^{-6}$
Ø	BCD				BCD equivalent of BIN with leading zeros replaced by blanks.

Common storages used: None

Subroutines required: None

```
$IBMAP MC13I6  50,NODD
       TTL     I6 CONVERSION OF POSITIVE FIXED-POINT NUMBERS              BIBC0001
 BIBCD SAVE                                                               BIBC0002
       CLA*    3,4                                                        BIBC0003
       TNZ     NZ                                                         BIBC0004
       CLA     BLNKZ                                                      BIBC0005
 RETRN RETURN  BIBCD                                                      BIBC0006
 NZ    SSP                                                                BIBC0007
       STO     SV                                                         BIBC0008
       AXT     6,4                                                        BIBC0009
       CLA     TEMP                                                       BIBC0010
       STO     TEMP+7,4                                                   BIBC0011
       TIX     *-1,4,1                                                    BIBC0012
       AXT     6,4                                                        BIBC0013
       LDQ     SV                                                         BIBC0014
 NDN   PXA     0,0                                                        BIBC0015
       DVP     TEN                                                        BIBC0016
       STQ     SV                                                         BIBC0017
       ALS     30                                                         BIBC0018
       STP     TEMP+7,4                                                   BIBC0019
       ORS     TEMP+7,4                                                   BIBC0020
       NZT     SV                                                         BIBC0021
       TRA     DN                                                         BIBC0022
       TIX     NDN,4,1                                                    BIBC0023
 DN    LDQ     MQTST                                                      BIBC0024
       CRQ     TEMP,,6                                                    BIBC0025
       XCA                                                                BIBC0026
       RETURN  BIBCD                                                      BIBC0027
 MQTST OCT     060504030201                                               BIBC0028
 SV    PZE     0                                                          BIBC0029
 BLNKZ OCT     606060606000                                               BIBC0030
 TEMP  SIX     *                                                          BIBC0031
       BSS     6                                                          BIBC0032
 TEN   PZE     10                                                         BIBC0033
       END
```

Subroutine: BLDCØV

Purpose: To build up square submatrices of the covariance matrix from standard deviations and normalized correlations and to build subsets of the vector of state variable BCD names from an array of possible names.

Calling Sequence: CALL BLDCØV(S,TITLE,KHØLD,NUMBER,LØCAT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	S	d			Vector of standard deviations and correlations.
I	TITLE				Vector of possible state names.
I	KHØLD				Construction of P will begin with element P(KHOLD+1,KHOLD+1).
I	NUMBER				Dimension of submatrix P to be built on this entry.
I	LØCAT				Array associated with S, used for choosing the desired element of S.

Common Storages Used: /ES1CØM/
Subroutines Required: None.

1. /ES1CØM/. The arrays of ES1CØM (q.v.) used by BLDCØV are

Ø PNEW	d(N,N)	Covariance matrix, P
Ø STNAMN	(N)	Vector of state names
I ILØCN	(N)	Vector of state element idendification indices

where N is the maximum length of the state vector.

2. Usage

The allowed state variables for the Differential Correction Program are divided into logically related groups. BLDCØV is called for each of the variable groups to be included in the state, and for which the related portion of the covariance matrix is to be built up from block data arrays of standard deviations and normalized correlations. BLDCØV is used only at case initialization and is called only by SETCAS. The resulting initial covariance matrix for a case allows correlations between variables within the same variable group; correlations between variables in different groups are set to zero by SETCAS. For example, if the state consisted of variables from each of three groups, the built-up initial covariance matrix would have the following form:

P_1	ϕ	ϕ
ϕ	P_2	ϕ
ϕ	ϕ	P_3

where ϕ represents the null matrix. Each of the blocks P_i, i = 1,3 is built separately. Each requires a call of BLDCØV if it is to be built from stored block data. (Such a block might, instead, be lifted intact from a previous tape-stored estimate. This choice is exercised in SETCAS.) The variable groups and the related arrays used in the calls of BLDCØV are as follows:

Group Code Name	Description	S Vector of Standard Deviations and Correlations	TITLE Vector of BCD Element Names	LØCAT Vector of Indices for Using S
SPCDAT	Spacecraft position and velocity	SPCDAT	SPCNAM	LØCSPC
EFEMRS	Planetary mu's and ephemeris parameters	EFEDAT	EFENAM	LØCEFE
PRESUR	Solar pressure, drag, venting	PREDAT	PRENAM	LØCPRE
DELTIM	Ephemeris to universal time conversion	DELDAT	DELNAM	LØCDEL
HARM03	Gravitational harmonics for Earth	EHADAT	HARNAM	LØCHAR
HARM11	Moon harmonics	MHADAT	HARNAM	LØCHAR
HARMmn[1]	Extra body harmonics	XHADAT	HARNAM	LØCHAR
(Name)[2]	Tracking station biases	SEN(1,K)[3]	STBNAM	LØCSTA

[1] Where mn is a body number other than 03 or 11.

[2] Any valid station name.

[3] Where K is the internal number assigned to the station by subroutine SETSTA.

At each call BLDCØV will construct the square portion beginning in P(KHØLD+1,KHØLD+1) and ending in P(KHØLD+NUMBER,KHØLD+NUMBER). BLDCØV will construct the subset from STNAMN(KHØLD+1) to STNAMN(KHØLD+NUMBER). At any call of BLDCØV, KHØLD is the size of the covariance matrix already built, NUMBER is the number of additional rows and columns to be added by this call.

The use of the arrays S, TITLE, ILØCN, and LØCAT is illustrated by the following example:

Suppose that the first six rows and columns of P have already been built. Then KHØLD = 6. The first elements of ILØCN contain the identifying numbers for the six state variables for which P has already been built. Now, suppose that the next group to be added is the solar pressure, drag, vent group. Let us assume that the two drag parameters are to be included. Then NUMBER = 2 and ILØCN(7) = 2, ILØCN(8) = 3. The various arrays are as follows:

ILØCN = (k_1, k_2, k_3, k_4, k_5, k_6, 2, 3, n_1, m_2, . . .)

TITLE = (3HSPR, 3HDR1, 3HDR2, 3HVNT) = PRENAM

KHØLD = 6

NUMBER = 2

LØCDAT = (7, 9, 10) = LØCPRE

S = PREDAT:

```
    S(1) = Solar Pressure value
     (2) = Drag 1           "
     (3) = Drag 2           "
     (4) = Venting          "
     (5) = Solar Pressure Standard Deviation
     (6) = Drag 1             "         "
     (7) = Drag 2             "         "
     (8) = Venting            "         "
     (9) = C(1,2)   , correlation between Solar Pressure and Drag 1
    (10) = C(1,3)
    (11) = C(1,4)
    (12) = C(2,3)
    (13) = C(2,4)
    (14) = C(3,4)
```

The variable identifying numbers to be used are ILØCN(7) = 2 and ILØCN(8) = 3.

BDLCØV first picks up elements 2 and 3 from the TITLE array and puts them in positions 7 and 8 of the STNAMN array.

Next, the diagonal terms P(7,7) and P(8,8) of the covariance matrix are loaded with the correct standard deviations from the S array. For this purpose, the LØCAT array is used. Let K be the integer part of LØCAT(1)/2 + 1. That is, K = 4. The values 2 and 3 from the ILO¢N array are added to K to yield 6 and 7. Then P(7,7) and P(8,8) are loaded with S(6) and S(7), respectively. For P(7,8) we wish to load C(2,3). We use the first index of C, viz., 2, to find LØCAT(2)= 9. To this 9 we add the second index of C, viz., 3. This yields 12. Then S(12) is loaded into P(7,8). Also P(8,7) is set to P(7,8). Thus the values of the LØCDAT array are seen to describe the location of the various variables in the S array.

Finally the subset of P is un-normalized by multiplying the off-diagonal terms by their related standard deviations, after which all diagonal terms are squared to form variances.

```
$IBFTC MC13SX  XR3,M94,NODD,LIST
      SUBROUTINE BLDCOV (S,TITLE,KHOLD,NUMBER,LOCAT)                    BLDC0001
C     BUILDS PORTIONS OF COVARIANCE MATRIX                              BLDC0002
      DOUBLE PRECISION   S(1)                                           BLDC0003
      DIMENSION          LOCAT(1) ,TITLE(1)                             BLDC0004
      COMMON            /ES1COM/PNEW(30,30),STNAMN(30),TRAKER(30)       BLDC0005
     1                          ,ITRETN (30),ILOCN (30),KLOCN (54)      BLDC0006
     2                          ,NSN (12,20)                            BLDC0007
      DOUBLE PRECISION   PNEW                                           BLDC0008
C                                                                       BLDC0009
    1 DO 3 I=1,NUMBER                                                   BLDC0010
      IO = KHOLD+I                                                      BLDC0011
      KK = ILOCN(IO)+LOCAT(1)/2+1                                       BLDC0012
      PNEW(IO,IO) = S(KK)                                               BLDC0013
      KK = ILOCN(IO)                                                    BLDC0014
      STNAMN(IO) = TITLE(KK)                                            BLDC0015
      IF (I.GE.NUMBER) GO TO 3                                          BLDC0016
      LL = LOCAT(KK)                                                    BLDC0017
      II = I+1                                                          BLDC0018
      DO 2 J=II,NUMBER                                                  BLDC0019
      JO = KHOLD+J                                                      BLDC0020
      MM = ILOCN(JO)+LL                                                 BLDC0021
      PNEW(IO,JO) = S(MM)                                               BLDC0022
    2 PNEW(JO,IO) = PNEW(IO,JO)                                         BLDC0023
    3 CONTINUE                                                          BLDC0024
      DO 5 I=1,NUMBER                                                   BLDC0025
      IO = KHOLD+I                                                      BLDC0026
      DO 4 J=1,NUMBER                                                   BLDC0027
      JO = KHOLD+J                                                      BLDC0028
      IF (I.EQ.J) GO TO 4                                               BLDC0029
      PNEW(IO,JO) = PNEW(IO,JO)*PNEW(IO,IO)                             BLDC0030
      PNEW(JO,IO) = PNEW(IO,JO)                                         BLDC0031
    4 CONTINUE                                                          BLDC0032
    5 PNEW(IO,IO) = PNEW(IO,IO)**2                                      BLDC0033
  999 RETURN                                                            BLDC0034
      END
```

Subroutine: BMEAS

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

Calling Sequence: CALL BMEAS(NBEAC,BRBD,XØUT,RMEAS)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NBEAC				Beacon identifying number
I	BRDB	(3)			Semi-major axis, semi-minor axis, spin rate for body on which beacons are located.
I	XØUT	(6)			Spacecraft position, velocity with respect to body on which beacons are located.
Ø	RMEAS	(5)			1. Time 2. Range 3. Range rate 4. Angle 1 5. Angle 2

Common storages used: /INPCØM/,/WCØM/

Subroutines required: ARKTNS,BARN,CRØSS,DØT,MNA,MTRN,NUTAIT,
STAT,VNØRM,VTRN

```
$IBFTC MC13MB  NOREF,M94,NODD,XR3
CMC13MB  BMEAS   LIKE SMEAS BUT FOR BEACON                              BMES0001
      SUBROUTINE BMEAS(NBEAC,BRDB,XOUT,RMEAS)                           BMES0002
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)                        BMES0003
      DIMENSION    OMG(3),    RMEAS(5), BRDB(3),   B(91),    TM(9)      BMES0004
     1,            RT(3),     SC(5),    XOUT(6),   DUM(3),   AN(3,3)    BMES0005
     2,            TI(2),     EN(3,3),  A(3,3),    X(6),     XS(6)      BMES0006
     3,            US(3),     XI(3),    XJ(3),     XK(3),    R(4),  IB(36)BMES0007
      EQUIVALENCE  (C(9),SPDLT),    (C(608),EMP24),  (C(476),B)        BMES0008
     1,            (CW(1140),TSEC),(CW(1160),TI), (CW(1163),AN)         BMES0009
     2,            (CW(1172),TSECO),(CW(1173),GHAR)                     BMES0010
     3,            (IW(53),IEMP24),(IW(312),IB),  (IW(347),NB)          BMES0011
      EQUIVALENCE  (R(1),RAT),   (R(2),RDOT),   (R(3),ANG1),   (R(4),ANG2) BMES0012
     1,            (TM(1),XI),   (TM(4),XJ),    (TM(7),XK),    (SC(1),US)  BMES0013
C     SUBROUTINES  REQUIRED                                             BMES0014
C      ARKTNS                                                           BMES0015
C     BARN                                                              BMES0016
C      CROSS                                                            BMES0017
C      DOT,VNORM                                                        BMES0018
C     MNA                                                               BMES0019
C      MTRN,VTRN                                                        BMES0020
C     NUTAIT                                                            BMES0021
C      STAT                                                             BMES0022
      DATA (OMG(I),I=1,2)/2*0./                                         BMES0023
      RMEAS(1)=TSEC                                                     BMES0024
      OMG(3)=BRDB(3)                                                    BMES0025
      GHAB=GHAR+BRDB(3)*(TSEC-TSECO)                                    BMES0026
      IF(NB-2) 1,2,3                                                    BMES0027
    1 CALL VTRN(AN,XOUT,X)                                              BMES0028
      CALL VTRN(AN,XOUT(4),X(4))                                        BMES0029
      GO TO 5                                                           BMES0030
2     GHAB=0.                                                           BMES0031
      TIME=TI(1)+TI(2)                                                  BMES0032
      CALL NUTAIT(TIME,WM,CR,DA,EN,EPSIL)                               BMES0033
      CALL MNA(    TIME,WM,CR,DA,EPSIL,RR,G,GP,WW,EN)                   BMES0034
      CALL MTRN(EN,AN,A)                                                BMES0035
      CALL VTRN(AN,XOUT,X)                                              BMES0036
      CALL VTRN(AN,XOUT(4),X(4))                                        BMES0037
      GO TO 5                                                           BMES0038
3     DO 4 I=1,6                                                        BMES0039
4     X(I)=XOUT(I)                                                      BMES0040
5     CONTINUE                                                          BMES0041
      KBEAC=8*NBEAC-6                                                   BMES0042
      CALL STAT(B(KBEAC),GHAB,TM,RT,SC,BRDB)                            BMES0043
      CALL CROSS(OMG,RT,DUM)                                            BMES0044
      DO 6 I=1,3                                                        BMES0045
      J=I+3                                                             BMES0046
      XS(I)=RT(I)-X(I)                                                  BMES0047
6     XS(J)=DUM(I)-X(J)                                                 BMES0048
      RAT=VNORM(XS,US)                                                  BMES0049
      RDOT=DOT(US,XS(4))                                                BMES0050
      DEN=VNORM(X,US)                                                   BMES0051
      DO 7 I=1,3                                                        BMES0052
7     XK(I)=-US(I)                                                      BMES0053
      CALL CROSS(XK,X(4),XJ)                                            BMES0054
      DEN=VNORM(XJ,XJ)                                                  BMES0055
      CALL CROSS(XJ,XK,XI)                                              BMES0056
      ANG1=ARKTNS(360,DOT(XS,XI),DOT(XS,XJ))                            BMES0057
      CALL CROSS(XS,XK,DUM)                                             BMES0058
      DEN=VNORM(DUM,DUM)                                                BMES0059
      ANG2=ATAN(DOT(XS,XK)/DEN)                                         BMES0060
      IF(IEMP24.NE.0) R(1)=R(1)+EMP24*R(1)/SPDLT                        BMES0061
      DO 9 I=1,4                                                        BMES0062
      J=I+30                                                            BMES0063
      IF(IB(J).EQ.0) GO TO 9                                            BMES0064
      IF(IB(J).EQ.2) GO TO 8                                            BMES0065
      L=86+I                                                            BMES0066
      R(I)=R(I)+B (L)                                                   BMES0067
8     K=82+I                                                            BMES0068
      M=I+1                                                             BMES0069
      RMEAS(M)=R(I)+B (K)*BARN(-1)                                      BMES0070
9     CONTINUE                                                          BMES0071
      IF(IB(35).NE.0) RMEAS(1)=RMEAS(1)+B (91)                          BMES0072
      RETURN                                                            BMES0073
      END                                                               BMES0074
```

Subroutine: CARDIN

Purpose: To read change cards and following data cards.

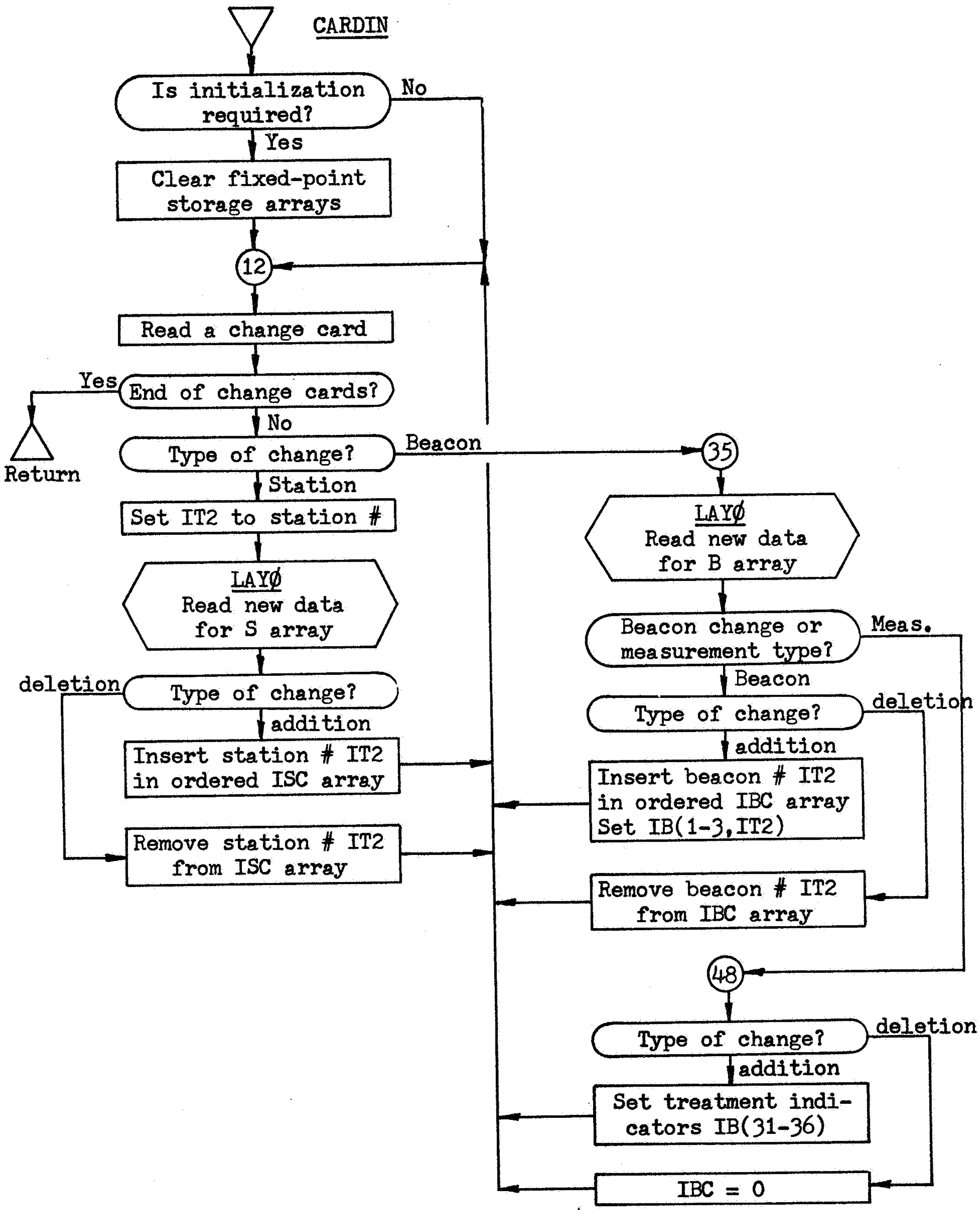
Calling Sequence: CALL CARDIN(INITS)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	INITS				Initialization key; initializes if INITS = -1.

Common storages used: /INPCØM/,/WCØM/

Subroutines required: LAYØ


CARDIN
Is initialization required?
No
Yes
Clear fixed-point storage arrays
12
Read a change card
Yes
End of change cards?
Return
No
Type of change?
Beacon
Station
Set IT2 to station #
LAYØ
Read new data for S array
deletion
Type of change?
addition
Insert station # IT2 in ordered ISC array
Remove station # IT2 from ISC array
35
LAYØ
Read new data for B array
Beacon change or measurement type?
Meas.
Beacon
Type of change?
deletion
addition
Insert beacon # IT2 in ordered IBC array Set IB(1-3,IT2)
Remove beacon # IT2 from IBC array
48
Type of change?
deletion
addition
Set treatment indicators IB(31-36)
IBC = 0

```
$IBFTC MC13AR  NOREF,M94,NODD,XR3
CMC13AR  CARDIN   SAME AS PUTIN BUT CALLS LAYO                          CRDN0001
      SUBROUTINE CARDIN(INITS)                                          CRDN0002
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)                        CRDN0003
      DIMENSION    EMP(24),    S(23,12),     B(91),        OB(18)       CRDN0004
     1,            IEMP(24),   ISC(12),      IBC(10),      IOBR(22)     CRDN0005
     2,            ITEMP(25),  IS(11,12),    IB(36)                     CRDN0006
      EQUIVALENCE  (IW(8),ISC),    (IW(20),IBC),     (IW(30),IEMP)      CRDN0007
     1,             (IW(54),IOBR),  (IW(180),IS),     (IW(312),IB)      CRDN0008
     2,             (C(200),S),     (C(476),B),       (C(567),OB)       CRDN0009
     3,             (C(585),EMP)                                        CRDN0010
C              REQUIRED SUBROUTINES                                     CRDN0011
C                  LAYO                                                 CRDN0012
      IF(INITS.GE.0) GO TO 12                                           CRDN0013
      DO 1 I=8,75                                                       CRDN0014
1     IW(I)=0                                                           CRDN0015
   12 CONTINUE                                                          CRDN0016
      READ (  5,102) NCH,SNAME,ITEMP                                    CRDN0017
  102 FORMAT(I1,A6,25I2)                                                CRDN0018
      IF(NCH) 100,100,13                                                CRDN0019
   13 GO TO (20,35,55,70),NCH                                           CRDN0020
   20 CONTINUE                                                          CRDN0021
      IT2=ITEMP(2)                                                      CRDN0022
      CALL LAYO   (S(1,IT2),NCH,ITEMP     )                             CRDN0023
      S(1,IT2)=SNAME                                                    CRDN0024
      IF(ITEMP) 28,28,21                                                CRDN0025
   21 IF(ISC.EQ.IT2) GO TO 31                                           CRDN0026
      NIT=0                                                             CRDN0027
      DO 27 I=1,12                                                      CRDN0028
      IF(ISC(I)) 23,23,24                                               CRDN0029
   23 ISC(I)=IT2                                                        CRDN0030
      GO TO 31                                                          CRDN0031
   24 IF(IT2-ISC(I)) 26,31,27                                           CRDN0032
   25 IST=ISC(I)                                                        CRDN0033
      NIT=IT2                                                           CRDN0034
      ISC(I)=IT2                                                        CRDN0035
   26 IF(NIT.NE.IT2) GO TO 25                                           CRDN0036
      ITS=ISC(I+1)                                                      CRDN0037
      ISC(I+1)=IST                                                      CRDN0038
      IF(ITS.EQ.0) GO TO 31                                             CRDN0039
      IST=ITS                                                           CRDN0040
   27 CONTINUE                                                          CRDN0041
      GO TO 31                                                          CRDN0042
   28 CONTINUE                                                          CRDN0043
      DO 29 I=1,11                                                      CRDN0044
      IF(IT2-ISC(I)) 12,30,29                                           CRDN0045
   29 CONTINUE                                                          CRDN0046
      ISC(12)=0                                                         CRDN0047
      GO TO 12                                                          CRDN0048
   30 DO 301 J=I,11                                                     CRDN0049
  301 ISC(J)=ISC(J+1)                                                   CRDN0050
      GO TO 12                                                          CRDN0051
   31 CONTINUE                                                          CRDN0052
      DO 32 I=1,11                                                      CRDN0053
   32 IS(I,IT2)=ITEMP(I+2)                                              CRDN0054
      GO TO 12                                                          CRDN0055
   35 CONTINUE                                                          CRDN0056
      IT2=8*(ITEMP(2)-1)+1                                              CRDN0057
      IF(ITEMP(2).EQ.0) IT2=81                                          CRDN0058
      CALL LAYO   (B(IT2),NCH,ITEMP     )                               CRDN0059
      IF(IT2.NE.81) B(IT2)=SNAME                                        CRDN0060
      IT2=ITEMP(2)                                                      CRDN0061
C     ITEMP(2)=0 INDICATES A MEASUREMENT TYPE CHANGE CARD               CRDN0062
      IF(IT2.EQ.0) GO TO 48                                             CRDN0063
      IF(ITEMP)43,43,36                                                 CRDN0064
   36 IF(IBC.EQ.IT2) GO TO 46                                           CRDN0065
      NIT=0                                                             CRDN0066
      DO 42 I=1,9                                                       CRDN0067
      IF(IBC(I)) 38,38,39                                               CRDN0068
   38 IBC(I)=IT2                                                        CRDN0069
      GO TO 46                                                          CRDN0070
   39 IF(IT2-IBC(I)) 41,46,42                                           CRDN0071
   40 IST=IBC(I)                                                        CRDN0072
      NIT=IT2                                                           CRDN0073
      IBC(I)=IT2                                                        CRDN0074
```

```
   41 IF(NIT.NE.IT2) GO TO 40                                             CRDN0075
      ITS=IBC(I+1)                                                        CRDN0076
      IBC(I+1)=IST                                                        CRDN0077
      IF(ITS.EQ.0) GO TO 46                                               CRDN0078
      IST=ITS                                                             CRDN0079
   42 CONTINUE                                                            CRDN0080
      GO TO 46                                                            CRDN0081
   43 CONTINUE                                                            CRDN0082
      DO 44 I=1,9                                                         CRDN0083
      IF(IT2-IBC(I)) 12,45,44                                             CRDN0084
   44 CONTINUE                                                            CRDN0085
      IBC(10)=0                                                           CRDN0086
      GO TO 12                                                            CRDN0087
   45 DO 451 J=I,9                                                        CRDN0088
  451 IBC(J)=IBC(J+1)                                                     CRDN0089
      GO TO 12                                                            CRDN0090
   46 DO 47 I=1,3                                                         CRDN0091
      J=3*(IT2-1)+I                                                       CRDN0092
   47 IB(J)=ITEMP(I+3)                                                    CRDN0093
      GO TO 12                                                            CRDN0094
   48 IF(ITEMP)49,49,51                                                   CRDN0095
   49 DO 50 I=1,10                                                        CRDN0096
   50 IBC(I)=0                                                            CRDN0097
      GO TO 12                                                            CRDN0098
   51 CONTINUE                                                            CRDN0099
      IB(36)=ITEMP(3)                                                     CRDN0100
      DO 52 I=1,5                                                         CRDN0101
      J=30+I                                                              CRDN0102
   52 IB(J)=ITEMP(I+3)                                                    CRDN0103
      GO TO 12                                                            CRDN0104
   55 CONTINUE                                                            CRDN0105
      IF(ITEMP)56,56,59                                                   CRDN0106
 56   DO 57 I=1,22                                                        CRDN0107
   57 IOBR(I)=0                                                           CRDN0108
      GO TO 69                                                            CRDN0109
   59 CONTINUE                                                            CRDN0110
      DO 60 I=1,22                                                        CRDN0111
   60 IOBR(I)=ITEMP(I+1)                                                  CRDN0112
   69 CONTINUE                                                            CRDN0113
      CALL LAYO  (OB,NCH,ITEMP      )                                     CRDN0114
      GO TO 12                                                            CRDN0115
   70 CONTINUE                                                            CRDN0116
      DO 71 I=1,24                                                        CRDN0117
   71 IEMP(I)=ITEMP(I)                                                    CRDN0118
      CALL LAYO  (EMP,NCH,ITEMP     )                                     CRDN0119
      GO TO 12                                                            CRDN0120
  100 RETURN                                                              CRDN0121
       END                                                                CRDN0122
```

Subroutine: CBDAT

Purpose: Computes three C-Band measurements (azimuth, elevation and range).

Calling Sequence: CALL CBDAT

Common storages used: /DATCØM/

Subroutines required: DMVTRN,DNØRM

Usage

The equations describing the measurement model are given in Appendix C of Reference 1.

The principal communication between the data subroutines (CBDAT, CBDATP, etc.) with their using programs is through the labelled common DATCØM. The quantities used by each subroutine are described in tabular form in Table 1.

The output array ØBS is described in Table 1, below.

COMMON LOCATIONS

COMMON	LOCATION	NAME	DIMENSION	DESCRIPTION
DATCØM	C(139)	STA	d(5)	STA(1-3), receiving station position in B-frame, (km). STA (4-5), refraction constants.
	C(161)	TB2CØ	d(18)	TB2CØ (1-9), B-frame to C-frame transform at time of measurements. TB2CØ (10-18) are not used.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors at station in B-frame.
	C(269)	XV	d(3)	Spacecraft postion in C-frame at time received signal left the spacecraft, (km).
	C(299)	NFRAC		Refraction key: +1, include refraction corrections; +2, omit.
	C(5) C(37) C(69)	ØBS	d(64)	ØBS(1), azimuth (rad) ØBS(17), elevation (rad) ØBS(33), range (km)

Table 1

α_1 Azimuth, x-Angle, or Hour Angle	α_2 Elevation, y-Angle, or Declination	ρ Ranging Observable	$\dot{\rho}$ Doppler Observable	COMMENTS
1 α_1	17 α_2	33 ρ	49 $\dot{\rho}$	Measurements
$\frac{\partial \alpha_1}{\partial x_v}$	$\frac{\partial \alpha_2}{\partial x_v}$	$\frac{\partial \rho}{\partial x_v}$	$\frac{\partial \dot{\rho}}{\partial x_v}$	Vehicle Position Partials
$\frac{\partial \alpha_1}{\partial \dot{x}_v}$	$\frac{\partial \alpha_2}{\partial \dot{x}_v}$	$\frac{\partial \rho}{\partial \dot{x}_v}$	$\frac{\partial \dot{\rho}}{\partial \dot{x}_v}$	Vehicle Velocity Partials
$\frac{\partial \alpha_1}{\partial \alpha_1}$	$\frac{\partial \alpha_2}{\partial \alpha_2}$	$\frac{\partial \rho}{\partial \rho}$	$\frac{\partial \dot{\rho}}{\partial \omega_3}$	Measurement Bias Partials
$\frac{\partial \alpha_1}{\partial \tau}$	$\frac{\partial \alpha_2}{\partial \tau}$	$\frac{\partial \rho}{\partial \tau}$	$\frac{\partial \dot{\rho}}{\partial \tau}$	Observating Station Clock Bias Partials
$\frac{\partial \alpha_1}{\partial c}$	$\frac{\partial \alpha_2}{\partial c}$	$\frac{\partial \rho}{\partial c}$	$\frac{\partial \dot{\rho}}{\partial \rho}$	Speed of Light Partials
$\frac{\partial \alpha_1}{\partial x_R}$	$\frac{\partial \alpha_2}{\partial x_R}$	$\frac{\partial \rho}{\partial x_R}$	$\frac{\partial \dot{\rho}}{\partial x_R}$	Receiving Station Position Partials
0 0 16 0	0 0 32 0	48 $\frac{\partial \rho}{\partial x_T}$	64 $\frac{\partial \dot{\rho}}{\partial x_T}$	Transmitting Station Position Partials (When operating in three-way doppler mode)

Layout of ØBS-Vector in DATCØM. Used by all data and partial subroutines. Only those portions of the ØBS-Vector relevant to the data type being computed are loaded by the various data and partial subroutines; (e.g., CBDAT uses only ØBS(1), ØBS(17), ØBS(33)).

```
$IBFTC MC13CE  M94,NODD,XR3
CMC13CE   C-BAND MEASURMENTS
      SUBROUTINE CBDAT
      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  RT(3),      SX(3),      TT2C(9),    XS(3)
     1,                 BIAS,    CE,       OBS,                DELTE
     2,                 DNORM,   E,        F,        FTR,      OMEGA
     3,                 RAT,     RTMG,     SE,       SPDLT,    STA
     4,                 T,       TAU,      TB2CO,    TB2CT,    TT2BO
     5,                 TT2BT,   XTD,      XTE,      XTN,      XV
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION  DATAN2,DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE  (SX(1),XTN),      (SX(2),XTE),      (SX(3),XTD)
C     N,E,D VECTORS IN C FRAME
      CALL DMVTRN(TB2CO,TT2BO,TT2C,1,3)
C     STATION VECTOR IN C FRAME
      CALL DMVTRN(TB2CO,STA,RT,1,1)
C     RANGE VECTOR IN C FRAME
      DO 1 I=1,3
    1 XS(I)=XV(I)-RT(I)
C  .  RANGE VECTOR IN TOPOCENTRIC COORDINATES
      CALL DMVTRN(TT2C,XS,SX,2,1)
C     STATION AND RANGE VECTOR MAGNITUDES
      RTMG=DNORM(RT)
      RAT=DNORM(XS)
C     AZIMUTH, ELEVATION, AND RANGE
      OBS(1)=DATAN2(XTE,XTN)
      OBS(17)=DATAN(-XTD/DSQRT(XTN*XTN+XTE*XTE))
      OBS(33)=RAT
      GO TO (2,7),NFRAC
C     REFRACTION CORRECTIONS
C     ELEVATION CORRECTION
    2 E=OBS(17)
      IF(E.LT..01) GO TO 606
      SE=DSIN(E)
      CE=DCOS(E)
      IF(E.GT..17452393D0) GO TO 5
      T=(1.03585796D0-(.01072014D0-(.1279119D-7-.1227363D-7/E)/E)/E)*STA
     1(4)*CE/SE
    3 F=RTMG/RAT
    4 DELTE=T-F*((STA(4)+T*T/2.)*CE-T*SE)
      GO TO 6
    5 DELTE=STA(4)*CE/SE
    6 OBS(17)=OBS(17)+DELTE
C     RANGE CORRECTION
  606 OBS(33)=OBS(33)+(STA(4)/(STA(5)*DSIN(OBS(17))))*1.D-3
C     AZIMUTH IS NOT CORRECTED FOR REFRACTION
    7 CONTINUE
      IF(OBS(1).LT.0.D0) OBS(1)=OBS(1)+6.2831853071795864
      IF(OBS(1).GT.6.2831853071795864) OBS(1)=OBS(1)-6.2831853071795864
      RETURN
      END
```

Subroutine: CBDATP

Purpose: Computes three C-Band measurements and their partial derivatives.

Calling Sequence: CALL CBDATP(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(135)	OMEGA	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(161)	TB2CØ	d(9)	B-frame to C-frame transform at time of measurements.
	C(233)	TT2BØ	d(9)	Unit North, East, Down vectors at station in B-frame.
	C(269)	XV	d(6)	Spacecraft position and velocity in C-frame at time received signal left the spacecraft (km, km/sec).
	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(299)	NFRAC		Refraction correction option key*.
				*When option key = 1, option will be included; option will be omitted if key = 2.
	C(5)	ØBS	d(64)	Measurement and partial output vector. See Table 1 in description of CDBAT.

```
$IBFTC MC13CD  XR3,M94,NODD,LIST
CMC13CD    C-BAND MEASUREMENTS AND PARTIAL DERIVATIVES
      SUBROUTINE CBDATP(R2)
      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  DCXS(3),         DN(3),          DUM(3)
     1,                 EA(3),           EN(3),          OMG(3)
     2,                 RT(3),           SX(3),          TT2C(9)
     3,                 XS(6),           Z1(3)
      DOUBLE PRECISION  BIAS,     CE,      OBS,                DDOT
     1,                 DELTE,    DEN,     DNORM,    E,        E2
     2,                 E3,       F,       FTR,      OMEGA,    RAT
     3,                 RTMG,     R2,      SE,       SPDLT,    STA
     4,                 S1,       S2,      T,        TAU,      TB2CO
     5,                 TB2CT,    TT2BO,   TT2BT,    XTD,      XTE
     6,                 XTN,      XV
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION  DATAN2,DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE  (TT2C(1),EN),     (TT2C(4),EA),     (TT2C(7),DN)
     1,            (SX(1),XTN),      (SX(2),XTE),      (SX(3),XTD)
      CALL DMVTRN(TB2CO,STA,RT,1,1)
      CALL DMVTRN(TB2CO,TT2BO,TT2C,1,3)
      DO 1 I=1,3
    1 OMG(I)=TB2CO(I+6)*OMEGA
      CALL DCROSS(OMG,RT,DUM)
      DO 2 I=1,3
      XS(I)=XV(I)-RT(I)
    2 XS(I+3)=XV(I+3)-DUM(I)
      CALL DMVTRN(TT2C,XS,SX,2,1)
      RTMG=DNORM(RT)
      R2=DDOT(XS,XS)
      RAT=DSQRT(R2)
C     COMPUTE AZIMUTH, ELEVATION, AND RANGE
      OBS(1)=DATAN2(XTE,XTN)
      OBS(17)=DATAN(-XTD/DSQRT(XTN*XTN+XTE*XTE))
      OBS(33)=RAT
   20 E=OBS(17)
      IF(E.LT..01) GO TO 56
      SE=DSIN(E)
      CE=DCOS(E)
      IF(E.GE..17453293D0) GO TO 5
      E2=E*E
      E3=E2*E
      T=(1.03585796D0-.01072014D0/E+.1279119D-7/E2-.1227363D-7/E3)*STA(4
     1)*CE/SE
    3 F=RTMG/RAT
    4 DELTE=T-F*((STA(4)+T*T/2.)*CE-T*SE)
      GO TO 6
    5 DELTE=STA(4)*CE/SE
    6 OBS(17)=OBS(17)+DELTE
C     RANGE CORRECTION
   56 OBS(33)=OBS(33)+(STA(4)/(STA(5)*DSIN(OBS(17))))*1.D-3
C     AZIMUTH IS NOT CORRECTED FOR REFRACTION
   60 CONTINUE
      IF(OBS(1).LT.0.D0) OBS(1)=OBS(1)+6.2831853071795864
      IF(OBS(1).GT.6.2831853071795864) OBS(1)=OBS(1)-6.2831853071795864
C     COMPUTE PARTIAL DERIVATIVES
C     VEHICLE STATE PARTIALS
      CALL DCROSS(DN,XS,DCXS)
      S2=DDOT(DCXS,DCXS)
      S1=DSQRT(S2)
      CALL DCROSS(DCXS,XS,OBS(18))
      DO 7 I=1,3
      OBS(I+1)=DCXS(I)/S2
      OBS(I+17)=OBS(I+17)/(R2*S1)
       OBS(I+33)=XS(I)/RAT
      OBS(I+4)=0.D0
      OBS(I+20)=0.D0
    7 OBS(I+36)=0.D0
C     MEASUREMENT BIAS PARTIALS
```

```
      OBS(8)=1.D0
      OBS(24)=1.D0
      OBS(40)=1.D0
      GO TO (8,10),MTIM
C     OBSERVING STATION TIME BIAS PARTIALS
    8 CALL DCROSS(XS,OMG,Z1)
      DO 9 I=1,3
    9 Z1(I)=Z1(I)+XS(I+3)
      OBS(9)=DDOT(OBS(2),Z1)
      OBS(25)=DDOT(OBS(18),Z1)
      OBS(41)=DDOT(XS,XS(4))/RAT
   10 GO TO (11,12),MLT
C     SPEED OF LIGHT PARTIALS
   11 DEN=SPDLT*SPDLT
      OBS(10)=DDOT(OBS(2),XV(4))*RAT/DEN
      OBS(26)=DDOT(OBS(18),XV(4))*RAT/DEN
      OBS(42)=DDOT(XS,XV(4))/DEN-RAT/SPDLT
   12 GO TO (13,16),MSTA
C     OBSERVING STATION LOCATION ERROR PARTIALS
   13 DO 14 I=2,34,16
   14 CALL DMVTRN(TB2CO,OBS(I),OBS(I+9),2,1)
      DO 15 I=11,13
      OBS(I)=-OBS(I)
      OBS(I+16)=-OBS(I+16)
   15 OBS(I+32)=-OBS(I+32)
   16 RETURN
      END
```

Subroutine: CBTEST

Purpose: To read C-Band 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.

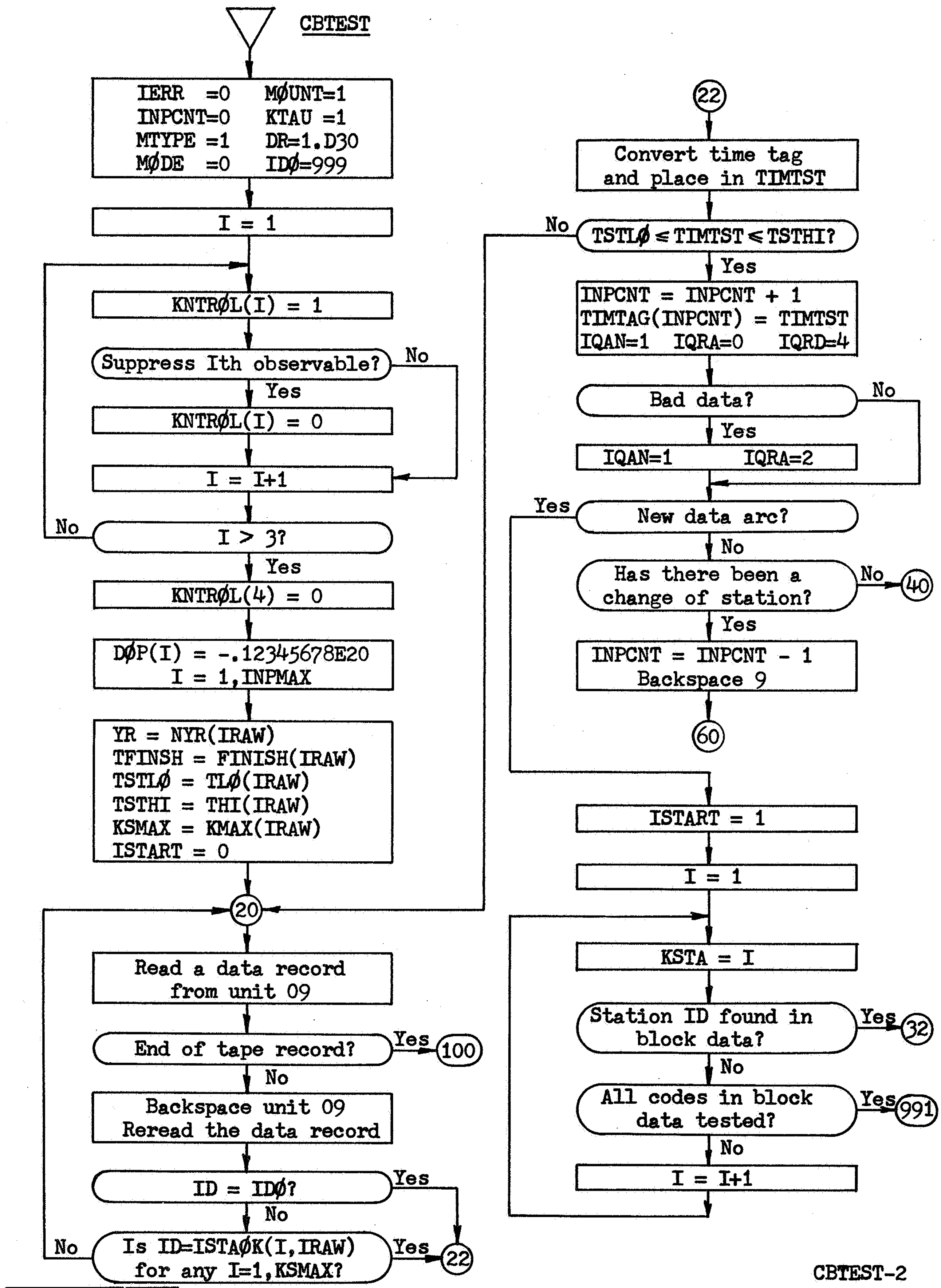
Calling Sequence: CALL CBTEST (IERR)

Input and Output:

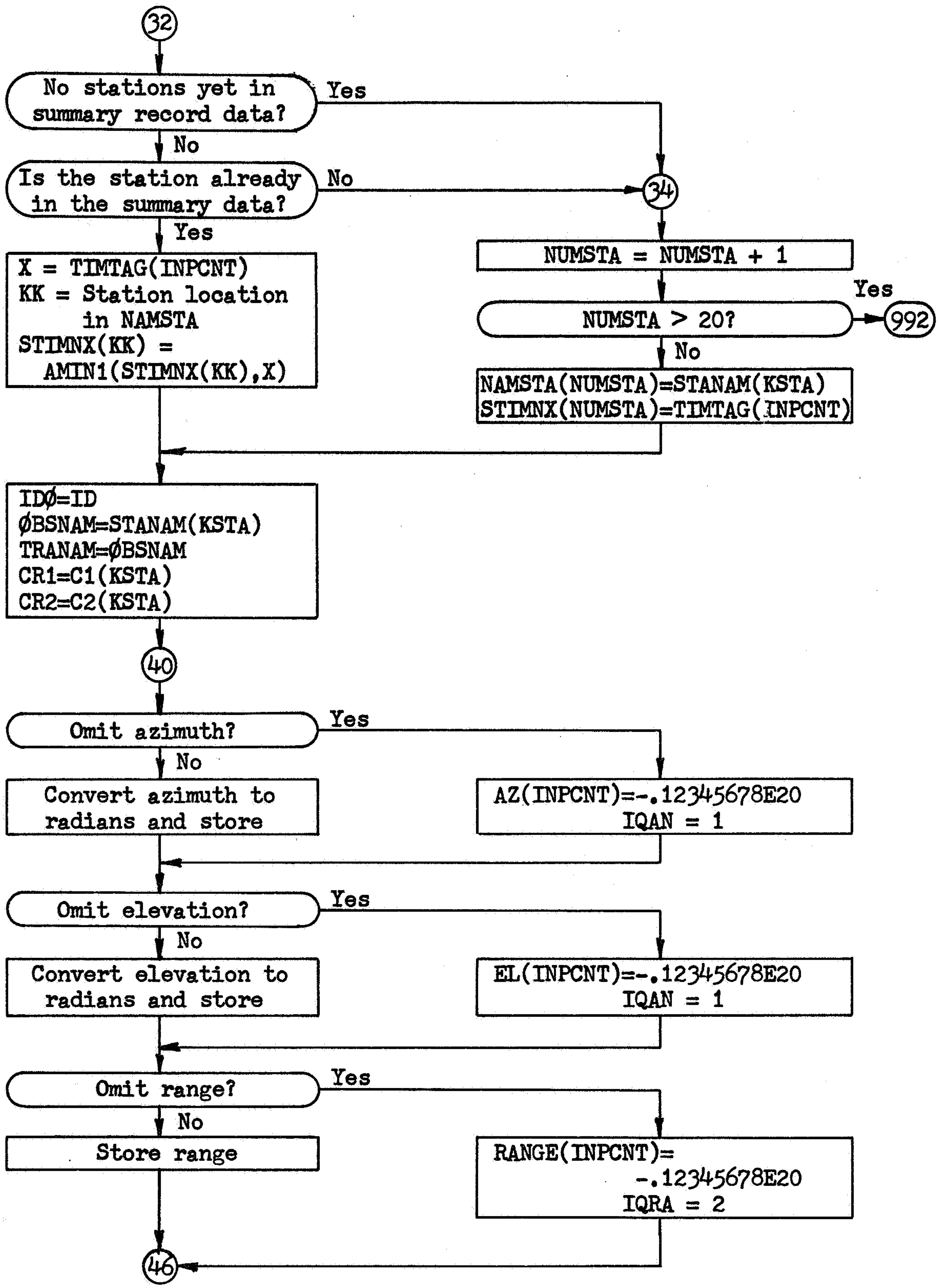
I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	IERR				Error flag. Set = 2 by CBTEST if more than 20 stations are accumulated. Set = 0 otherwise.

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

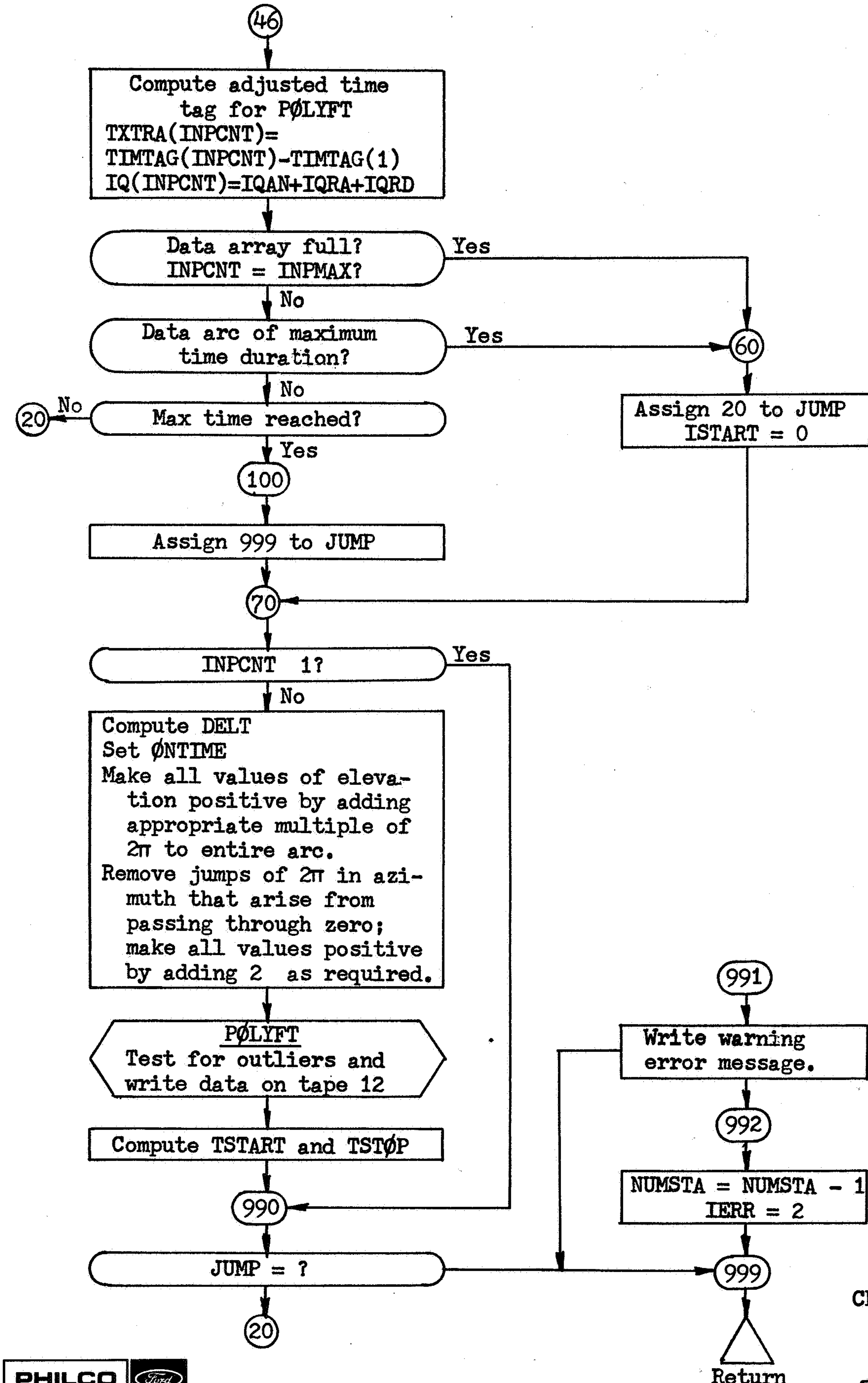
Subroutines required: DATINP, PØLYFT


CBTEST
IERR =0 MØUNT=1
INPCNT=0 KTAU =1
MTYPE =1 DR=1.D30
MØDE =0 IDØ=999
I = 1
KNTRØL(I) = 1
Suppress Ith observable?
No
Yes
KNTRØL(I) = 0
I = I+1
No
I > 3?
Yes
KNTRØL(4) = 0
DØP(I) = -.12345678E20
I = 1,INPMAX
YR = NYR(IRAW)
TFINSH = FINISH(IRAW)
TSTLØ = TLØ(IRAW)
TSTHI = THI(IRAW)
KSMAX = KMAX(IRAW)
ISTART = 0
20
Read a data record
from unit 09
End of tape record?
Yes
100
No
Backspace unit 09
Reread the data record
ID = IDØ?
Yes
No
No
Is ID=ISTAØK(I,IRAW)
for any I=1,KSMAX?
Yes
22
22
Convert time tag
and place in TIMTST
No
TSTLØ ≤ TIMTST ≤ TSTHI?
Yes
INPCNT = INPCNT + 1
TIMTAG(INPCNT) = TIMTST
IQAN=1 IQRA=0 IQRD=4
Bad data?
No
Yes
IQAN=1 IQRA=2
Yes
New data arc?
No
Has there been a
change of station?
No
40
Yes
INPCNT = INPCNT - 1
Backspace 9
60
ISTART = 1
I = 1
KSTA = I
Station ID found in
block data?
Yes
32
No
All codes in block
data tested?
Yes
991
No
I = I+1

CBTEST-2


32
No stations yet in summary record data?
Yes
No
Is the station already in the summary data?
No
34
Yes
X = TIMTAG(INPCNT)
KK = Station location in NAMSTA
STIMNX(KK) = AMIN1(STIMNX(KK),X)
NUMSTA = NUMSTA + 1
NUMSTA > 20?
Yes
992
No
NAMSTA(NUMSTA)=STANAM(KSTA)
STIMNX(NUMSTA)=TIMTAG(INPCNT)
IDØ=ID
ØBSNAM=STANAM(KSTA)
TRANAM=ØBSNAM
CR1=C1(KSTA)
CR2=C2(KSTA)
40
Omit azimuth?
Yes
No
Convert azimuth to radians and store
AZ(INPCNT)=-.12345678E20
IQAN = 1
Omit elevation?
Yes
No
Convert elevation to radians and store
EL(INPCNT)=-.12345678E20
IQAN = 1
Omit range?
Yes
No
Store range
RANGE(INPCNT)= -.12345678E20
IQRA = 2
46

CBTEST-3


46
Compute adjusted time tag for PØLYFT
TXTRA(INPCNT)=
TIMTAG(INPCNT)-TIMTAG(1)
IQ(INPCNT)=IQAN+IQRA+IQRD
Data array full?
INPCNT = INPMAX?
Yes
No
Data arc of maximum time duration?
Yes
60
No
Max time reached?
No
20
Yes
100
Assign 20 to JUMP
ISTART = 0
Assign 999 to JUMP
70
INPCNT 1?
Yes
No
Compute DELT
Set ØNTIME
Make all values of elevation positive by adding appropriate multiple of 2π to entire arc.
Remove jumps of 2π in azimuth that arise from passing through zero; make all values positive by adding 2 as required.
PØLYFT
Test for outliers and write data on tape 12
Compute TSTART and TSTØP
990
JUMP = ?
20
991
Write warning error message.
992
NUMSTA = NUMSTA - 1
IERR = 2
999
Return

CBTEST-4

```
$IBFTC MC134Y  M94,NODD,XR3
CMC134Y    C-BAND RAW DATA PROCESSOR
      SUBROUTINE CBTEST(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),MODE),      (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),       AZ(300),       EL(300)
     1,           RANGE(300),    DOP(300)
      EQUIVALENCE               (CDAT(1),TIMTAG),     (CDAT(601),TXTRA)
     1,  (CDAT(901),IQ),      (CDAT(1201),AZ),      (CDAT(1501),EL)
     2,  (CDAT(1801),RANGE),  (CDAT(2101),DOP)
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
      DOUBLE PRECISION  TFINSH,   TMAX,     TIMTST,   TSTLO,    TSTHI
      DIMENSION         DI(3)
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)
  605 FORMAT(I2,2X,O1,3X,F3.0,1X,2F2.0,1X,F6.3,2X,F7.3,2X,F7.3,2X,F12.6)
  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
```

```
      INPCNT=0                                                          CBTS0075
      MTYPE=1                                                           CBTS0076
      MOUNT=1                                                           CBTS0077
      MODE=0                                                            CBTS0078
      KTAU=1                                                            CBTS0079
      IDO=999                                                           CBTS0080
C     SET RANGE AMBIGUITY LARGE                                         CBTS0081
C     IF RANGE REALLY DOES RECYCLE, YOU WILL CHANGE THIS                CBTS0082
      DR=1.D30                                                          CBTS0083
      DO 1 I=1,3                                                        CBTS0084
      KNTROL(I)=1                                                       CBTS0085
      IF(IFOMIT(I,IRAW).NE.0) KNTROL(I)=0                               CBTS0086
    1 CONTINUE                                                          CBTS0087
      KNTROL(4)=0                                                       CBTS0088
      DO 2 I=1,INPMAX                                                   CBTS0089
    2 DOP(I)=TEST                                                       CBTS0090
      YR=NYR(IRAW)                                                      CBTS0091
      TFINSH=FINISH(IRAW)                                               CBTS0092
      TSTLO=TLO(IRAW)                                                   CBTS0093
      TSTHI=THI(IRAW)                                                   CBTS0094
      KSMAX=KMAX(IRAW)                                                  CBTS0095
      ISTART=0                                                          CBTS0096
C     TEST FOR END OF TAPE INDICATOR                                    CBTS0097
   20 READ(9,602) ARCIND                                                CBTS0098
      IF(ARCIND.EQ.EOTIND) GO TO 100                                    CBTS0099
      BACKSPACE 9                                                       CBTS0100
C     READ A LINE OF DATA                                               CBTS0101
      READ(9,605) ID,IDC,DAY,HR,XMIN,SEC,AZX,ELX,RAX                    CBTS0102
C     FIND IF STATION NUMBER IS UNCHANGED                               CBTS0103
      IF(ID.EQ.IDO) GO TO 22                                            CBTS0124
C     TEST IF STATION ID VALID                                          CBTS0125
      DO 21 I=1,KSMAX                                                   CBTS0126
      IF(ID.EQ.ISTAOK(I,IRAW)) GO TO 22                                 CBTS0127
   21 CONTINUE                                                          CBTS0128
      GO TO 20                                                          CBTS0129
   22 CONTINUE                                                          CBTS0130
C     DECODE DATA KEYS AND CONVERT TIME TAG                             CBTS0131
      DI(1)=YR*100.+1.                                                  CBTS0132
      DI(2)=(DAY*100.+HR)*100.+XMIN                                     CBTS0133
      DI(3)=SEC                                                         CBTS0134
      CALL DATINP(DI,TIMTST)                                            CBTS0135
C     TEST TIME TAG FOR VALID RANGE OF TIMES                            CBTS0136
      IF(TIMTST.GE.TSTLO.AND.TIMTST.LE.TSTHI) GO TO 220                 CBTS0137
      GO TO 20                                                          CBTS0138
  220 INPCNT=INPCNT+1                                                   CBTS0139
      IQ(INPCNT)=0                                                      CBTS0140
      TIMTAG(INPCNT)=TIMTST                                             CBTS0141
      IQAN=0                                                            CBTS0142
      IQRA=0                                                            CBTS0143
      IQRD=4                                                            CBTS0144
      IF(IDC.NE.0) GO TO 23                                             CBTS0145
      IQAN=1                                                            CBTS0146
      IQRA=2                                                            CBTS0147
   23 IF(ISTART.EQ.0) GO TO 30                                          CBTS0148
C     IF STATION CHANGE, A NEW DATA ARC IS FORCED                       CBTS0149
      IF(ID.EQ.IDO) GO TO 40                                            CBTS0150
      INPCNT=INPCNT-1                                                   CBTS0151
      BACKSPACE 9                                                       CBTS0152
      GO TO 60                                                          CBTS0153
C     NEW DATA ARC                                                      CBTS0154
   30 ISTART=1                                                          CBTS0155
C     FIND STATION ID IN BLOCK DATA                                     CBTS0156
      DO 31 I=1,50                                                      CBTS0157
      KSTA=I                                                            CBTS0158
      IF(ID.EQ.KODSTA(I)) GO TO 32                                      CBTS0159
   31 CONTINUE                                                          CBTS0160
      GO TO 991                                                         CBTS0161
   32 IF(NUMSTA.EQ.0) GO TO 34                                          CBTS0162
C     IS STATION NAME ALREADY IN SUMMARY DATA                           CBTS0163
      DO 33 I=1,NUMSTA                                                  CBTS0164
      KK=I                                                              CBTS0165
      IF(NAMSTA(I).EQ.STANAM(KSTA)) GO TO 35                            CBTS0166
   33 CONTINUE                                                          CBTS0167
C     NEW STATION NAME                                                  CBTS0168
   34 NUMSTA=NUMSTA+1                                                   CBTS0169
```

```
      IF(NUMSTA.GT.20) GO TO 992                                        CBTS0170
      NAMSTA(NUMSTA)=STANAM(KSTA)                                       CBTS0171
      STIMNX(NUMSTA)=TIMTAG(INPCNT)                                     CBTS0172
      GO TO 36                                                          CBTS0173
C     OLD STATION NAME                                                  CBTS0174
C     UPDATE STATION FIRST ON TIME                                      CBTS0175
   35 X=TIMTAG(INPCNT)                                                  CBTS0176
      STIMNX(KK)=AMIN1(STIMNX(KK),X)                                    CBTS0177
   36 IDO=ID                                                            CBTS0178
      OBSNAM=STANAM(KSTA)                                               CBTS0179
      TRANAM=OBSNAM                                                     CBTS0180
      CR1=C1(KSTA)                                                      CBTS0181
      CR2=C2(KSTA)                                                      CBTS0182
   40 CONTINUE                                                          CBTS0183
C     LOAD ROW OF DATA ARRAY                                            CBTS0184
      IF(KNTROL(1).NE.0) GO TO 41                                       CBTS0185
      AZ(INPCNT)=TEST                                                   CBTS0186
      IQAN=1                                                            CBTS0187
      GO TO 42                                                          CBTS0188
   41 AZ(INPCNT)=AZX*.0174532926                                        CBTS0189
   42 IF(KNTROL(2).NE.0) GO TO 43                                       CBTS0190
      EL(INPCNT)=TEST                                                   CBTS0191
      IQAN=1                                                            CBTS0192
      GO TO 44                                                          CBTS0193
   43 EL(INPCNT)=ELX*.0174532926                                        CBTS0194
   44 IF(KNTROL(3).NE.0.AND.RAX.NE.0.) GO TO 45                         CBTS0195
      RANGE(INPCNT)=TEST                                                CBTS0196
      IQRA=2                                                            CBTS0197
      GO TO 46                                                          CBTS0198
   45 RANGE(INPCNT)=RAX                                                 CBTS0199
   46 TXTRA(INPCNT)=TIMTAG(INPCNT)-TIMTAG(1)                            CBTS0200
      IQ(INPCNT)=IQAN+IQRA+IQRD                                         CBTS0201
      IF(INPCNT.EQ.INPMAX) GO TO 60                                     CBTS0202
      IF(TXTRA(INPCNT).GE.TMAX) GO TO 60                                CBTS0203
      IF(TIMTAG(INPCNT).GE.TFINSH) GO TO 100                            CBTS0204
      GO TO 20                                                          CBTS0205
   60 ASSIGN 20 TO JUMP                                                 CBTS0206
      ISTART=0                                                          CBTS0207
      GO TO 70                                                          CBTS0208
  100 ASSIGN 999 TO JUMP                                                CBTS0209
   70 IF(INPCNT.LE.1) GO TO 990                                         CBTS0210
C     FROM HERE TO EFN 705 IS ALL TO FIND THE MOST COMMON TIME INTERVAL CBTS0211
C     BETWEEN MEASUREMENTS.  THIS WILL BE PUT INTO DELT                 CBTS0212
      MANY=1                                                            CBTS0213
      ITEM(1)=1                                                         CBTS0214
      HOLD(1)=TIMTAG(2)-TIMTAG(1)                                       CBTS0215
      IF(INPCNT.LE.2) GO TO 704                                         CBTS0216
      DO 703 I=3,INPCNT                                                 CBTS0217
      HERE=TIMTAG(I)-TIMTAG(I-1)                                        CBTS0218
      DO 701 J=1,MANY                                                   CBTS0219
      IF(HERE.EQ.HOLD(J)) GO TO 702                                     CBTS0220
  701 CONTINUE                                                          CBTS0221
      MANY=MANY+1                                                       CBTS0222
      HOLD(MANY)=HERE                                                   CBTS0223
      ITEM(MANY)=1                                                      CBTS0224
      GO TO 703                                                         CBTS0225
  702 ITEM(J)=ITEM(J)+1                                                 CBTS0226
  703 CONTINUE                                                          CBTS0227
  704 MOST=1                                                            CBTS0228
      ISRCH=1                                                           CBTS0229
      DO 705 I=1,MANY                                                   CBTS0230
      IF(MOST.GE.ITEM(I)) GO TO 705                                     CBTS0231
      MOST=ITEM(I)                                                      CBTS0232
      ISRCH=I                                                           CBTS0233
  705 CONTINUE                                                          CBTS0234
      DELT=HOLD(ISRCH)                                                  CBTS0235
      ONTIME=TIMTAG(1)                                                  CBTS0236
C     MAKE ALL ELEVATIONS POSITIVE IF NECESSARY                         CBTS0237
C     BUT RETAIN CONTINUITY                                             CBTS0238
      DO 710 I=1,INPCNT                                                 CBTS0239
      IF(EL(I).EQ.TEST) GO TO 710                                       CBTS0240
      IF(EL(I).GE.0.) GO TO 710                                         CBTS0241
      DO 71 L=1,INPCNT                                                  CBTS0242
      IF(EL(L).EQ.TEST) GO TO 71                                        CBTS0243
      EL(L)=EL(L)+6.28318531                                            CBTS0244
```

```
   71 CONTINUE                                                          CBTS0245
  710 CONTINUE                                                          CBTS0246
C     SMOOTH OUT THE AZIMUTH DISCONTINUITIES AT ZERO=TWO PI             CBTS0247
      DO 72 I=1,INPCNT                                                  CBTS0248
      KK=I+1                                                            CBTS0249
      IF(AZ(I).NE.TEST) GO TO 73                                        CBTS0250
   72 CONTINUE                                                          CBTS0251
      GO TO 79                                                          CBTS0252
   73 IF(KK.GT.INPCNT) GO TO 79                                         CBTS0253
      KHOLD=KK-1                                                        CBTS0254
      DO 78 I=KK,INPCNT                                                 CBTS0255
      IF(AZ(I).EQ.TEST) GO TO 78                                        CBTS0256
      DIFF=AZ(I)-AZ(KHOLD)                                              CBTS0257
      IF(ABS(DIFF).LT.3.14159266) GO TO 77                              CBTS0258
      IF(DIFF.LT.0.) GO TO 74                                           CBTS0259
      II=KK                                                             CBTS0260
      LL=KHOLD                                                          CBTS0261
      GO TO 75                                                          CBTS0262
   74 II=I                                                              CBTS0263
      LL=INPCNT                                                         CBTS0264
   75 DO 76 M=II,LL                                                     CBTS0265
      IF(AZ(M).EQ.TEST) GO TO 76                                        CBTS0266
      AZ(M)=AZ(M)+6.28318531                                            CBTS0267
   76 CONTINUE                                                          CBTS0268
   77 KHOLD=I                                                           CBTS0269
   78 CONTINUE                                                          CBTS0270
   79 CONTINUE                                                          CBTS0271
C     DO POLYNOMIAL FITS, TEST FOR OUTLIERS                             CBTS0272
C     PUT DATA ON TAPE 12                                               CBTS0273
      CALL POLYFT(INPCNT)                                               CBTS0274
      TSTART=DMIN1(TSTART,TIMTAG(1))                                    CBTS0275
      TSTOP=DMAX1(TSTOP,TIMTAG(INPCNT))                                 CBTS0276
  990 INPCNT=0                                                          CBTS0277
      GO TO JUMP,(20,999)                                               CBTS0278
  991 WRITE(6,901)ID                                                    CBTS0279
      GO TO 999                                                         CBTS0280
  992 WRITE(6,902)                                                      CBTS0281
      NUMSTA=NUMSTA-1                                                   CBTS0282
      IERR=2                                                            CBTS0283
  999 RETURN                                                            CBTS0284
      END                                                               CBTS0285
```

Subroutine: CØVØUT

Purpose: To output the normalized covariance matrix in the base coordinate system and, optionally, either Darboux or local tangent plane coordinates.

Calling Sequence: CALL CØVØUT (NNP,X,LD)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NNP				Dimension of upper-left portion of P to be output.
I	X	d(6)		km km/sec	Vehicle position and velocity in base co-ordinate system (body centered).
I	LD				LD=1, P will (also) be output in Darboux, or; LD=2, P will be output in local tanget plane.

Common Storages used: //248 cells,/ES1CØM/

Subroutines Required: DTRANP, DTRDB, DMVTRN

Organization of ILØC and KLØC

These index arrays are used throughout the Differential Correction Program to identify the elements of the state vector. The allowed state variables are divided logically into type groups; each group has its identifying code number as follows:

Code	Description	Maximum Number of Elements
1	Spacecraft position and velocity	6
2	Planetary mu's and ephemeris parameters	14
3	Solar pressure, drag, and venting	4
4	Ephemeris time to universal time conversion	2
5	Earth gravitational harmonics	24
6	Moon gravitational harmonics	24
7	Extra body gravitational harmonics	24
100+k	Tracker location and measurement bias for station k. (k is an internal number assigned by the D.C.P.)	10

The values of KLØC are upper loaded and are considered in pairs. The first value of a pair is one of the type group codes described above; the second value of a pair is the number of variables from that type group that are included in the state. The specific variables are identified by values in the ILØC array. The following example shows how the ILØC and KLØC arrays are built up, and should make their use clear:

Suppose the state is to consist, first of all, of all six spacecraft position and velocity elements. Then KLØC(1) = 1, identifying the spacecraft group; KLØC(2) = 6, indicating a total of six spacecraft elements. Then the first 6 elements of ILØC are the identifying numbers of the elements chosen from the spacecraft group, in this case, the numbers 1-6. Thus, having inserted zeros in the unused locations:

KLØC

$$\begin{matrix}1\\ \\ \\ \\ \\ \\ \\ 54\end{matrix}\quad\begin{pmatrix}1\\6\\0\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}$$

ILØC

$$\begin{matrix}1\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ 30\end{matrix}\quad\begin{pmatrix}1\\2\\3\\4\\5\\6\\0\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}$$

Suppose we wish also the mu's of the earth and moon to be included in the state. Then KLØC(3) = 2, identifying the planetary mu group; KLØC(4) = 2, indicating two elements from that group; ILØC(7) = 3, indicating earth mu; and ILØC(8) = 11, indicating moon mu. Thus, we have:

KLØC

$$\begin{matrix}1\\ \\ \\ \\ \\ \\ \\ \\ 54\end{matrix}\quad\begin{pmatrix}1\\6\\2\\2\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}$$

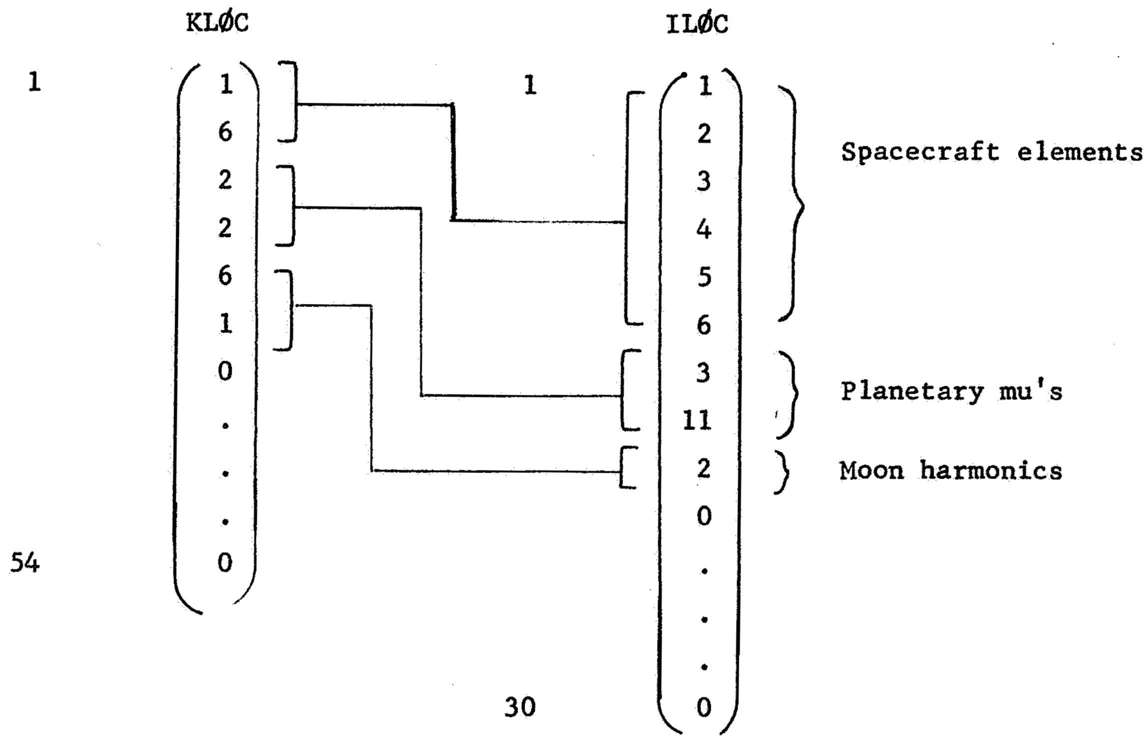
ILØC

$$\begin{pmatrix}1\\2\\3\\4\\5\\6\\3\\11\\0\\0\\\cdot\\\cdot\\\cdot\\0\end{pmatrix}\begin{matrix}\left.\begin{matrix}\\ \\ \\ \\ \\ \\ \end{matrix}\right\}\text{Spacecraft elements}\\ \left.\begin{matrix}\\ \\ \end{matrix}\right\}\text{Planetary mu elements}\\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

Suppose we wish also to include one gravitational harmonic for the moon, namely, $J_{3,0}$.

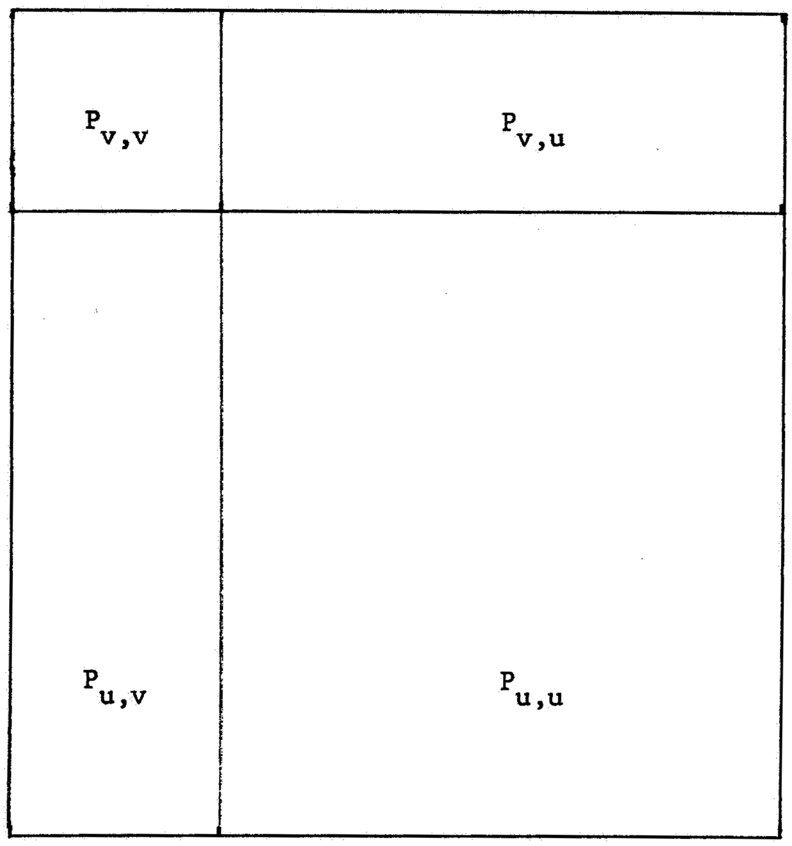
Then, KLØC(5) = 6, indicating the moon harmonic group; KLØC(6) = 1, indicating only one element from that group, and ILØC(9) = 2, identifying the second element from that group, **viz.**, $J_{3,0}$.

Then, we have, showing the correspondence,

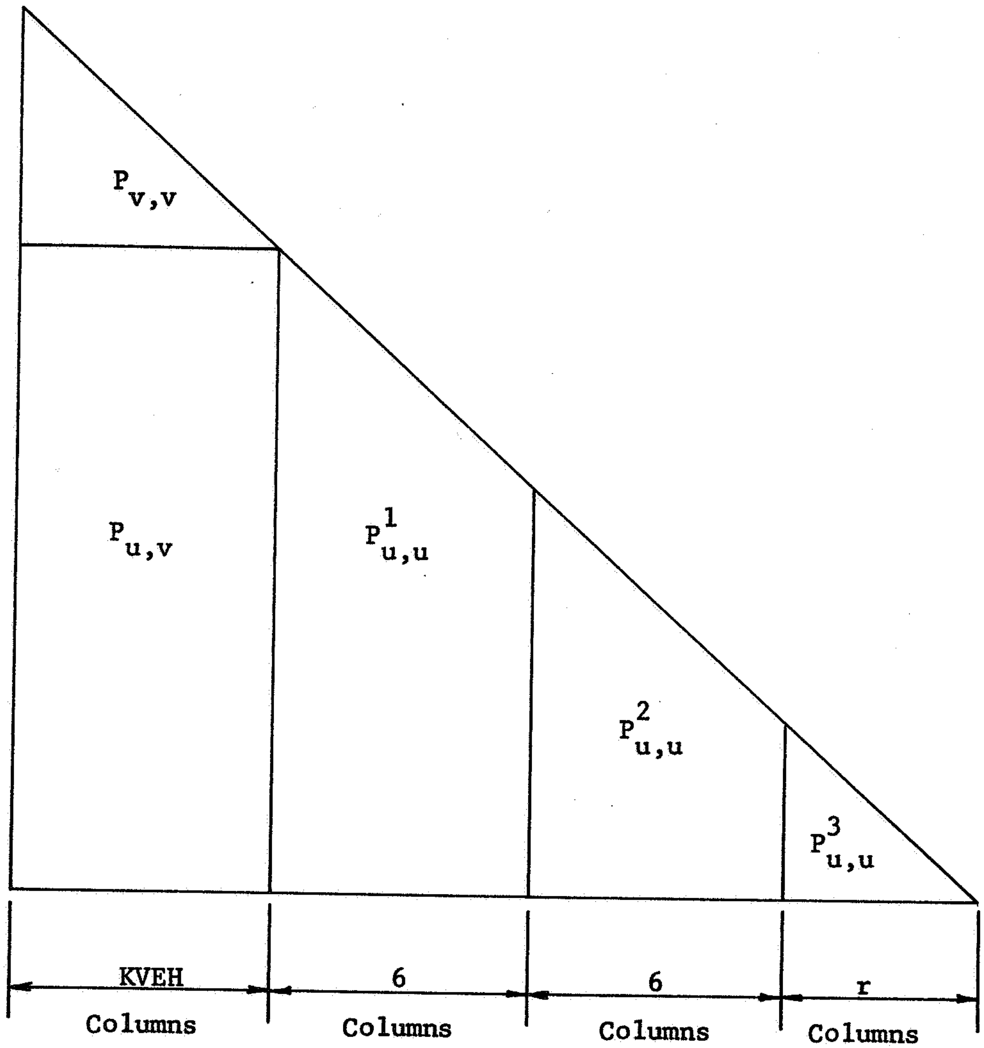


Output Format for Normalized Covariance Matrix

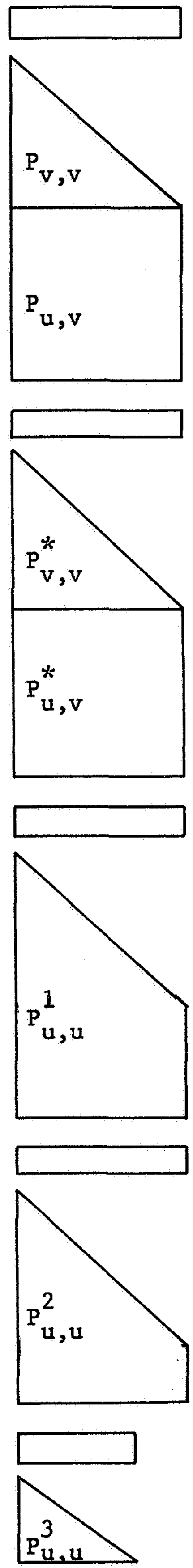
Since the covariance matrix is symmetric, it is unnecessary to display the entire matrix; CØVØUT prints out the lower-left triangular part of the covariance matrix. Let the matrix be considered as partitioned as shown below:



The matrix $P_{v,v}$ is concerned exculsively with spacecraft position and velocity, $P_{u,u}$ exclusively with non-spacecraft elements of the state vector, and $P_{u,v}$ and $P_{v,u}$ show correlations between these two groups. Further divide the lower-left triangular portion of P as follows:



If the matrix is n X n in size, and KVEH is the number of spacecraft position and velocity elements in the state, then r = (n - KVEH) modulo 6. Then the matrix is printed out in the following order:



Standard deviations from $P_{v,v}$ coordinates of estimate

Normalized $P_{v,v}$ in corrdinates of estimate

Normalized $P_{u,v}$ in coordinates of estimate

Standard deviations from $P_{v,v}$ after transformation into alternate coordinates (Darboux or tangent plane)

Normalized $P_{v,v}$ transformed into alternate coordinates

Normalized $P_{u,v}$ in alternate coordinates

Standard deviations from $P^1_{u,u}$

Normalized $P^1_{u,u}$

Standard deviations from $P^2_{u,u}$

Normalized $P^2_{u,u}$

Standard deviations from $P^3_{u,u}$

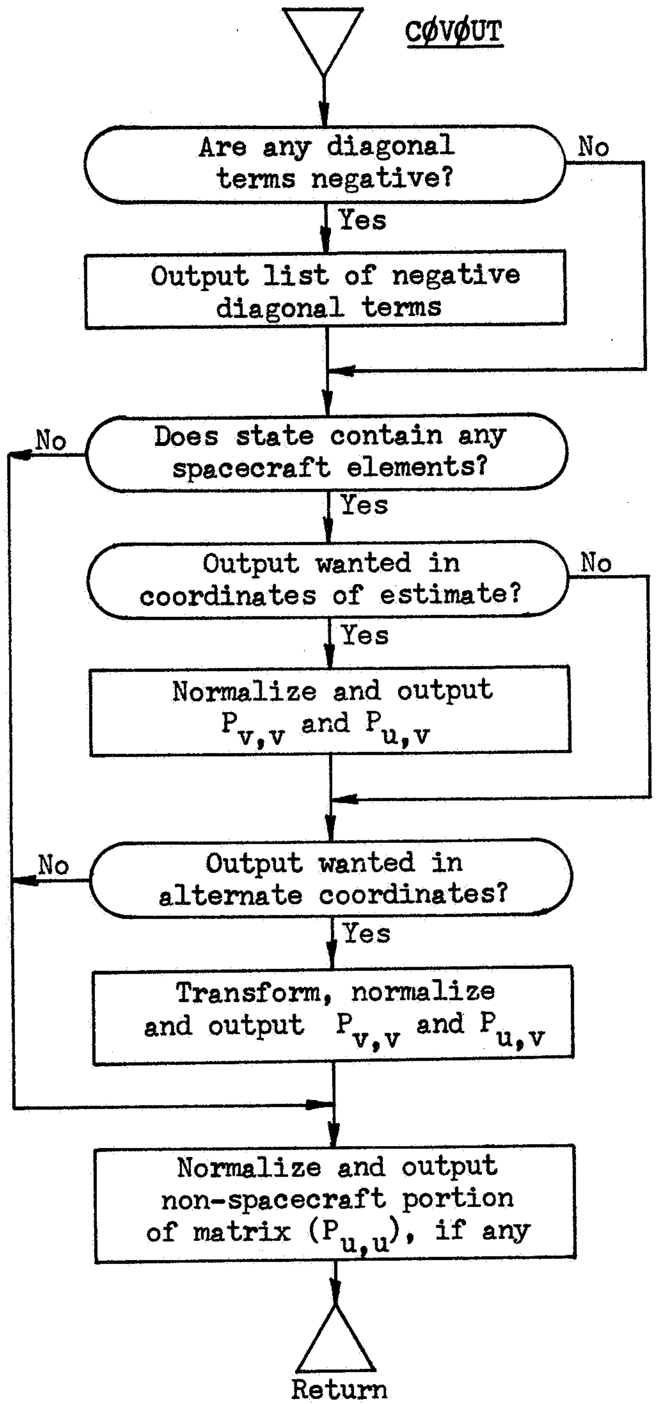
Normalized $P^3_{u,u}$

The first block of output will be KVEH Columns wide, one Column for each spacecraft position or velocity element included in the state. The second block of output, representing the first block expressed in alternate coordinates, will be either three or six Columns wide. If the spacecraft portion of the state consists of only position elements <u>or</u> only velocity elements, the transformation will cause the output to occupy three Columns. If the spacecraft portion of the state consists of at least one position element <u>and</u> at least one velocity element, the transformation will cause the output to occupy six Columns.

The remainder of the output is printed six Columns at a time. (Obviously the last block may occupy fewer than six Columns.)

Simplified CØVØUT

The DCP is restricted in the selection of state elements in that either all position and velocity components must be included or all must be omitted. A simplified CØVØUT (deck MC13CV) which uses this restriction to simplify and contract the subroutine is provided. It is completely interchangeable with the complete version (MC13CØ).


CØVØUT
Are any diagonal terms negative?
No
Yes
Output list of negative diagonal terms
Does state contain any spacecraft elements?
No
Yes
Output wanted in coordinates of estimate?
No
Yes
Normalize and output $P_{v,v}$ and $P_{u,v}$
Output wanted in alternate coordinates?
No
Yes
Transform, normalize and output $P_{v,v}$ and $P_{u,v}$
Normalize and output non-spacecraft portion of matrix ($P_{u,u}$), if any
Return

```
$IBFTC MC13CO  XR3,M94,NODD,LIST
      SUBROUTINE COVOUT (NNP,X,LD)
C     OUTPUTS NORMALIZED COVARIANCE MATRIX
      DOUBLE PRECISION   X(6) ,DSQRT
      COMMON             /ES1COM/PNEW(30,30),STNAMN(30),TRAKER(30)
     1                           ,ITRETN (30),ILOCN (30),KLOCN (54)
     2                           ,NSN (12,20)
      DOUBLE PRECISION   PNEW
C
      COMMON             D,A,PD,PP,T
      DOUBLE PRECISION   D(34) ,A(3,3) ,PD(6,6) ,PP(6,6) ,T(3,3)
C
      DIMENSION BN(9)
      DATA      BN/2H N,2H V,2H W,2HND,2HVD,2HWD
     1             ,6H      ,6HODARBO,6HOTANGT/
C
  600 FORMAT(1H0,51X,17HCOVARIANCE MATRIX)
  601 FORMAT(1H0,37X,44HTHE FOLLOWING STATES HAVE NEGATIVE DIAGONALS//47
     1X,5HSTATE,10X,8HVARIANCE)
  602 FORMAT(41X,A6,1X,A3,D22.8)
  603 FORMAT(1H0,7X,6(12X,A6))
  604 FORMAT(A6,6(15X,A3))
  605 FORMAT(8X,3HSTD,1X,6D18.8)
  606 FORMAT(1X,A6,1X,A3,4X,F12.9,5(6X,F12.9))
  607 FORMAT(1H )
C
C     OUTPUT WILL APPEAR IN COORDINATES OF ESTIMATE
C     IF LD.NE.0 OUTPUT WILL (ALSO) APPEAR IN ALTERNATE COORDINATES
C        LD=1, OUTPUT IN DARBOUX
C        LD=2, OUTPUT IN LOCAL TANGENT PLANE
C
      WRITE(6,600)
      NP=NNP
      KVEH=0
      KDEL=0
      KP=1
      DO 2 I=1,NP
      IF(PNEW(I,I).GE.0.) GO TO 2
      IF(KP.EQ.0) GO TO 1
      WRITE(6,601)
      KP=0
    1 WRITE(6,602) TRAKER(I),STNAMN(I),PNEW(I,I)
    2 D(I)=DSQRT(DABS(PNEW(I,I)))
      IF(KLOCN(1).NE.1) GO TO 20
C     OUTPUT PORTION OF P MATRIX ASSOCIATED WITH SPACECRAFT
C     OUTPUT IN COORDINATES OF ESTIMATE
      KVEH=KLOCN(2)
      DO 3 I=1,KVEH
    3 PD(I,I)=1.
      DO 4 J=2,KVEH
      DO 4 I=J,KVEH
    4 PD(I,J-1)=PNEW(I,J-1)/D(I)/D(J-1)
      WRITE(6,607)
      WRITE(6,604) BN(7),(STNAMN(I),I=1,KVEH)
      WRITE(6,605) (D(I),I=1,KVEH)
      WRITE(6,607)
      DO 5 I=1,KVEH
    5 WRITE(6,606) TRAKER(1),STNAMN(I),(PD(I,J),J=1,I)
      IF(KVEH.EQ.NP) GO TO 8
      KP=KVEH+1
      DO 7 I=KP,NP
      DO 6 J=1,KVEH
    6 PP(J,1)=PNEW(I,J)/D(I)/D(J)
    7 WRITE(6,606) TRAKER(I),STNAMN(I),(PP(J,1),J=1,KVEH)
    8 IF(LD.EQ.0) GO TO 20
C     OUTPUT IN DARBOUX OR LOCAL TANGENT PLANE
      KKJ=0
      KP=0
      KV=0
      KVEH=3
      K=KLOCN(2)
      DO 9 I=1,K
      IF(ILOCN(I).LE.3) KP=1
    9 IF(ILOCN(I).GE.4) KV=1
      IF(KP+KV.EQ.2) KVEH=6
```

```
      IF(KP.EQ.0.AND.KV.EQ.1) KKJ=3
      KDEL=KVEH-K
      NP=NP+KDEL
      DO 10 I=1,6
      DO 10 J=1,6
   10 PP(I,J)=0.
      DO 11 I=1,K
      KROW=ILOCN(I)
      DO 11 J=1,K
      KCOL=ILOCN(J)
   11 PP(KROW,KCOL)=PNEW(I,J)
      CALL DTRDB(X,X(4),T,LD)
      DO 12 I=1,3
      DO 12 J=1,3
   12 A(I,J)=T(J,I)
      CALL DTRANP(PP,6,A,A,PD,0)
      DO 13 I=1,KVEH
      D(I)=DSQRT(DABS(PD(I,I)))
   13 PD(I,I)=1.
      DO 14 I=2,KVEH
      DO 14 J=I,KVEH
   14 PD(J,I-1)=PD(J,I-1)/D(J)/D(I-1)
      M=KKJ+1
      M1=KKJ+KVEH
      WRITE(6,604) BN(LD+7),(BN(I),I=M,M1)
      WRITE(6,605) (D(I),I=1,KVEH)
      WRITE(6,607)
      DO 15 I=1,KVEH
      M=KKJ+I
   15 WRITE(6,606) TRAKER(1),BN(M),(PD(I,J),J=1,I)
      IF(KVEH.EQ.NP) GO TO 99
      KP=KVEH+1
      DO 19 I=KP,NP
      KROW=I-KDEL
      D(I)=DSQRT(DABS(PNEW(KROW,KROW)))
      DO 16 J=1,KVEH
   16 PP(J,2)=0.
      DO 17 J=1,K
      KCOL=ILOCN(J)
   17 PP(KCOL,2)=PNEW(KROW,J)
      CALL DMVTRN(A,PP(1,2),PP,1,1)
      IF(KVEH.EQ.6) CALL DMVTRN(A,PP(4,2),PP(4,1),1,1)
      DO 18 J=1,KVEH
   18 PP(J,1)=PP(J,1)/D(I)/D(J)
   19 WRITE(6,606) TRAKER(KROW),STNAMN(KROW),(PP(J,1),J=1,KVEH)
   20 CONTINUE
C     OUTPUT PORTION NOT ASSOCIATED WITH SPACECRAFT
      K=KVEH
   21 IF(NP.LE.K) GO TO 99
      KV=NP-K
      JJ=6
      IF(KV.LT.6) JJ=KV
      DO 23 I=1,JJ
      L=K+I
      L1=L-KDEL
      DO 22 J=1,I
      M=K+J
      M1=M-KDEL
   22 PD(I,J)=PNEW(L1,M1)/D(L)/D(M)
   23 PD(I,I)=1.
      KV=L-JJ+1
      KKJ=KV-KDEL
      LLJ=L-KDEL
      WRITE(6,603) (TRAKER(I),I=KKJ,LLJ)
      WRITE(6,604) BN(7),(STNAMN(I),I=KKJ,LLJ)
      WRITE(6,605) (D(I),I=KV,L)
      WRITE(6,607)
      KV=KV-1
      DO 24 I=1,JJ
      KP=KV+I
      KKJ=KP-KDEL
   24 WRITE(6,606) TRAKER(KKJ),STNAMN(KKJ),(PD(I,J),J=1,I)
      IF(NP.EQ.KP) GO TO 99
      KP=KP+1
      DO 26 I=KP,NP
```

```
      I1=I-KDEL                                                         COVT0150
      DO 25 J=1,JJ                                                      COVT0151
      L=KV+J                                                            COVT0152
      L1=L-KDEL                                                         COVT0153
   25 PP(J,1)=PNEW(L1,I1)/D(L)/D(I)                                     COVT0154
   26 WRITE(6,606) TRAKER(I1),STNAMN(I1),(PP(J,1),J=1,JJ)               COVT0155
      K=K+JJ                                                            COVT0156
      GO TO 21                                                          COVT0157
   99 RETURN                                                            COVT0158
      END
```

```
$IBFTC MC13CV  XR3,M94,NODD,LIST
      SUBROUTINE COVOUT (NNP,X,LD)                                      COVT0001
C     OUTPUTS NORMALIZED COVARIANCE MATRIX -- SIMPLIFIED VERSION        COVT0002
C     ASSUMES EITHER A SIX-ELEMENT OR ZERO-ELEMENT VEHICLE STATE        COVT0003
      DOUBLE PRECISION   X(6) ,DSQRT                                    COVT0004
      COMMON            /ES1COM/PNEW(30,30),STNAMN(30),TRAKER(30)       COVT0005
     1                         ,ITRETN (30),ILOCN (30),KLOCN (54)       COVT0006
     2                         ,NSN (12,20)                             COVT0007
      DOUBLE PRECISION   PNEW                                           COVT0008
C                                                                       COVT0009
      COMMON             SAVE(12),B,DD,T,A,D,PD                         COVT0010
      DOUBLE PRECISION   A(3,3) ,B(6) ,D(30) ,DD(6) ,PD(6,6) ,T(3,3)    COVT0011
C                                                                       COVT0012
      DIMENSION BN(9)                                                   COVT0013
      DATA       BN/2H N,2H V,2H W,2HND,2HVD,2HWD                       COVT0014
     1              ,6H        ,6HODARBO,6HOTANGT/                      COVT0015
C                                                                       COVT0016
  600 FORMAT(1H0,51X,17HCOVARIANCE MATRIX)                              COVT0017
  601 FORMAT(1H0,37X,44HTHE FOLLOWING STATES HAVE NEGATIVE DIAGONALS//47COVT0018
     1X,5HSTATE,10X,8HVARIANCE)                                         COVT0019
  602 FORMAT(41X,A6,1X,A3,D22.8)                                        COVT0020
  603 FORMAT(1H0,7X,6(12X,A6))                                          COVT0021
  604 FORMAT(A6,6(15X,A3))                                              COVT0022
  605 FORMAT(8X,3HSTD,1X,6D18.8)                                        COVT0023
  606 FORMAT(1X,A6,1X,A3,4X,F12.9,5(6X,F12.9))                          COVT0024
  607 FORMAT(1H )                                                       COVT0025
C                                                                       COVT0026
C     OUTPUT WILL APPEAR UNCONDITIONALLY IN COORDINATES OF ESTIMATE     COVT0027
C     IF LD.NE.0 OUTPUT WILL FIRST APPEAR IN ALTERNATE COORDINATES      COVT0028
C                IF SPACECRAFT STATE ELEMENTS ARE PRESENT               COVT0029
C        LD=1, OUTPUT IN DARBOUX                                        COVT0030
C        LD=2, OUTPUT IN LOCAL TANGENT PLANE                            COVT0031
C                                                                       COVT0032
      WRITE(6,600)                                                      COVT0033
      NP=NNP                                                            COVT0034
      KP=1                                                              COVT0035
      DO 2 I=1,NP                                                       COVT0036
      IF(PNEW(I,I).GT.0.) GO TO 2                                       COVT0037
      IF(KP.EQ.0) GO TO 1                                               COVT0038
      WRITE(6,601)                                                      COVT0039
      KP=0                                                              COVT0040
    1 WRITE(6,602) TRAKER(I),STNAMN(I),PNEW(I,I)                        COVT0041
    2 D(I)=DSQRT(DABS(PNEW(I,I)))                                       COVT0042
      IF(KLOCN(1).NE.1) GO TO 20                                        COVT0043
      IF(LD.EQ.0) GO TO 20                                              COVT0044
C     OUTPUT IN ALTERNATE COORDINATES                                   COVT0045
      CALL DTRDB(X,X(4),T,LD)                                           COVT0046
      DO 12 I=1,3                                                       COVT0047
      DO 12 J=1,3                                                       COVT0048
   12 A(I,J)=T(J,I)                                                     COVT0049
      CALL DTRANP(PNEW,30,A,A,PD,0)                                     COVT0050
      DO 13 I=1,6                                                       COVT0051
      DD(I)=DSQRT(DABS(PD(I,I)))                                        COVT0052
   13 PD(I,I)=1.                                                        COVT0053
      DO 14 I=2,6                                                       COVT0054
      DO 14 J=I,6                                                       COVT0055
   14 PD(J,I-1)=PD(J,I-1)/D(J)/D(I-1)                                   COVT0056
      WRITE(6,604) BN(LD+7),(BN(I),I=1,6)                               COVT0057
      WRITE(6,605) (DD(I),I=1,6)                                        COVT0058
      WRITE(6,607)                                                      COVT0059
      DO 15 I=1,6                                                       COVT0060
   15 WRITE(6,606) TRAKER(I),BN(I),(PD(I,J),J=1,I)                      COVT0061
      IF(NP.EQ.6) GO TO 20                                              COVT0062
      DO 19 I=7,NP                                                      COVT0063
      CALL DMVTRN(A,PNEW(1,I),1,2)                                      COVT0064
      DO 18 J=1,6                                                       COVT0065
   18 B(J)=B(J)/D(I)/D(J)                                               COVT0066
   19 WRITE(6,606) TRAKER(I),STNAMN(I),(B(J),J=1,6)                     COVT0067
   20 CONTINUE                                                          COVT0068
C     COORDINATES OF ESTIMATE                                           COVT0069
      K=0                                                               COVT0070
   21 IF(NP.LE.K) GO TO 99                                              COVT0071
      KV=NP-K                                                           COVT0072
      JJ=6                                                              COVT0073
      IF(KV.LT.6) JJ=KV                                                 COVT0074
```

```
      DO 23 I=1,JJ                                                      COVT0075
      L=K+I                                                             COVT0076
      DO 22 J=1,I                                                       COVT0077
      M=K+J                                                             COVT0078
   22 PD(I,J)=PNEW(L,M)/D(L)/D(M)                                       COVT0079
   23 PD(I,I)=1.                                                        COVT0080
      KV=L-JJ+1                                                         COVT0081
      WRITE(6,603) (TRAKER(I),I=KV,L)                                   COVT0082
      WRITE(6,604) BN(7),(STNAMN(I),I=KV,L)                             COVT0083
      WRITE(6,605) (D(I),I=KV,L)                                        COVT0084
      WRITE(6,607)                                                      COVT0085
      KV=KV-1                                                           COVT0086
      DO 24 I=1,JJ                                                      COVT0087
      KP=KV+I                                                           COVT0088
   24 WRITE(6,606) TRAKER(KP),STNAMN(KP),(PD(I,J),J=1,I)                COVT0089
      IF(KP.EQ.NP) GO TO 99                                             COVT0090
      KP=KP+1                                                           COVT0091
      DO 26 I=KP,NP                                                     COVT0092
      DO 25 J=1,JJ                                                      COVT0093
      L=KV+J                                                            COVT0094
   25 B(J)=PNEW(L,I)/D(L)/D(I)                                          COVT0095
   26 WRITE(6,606) TRAKER(I),STNAMN(I),(B(J),J=1,JJ)                    COVT0096
      K=K+JJ                                                            COVT0097
      GO TO 21                                                          COVT0098
   99 RETURN                                                            COVT0099
      END
```

Subroutine: CRITA

Purpose: To output the time at which a body starts or stops occulting the vehicle.

Calling Sequence: CALL CRITA(TIME,BNAM,CNAM,NØUT,KSW)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TIME			Seconds	Time for output.
I	BNAM				Name of occulting body (BCD).
I	CNAM				Name of body being occulted.
I	NØUT				Output tape unit.
I	KSW				If KSN = 1, occulting just starting. KSN = 2, occulting just stopped.

Common storages used: None

Subroutines required: TIMES

```
$IBFTC MC13CA  NOREF,M94,NODD,XR3
CMC13CA   CRITA  OUTPUTS OCCULTING BODY                                   CRIA000
      SUBROUTINE CRITA(TIME,BNAM,CNAM,NOUT,KSW)                          CRIA001
      DIMENSION ANAM(2),  A(6)                                           CRIA002
      DATA (ANAM(J),J=1,2)/6HSTARTS,6H STOPS/                            CRIA003
1     CALL TIMES(TIME,D,A)                                               CRIA004
      SW=ANAM(KSW)                                                       CRIA005
      WRITE (NOUT,100) A,BNAM,SW,CNAM                                    CRIA006
100   FORMAT(6A6,1X,1A6,1X,1A6,1X,22HOCCULTING VEHICLE FROM,1X,1A6)      CRIA007
      RETURN                                                             CRIA008
      END                                                                CRIA009
```

Subroutine: CRITIC

Purpose: Takes station-related or beacon-related critical events from the STIME or BTIME arrays and forms arrays of station on-off times or beacon on-off times - SSTART,SSTØP orBSTART,BSTØP.

Calling sequence: CALL CRITIC(NCX,IXC,XTIME,IXTIME,IXEE,KXMAX, TIM,XSART,XSTØP,KXN,KAMAX)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	NCX				NCX = 10, beacons = 12, stations.
I	IXC	As needed			Array of station numbers, ISC or Array of beacon numbers, IBC.
I	XTIME	As needed			Array of critical event times from SBEV2.
I	IXTIME	As needed			Array of critical event types, corresponds to XTIME.
I	IXEE	As			ISEE or IBEE array;in view indication for stations/beacons listed in IXC array.
I	KXMAX				Number of critical events in XTIME array.
I	TIM	(2)			Time span over which critical events were found.
Ø Ø Ø	XSTART XSTOP KXN	(75) (75) (75)			On times off times, and related station/beacon number.
Ø	KAMAX				Number of on-off pairs in XSTART,XSTØP.

Common storages used: /INPCØM/

Subroutines required: BARN, SORDR2

Functions of CRITIC

CRITIC receives as inputs a list, IXTIM, of critical event types such as station/beacon into view, station/beacon out of view, occulting starts, and occulting stops; a list, XTIME, of times associated with these events; a list of station/beacon numbers, IXC; a corresponding list of in-view indicators, IXEE, that describe conditions at the time TIM(1), which is the start time for the list of critical events; and a stop time, TIM(2), which is the end time for the list of critical events. From these input data, CRITIC forms three corresponding lists: XSTART, a list of station/beacon on times, XSTØP, a list of station/beacon off times, and KXN, a list of station/beacon numbers. On times for which no corresponding off time is found are assigned off times of TIM(2). One extra on time for each station/beacon is added to the list; this on time is set very large and merely fulfills a program logic requirement.

To each on time is added a small, "random" number representing the acquisition delay after the station/beacon in-view event. The standard deviation for this delay is found in C(114) of INPCØM. Subroutine BARN is the random number generator.

The Differential Correction Program requires that data arcs not exceed a maximum time span. Therefore, CRITIC examines all on-off time pairs and appropriately breaks up, by inserting additional off and on times, any station/beacon on periods longer than XMAX, which is stored in C(113) of INPCØM.

Next, CRITIC eliminates on periods shorter than XMIN (found in C(112)).

Finally, CRITIC sorts all three output lists by ascending order of on times in list XSTART.

```
$IBFTC MC13CC  NOREF,M94,NODD,XR3
CMC13CC  CRITIC  SORTS START-STOP ARRAYS FROM STIME OR BTIME ARRAY      CRIC0001
      SUBROUTINE CRITIC(NCX,IXC,XTIME,IXTIM,IXEE,KXMAX,TIM,XSTART,XSTOP,CRIC0002
     1KXN,KAMAX)                                                        CRIC0003
C  SUBROUTINES REQUIRED     SORDR2  BARN                                CRIC0004
      COMMON/INPCOM/C(700)                                              CRIC0005
      DIMENSION   XTIME(1),     IXTIM(1),      IXC(1),   IXEE(1)        CRIC0006
     1,             KXN(1),        XSTART(1),      XSTOP(1)             CRIC0007
     2,              TIM(2)                                             CRIC0008
      EQUIVALENCE   (C(112),XMIN),  (C(113),XMAX),  (C(114),RDELAY)     CRIC0009
      LOGICAL  NBEAC,STOP                                               CRIC0010
      DATA C2XR3/ 3.46410162/                                           CRIC0011
      DELA=C2XR3*RDELAY                                                 CRIC0012
      NBEAC=NCX.EQ.10                                                   CRIC0013
      TIMEND=XTIME(KXMAX)                                               CRIC0014
      STOP=TIMEND.NE.TIM(2)+99999.                                      CRIC0015
      IF(STOP) GO TO 50                                                 CRIC0016
      TIMEND=TIM(2)                                                     CRIC0017
      KXMAX=KXMAX-1                                                     CRIC0018
50    CONTINUE                                                          CRIC0019
      KA=1                                                              CRIC0020
      KM=0                                                              CRIC0021
      IF(NBEAC) GO TO 1                                                 CRIC0022
      IF(IXTIM.EQ.13.AND.XTIME.EQ.TIM) GO TO 19                         CRIC0023
1     DO 5 I=1,NCX                                                      CRIC0024
      K=IXC(I)                                                          CRIC0025
      IF(K.EQ.0) GO TO 6                                                CRIC0026
      IF(IXEE(I)) 5,2,2                                                 CRIC0027
2     KXN(KA)=I                                                         CRIC0028
      XSTART(KA)=XTIME(KM)                                              CRIC0029
      IF(KM.EQ.0) XSTART(KA)=TIM                                        CRIC0030
      IF(KM.EQ.KXMAX) GO TO 35                                          CRIC0031
      KMP1=KM+1                                                         CRIC0032
      DO 3 L=KMP1,KXMAX                                                 CRIC0033
      IF(IXTIM(L).EQ.13) GO TO 4                                        CRIC0034
      IF(IXTIM(L)+I) 3,4,3                                              CRIC0035
3     CONTINUE                                                          CRIC0036
35    CONTINUE                                                          CRIC0037
      XSTOP(KA)=TIMEND                                                  CRIC0038
      GO TO 41                                                          CRIC0039
4     XSTOP(KA)=XTIME(L)                                                CRIC0040
41    KA=KA+1                                                           CRIC0041
5     CONTINUE                                                          CRIC0042
6     CONTINUE                                                          CRIC0043
      IF(KM.EQ.KXMAX) GO TO 100                                         CRIC0044
      KM=KM+1                                                           CRIC0045
      K=IXTIM(KM)                                                       CRIC0046
      IF(K.EQ.13) GO TO 20                                              CRIC0047
      IF(K) 17,1000,8                                                   CRIC0048
8     IXEE(K)=0                                                         CRIC0049
      XSTART(KA)=XTIME(KM)                                              CRIC0050
      KXN(KA)=K                                                         CRIC0051
      IF(KM.EQ.KXMAX) GO TO 70                                          CRIC0052
      KM1=KM+1                                                          CRIC0053
      DO 10 L=KM1,KXMAX                                                 CRIC0054
      IF(IXTIM(L)  - 13) 9,12,9                                         CRIC0055
9     IF(IXTIM(L)  +  K) 10,12,10                                       CRIC0056
10    CONTINUE                                                          CRIC0057
70    XSTOP(KA)=TIMEND                                                  CRIC0058
11    KA=KA+1                                                           CRIC0059
      GO TO 6                                                           CRIC0060
12    XSTOP(KA)=XTIME(L)                                                CRIC0061
      GO TO 11                                                          CRIC0062
17    JK=-K                                                             CRIC0063
      IXEE(JK)=-1                                                       CRIC0064
      GO TO 6                                                           CRIC0065
19    KM=1                                                              CRIC0066
20    CONTINUE                                                          CRIC0067
      IF(KM.EQ.KXMAX) GO TO 100                                         CRIC0068
      KM=KM+1                                                           CRIC0069
      K=IXTIM(KM)                                                       CRIC0070
      IF(K.EQ.14) GO TO 1                                               CRIC0071
      IF(K) 21,1000,22                                                  CRIC0072
21    JK=-K                                                             CRIC0073
      IXEE(JK)=-1                                                       CRIC0074
```

```
      GO TO 20
22    IXEE(K)=0
      GO TO 20
100   KAMAX=KA-1
      KDLET=0
      DO 110 I=1,KAMAX
110   XSTART(I)=XSTART(I)+DELA*    BARN(1)
      KA=KAMAX
      DO 120 I=1,KA
      IF(XSTOP(I)-XSTART(I)-XMIN) 101,102,102
101   XSTART(I)=TIM(2)+9999.
      KDLET=KDLET+1
      GO TO 120
102   IF(XSTOP(I)-XSTART(I)-XMAX) 120,120,103
103   XSTART(KAMAX+1)=XSTART(I)+XMAX
      XSTOP(KAMAX+1) = XSTOP(I)
      XSTOP(I)=XSTART(KAMAX+1)
      KAMAX=KAMAX+1
      KXN(KAMAX)=KXN(I)
105   IF(XSTOP(KAMAX)-XSTART(KAMAX)-XMAX) 119,120,104
104   XSTART(KAMAX+1)=XSTART(KAMAX)+XMAX
      XSTOP(KAMAX+1)=XSTOP(KAMAX)
      XSTOP(KAMAX)=  XSTART(KAMAX+1)
      KAMAX=KAMAX+1
      KXN(KAMAX)=KXN(I)
      GO TO 105
119   IF(XSTOP(KAMAX)-XSTART(KAMAX)-XMIN.LT.0.) KAMAX=KAMAX-1
120   CONTINUE
      CALL SORDR2(XSTART,KXN,KAMAX,2,XSTOP)
      KAMAX=KAMAX-KDLET
      IF(STOP) GO TO 200
201   KAMAX=KAMAX+1
      XSTART(KAMAX)=XTIME(KXMAX+1)
      GO TO 200
1000  WRITE (6,700)
700   FORMAT(1H0,20H****CRITIC ERROR****)
200   RETURN
      END
```

Subroutine: CRITØ

Purpose: To output at a station (or beacon) critical event.

Calling Sequence: CALL CRITØ (TIME,SNAME,K,R,EL,NØUT,NHEAD)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TIME			Seconds	Time of critical event for output.
I	SNAME				Station name.
I	K				If K = 1, station is turned off K = 2, station is turned on
I	R	(3)			Vehicle position with respect to station.
I	EL				Vehicle elevation.
I	NØUT				Output tape number
I/Ø	NHEAD				If NHEAD = 1, a header is output and NHEAD is set = 2. NHEAD = 2, no header is output.

Common storages used: None

Subroutines required: FNØRM,ARKTNS,TIMES

```
$IBFTC MC13CR  NOREF,M94,NODD,XR3
CMC13CR     CRITO
      SUBROUTINE CRITO(TIME,SNAME,K,R,EL,NOUT,NHEAD)
      DIMENSION     R(3),   A(6), SWTCH(2)
      DATA RTD,(SWTCH(J),J=1,2)/57.2957796,4H OFF,3H ON/
C     SUBROUTINES REQUIRED ARE  FNORM,ARKTNS,TIMES
      SW=SWTCH(K)
      GO TO (1,2),NHEAD
1     CONTINUE
      NHEAD=2
      WRITE (NOUT,100)
100   FORMAT(1H0,30X,37HSTATION CRITICAL EVENT AND CONDITIONS/1H ,9X,15H
     1TIME FROM EPOCH,20X,5HEVENT,17X,5HRANGE,10X,7HAZIMUTH,6X,9HELEVATI
     2ON)
2     CONTINUE
      RM=FNORM(R)
      AZ=ARKTNS(180,R(1),R(2))
      AZ=AZ*RTD
      ELD=EL*RTD
      CALL TIMES(TIME,D,A)
      WRITE (NOUT,101) A,SNAME,SW,RM,AZ,ELD
101   FORMAT(6A6,7X,2A6,8X,E15.8,3X,F8.3,6X,F7.3)
      RETURN
      END
```

Subroutine: CRØSIM

Purpose: Solves simultaneous linear algebraic equations in double precision.

Calling Sequence: CALL CRØSIM

Common storages used: /CRØCØM/

Subroutines required: None.

1. Use of CRØSIM

Common assignments, CCRØ(86)

Location	Name	Dimension	Description
1	A	d(6,7)	Input: Augmented matrix of coefficients. Output: Solution vector will appear in the first L elements of column L+1.
85	L		Number of rows of A to be considered.
86	ISING		Output singularity indicator. ISING = 0: Solution found ISING = 1: Singular matrix, no solution found.

CRØSIM will solve up to six linear equations. More equations may be accommodated by altering the dimensions of the input/output matrix A and making the appropriate changes in the common size and assignments. The upper left L by L subset of A must be loaded with the equation coefficients; the first L elements of column L+1 must be the constant terms. The original matrix is destroyed during computation. The solution is developed in the first L elements of column L+1. All elements of A and all internal computations are in double precision.

2. Method of Solution

Reference: Hildebrand, F.B.: *Methods of Applied Mathematics*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1952, p. 503, *et seq*.

The method of Crout, described in the reference, has been modified slightly to improve accuracy. The modification is a re-ordering of the equations as necessary to use the available diagonal element of greatest magnitude as a divisor in each step of the reduction. If, at any step, the diagonal element of greatest magnitude is identically zero, the singularity flag ISING is set to 1 and the reduction is abandoned. If no singularity condition occurs, ISING is set to zero.

Crout's method for systems with real coefficients proceeds from the augmented matrix of the system

$$M = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & c_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & c_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & c_n \end{array}\right) = (a \mid c),$$

which may be considered as partioned into the coefficient matrix a and the column vector c, to an auxiliary matrix

$$M' = \left(\begin{array}{cccc|c} a'_{11} & a'_{12} & \cdots & a'_{1n} & c'_1 \\ \vdots & \vdots & & \vdots & \vdots \\ a'_{21} & a'_{22} & \cdots & a'_{2n} & c'_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a'_{n1} & a'_{n2} & \cdots & a'_{nn} & c'_n \end{array}\right) = (a' \mid c')$$

of the same dimensions, and thence to the required solution vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix}$$

The procedure for finding the elements of M' from those of M is described by the following rules:

1. The elements of M' are determined in the following order: elements of the first column , then elements of the first row to the right of the first column; elements of the second column below the first row, then elements of the second row to the right of the second column; and so on until all elements are determined.

2. The column operations may be divided into three steps as follows:
a. Each element a'_{ij} on or below the principal diagonal of M' is obtained by subtracting from the corresponding a_{ij} of M the sum of the products of elements in the i^{th} row and corresponding elements in the j^{th} column of M', all uncalculated elements being imagined to be zeros. That is

$$a'_{ij} = a_{ij} - \sum_{k=1}^{j-1} a'_{ik}\, a'_{kj} \qquad (i \geq j) .$$

For the first column, this is clearly no operation at all.

b. The magnitude of the diagonal element a'_{ii} of the i^{th} column of M' is compared with the magnitudes of the elements of the i^{th} column below the diagonal a'_{ik}, k = i+1,n. If the largest of these, say a'_{ik}, is greater in magnitude than a'_{ii}, then the entire i^{th} and k^{th} rows of M (and M') are interchanged. This step is purely a numerical device to minimize round-off errors.

c. Each element a'_{ij} below the principal diagonal is finally determined by division by the diagonal element a'_{jj} of M'. That is

$$a'_{ij} = a'_{ij} / a'_{jj} \qquad (i > j)$$

3. Row operations consist of a single step: Each element a'_{ij} to the right of the principal diagonal is obtained by subtracting from the corresponding element a_{ij} of M the sum of the products of elements in the i^{th} row and corresponding elements of the j^{th} column of M', all calculated elements being imagined to be zeros. That is

$$a'_{ij} = a_{ij} - \sum_{k=1}^{i-1} a'_{ik}\, a'_{kj} \qquad (i < j)$$

Clearly for the first row this involves no operation at all.

The procedure for obtaining the final solution vector from the matrix a' and the vector c' into which M' is partitioned is described by the following rules:

1. The elements of x are determined in the reverse order x_n, x_{n-1}, . . . , x_1 from the last element to the first.

2. The last element x_n is determined by dividing the last element c'_n of c' by the last element a'_{nn} on the principal diagonal of M'. That is

$$x_n = c'_n / a'_{nn}$$

3. Each of the remaining elements x_i of x is determined by subtracting from the corresponding element c'_i of c' the sum of the products of elements in the i^{th} row of a' by corresponding elements of the column x, followed by a division by the diagonal element a'_{ii} of M', all uncalculated elements of x being imagined to be zeros. That is

$$x_i = \frac{c_i' - \sum_{k=i+1}^{n} a_{ik}' x_k}{a_{ii}'}$$

Core storage space is saved in CRØSIM by causing the matrices M and M' to share the same storage locations.

```
$IBFTC MC134Q  M94,NODD,XR3
CMC134Q    DOUBLE PRECISION SIMULTANEOUS LINEAR EQUATION SOLVER
      SUBROUTINE CROSIM
      COMMON  /CROCOM/CCRO(86)
      DOUBLE PRECISION  A(6,7)
      EQUIVALENCE  (CCRO(1),A),      (CCRO(85),L),      (CCRO(86),ISING)
C
      DOUBLE PRECISION  SUM
C
      ISING=0
      M=L
      N=M+1
      IK=1
    1 IL=IK
      SUM=DABS(A(IK,IK))
      DO 2 I=IK,M
      IF(SUM.GE.DABS(A(I,IK))) GO TO 2
      IL=I
      SUM=DABS(A(I,IK))
    2 CONTINUE
      IF(IK.EQ.IL) GO TO 4
      DO 3 J=1,N
      SUM=-A(IK,J)
      A(IK,J)=A(IL,J)
    3 A(IL,J)=SUM
    4 IL=IK+1
      IF(A(IK,IK).EQ.0.) GO TO 11
      DO 5 I=IL,M
    5 A(I,IK)=A(I,IK)/A(IK,IK)
    6 JJ=IK-1
      IL=IK+1
      IF(JJ.EQ.0) GO TO 8
      DO 7 J=IL,N
      DO 7 I=1,JJ
    7 A(IK,J)=A(IK,J)-A(IK,I)*A(I,J)
      IF(IK.EQ.M) GO TO 10
    8 JJ=IK
      IK=IK+1
      DO 9 I=IK,M
      DO 9 J=1,JJ
    9 A(I,IK)=A(I,IK)-A(I,J)*A(J,IK)
      IF(IK-M) 1,6,1
   10 IF(A(M,M).NE.0.) GO TO 12
   11 ISING=1
      GO TO 14
   12 DO 13 I=1,M
      JJ=M-I
      IL=JJ+1
      A(IL,N)=A(IL,N)/A(IL,IL)
      IF(JJ.EQ.0) GO TO 14
      DO 13 J=1,JJ
   13 A(J,N)=A(J,N)-A(IL,N)*A(J,IL)
   14 RETURN
      END
```

Subroutine: CRØSS

Purpose: Computes the vector cross product of two vectors.

Calling Sequence: CALL CRØSS (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)			Input vectors
Ø	C	(3)			A X B

Common storages used: None

Subroutines required: None

```
$IBFTC MC13MK  XR3,M94,NODD,LIST
      SUBROUTINE CROSS (A,B,C)                                          CROS0001
      DIMENSION  A(3),B(3),C(3),IN(4)                                   CROS0002
      DATA       IN/2,3,1,2/                                            CROS0003
      DO 1 I=1,3                                                        CROS0004
      J = IN(I)                                                         CROS0005
      K = IN(I+1)                                                       CROS0006
    1 C(I) = A(J)*B(K)-A(K)*B(J)                                        CROS0007
      RETURN                                                            CROS0008
       END
```

Subroutine: DATINP

Purpose: To convert from calendar date (in date format) to double-precision seconds from 0^h, Jan 1, 1950.

Calling Sequence: CALL DATINP(DI,DPSEC)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	DI	3			DI(1) = YR MO. DI(2) = DAY HR MIN. DI(3) = SEC. } Date Format
Ø	DPSEC	d			Double-precision seconds from 0^h, Jan 1, 1950.

Common storages used: 8 cells.

Subroutines required: None

Date Format:

The calendar date is represented by a three-element, single-precision array DI.

Let:

YR be the year number from 1900;
i.e., the year 1967 is represented as
67. ; the year 2001 as 101.

MO be the number of the month.
DAY be the day of the month.
HR be the whole hour on the 24-hour clock.
MIN be the whole minutes past the hour.
SEC be the whole and fractional seconds.

Then:

$DI(1) = YR. \times 10^2 + MO.$
$DI(2) = DAY \times 10^4 + HR. \times 10^2 + MIN.$
$DI(3) = SEC.$

Example:

JAN 24, 1967, 11^h 26^m 16.482^s is represented as:

DI(1) = 6701.
DI(2) = 241126.
DI(3) = 16.482

```
$IBFTC MC13TV  XR3,M94,NODD,LIST
      SUBROUTINE DATINP(DI,DPSEC)
C
C     DI(1) = YEAR MONTH.                                     (INPUT)
C     DI(2) = DAY HR MIN.                                     (INPUT)
C     DI(3) = SECONDS.                                        (INPUT)
C     DPSEC = DOUBLE PRECISION SECONDS FROM JAN 0, 1950       (OUTPUT)
C
      DOUBLE PRECISION  DPSEC
      COMMON            Y,I(5),JJ
      DIMENSION         J(11),DI(3)
      DATA              J/31,59,90,120,151,181,212,243,273,304,334/
C
      I=DI(1)
      I(2)=DI(2)
      IF(I.LT.5000.OR.I(2).LT.0) GO TO 2
      I(3)=I(2)/10000
      I(4)=I(2)-10000*I(3)
      I(2)=I(4)/100
      I(5)=I(4)-100*I(2)
      Y=60*(I(5)+60*I(2))
      DPSEC=Y+DI(3)
      GO TO 10
    2 WRITE(6,700) DI
  700 FORMAT(22H1DATINP EXIT,  YRMO. =,F10.2,12H DAYHRMIN. =,F12.2,
     16H SEC =,F9.6)
      STOP
   10 I(3)=I(3)-1
      I(4)=I(1)/100
      I(1)=I(1)-100*I(4)-1
      IF(I.GT.11) GO TO 2
      IF(I) 2,13,12
   12 JJ=I
      I(3)=I(3)+J(JJ)
   13 I(5)=I(4)-4*(I(4)/4)
      IF(I(1)/2+I(5).LE.0) I(3)=I(3)-1
      Y=I(3)+(2+1461*(I(4)-50))/4
      DPSEC=DPSEC+86400.*Y
      RETURN
      END
```

Subroutine: DATØUP

Purpose: To convert from double-precision seconds from 0^h Jan 1, 1950 to calendar date.

Calling Sequence: CALL DATØUP(DPSEC, D, K)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	DPSEC	d			Double-precision seconds from 0^h Jan 1, 1950
Ø	D	3			D(1) = YR MO. D(2) = DAY HR MIN. D(3) = SEC. } Date Format*
I	K				K = 0 causes the calendar and Julian dates to be written on the system output tape.

*See DATINP Description.

Common storages used: 10 cells.

Subroutines required: None.

```
$IBFTC MC13TW  XR3,M94,NODD,LIST
      SUBROUTINE DATOUP(DPSEC,D,K)
C
C     DPSEC = DOUBLE PRECISION SECONDS FROM 1950 (INPUT)
C     D(1) = YR MO.       FORMAT OF DATINP       (OUTPUT)
C     D(2) = DAY HR MIN.  FORMAT OF DATINP       (OUTPUT)
C     D(3) = SEC.         FORMAT OF DATINP       (OUTPUT)
C     K= 0 GIVES CALENDAR DATE FORMAT OUTPUT
C
      DOUBLE PRECISION   DPSEC,     TDUM
      DIMENSION          C(12), D(3), IY(4)
      COMMON             Y(8), TDUM
      EQUIVALENCE       (Y(2),IY)
C
      DATA C/  3HJAN, 3HFEB, 3HMAR, 3HAPR, 3HMAY, 3HJUN
     1,        3HJUL, 3HAUG, 3HSEP, 3HOCT, 3HNOV, 3HDEC /
C
  601 FORMAT(1H ,A3,I3,1H,,I5,1H,,I3,5H HRS,,I3,5H MIN,,F9.5,4H SEC,48X,
     111HJULIAN DATE,F10.0,F8.8)
  602 FORMAT(21H0NEG DAYS IN DATOUP =,F8.0)
C
      TDUM=DPSEC
      Y(7)=AINT(TDUM/86400.D0)
      IF(TDUM.GT.0.)GO TO 10
      WRITE(6,602) Y(7)
      GO TO 999
   10 CONTINUE
      TDUM=TDUM-Y(7)*86400.
      Y(8)=TDUM/86400.D0
      IY(1) = Y(7)
      IY(2) = IY(1)/365
   11 IY(3) = IY(1)-(1461*IY(2)+1)/4
      IF (IY(3).GE.0) GO TO 12
      IY(2) = IY(2)-1
      GO TO 11
   12 IY(4) = IY(2)-2-4*(IY(2)/4)
      IY(2) = IY(2)+50
      JJ = 0
      KD = 0
   13 CONTINUE
      MD = KD
      JJ = JJ+1
      GO TO (14,16,14,15,14,15,14,14,15,14,15,14) ,JJ
   14 KD = KD+31
      GO TO 17
   15 KD = KD+30
      GO TO 17
   16 IF (IY(4).EQ.0) KD=KD+1
      KD = KD+28
   17 IF (KD.LE.IY(3)) GO TO 13
   20 IY(1) = IY(3)-MD+1
      Y(1) = C(JJ)
      D(1)=JJ+100*IY(2)
      IY(2) = IY(2)+1900
C
   30 CONTINUE
      Y(4)=AINT(TDUM/3600.D0)
      TDUM=TDUM-Y(4)*3600.
      Y(5)=AINT(TDUM/60.D0)
      TDUM=TDUM-Y(5)*60.
      IY(3)=Y(4)
      IY(4)=Y(5)
      Y(6)=TDUM
      YY=IY(4)+100*(IY(3)+100*IY(1))
      D(2)=YY
      D(3)=Y(6)
C
   40 CONTINUE
      Y(8) = Y(8)+0.5
      YY = AINT(Y(8))
      Y(8) = Y(8)-YY
      Y(7) = Y(7)+YY+2433282.
      IF (K.NE.0) GO TO 999
      WRITE (6,601) Y
  999 RETURN

      END
```

Subroutine: DATØUT

Purpose: Converts whole and fractional days since 1950 Jan 0.0 to date format. Optionally writes out calendar and Julian dates.

Calling Sequence: CALL DATØUT (TW,TF,YW,YF,K)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	TW			days	Whole days since 1950 Jan 0.0
I	TF			days	Fractional days
Ø	YW				Calendar date
Ø	YF				Time of day
I	K				If K = 0, date is written out If K ≠ 0, date is not written

Common storages used: 8 cells

Subroutines required: ERRØUT, TFRAC

Restrictions:

The permissible time range is from 1950 Jan 0.0 to but not including 2100 Jan 0.0. Times outside this range cause exit through ERRØUT.

Date Format:

The output quantities YW, YF are the floating point numbers

YW = YYMM.DD

YF = HHNN.SSSSS

where

YY = Year - 1900

MM = Month (number)

DD = Day of the month

HH = Hours since midnight

NN = Minutes

SSSSS = Seconds x 10^3

```
$IBFTC MC13DA  XR3,M94,NODD,LIST
      SUBROUTINE DATOUT (TW,TF,YW,YF,K)
      COMMON     Y(8)
      DIMENSION  C(12),F(10),IY(4)
      EQUIVALENCE(Y(2),IY)
      DATA  C(1)/36H    JAN    FEB    MAR    APR    MAY    JUN
     1     /,C(7)/36H    JUL    AUG    SEP    OCT    NOV    DEC
     2     /,F(1)/30H(6X,20HDATE 1950 OR BEFORE.)
     3     /,F(6)/30H(6X,19HDATE 2100 OR LATER.)          /
  601 FORMAT(A6,I3,1H,,I5,1H,,I3,5H HRS,,I3,5H MIN,,F7.3,4H SEC,48X,
     111HJULIAN DATE,F10.0,F8.8)
C
    1 CALL TFRAC (TW,TF,Y(7),Y(8))
      JJ = 1
      IF (Y(7).LT.0.) GO TO 3
      IF (Y(7).LT.54787.) GO TO 10
    2 JJ = 6
    3 CALL ERROUT (1,F(JJ))
C
   10 CONTINUE
      IY(1) = Y(7)
      IY(2) = IY(1)/365
   11 IY(3) = IY(1)-(1461*IY(2)+1)/4
      IF (IY(3).GE.0) GO TO 12
      IY(2) = IY(2)-1
      GO TO 11
   12 IY(4) = IY(2)-2-4*(IY(2)/4)
      IY(2) = IY(2)+50
      JJ = 0
      KD = 0
   13 CONTINUE
      MD = KD
      JJ = JJ+1
      GO TO (14,16,14,15,14,15,14,14,15,14,15,14) ,JJ
   14 KD = KD+31
      GO TO 17
   15 KD = KD+30
      GO TO 17
   16 IF (IY(4).EQ.0) KD=KD+1
      KD = KD+28
   17 IF (KD.LE.IY(3)) GO TO 13
C
   20 IY(1) = IY(3)-MD+1
      Y(1) = C(JJ)
      YW = FLOAT(IY(1)+100*(JJ+100*IY(2)))/100.
      IY(2) = IY(2)+1900
C
   30 Y(6)  = Y(8)*24.
      IY(3) = Y(6)
      Y(6)  = 60.*(Y(6)-FLOAT(IY(3)))
      IY(4) = Y(6)
      Y(6)  = 60.*(Y(6)-FLOAT(IY(4)))
      YY = IY(4)+100*IY(3)
      YF = YY+Y(6)/100.
C
   40 IF (K.NE.0) GO TO 999
      Y(8) = Y(8)+0.5
      YY = AINT(Y(8))
      Y(8) = Y(8)-YY
      Y(7) = Y(7)+YY+2433282.
      WRITE (6,601) Y
  999 RETURN
      END
```

Subroutine: DCP

Purpose: Main program for the Differential Correction Program.

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

Subroutines required: DATINP,DIFCØR,MLESTT,RSIDUL,SETCAS,SETSTA,SETTAP

Discussion:

The Differential Correction Program was designed as a flexible post-flight data analysis program. It is capable of processing given data in any desired order, of generating an a priori estimate from the data itself, of propagating the state and covariance to a desired event, and of outputting residuals in a variety of forms.

The sequence of steps taken to process a given set of data is controlled at two levels. The case level is used for the definition of the quantities to be included in the state for the ensuing steps, and the process level is used to accomplish specific steps without redefinition of the state composition.

The main program described here controls the program flow as required to accomplish the specified tasks. The subroutines and secondary control routines used are listed below with functional descriptions (see the Flow Diagram below).

SETTAP — Reads tape identification cards and checks or writes tape headers as required. Outputs tape identification cards. Sets IERR = 1 if any tape header disagrees with the corresponding identification card.

SETSTA — Locates the stations on the data tape, and loads working arrays for those stations from a larger array containing data for up to 50 stations. Computes station location vectors and transformations. Sets IERR = 2 if any station is not found in the array of available stations.

SETCAS — Reads option cards and sets internal option variables. Reads state definition cards and the associated overlay cards. Combines overlayed block data with a previously computed estimate from the estimation tape to form an a priori estimate of the state and its covariance. Sets internal indices used for interpretation of the state. Sets IERR for any invalid request in the state definition cards.

MLESTT — Computes an estimate of the state from specified data for use as an a priori estimate, using maximum likelihood estimation in measurement space. Sets IERR = 15 if the MLE does not converge.

DIFCØR — Controls the integration of vehicle equations of motion and variational equations for propagation or estimation of the vehicle state. Controls differential correction of the state. Sets IERR for any error noted.

RSIDUL — Computes and outputs residuals of specified data from a specified state estimate.

Subroutines SETTAP and SETSTA comprise the program initialization for a given run, and are called only once during the run.

The DCP reads a case header card with the following contents:

Col 1-6 — Number of estimate on the estimation tape to be used in building the a priori estimate.

Col 7-72 — Hollerith case identification for user reference.

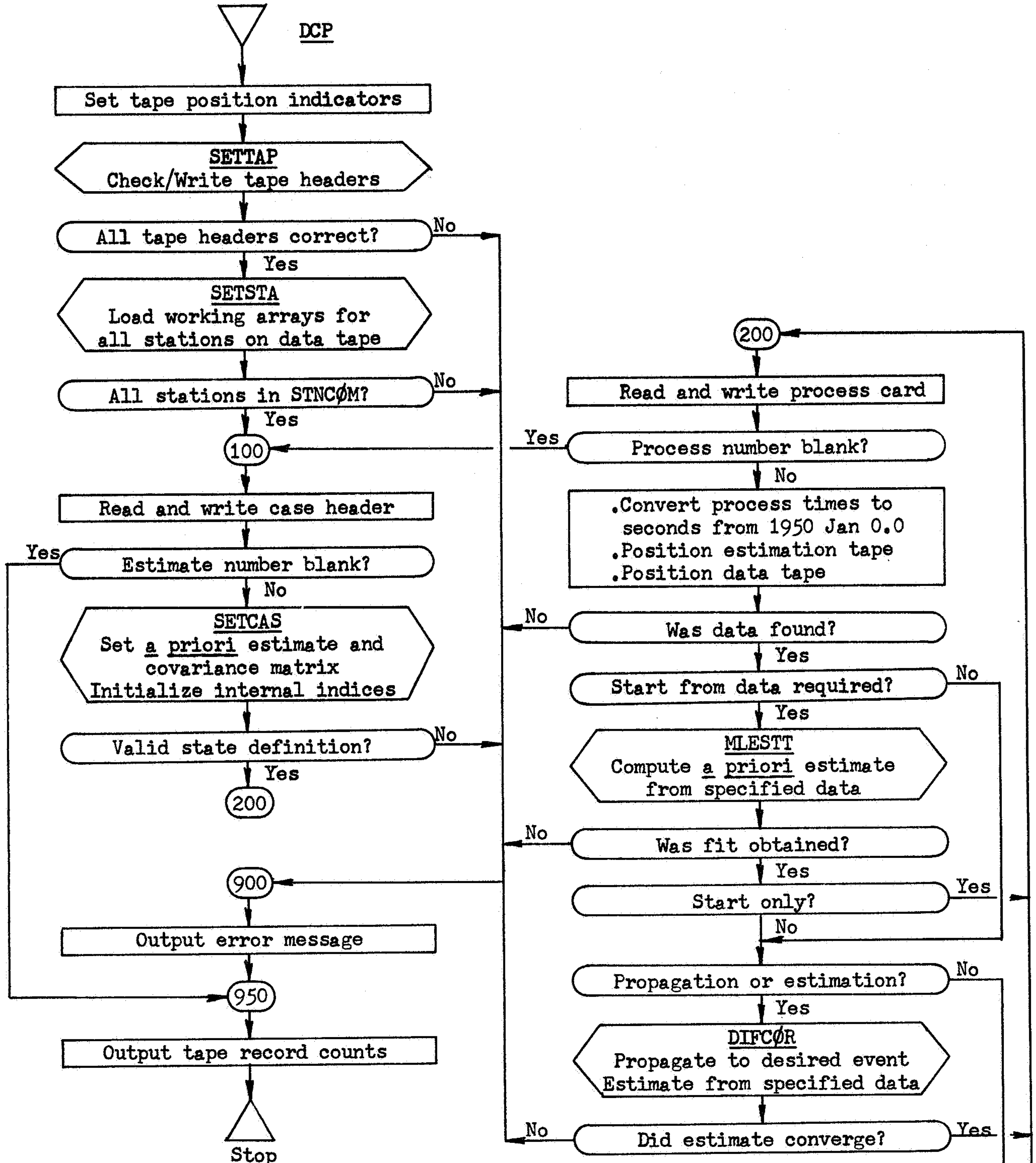
If the program encounters a blank card, or a card with a blank or 0 in 1-6, when it expects a case card, the run is terminated. The subroutine SETCAS performs case-initialization, and is called once per case.

Following case initialization, the program reads a set of process control cards. For each, it searches the data tape for the data requested on the card and reads the first block of data into BUFDAT (EDTCØM). It then calls the appropriate data processing subroutine (MLESTT, DIFCØR,RSIDUL) and proceeds with the next process card. If the program encounters a card with a blank or 0 in column 1 when it expects a process card, the case is terminated and the next case card is read.

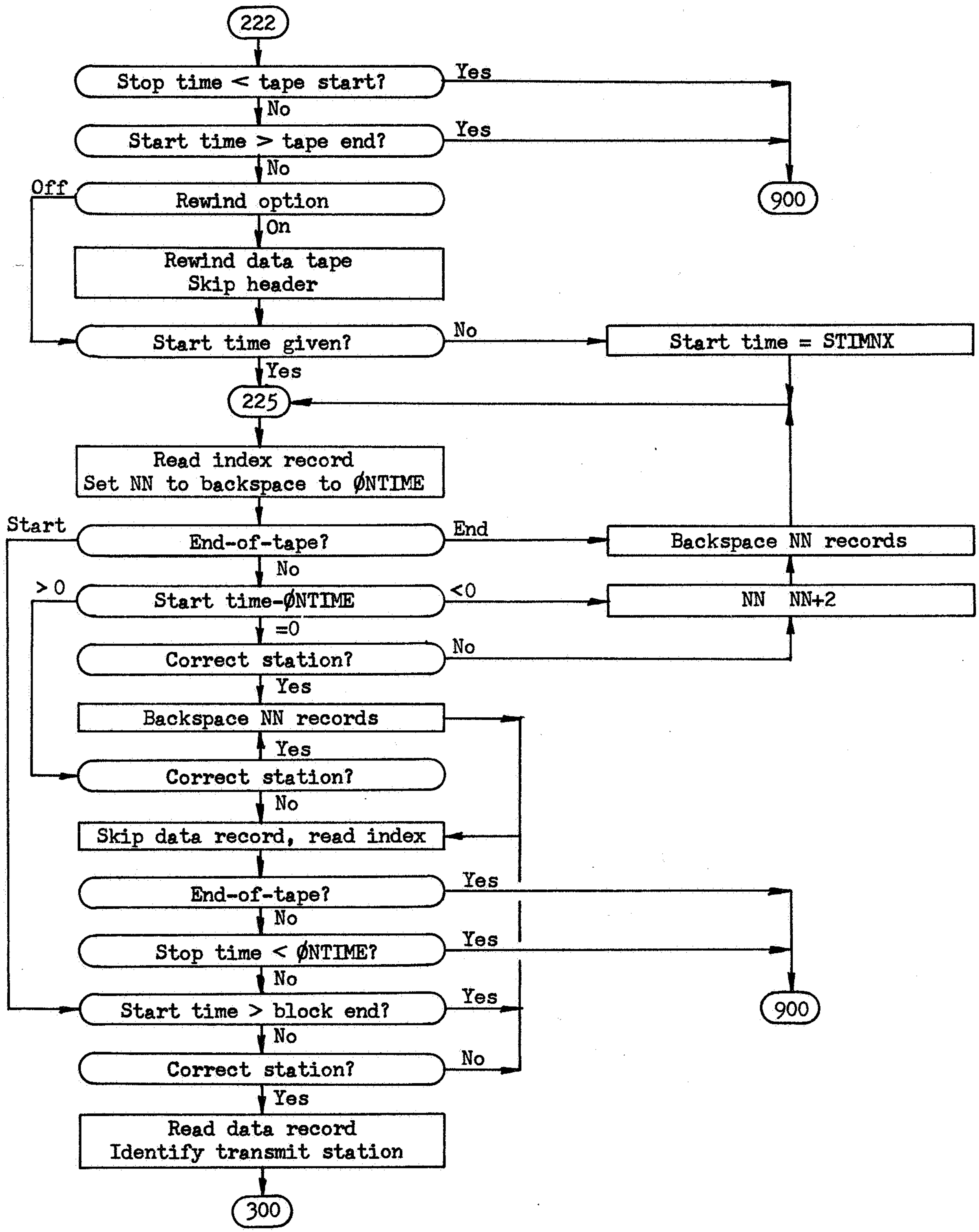
The process control card has the format

(2I1,A6,2(I3,I7,I2),I4,18I2)

Its contents are read into the array NPRØC. The components of NPRØC, their card locations, and their meanings are described in Reference 1.


DCP
Set tape position indicators
SETTAP
Check/Write tape headers
All tape headers correct?
No
Yes
SETSTA
Load working arrays for all stations on data tape
All stations in STNCØM?
No
Yes
100
Read and write case header
Estimate number blank?
Yes
No
SETCAS
Set a priori estimate and covariance matrix
Initialize internal indices
Valid state definition?
No
Yes
200
900
Output error message
950
Output tape record counts
Stop
200
Read and write process card
Process number blank?
Yes
No
.Convert process times to seconds from 1950 Jan 0.0
.Position estimation tape
.Position data tape
Was data found?
No
Yes
Start from data required?
No
Yes
MLESTT
Compute a priori estimate from specified data
Was fit obtained?
No
Yes
Start only?
Yes
No
Propagation or estimation?
No
Yes
DIFCØR
Propagate to desired event
Estimate from specified data
Did estimate converge?
No
Yes
RSIDUL
Compute residuals of given data from current estimate

DCP-5



DCP - Data Tape Search

```
$IBFTC MC13M3  XR3,M94,NODD,LIST
C      DCP - DIFFERENTIAL CORRECTION 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            /DCPCOM/CDCP(900)
      DIMENSION          IPROC (5) ,NDATE(3,2) ,STIMNX(20)
     1                  ,NPROC(22) ,PDATE(3,2)
      DOUBLE PRECISION   STIMR (2) ,TBEGIN      ,TEND
      EQUIVALENCE
     1       (CDCP(111),IERR   ) ,(IPROC( 1),IPROCS) ,(NPROC( 1),NPROCS)
     2      ,(CDCP(756),IPROC  ) ,(IPROC( 2),ITRSTA) ,(NPROC( 2),NPROVR)
     3      ,(CDCP(117),NESEND) ,(IPROC( 3),INDSTA) ,(NPROC( 3),NPRSTA)
     4      ,(CDCP(118),NEST   ) ,(IPROC( 4),IFTSTT) ,(NPROC( 4),NDATE )
     5      ,(CDCP(731),NPROC  ) ,(IPROC( 5),IFTSTP) ,(NPROC( 4),PDATE )
     6      ,(CDCP(116),NRSEND)                      ,(NPROC(10),NPREST)
     7      ,(CDCP(123),STIMNX)                      ,(NPROC(11),NREWND)
     8      ,(CDCP(781),STIMR )
     9      ,(CDCP(119),TBEGIN)
     1      ,(CDCP(121),TEND   )
C
      COMMON            /ESTCOM/CEST(804)
      DIMENSION          NAMSTA(20)
      EQUIVALENCE       (CEST( 7),NAMSTA) ,(CEST( 1),NESPOS)
     1                 ,(CEST( 6),NUMSTA)
C
      COMMON            /EDTCOM/INDDAT(40),BUFDAT(85,6)
      DOUBLE PRECISION   ONTIME ,TLAST
      EQUIVALENCE       (INDDAT( 2),NEOT   ) ,(INDDAT( 4),NTSTA )
     1                 ,(INDDAT( 5),NRCD   ) ,(INDDAT(13),ONTIME)
     2                 ,(INDDAT( 3),NRSTA  ) ,(INDDAT(17),TLAST )
C
      COMMON             HEAD(11)
C
      DIMENSION          DIAGN(132),NERR(19)
      DATA   DIAGN(  1)/36HTAPE HEADER DOES NOT AGREE.
     1    /,DIAGN(  6)/42HSTATION CANNOT BE FOUND IN BLOCK DATA.
     2    /,DIAGN( 13)/42HSTATE REQUESTED FROM TAPE WITH NEGATIVE ES
     2    /,DIAGN( 20)/18HTIMATE NUMBER.
     3    /,DIAGN( 23)/36HSTATE TYPE NAMED DOES NOT EXIST.
     4    /,DIAGN( 29)/24HSTATE TYPE DUPLICATED.
     5    /,DIAGN( 33)/42HINCLUSION OF HARMONICS FOR TWO BODIES IS N
     5    /,DIAGN( 40)/18HOT PERMITTED.
     6    /,DIAGN( 43)/42HSTATE REQUESTED FROM TAPE ESTIMATE DOES NO
     6    /,DIAGN( 50)/18HT EXIST THEREIN.
     7    /,DIAGN( 53)/42HSTATION PARAMETERS REQUESTED FROM TAPE, ST
     7    /,DIAGN( 60)/24HATION NOT IN ESTIMATE.
     8    /,DIAGN( 64)/30HTOO MANY STATE VARIABLES.
     9    /,DIAGN( 69)/42HTOO MANY EQUATION OF MOTION PARAMETERS.    /
      DATA   DIAGN( 76)/42HCENTRAL BODY REQUESTED FROM TAPE ESTIMATE
     1    /,DIAGN( 83)/30HDOES NOT AGREE WITH TAPE.
     2    /,DIAGN( 88)/36HOBSERVING STATION NOT ON DATA TAPE.
     3    /,DIAGN( 94)/42HPROCESS INTERVAL OUTSIDE DATA TAPE RANGE.
     4    /,DIAGN(101)/42HPROCESS INTERVAL CONTAINS NO DATA FOR REQU
     4    /,DIAGN(108)/18HESTED STATION.
     5    /,DIAGN(111)/30HDATA START FAILED TO CONVERGE.
     6    /,DIAGN(116)/36HTIME PRECEDES EPHEMERIS TAPE RANGE.
     7    /,DIAGN(122)/36HTIME FOLLOWS EPHEMERIS TAPE RANGE.
     8    /,DIAGN(128)/30HESTIMATION FAILED TO CONVERGE.             /
      DATA   NERR/  1,  6, 13, 23, 29, 33, 43, 53, 64, 69
     1           , 76, 88, 94,101,111,116,122,128,133    /
C
 5101 FORMAT(I6,11A6)
 5201 FORMAT(2I1,A6,2(I3,I7,I2),I4,18I2)
```

```
6101 FORMAT(1H1,45X,27H*** CASE INITIALIZATION ***/9X,5HCASE ,11A6,15H,
    1 TAPE ESTIMATE,I6,10H REQUESTED)
6201 FORMAT(1H1,48X,20H*** PROCESS CARD ***/103H0 PROC  OVR  STATION
    1      START         STOP        NEST   1   2   3   4   5   6   7   8
    2 9  10  11  12/2I5,4X,A6,3X,2(I4,I7,I2),I6,12I4)
6901 FORMAT(15H0*** ERROR STOP/1X,11A6)
6902 FORMAT(20H0TAPE STATUS SUMMARY/26H RESIDUAL   TAPE CONTAINS ,I3,13
    1H RECORD PAIRS/26H ESTIMATION TAPE CONTAINS ,I3,13H RECORD PAIRS)
C
C**** RUN INITIALIZATION
C
C     CHECK/WRITE TAPE HEADERS
    1 CONTINUE
      NESPOS = 0
      NRSPOS = 0
      CALL SETTAP
      IF (IERR.NE.0)   GO TO 900
C
C     SET UP STATION DATA
    2 CONTINUE
      CALL SETSTA
      IF (IERR.NE.0)   GO TO 900
C
C**** CASE INITIALIZATION
C
C     READ CASE HEADER
  100 CONTINUE
      READ  (5,5101)   NEST,HEAD
      IF (NEST.EQ.0)   GO TO 950
      WRITE (6,6101)   HEAD,NEST
C
C     SET UP A PRIORI STATE AND COVARIANCE
      CALL SETCAS
      IF (IERR.NE.0)   GO TO 900
C
C**** PROCESS CONTROL
C
C     READ AND INTERPRET PROCESS CARD
  200 CONTINUE
      READ  (5,5201)   NPROC
      IF (NPROCS.EQ.0) GO TO 100
      WRITE (6,6201)   NPROC
      IFDSTT = NPROCS/4
      IPROCS = NPROCS-4*IFDSTT
      IF (NPREST.GT.NESEND)   NPREST = NESEND
C
C     CONVERT PROCESS TIMES TO SECONDS
      DO 201 I=1,2
      IPROC(I+3) = 2
      IF (NDATE(2,I).LE.0)    GO TO 201
      PDATE(1,I) = 100*NDATE(1,I)+1
      PDATE(2,I) = NDATE(2,I)
      PDATE(3,I) = NDATE(3,I)
      CALL DATINP (PDATE(1,I),STIMR(I))
      IPROC(I+3) = 1
  201 CONTINUE
C
C     POSITION ESTIMATION TAPE
  210 IF (NPREST.LE.0) GO TO 220
      NN = NPREST-NESPOS-1
      IF (NN) 211,215,213
  211 NN =-2*NN
      DO 212 I=1,NN
  212 BACKSPACE 12
      GO TO 215
  213 DO 214 I=1,NN
      READ  (12) SKIP
  214 READ  (12) SKIP
  215 CONTINUE
      READ  (12) CEST
C
C     POSITION EDITED DATA TAPE
  220 CONTINUE
      IF (NPROCS.EQ.1)        GO TO 300
      IERR = 13
```

```
      DO 221 INDSTA=1,NUMSTA
  221 IF (NPRSTA.EQ.NAMSTA(INDSTA)) GO TO (222,223) ,IFTSTP
      IERR = 12
      GO TO 900
  222 IF (STIMR(2).LT.TBEGIN) GO TO 900
  223 IF (STIMR(1).GT.TEND)   GO TO 900
      IF (NREWND.EQ.0)        GO TO 224
      REWIND 10
      READ  (10) SKIP
  224 IF (NDATF(2,1).LE.0)    STIMR(1) = STIMNX(INDSTA)
  225 READ  (10) INDDAT
      NM = 1
      NN = 2*NRCD-1
      IF (NEOT) 232,226,227
  226 IF (NRSTA.EQ.NPRSTA)   NM = 2
      IF (STIMR(1).LT.ONTIME) GO TO 227
      IF (STIMR(1).EQ.ONTIME) GO TO (227,228) ,NM
      GO TO (230,228) ,NM
  227 NM = 1
      NN = NN+2
  228 DO 229 I=1,NN
  229 BACKSPACE 10
      GO TO (225,231) ,NM
  230 CONTINUE
      READ  (10) SKIP
  231 READ  (10) INDDAT
      IERR = 14
      IF (NEOT.GT.0)          GO TO 900
      IF (STIMR(2).LT.ONTIME) GO TO (900,232) ,IFTSTP
  232 IF (STIMR(1).GT.TLAST)  GO TO 230
      IF (NRSTA.NE.NPRSTA)    GO TO 230
      IERR = 0
      READ  (10) BUFDAT
      DO 233 ITRSTA=1,NUMSTA
  233 IF (NTSTA.EQ.NAMSTA(ITRSTA)) GO TO 300
C
C**** DATA START
C
  300 IF (IFDSTT.EQ.0) GO TO 350
      CALL MLESTT
      IF (IERR.NE.0)   GO TO 900
  350 IF (IPROCS.EQ.0) GO TO 200
      GO TO (400,400,500) ,IPROCS
C
C**** PROPAGATE/ESTIMATE
C
  400 CONTINUE
      CALL DIFCOR
      IF (IERR.NE.0)   GO TO 900
      GO TO 200
C
C**** OUTPUT RESIDUALS
C
  500 CONTINUE
      CALL RSIDUL (NRSPOS)
      IF (IERR.NE.0)   GO TO 900
      GO TO 200
C
C**** PROGRAM STOPS
C
C     ERROR STOP
  900 CONTINUE
      NN = NERR(IERR)
      NM = NERR(IERR+1)-1
      WRITE (6,6901)   (DIAGN(I),I=NN,NM)
C
C     NORMAL STOP
  950 CONTINUE
      WRITE (6,6902)   NRSEND,NESEND
      STOP
      END
```

Subroutine: DCRØSS

Purpose: To form the double precision cross product of two vectors.

Calling Sequence: CALL DCRØSS(A,B,C)

Input and Output:

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d(3)			Pre-vector
I	B	d(3)			Post-vector
Ø	C	d(3)			A×B

Common storages used: None

Subroutines required: None

```
$IBFTC MC13M2  XR3,M94,NODD,LIST
      SUBROUTINE DCROSS(A,B,C)                                          DCRS0000
      DOUBLE PRECISION A,B,C                                            DCRS0001
      DIMENSION  A(3),B(3),C(3),IN(4)                                   DCRS0002
      DATA       IN/2,3,1,2/                                            DCRS0003
      DO 1 I=1,3                                                        DCRS0004
      J = IN(I)                                                         DCRS0005
      K = IN(I+1)                                                       DCRS0006
    1 C(I) = A(J)*B(K)-A(K)*B(J)                                        DCRS0007
      RETURN                                                            DCRS0008
      END
```

Subroutine: DDØT

Purpose: Computes the inner product of two 3-vectors in double precision.

Calling Sequence: Z = DDØT (X, Y)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	d(3)			Pre-vector.
I	Y	d(3)			Post-vector.
Ø	DDØT	d			Inner product X·Y

Common storages used: None

Subroutines required: None

```
$IBFTC MC132D  XR3,M94,NODD,LIST
      DOUBLE PRECISION FUNCTION DDOT (X,Y)                              DDOT0001
      DOUBLE PRECISION          X(3),Y(3),SUM                           DDOT0002
    1 SUM = 0.                                                          DDOT0003
      DO 2 I=1,3                                                        DDOT0004
    2 SUM = SUM+X(I)*Y(I)                                               DDOT0005
      DDOT = SUM                                                        DDOT0006
      RETURN                                                            DDOT0007
      END
```

Subroutine: DEHA

Purpose: To compute the Greenwich hour angle of the true equinox of date and the Earth's angular velocity.

Calling Sequence: CALL DEHA (T,DA,RA,ØM)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T	d	T	sec	Time from 1950 January 0.0 UT
I	DA	d	$\delta\alpha$	rad	Nutation in right ascension
Ø	RA	d	♈	rad	Hour angle of the true equinox of date
Ø	ØM	d	ω	rad/sec	Earth's angular velocity

Common storages used: None

Subroutines required: None

Method

Universal time is defined as 12^h + Greenwich hour angle of a point on the equator whose longitude is

$$L = 18^h\ 38^m\ 45^s\!.836 + 8640184^s\!.542\ \acute{T}_u + 0^s\!.0929\ \acute{T}_u^{\,2}$$

measured from the mean equinox of date, where $\acute{T}_u$ is measured in Julian centuries from 1900 Jan 0.5 UT. Setting T_u = Julian centuries from 1950 Jan 0.0 UT,

$$L-12^h = 1.746\ 647\ 719\ 162\ 569 + 0.0172\ 027\ 914\ 509\ 729\ 83\ [d_u]$$

$$+\ 0.506\ 408\ 971\ 100224\ \ 10 \times 10^{-14}\ [d_u]^2 + \omega t \text{ radians}$$

$[d_u]$ = integer part of $[36525\ T_u]$

= whole days since 1950 Jan 0.0 UT

t = seconds from 0^h of date

ω = Earth's angular velocity

$$= \left\{ (L-12^h) \Big|_{[d_u]}^{[d_u]+1} + 2\pi \right\} \Big/ 86400 \text{ radians/second}$$

The G.H.A. of the true vernal equinox, ♈, is

$$♈ = L-12^h + \delta\alpha$$

where $\delta\alpha$ is the nutation in right ascension.

```
$IBFTC MC132Y  XR3,M94,NODD,LIST
      SUBROUTINE DEHA (T,DA,RA,OM)                                      DEHA0001
C     HOUR ANGLE OF THE TRUE EQUINOX                                    DEHA0002
      DOUBLE PRECISION T,DA,RA,OM                                       DEHA0003
     1                ,C(7),DU,TS,DMOD                                  DEHA0004
      DATA   C/1.746647719162569     ,1.72027914509729 D-02             DEHA0005
     1         ,5.064089711002241D-15,7.292115854896487D-05             DEHA0006
     2         ,1.172242988657926D-19,6.283185307179586                 DEHA0007
     3         ,8.64                 D+04/                              DEHA0008
C                                                                       DEHA0009
    1 DU = AINT(T/C(7))                                                 DEHA0010
      TS = T-C(7)*DU                                                    DEHA0011
      OM = C(4)+C(5)*DU                                                 DEHA0012
      TS = DA+OM*TS+C(1)+DU*(C(2)+DU*C(3))                              DEHA0013
      RA = DMOD(TS,C(6))                                                DEHA0014
  999 RETURN                                                            DEHA0015
      END
```

Subroutine: DELEV

Purpose: To sort the critical event array in ascending order and determine the measurement interval DEL.

Calling Sequence: CALL DELEV (EVNT,KEV,DEL,KSTØP,NSTA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	EVNT	(6)		seconds	Critical event times. All except the first must be in ascending order on entry.
IØ	KEV	(6)			Critical event keys; KEV(I) is the key for EVNT(I).
Ø	DEL			seconds	Measurement interval.
Ø	KSTØP				Key indicating type of first of ordered output critical events. KSTØP = KEV(1)
I	NSTA				Number of station having earliest next on time.

Common storages used: /INPCØM/,/WCØM/

Subroutines required: None

```
$IBFTC MC13DL   NOREF,M94,NODD,XR3
CMC13DL  DELEV   EVDEL WITH DIFFERENT LOGIC FOR FINDING DEL
      SUBROUTINE DELEV(EVNT,KEV,DEL,KSTOP,NSTA)
      COMMON/INPCOM/C(700)/WCOM/IW(550),CW(1450)
      DIMENSION  EVNT(6),  KEV(6),    S(23,12),    ISC(12)
      EQUIVALENCE  (C(200),S),  (C(556),DELBEC),  (C(573),DELRAD)
     1,           (C(580),DELOPT),   (IW(8),ISC)
      IF(EVNT(1).LE.EVNT(2)) GO TO 4
      ESAV=EVNT(1)
      KSAV=KEV(1)
      DO 1 J=1,4
      I=J+2
      IF(ESAV-EVNT(I)) 2,2,1
1     CONTINUE
      J=J+1
2     CONTINUE
      DO 3 I=1,J
      EVNT(I)=EVNT(I+1)
3     KEV(I)=KEV(I+1)
      J=J+1
      EVNT(J)=ESAV
      KEV(J)=KSAV
4     CONTINUE
      KSTOP=KEV(1)
      GO TO (20,30,40,50,100,100),KSTOP
20    N=ISC(NSTA)
      DEL=S(2,N)
      GO TO 100
30    DEL=DELBEC
      GO TO 100
40    DEL=DELOPT
      GO TO 100
50    DEL=DELRAD
100   RETURN
       END
```

Subroutine: DEQD

Purpose: Integrates second-order differential equations in double precision.

Calling Sequence: CALL DEQD(FSUB,ØSUB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
	FSUB				Name of subroutine which evaluates derivatives
	ØSUB				Name of subroutine which controls output.

Common storages used: //(40),/DQDCØM/

Subroutines required: FSUB,ØSUB

Restrictions

An "EXTERNAL FSUB,ØSUB" statement must be included in the calling program. The subroutines FSUB,ØSUB must have zero length call lists. For use with ØVERLAY, FSUB,ØSUB and the labelled common DQDCØM must be included in the link containing DEQD or in a higher link.

The functions of these subroutines are

FSUB Computes the second derivatives of the dependent variables, and any stopping functions.

ØSUB Provides for output at completion of integration of each interval (regular points) and at certain points obtained by interpolation.

The Kutta mode may not be used if any interpolation is required (i.e., even-interval output or stopping functions).

Usage

DEQD is designed to integrate the second-order differential equations

$$\frac{d}{dx}(\mathrm{RDEQ}(i)) = \mathrm{VDEQ}(i)$$

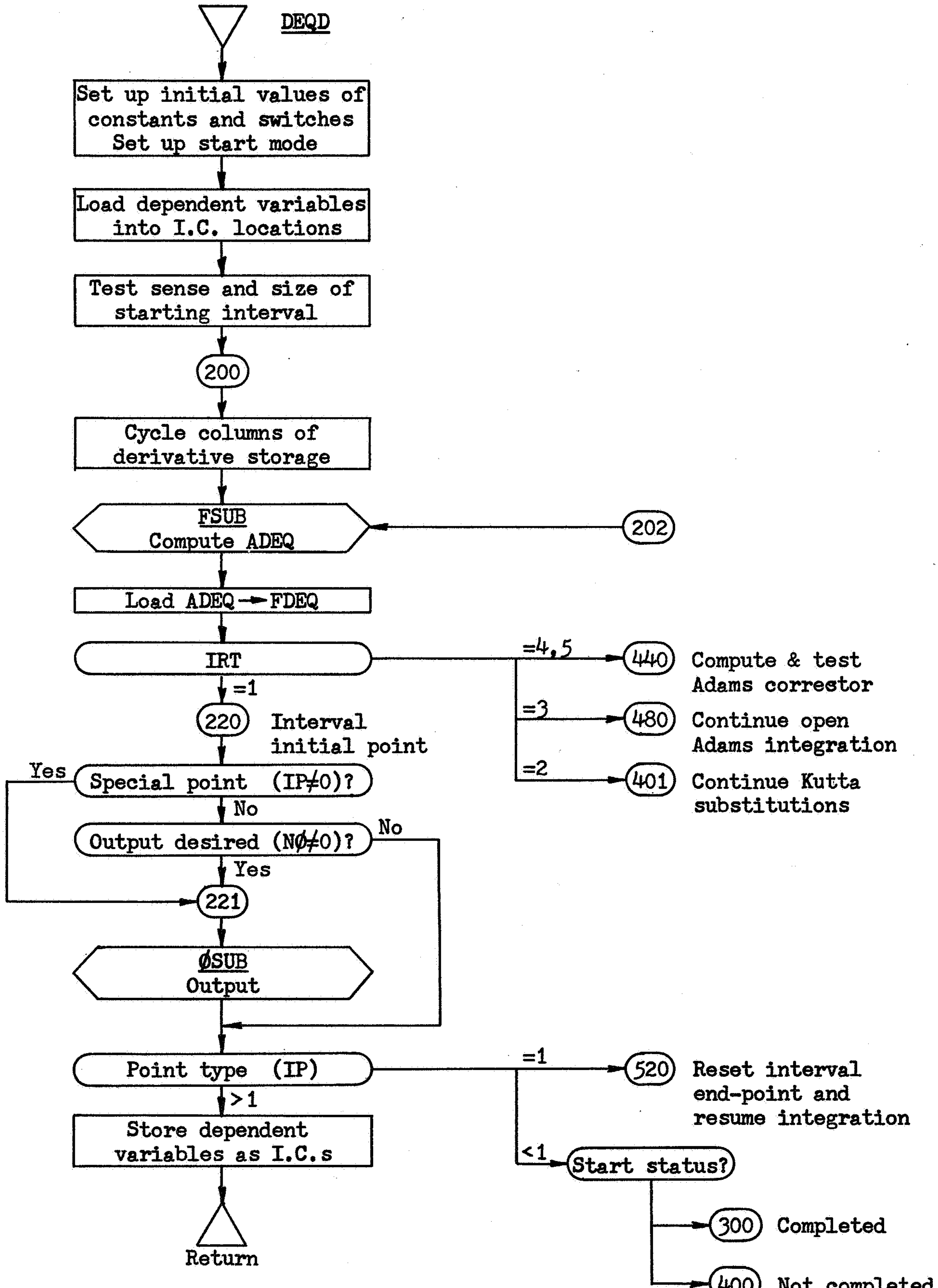
$$\frac{d}{dx}(\mathrm{VDEQ}(i)) = \mathrm{ADEQ}(i)$$

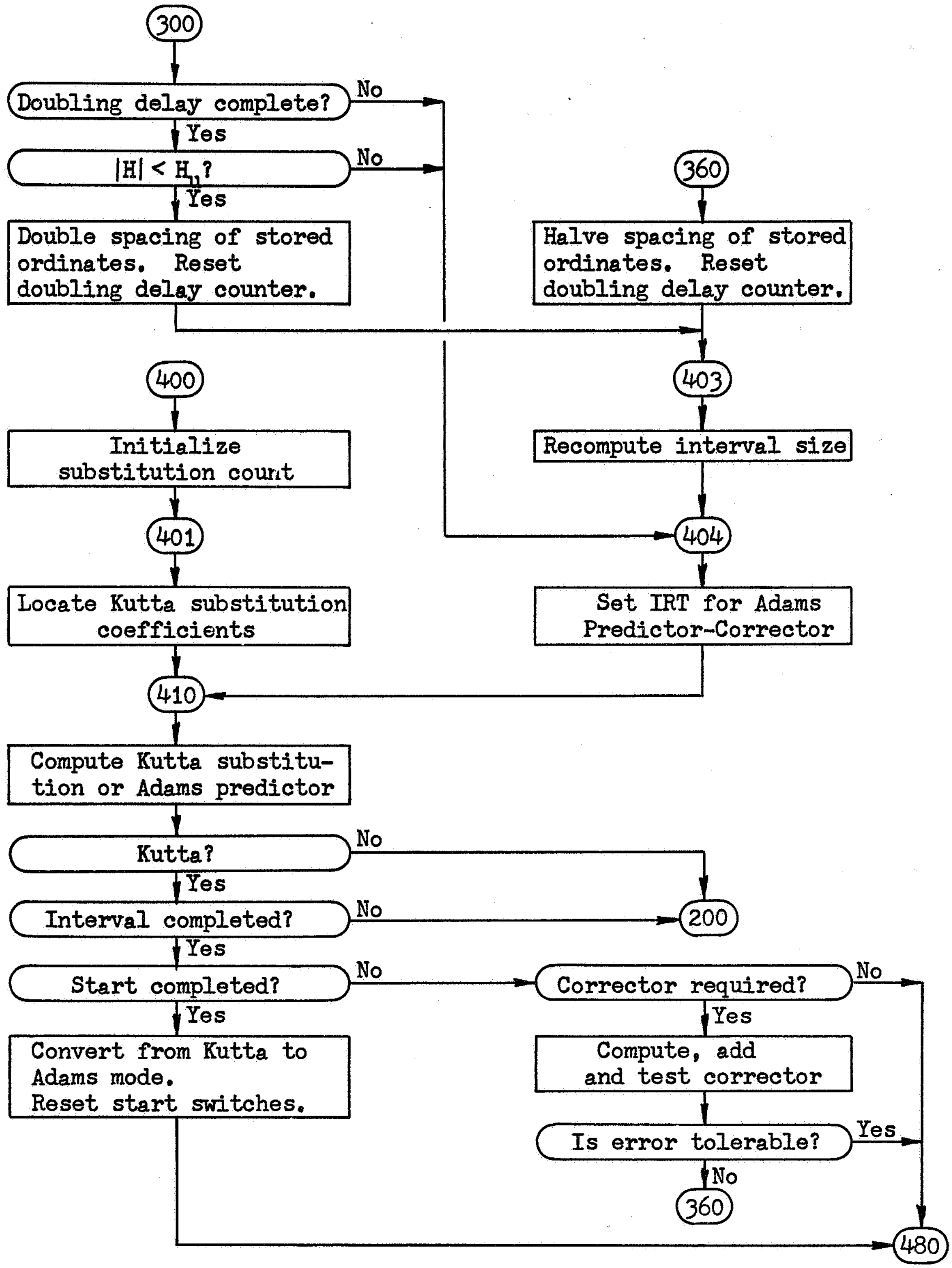
from initial conditions pre-loaded into RDEQ,VDEQ, and derivatives loaded into ADEQ by the subroutine "FSUB". NE upper-loaded equations are integrated from the pre-loaded x = XDEQ(1) to the specified final point XDEQ(2).

If NS $\neq$ 0, NS functions are loaded into ADEQ by "FSUB", starting at ADEQ(NE+1). The first zero of these functions, if any, causes a return from DEQD with XDEQ(1) at the zero, and IP = 3. The function causing the return is ADEQ(IT).

Interpolated values of RDEQ,VDEQ may be obtained at even intervals for output purposes during Adams integration. The first point interpolated is the pre-loaded XDEQ(3) and others follow at the interval HE.

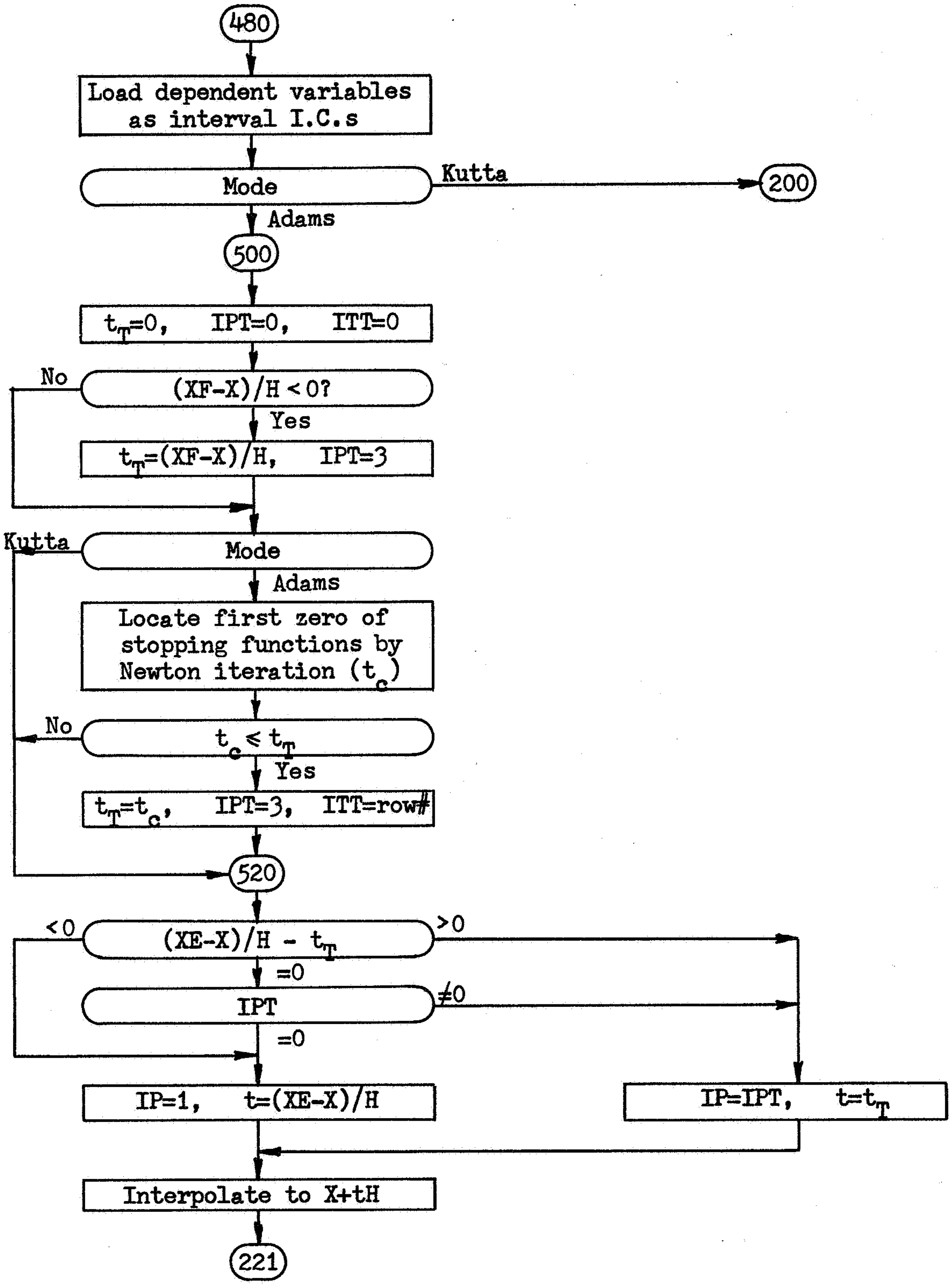
The integration, interpolation, and accuracy formulas are described in Appendix D of Reference 1.


DEQD
Set up initial values of constants and switches
Set up start mode
Load dependent variables into I.C. locations
Test sense and size of starting interval
200
Cycle columns of derivative storage
FSUB
Compute ADEQ
202
Load ADEQ → FDEQ
IRT
=4,5
440
Compute & test Adams corrector
=3
480
Continue open Adams integration
=2
401
Continue Kutta substitutions
=1
220
Interval initial point
Yes
Special point (IP≠0)?
No
Output desired (NØ≠0)?
No
Yes
221
ØSUB
Output
Point type (IP)
=1
520
Reset interval end-point and resume integration
>1
Store dependent variables as I.C.s
<1
Start status?
300
Completed
400
Not completed
Return



300
Doubling delay complete?
No
Yes
|H| < H_u?
No
Yes
Double spacing of stored ordinates. Reset doubling delay counter.
360
Halve spacing of stored ordinates. Reset doubling delay counter.
400
Initialize substitution count
403
Recompute interval size
401
404
Locate Kutta substitution coefficients
Set IRT for Adams Predictor-Corrector
410
Compute Kutta substitution or Adams predictor
Kutta?
No
Yes
Interval completed?
No
200
Yes
Start completed?
No
Corrector required?
No
Yes
Yes
Convert from Kutta to Adams mode. Reset start switches.
Compute, add and test corrector
Is error tolerable?
Yes
No
360
480

DEQD-5


480
Load dependent variables as interval I.C.s
Mode
Kutta
200
Adams
500
t_T=0, IPT=0, ITT=0
No
(XF-X)/H < 0?
Yes
t_T=(XF-X)/H, IPT=3
Kutta
Mode
Adams
Locate first zero of stopping functions by Newton iteration (t_c)
No
$t_c \leqslant t_T$
Yes
t_T=t_c, IPT=3, ITT=row#
520
<0
(XE-X)/H - t_T
>0
=0
IPT
≠0
=0
IP=1, t=(XE-X)/H
IP=IPT, t=t_T
Interpolate to X+tH
221

DEQD-6

```
$IBFTC MC133D  XR3,M94,NODD,LIST
      SUBROUTINE DEQD (FSUB,OSUB)
C     DEQD - DOUBLE PRECISION INTEGRATION
C              OF SECOND ORDER EQUATIONS
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
     1                   ,X,XF,XE,X0
      EQUIVALENCE        (NDEQ( 1),NI),(CDEQ( 1),H ),(XDEQ( 1),X )
     1                  ,(NDEQ( 2),NO),(CDEQ( 2),HP),(XDEQ( 2),XF)
     2                  ,(NDEQ( 3),NS),(CDEQ( 3),H0),(XDEQ( 3),XE)
     3                  ,(NDEQ( 4),NA),(CDEQ( 4),HU),(XDEQ( 4),X0)
     4                  ,(NDEQ( 5),NC),(CDEQ( 5),HL)
     5                  ,(NDEQ( 6),NE),(CDEQ( 6),HE)
     6                  ,(NDEQ( 7),NT),(CDEQ( 7),EU)
     7                  ,(NDEQ( 8),IP),(CDEQ( 8),EL)
     8                  ,(NDEQ( 9),IT)
C
      COMMON              STP(40)
      DOUBLE PRECISION    TMP(20)
      EQUIVALENCE        (STP(1),TMP)
C
      DOUBLE PRECISION    ALP(4),CQD(12),CQX(4),DCN(2),DX,FLT(7)
      DIMENSION           CAE  (7),CDB(7,2),CQK (20),SFL (14)
     1                   ,CAI(6,6),CHV(6,3),IN   (8),SLP  (8)
     2                   ,CCV(8,6),CQA (12)
      EQUIVALENCE        (ALP(1),SLP),(FLT(1),SFL)
C
      DATA CAE/  1., -6., 15.,-20., 15., -6., 1./
      DATA CAI/5*0.,60.,  -12.,   75., -200.,  300., -300.,  137.
     1                 ,  -50.,  305., -780., 1070., -770.,  225.
     2                 , -105.,  615.,-1470., 1770.,-1065.,  255.
     3                 , -120.,  660.,-1440., 1560., -840.,  180.
     4                 ,  -60.,  300., -600.,  600., -300.,   60./
      DATA CCV/ 3864.,-3200.,-6496.,-1539.,12327.,-4300., -224.,  324.
     1        , 1176., -400.,-3584., -972., 7812.,-3600.,    0.,  324.
     2        ,  126.,  300.,-1344., -405., 3717.,-1800.,    0.,  162.
     3        ,    0.,    0.,    0.,    0.,  756.,    0.,    0.,    0.
     4        ,   84., -200.,  224.,   81., -357.,  700.,  224.,    0.
     5        , -336.,  800., -896., -324., 1092., -800.,  896.,  324./
      DATA CDB/ -224., -112.,  -48.,   28.,  210.,  140.,    7.
     1        ,-2400.,-1008., -560.,  210., 2100., 1575.,   84./
      DATA CHV/-0.01171875, 0.08203125,-0.2734375 , 0.8203125
     1        , 0.41015625,-0.02734375, 0.01171875,-0.09865625
     2        , 0.5859375 , 0.5859375 ,-0.09865625, 0.01171875
     3        , 0.02737375,-0.17578125, 0.4921875 ,-0.8203125
     4        , 1.23046875, 0.24609375/
      DATA CQA/  -863.,  5260.,-13474., 18752.,-15487., 10852.
     1        , -3325., 20139.,-51086., 69874.,-55461., 29939./
      DATA CQK/  2268.,     0.,  2268.,  1680.,  -196.,   784.,   420.
     1        ,  1400.,   448.,     0.,  4536., -3969.,  8505., 13944.
     2        ,-15680.,  6272.,   420.,  2000.,  1792.,   324./
      DATA CQX/ 1.0D+0, 1.0D+0, .75D+0, .30D+0/
      DATA DCN/ 3.168648818567611, 3.966129870129870/
      DATA FLT/ 1.D0, 2.D0, 3.D0, 4.D0, 5.D0, 6.D0, 7.D0/
C
C**** PROLOGUE          SET FIXED DATA AND SWITCHES
C
  100 DO 101 I=1,8
  101 IN(I) = I
      HP = 0.
      EL = EU/128.
      ASSIGN 234 TO LLA
      ASSIGN 480 TO LLB
      ASSIGN 480 TO LLC
      ASSIGN 200 TO LLD
      ASSIGN 521 TO LLE
      ASSIGN 200 TO LLF
      IF (NI.EQ.0) GO TO 103
      IF (NS.NE.0) ASSIGN 502 TO LLE
      IF (NA.NE.0) ASSIGN 450 TO LLC
      ASSIGN 202 TO LLF
      IF (NC.NE.2) ASSIGN 220 TO LLF
  103 IRT = 1
```

```
      IRS = 0
      IPT = 0
      IP  = 2
      H   = 0.
      ALP(4) = 0.D0
      GO TO 480
  140 H   = H0
      TMP(1) = XF-X
      IF (STP(1).LT.0.) H=-H
      IF (STP(1)/H.GT.2.) GO TO 141
      H = TMP(1)/2.D0
      IPT = 3
  141 IP = 3
      IF (ABS(H).EQ.0.) GO TO 221
      IP = 0
      IT = 0
      N1 = NE+1
      NT = NE+NS
C
C**** CONTROL             SELECT COMPUTATIONAL PATH
C
C     COMPUTE AND LOAD DERIVATIVES
  200 I = IN(1)
      DO 201 J=1,7
  201 IN(J) = IN(J+1)
      IN(8) = I
  202 CALL FSUB
      I = IN(8)
      DO 203 J=1,NT
  203 FDEQ(J,I) = ADEQ(J)
      GO TO (220,401,480,440) ,IRT
C
C     OUTPUT CONTROL
  220 IF (IP.NE.0) GO TO 221
      IF (NO.EQ.0) GO TO 233
  221 CALL OSUB
C
C     IDENTIFY POINT TYPE AND SELECT PATH
  230 IF (IP) 231,232,232
  231 IPT = 0
      NS = 0
      NT = NE
      XF = XF+FLT(7)*DX
      GO TO 300
  232 IF (IP-1) 233,520,480
  233 GO TO LLA, (234,400,300)
  234 IF (HP.EQ.0.) GO TO 400
      ASSIGN 400 TO LLA
      IF (NI.EQ.0) GO TO 400
      ASSIGN 300 TO LLA
      ASSIGN 420 TO LLB
      ASSIGN 500 TO LLD
      GO TO 400
C
C**** NEXT                INTERVAL SIZE CONTROL
C
C     SELECT NEXT INTERVAL SIZE
  300 IF (IDL.GT.0)      GO TO 301
      IF (ABS(H).GE.HU) GO TO 404
      GO TO 340
  301 IDL = IDL-1
      GO TO 404
C
C     DOUBLE SPACING OF ADAMS ORDINATES
  340 H = H*SFL(3)
      I = IN(7)
      J = IN(5)
      IN(7) = IN(6)
      IN(6) = IN(4)
      IN(5) = IN(2)
      IN(4) = I
      IN(2) = J
      DO 342 I=1,NT
      TMP(1) = 0.D0
      TMP(2) = 0.D0
```

```
      DO 341 J=1,7
      L = IN(J+1)
      TMP(1) = TMP(1)+CDB(J,1)*FDEQ(I,L)
  341 TMP(2) = TMP(2)+CDB(J,2)*FDEQ(I,L)
      L = IN(4)
      FDEQ(I,L) = TMP(1)
      L = IN(3)
  342 FDEQ(I,L) = TMP(2)
      GO TO 403
C
C     HALVE SPACING OF ADAMS ORDINATES
  360 H = H/SFL(3)
      DO 363 I=1,NT
      DO 361 J=1,6
  361 STP(J) = 0.
      DO 362 J=1,6
      L = IN(J+1)
      TMP(4) = FDEQ(I,L)
      DO 362 K=1,3
  362 TMP(K) = TMP(K)+TMP(4)*CHV(J,K)
      DO 363 J=1,3
      L = IN(J+1)
  363 FDEQ(I,L) = TMP(J)
      I = IN(8)
      IN(8) = IN(7)
      IN(7) = IN(4)
      IN(4) = IN(5)
      IN(5) = IN(2)
      IN(2) = I
      GO TO 403
C
C**** QUADRATURE         INTEGRATE TO X0+H
C
C     LOCATE KUTTA SUBSTITUTION COEFFICIENTS
  400 IRT = 2
      ISB = 9
      ICN = 1
  401 ISB = ISB-1
      DX  = H*CQX(ISB-4)
      DO 402 I=ISB,8
      CQD(I-2) = CQK(ICN)/4536.D0
      CQD(I+4) = CQK(ICN+10)/4536.D0
  402 ICN = ICN+1
      GO TO 410
  403 IDL = 2
      DX = H
      ALP(2) = 60.D0/DX
      ALP(3) = ALP(2)/DX
  404 IRT = NRT
C
C     INTEGRATE/PREDICT
  410 X = X0+DX
      DO 413 I=1,NE
      TMP(1) = 0.D0
      TMP(2) = 0.D0
      DO 411 J=ISB,8
      L = IN(J)
      TMP(1) = TMP(1)+CQD(J-2)*FDEQ(I,L)
  411 TMP(2) = TMP(2)+CQD(J+4)*FDEQ(I,L)
      DO 412 J=1,2
  412 TMP(J) = FDEQ(I,10)+DX*TMP(J)
      RDEQ(I) = FDEQ(I,9)+DX*TMP(1)
  413 VDEQ(I) = TMP(2)
      IF (IRT.NE.2) GO TO 200
      IF (ISB.GT.5) GO TO 200
      GO TO LLB, (480,420,440,432)
C
C**** CONVERT            CHANGE FROM START TO ADAMS MODE
C
C     COMPUTE ADAMS ORDINATES
  420 H = H/SFL(3)
      DO 422 I=1,NT
      DO 421 J=1,6
      TMP(J) = 0.D0
      DO 421 L=1,8
```

```
  421 TMP(J) = TMP(J)+CCV(L,J)*FDEQ(I,L)
      DO 422 J=3,8
  422 FDEQ(I,J) = TMP(J-2)/756.D0
C
C     SET ADAMS INTEGRATION PARAMETERS
  430 ISB = 3
      NRT = 3
      ASSIGN 480 TO LLB
      DO 431 I=1,12
  431 CQD(I) = CQA(I)/10080.D0
      IF (NA+NC.EQ.0) GO TO 480
      NRT = 4
      IDL = 4
      IP = IPT
      ASSIGN 432 TO LLB
      IF (IP.NE.0) GO TO 221
      GO TO 480
  432 IF (IDL.GT.1) GO TO 480
      ASSIGN 440 TO LLB
C
C**** CORRECT             TEST ACCURACY AND ADD ADAMS CORRECTOR
C
C     ADD CORRECTOR AND COMPUTE MAXIMUM
  440 STP(1) = 0.
      DO 442 I=1,NE
      STP(2) = ABS(VDEQ(I))
      IF (SFL(1).GT.STP(2)) STP(2) = SFL(1)
      TMP(3) = 0.D0
      DO 441 J=1,7
      L = IN(J+1)
  441 TMP(3) = TMP(3)+CAE(J)*FDEQ(I,L)
      TMP(3) = DX*TMP(3)/DCN(1)
      TMP(2) = DX*TMP(3)/DCN(2)
      STP(2) = ABS(STP(5))/STP(2)
      IF (STP(2).GT.STP(1).AND.I.LE.3) STP(1) = STP(2)
      IF (NC.EQ.0) GO TO 442
      RDEQ(I) = RDEQ(I)+TMP(2)
      VDEQ(I) = VDEQ(I)+TMP(3)
  442 CONTINUE
      GO TO LLC, (480,450)
C
C     TEST ACCURACY
  450 IF (STP(1).LT.EL) GO TO 480
      IDL = IDL+1
      IF (STP(1).LT.EU) GO TO 480
      IF (ABS(H).LE.HL) GO TO 480
      GO TO 360
C
C**** ADVANCE             STORE ICS FOR NEXT INTERVAL
C
  480 IRT = 1
      HP = H
      DO 481 I=1,NE
      FDEQ(I, 9) = RDEQ(I)
  481 FDEQ(I,10) = VDEQ(I)
      X0 = X
      IF (IP-2) 482,140,999
  482 GO TO LLD, (500,200)
C
C**** TEST                LOCATE AND ORDER SPECIAL POINTS
C
C     FIND STOPPING POINTS
  500 ALP(4) = 0.D0
      IPT = 0
      ITT = 0
      ALP(1) = (XF-X)/DX
      IF (SLP(1).GT.0.) GO TO 501
      ALP(4) = ALP(1)
      IPT = 3
  501 GO TO LLE, (502,521)
  502 K = IN(8)
      L = IN(7)
      DO 512 I=N1,NT
      TMP(2) = FDEQ(I,L)-FDEQ(I,K)
      M = 3
```

```
      GO TO 504
  503 K = IN(6)
      M = 2
      TMP(2) = (TMP(2)+FDEQ(I,K)-FDEQ(I,L))/FLT(2)
  504 IF (STP(3).EQ.0.) GO TO 506
      TMP(1) = FDEQ(I,L)/TMP(2)-FLT(1)
  505 IF (STP(1).GT.SLP(7)) GO TO 506
      IF (STP(1)+SFL(3).LE.0.) GO TO 506
      GO TO (512,507,507) ,M
  506 GO TO (512,512,503) ,M
  507 M = 1
      DO 508 J=1,12
  508 STP(J+6) = 0.
      DO 509 J=1,6
      L = IN(J+2)
      TMP(10) = FDEQ(I,L)
      DO 509 K=1,6
  509 TMP(K+3) = TMP(K+3)+TMP(10)*CAI(J,K)
      DO 510 J=1,5
      K = 5-J
  510 TMP(9) = TMP(K+4)+TMP(9)*TMP(1)/FLT(K+1)
      TMP(3) = TMP(9)/TMP(2)/60.D0
      TMP(1) = TMP(1)+TMP(3)
      IF (ABS(STP(5)).GT..1E-11) GO TO 505
      IF (TMP(1)-ALP(4)) 511,511,512
  511 IPT = 3
      ITT = I
      ALP(4) = TMP(1)
  512 CONTINUE
      GO TO 521
C
C     FIND EVEN-INTERVAL POINT
  520 IRS = 1
  521 ALP(1) = (XE-X)/DX
      IF (ALP(1)-ALP(4)) 523,522,524
  522 IF (IPT.NE.0) GO TO 524
  523 IP = 1
      XE = XE+HE
      GO TO 525
  524 IP = IPT
      IT = ITT
      ALP(1) = ALP(4)
  525 IF (SLP(1).NE.0.) GO TO 610
      IF (IPT.NE.0) GO TO 221
      IF (IRS) 600,600,540
C
C**** RESET              RESTORE INTERVAL END-POINT
C
  540 IRS = 0
      X = X0
      L = IN(8)
      DO 541 I=1,NE
      RDEQ(I) = FDEQ(I, 9)
  541 VDEQ(I) = FDEQ(I,10)
      DO 542 I=1,NT
  542 ADEQ(I) = FDEQ(I,L)
C
C**** INTERPOLATE        COMPUTE INTEGRALS AT X0+T*H
C
  600 IF (SLP(1).EQ.0.) GO TO LLF, (200,202,220)
C
C     COMPUTE INTERPOLATION COEFFICIENTS
  610 TMP(1) = ALP(1)
      DO 611 I=1,6
  611 TMP(I+1) = TMP(1)*TMP(I)/FLT(I+1)
      DO 612 I=17,40
  612 STP(I) = 0.
      DO 614 I=1,6
      DO 613 J=1,6
      TMP(8) = CAI(I,J)
      TMP(I+ 8) = TMP(I+ 8)+TMP(8)*TMP(J)
  613 TMP(I+14) = TMP(I+14)+TMP(8)*TMP(J+1)
      TMP(I+ 8) = TMP(I+ 8)/ALP(2)
  614 TMP(I+14) = TMP(I+14)/ALP(3)
C
```

```
C     INTERPOLATE TO X0+T*H                                              DEQD0375
  620 TMP(6) = DX*TMP(1)                                                 DEQD0376
      X = X0+TMP(6)                                                      DEQD0377
      DO 621 I=1,NE                                                      DEQD0378
      VDEQ(I) = FDEQ(I,10)                                               DEQD0379
      RDEQ(I) = FDEQ(I, 9)+DX*FDEQ(I,10)                                 DEQD0380
      DO 621 J=1,6                                                       DEQD0381
      L = IN(J+2)                                                        DEQD0382
      VDEQ(I) = VDEQ(I)+FDEQ(I,L)*TMP(J+ 8)                              DEQD0383
  621 RDEQ(I) = RDEQ(I)+FDEQ(I,L)*TMP(J+14)                              DEQD0384
      GO TO 221                                                          DEQD0385
C                                                                        DEQD0386
C**** RETURN             STOP POINT REACHED                              DEQD0387
C                                                                        DEQD0388
  999 RETURN                                                             DEQD0389
      END
```

Subroutine: DEQTR

Purpose: To compute parameters describing the precession of the mean equator of date.

Calling Sequence: CALL DEQTR (T,A)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Description
I	T	d		sec	Time from 1950 Jan 0.0 (ET)
Ø	A(1)	d	$\cos\alpha$		Cosine of precession angle
Ø	A(2)	d	$\sin\alpha$		Sine of precession angle
Ø	A(3)	d	$\bar{\epsilon}$		Mean obliquity of the ecliptic
Ø	A(4-6)	d(3)	w		Vector along the axis of precession.

Common storages used: None

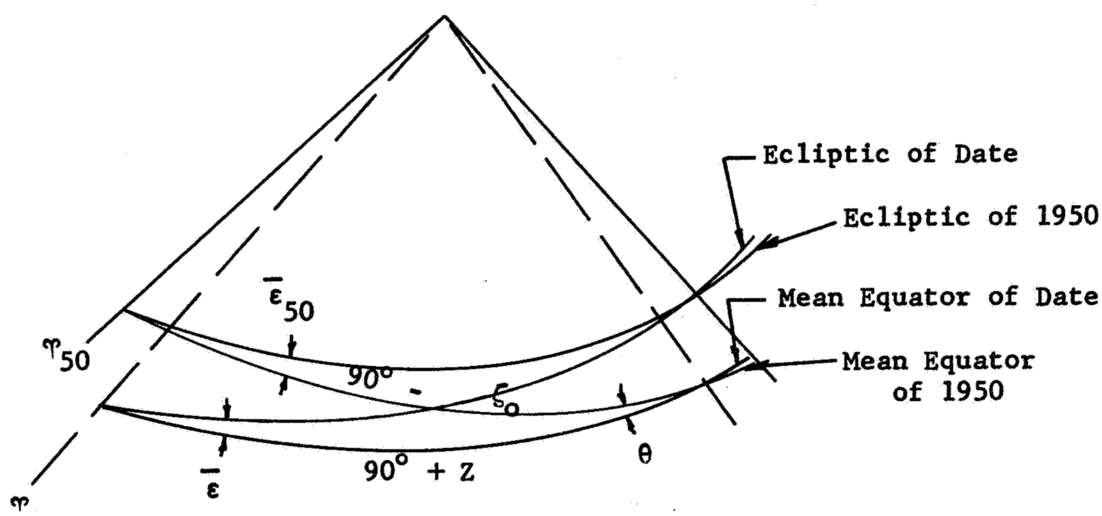
Subroutines required: None

Method

The precession of the Earth's mean equator of date is described by the three small angles ζ_o, Z, θ, where

$$\zeta_o + Z = \text{precession in right ascension}$$

$$\theta = \text{precession in declination}$$



Precession of the Mean Equator

Numerical values for ζ_o, Z, θ are given by reference 3, p. 30, in tropical centuries (of 36524.21988 ephemeris days) from 1900.0 (1900 Jan 0.814 ET). The base coordinate system is equator and equinox of 1950.0 (1949 Dec 30.924 ET), and the corresponding expressions in Julian centuries from T_{E1} are

T_E = Julian centuries from T_{E1}

$$T = T_E + 0.20824 \times 10^{-5}$$

$$\zeta_o = 2304''.997T + 0''.302T^2 + 0''.018T^3$$

$$Z = 2304''.997T + 1''.093T^2 + 0''.018T^3$$

$$\theta = 2004''.298T - 0''.426T^2 - 0''.042T^3$$

The mean obliquity of the ecliptic, $\bar{\epsilon}$, from reference 3, p.38, is

$$\bar{\epsilon} = 23''.4457878 - 0^\circ.01301376T_E - 0^\circ.8855 \times 10^{-6}T_E^2 + 0^\circ.503 \times 10^{-1}T_E^3$$

The transformation from mean EE of 1950.0 (C-frame) to mean EE of date (D_o-frame) is

$$T_{C2D_o} = T_W(\alpha) = T_3(-90^\circ - Z)\ T_1(\theta)\ T_3(90^\circ - \zeta_o)$$

Since retention of orthogonality in double precision calculation of T_{C2D_o} from the expansion of the right-hand side requires large polynomials in T (through terms in T^3), the eigenvector, W, and rotation angle, α, together with the mean obliquity are computed by DEQTR, using

$$\sin\alpha = .0243708730T + .65233 \times 10^{-5}T^2 - .493T^3$$

$$\cos\alpha = \sqrt{1 - \sin^2\alpha}$$

$$W_1 = -.83384 \times 10^{-5}T^2 + .41 \times 10^{-8}\ T^3$$

$$W_2 = .4347723663 - .2119164 \times 10^{-3}T + .12169 \times 10^{-5}T^2 + .32 \times 10^{-8}T^3$$

$$W_3 = -1.$$

```
$IBFTC MC132Z  XR3,M94,NODD,LIST
      SUBROUTINE DEQTR (T,A)
C     PRECESSION OF THE MEAN EQUATOR
      DOUBLE PRECISION  T,A(6),C(4,4),DSQRT
      DATA   C/-.493 D-7, .65233D-5, .24370873 D-1,0.D0
     1        , .8779D-8,-.15455D-7,-.227133008D-3, .409206191890
     2        , .41  D-8,-.83384D-5,0.D0          ,0.D0
     3        , .32  D-8, .12169D-5,-.2119164  D-3, .4347723663/
C
    1 TC = T/3.15576D+9+0.20824D-05
      DO 2 I=1,4
    2 A(I+1) = C(1,I)
      DO 3 I=2,4
      DO 3 J=1,4
    3 A(J+1) = C(I,J)+TC*A(J+1)
      A(1) = DSQRT(1.D0-A(2)*A(2))
      A(6) =-1.D0
  999 RETURN
      END
```

Subroutine: DEQ3

Purpose: Integrates three second-order equations in double precision. (See also DEQD).

Calling Sequence: CALL DEQ3 (FSUB,ØSUB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
	FSUB				Name of subroutine which evaluates derivatives
	ØSUB				Name of subroutine which controls output.

Common storages used: //(40),/DQ3CØM/

Subroutines required: FSUB,ØSUB

Restrictions

An "EXTERNAL FSUB,ØSUB" statement must be included in the calling program. The subroutines FSUB, ØSUB must have zero length call lists. For use with OVERLAY, FSUB,ØSUB and the labelled common DQ3CØM must be included in the link containing DEQ3 or in a higher link.

The functions of these subroutines are

FSUB Computes the second derivatives of the dependent variables, and any stopping functions.

ØSUB Provides for output at completion of integration of each interval (regular points) and at certain points obtained by interpolation.

The Kutta mode may not be used if any interpolation is required (i.e., even interval output or stopping functions).

```
$IBFTC MC133E  XR3,M94,NODD,LIST
      SUBROUTINE DEQ3 (FSUB,OSUB)
C     DEQ3 - DOUBLE PRECISION INTEGRATION
C              OF EQUATIONS OF STATE
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
     1                   ,X,XF,XE,X0
      EQUIVALENCE        (NDEQ( 1),NI),(CDEQ( 1),H ),(XDEQ( 1),X )
     1                  ,(NDEQ( 2),NO),(CDEQ( 2),HP),(XDEQ( 2),XF)
     2                  ,(NDEQ( 3),NS),(CDEQ( 3),HO),(XDEQ( 3),XE)
     3                  ,(NDEQ( 4),NA),(CDEQ( 4),HU),(XDEQ( 4),X0)
     4                  ,(NDEQ( 5),NC),(CDEQ( 5),HL)
     5                  ,(NDEQ( 6),NE),(CDEQ( 6),HE)
     6                  ,(NDEQ( 7),NT),(CDEQ( 7),EU)
     7                  ,(NDEQ( 8),IP),(CDEQ( 8),EL)
     8                  ,(NDEQ( 9),IT)
C
      COMMON              STP(40)
      DOUBLE PRECISION    TMP(20)
      EQUIVALENCE        (STP(1),TMP)
C
      DOUBLE PRECISION    ALP(4),CQD(12),CQX(4),DCN(2),DX,FLT(7)
      DIMENSION           CAE  (7),CDB(7,2),CQK (20),SFL (14)
     1                   ,CAI(6,6),CHV(6,3),IN   (8),SLP  (8)
     2                   ,CCV(8,6),CQA (12)
      EQUIVALENCE        (ALP(1),SLP),(FLT(1),SFL)
C
      DATA CAE/  1., -6., 15.,-20., 15., -6., 1./
      DATA CAI/5*0.,60.,  -12.,    75., -200.,  300., -300.,  137.
     1                 ,  -50.,  305., -780., 1070., -770.,  225.
     2                 , -105.,  615.,-1470., 1770.,-1065.,  255.
     3                 , -120.,  660.,-1440., 1560., -840.,  180.
     4                 ,  -60.,  300., -600.,  600., -300.,   60./
      DATA CCV/ 3864.,-3200.,-6496.,-1539.,12327.,-4300., -224.,  324.
     1        , 1176., -400.,-3584., -972., 7812.,-3600.,    0.,  324.
     2        ,  126.,  300.,-1344., -405., 3717.,-1800.,    0.,  162.
     3        ,    0.,    0.,    0.,    0.,  756.,    0.,    0.,    0.
     4        ,   84., -200.,  224.,   81., -357.,  700.,  224.,    0.
     5        , -336.,  800., -896., -324., 1092., -800.,  896.,  324./
      DATA CDB/ -224., -112.,  -48.,   28.,  210.,  140.,    7.
     1        ,-2400.,-1008., -560.,  210., 2100., 1575.,   84./
      DATA CHV/-0.01171875, 0.08203125,-0.2734375 , 0.8203125
     1        , 0.41015625,-0.02734375, 0.01171875,-0.09865625
     2        , 0.5859375 , 0.5859375 ,-0.09865625, 0.01171875
     3        , 0.02734375,-0.17578125, 0.4921875 ,-0.8203125
     4        , 1.23046875, 0.24609375/
      DATA CQA/  -863.,  5260.,-13474., 18752.,-15487., 10852.
     1        , -3325., 20139.,-51086., 69874.,-55461., 29939./
      DATA CQK/  2268.,     0.,  2268.,  1680.,  -196.,   784.,   420.
     1        ,  1400.,   448.,     0.,  4536., -3969.,  8505., 13944.
     2        ,-15680.,  6272.,   420.,  2000.,  1792.,   324./
      DATA CQX/ 1.0D+0, 1.0D+0, .75D+0, .30D+0/
      DATA DCN/ 3.168648818567611, 3.966129870129870/
      DATA FLT/ 1.D0, 2.D0, 3.D0, 4.D0, 5.D0, 6.D0, 7.D0/
C
C**** PROLOGUE            SET FIXED DATA AND SWITCHES
C
  100 DO 101 I=1,8
  101 IN(I) = I
      HP = 0.
      EL = EU/128.
      ASSIGN 234 TO LLA
      ASSIGN 480 TO LLB
      ASSIGN 480 TO LLC
      ASSIGN 200 TO LLD
      ASSIGN 521 TO LLE
      ASSIGN 200 TO LLF
      IF (NI.EQ.0) GO TO 103
      IF (NS.NE.0) ASSIGN 502 TO LLE
      IF (NA.NE.0) ASSIGN 450 TO LLC
      ASSIGN 202 TO LLF
      IF (NC.NE.2) ASSIGN 220 TO LLF
  103 IRT = 1
```

```
      IRS = 0
      IPT = 0
      IP = 2
      H  = 0.
      ALP(4) = 0.D0
      GO TO 480
  140 H  = HO
      TMP(1) = XF-X
      IF (STP(1).LT.0.) H=-H
      IF (STP(1)/H.GT.2.) GO TO 141
      H = TMP(1)/2.D0
      IPT = 3
  141 IP = 3
      IF (ABS(H).EQ.0.) GO TO 221
      IP = 0
      IT = 0
      N1 = NE+1
      NT = NE+NS
C
C**** CONTROL             SELECT COMPUTATIONAL PATH
C
C     COMPUTE AND LOAD DERIVATIVES
  200 I = IN(1)
      DO 201 J=1,7
  201 IN(J) = IN(J+1)
      IN(8) = I
  202 CALL FSUB
      I = IN(8)
      DO 203 J=1,NT
  203 FDEQ(J,I) = ADEQ(J)
      GO TO (220,401,480,440) ,IRT
C
C     OUTPUT CONTROL
  220 IF (IP.NE.0) GO TO 221
      IF (NO.EQ.0) GO TO 233
  221 CALL OSUB
C
C     IDENTIFY POINT TYPE AND SELECT PATH
  230 IF (IP) 231,232,232
  231 IPT = 0
      NS = 0
      NT = NE
      XF = XF+FLT(7)*DX
      GO TO 300
  232 IF (IP-1) 233,520,480
  233 GO TO LLA, (234,400,300)
  234 IF (HP.EQ.0.) GO TO 400
      ASSIGN 400 TO LLA
      IF (NI.EQ.0) GO TO 400
      ASSIGN 300 TO LLA
      ASSIGN 420 TO LLB
      ASSIGN 500 TO LLD
      GO TO 400
C
C**** NEXT                 INTERVAL SIZE CONTROL
C
C     SELECT NEXT INTERVAL SIZE
  300 IF (IDL.GT.0)       GO TO 301
      IF (ABS(H).GE.HU) GO TO 404
      GO TO 340
  301 IDL = IDL-1
      GO TO 404
C
C     DOUBLE SPACING OF ADAMS ORDINATES
  340 H = H*SFL(3)
      I = IN(7)
      J = IN(5)
      IN(7) = IN(6)
      IN(6) = IN(4)
      IN(5) = IN(2)
      IN(4) = I
      IN(2) = J
      DO 342 I=1,NT
      TMP(1) = 0.D0
      TMP(2) = 0.D0
```

```
      DO 341 J=1,7
      L = IN(J+1)
      TMP(1) = TMP(1)+CDB(J,1)*FDEQ(I,L)
  341 TMP(2) = TMP(2)+CDB(J,2)*FDEQ(I,L)
      L = IN(4)
      FDEQ(I,L) = TMP(1)
      L = IN(3)
  342 FDEQ(I,L) = TMP(2)
      GO TO 403
C
C     HALVE SPACING OF ADAMS ORDINATES
  360 H = H/SFL(3)
      DO 363 I=1,NT
      DO 361 J=1,6
  361 STP(J) = 0.
      DO 362 J=1,6
      L = IN(J+1)
      TMP(4) = FDEQ(I,L)
      DO 362 K=1,3
  362 TMP(K) = TMP(K)+TMP(4)*CHV(J,K)
      DO 363 J=1,3
      L = IN(J+1)
  363 FDEQ(I,L) = TMP(J)
      I = IN(8)
      IN(8) = IN(7)
      IN(7) = IN(4)
      IN(4) = IN(5)
      IN(5) = IN(2)
      IN(2) = I
      GO TO 403
C
C**** QUADRATURE        INTEGRATE TO X0+H
C
C     LOCATE KUTTA SUBSTITUTION COEFFICIENTS
  400 IRT = 2
      ISB = 9
      ICN = 1
  401 ISB = ISB-1
      DX  = H*CQX(ISB-4)
      DO 402 I=ISB,8
      CQD(I-2) = CQK(ICN)/4536.D0
      CQD(I+4) = CQK(ICN+10)/4536.D0
  402 ICN = ICN+1
      GO TO 410
  403 IDL = 2
      DX = H
      ALP(2) = 60.D0/DX
      ALP(3) = ALP(2)/DX
  404 IRT = NRT
C
C     INTEGRATE/PREDICT
  410 X = X0+DX
      DO 413 I=1,3
      TMP(1) = 0.D0
      TMP(2) = 0.D0
      DO 411 J=ISB,8
      L = IN(J)
      TMP(1) = TMP(1)+CQD(J-2)*FDEQ(I,L)
  411 TMP(2) = TMP(2)+CQD(J+4)*FDEQ(I,L)
      DO 412 J=1,2
  412 TMP(J) = FDEQ(I,10)+DX*TMP(J)
      RDEQ(I) = FDEQ(I,9)+DX*TMP(1)
  413 VDEQ(I) = TMP(2)
      IF (IRT.NE.2) GO TO 200
      IF (ISB.GT.5) GO TO 200
      GO TO LLB, (480,420,440,432)
C
C**** CONVERT           CHANGE FROM START TO ADAMS MODE
C
C     COMPUTE ADAMS ORDINATES
  420 H = H/SFL(3)
      DO 422 I=1,NT
      DO 421 J=1,6
      TMP(J) = 0.D0
      DO 421 L=1,8
```

```
  421 TMP(J) = TMP(J)+CCV(L,J)*FDEQ(I,L)
      DO 422 J=3,8
  422 FDEQ(I,J) = TMP(J-2)/756.D0
C
C     SET ADAMS INTEGRATION PARAMETERS
  430 ISB = 3
      NRT = 3
      ASSIGN 480 TO LLB
      DO 431 I=1,12
  431 CQD(I) = CQA(I)/10080.D0
      IF (NA+NC.EQ.0) GO TO 480
      NRT = 4
      IDL = 4
      IP = IPT
      ASSIGN 432 TO LLB
      IF (IP.NE.0) GO TO 221
      GO TO 480
  432 IF (IDL.GT.1) GO TO 480
      ASSIGN 440 TO LLB
C
C**** CORRECT              TEST ACCURACY AND ADD ADAMS CORRECTOR
C
C     ADD CORRECTOR AND COMPUTE MAXIMUM
  440 STP(1) = 0.
      DO 442 I=1,3
      STP(2) = ABS(VDEQ(I))
      IF (SFL(1).GT.STP(2)) STP(2) = SFL(1)
      TMP(3) = 0.D0
      DO 441 J=1,7
      L = IN(J+1)
  441 TMP(3) = TMP(3)+CAE(J)*FDEQ(I,L)
      TMP(3) = DX*TMP(3)/DCN(1)
      TMP(2) = DX*TMP(3)/DCN(2)
      STP(2) = ABS(STP(5))/STP(2)
      IF (STP(2).GT.STP(1)) STP(1) = STP(2)
      IF (NC.EQ.0) GO TO 442
      RDEQ(I) = RDEQ(I)+TMP(2)
      VDEQ(I) = VDEQ(I)+TMP(3)
  442 CONTINUE
      GO TO LLC, (480,450)
C
C     TEST ACCURACY
  450 IF (STP(1).LT.EL) GO TO 480
      IDL = IDL+1
      IF (STP(1).LT.EU) GO TO 480
      IF (ABS(H).LE.HL) GO TO 480
      GO TO 360
C
C**** ADVANCE              STORE ICS FOR NEXT INTERVAL
C
  480 IRT = 1
      HP = H
      DO 481 I=1,3
      FDEQ(I, 9) = RDEQ(I)
  481 FDEQ(I,10) = VDEQ(I)
      X0 = X
      IF (IP-2) 482,140,999
  482 GO TO LLD, (500,200)
C
C**** TEST                 LOCATE AND ORDER SPECIAL POINTS
C
C     FIND STOPPING POINTS
  500 ALP(4) = 0.D0
      IPT = 0
      ITT = 0
      ALP(1) = (XF-X)/DX
      IF (SLP(1).GT.0.) GO TO 501
      ALP(4) = ALP(1)
      IPT = 3
  501 GO TO LLE, (502,521)
  502 K = IN(8)
      L = IN(7)
      DO 512 I=N1,NT
      TMP(2) = FDEQ(I,L)-FDEQ(I,K)
      M = 3
```

```
      GO TO 504
  503 K = IN(6)
      M = 2
      TMP(2) = (TMP(2)+FDEQ(I,K)-FDEQ(I,L))/FLT(2)
  504 IF (STP(3).EQ.0.) GO TO 506
      TMP(1) = FDEQ(I,L)/TMP(2)-FLT(1)
  505 IF (STP(1).GT.SLP(7)) GO TO 506
      IF (STP(1)+SFL(3).LE.0.) GO TO 506
      GO TO (512,507,507) ,M
  506 GO TO (512,512,503) ,M
  507 M = 1
      DO 508 J=1,12
  508 STP(J+6) = 0.
      DO 509 J=1,6
      L = IN(J+2)
      TMP(10) = FDEQ(I,L)
      DO 509 K=1,6
  509 TMP(K+3) = TMP(K+3)+TMP(10)*CAI(J,K)
      DO 510 J=1,5
      K = 5-J
  510 TMP(9) = TMP(K+4)+TMP(9)*TMP(1)/FLT(K+1)
      TMP(3) = TMP(9)/TMP(2)/60.D0
      TMP(1) = TMP(1)+TMP(3)
      IF (ABS(STP(5)).GT..1E-11) GO TO 505
      IF (TMP(1)-ALP(4)) 511,511,512
  511 IPT = 3
      ITT = I
      ALP(4) = TMP(1)
  512 CONTINUE
      GO TO 521
C
C     FIND EVEN-INTERVAL POINT
  520 IRS = 1
  521 ALP(1) = (XE-X)/DX
      IF (ALP(1)-ALP(4)) 523,522,524
  522 IF (IPT.NE.0) GO TO 524
  523 IP = 1
      XE = XE+HE
      GO TO 525
  524 IP = IPT
      IT = ITT
      ALP(1) = ALP(4)
  525 IF (SLP(1).NE.0.) GO TO 610
      IF (IPT.NE.0) GO TO 221
      IF (IRS) 600,600,540
C
C**** RESET               RESTORE INTERVAL END-POINT
C
  540 IRS = 0
      X = X0
      L = IN(8)
      DO 541 I=1,3
      RDEQ(I) = FDEQ(I, 9)
  541 VDEQ(I) = FDEQ(I,10)
      DO 542 I=1,NT
  542 ADEQ(I) = FDEQ(I,L)
C
C**** INTERPOLATE         COMPUTE INTEGRALS AT X0+T*H
C
  600 IF (SLP(1).EQ.0.) GO TO LLF, (200,202,220)
C
C     COMPUTE INTERPOLATION COEFFICIENTS
  610 TMP(1) = ALP(1)
      DO 611 I=1,6
  611 TMP(I+1) = TMP(1)*TMP(I)/FLT(I+1)
      DO 612 I=17,40
  612 STP(I) = 0.
      DO 614 I=1,6
      DO 613 J=1,6
      TMP(8) = CAI(I,J)
      TMP(I+ 8) = TMP(I+ 8)+TMP(8)*TMP(J)
  613 TMP(I+14) = TMP(I+14)+TMP(8)*TMP(J+1)
      TMP(I+ 8) = TMP(I+ 8)/ALP(2)
  614 TMP(I+14) = TMP(I+14)/ALP(3)
C
```

```
C      INTERPOLATE TO X0+T*H
  620 TMP(6) = DX*TMP(1)
      X = X0+TMP(6)
      DO 621 I=1,3
      VDEQ(I) = FDEQ(I,10)
      RDEQ(I) = FDEQ(I, 9)+DX*FDEQ(I,10)
      DO 621 J=1,6
      L = IN(J+2)
      VDEQ(I) = VDEQ(I)+FDEQ(I,L)*TMP(J+ 8)
  621 RDEQ(I) = RDEQ(I)+FDEQ(I,L)*TMP(J+14)
      GO TO 221
C
C**** RETURN             STOP POINT REACHED
C
  999 RETURN
      END
```

Subroutine: DGTRN

Purpose: To compute the transformation matrix corresponding to: (1) a rotation about a given line, or (2) a sequence of rotations about the coordinate axes.

Calling Sequence: CALL DGTRN (T, N, A, M)
CALL DGTRN (T, 0, A, W)

Input and Output

I/Ø	Symbolic Name of Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	N	(M)			Axes of rotation.
I	A	d(M)			Angles of rotation.
I	M				Number of rotations.
I	W	d(3)			Vector along line of rotation.
Ø	T	d(3, 3)			Transformation matrix.

Common storages used: 72 cells

Subroutines required: DGTSN

```
$IBFTC MC132L  XR3,M94,NODD,LIST
      SUBROUTINE DGTRN (T,N,A,M)                                        DGTR0001
      DOUBLE PRECISION  A(10),CA,SA,DSIN,DCOS                           DGTR0002
      COMMON            SAV(32),CA(10),SA(10)                           DGTR0003
    1 MG = 1                                                            DGTR0004
      IF (N.NE.0) MG=M                                                  DGTR0005
      DO 2 I=1,MG                                                       DGTR0006
      CA(I) = DCOS(A(I))                                                DGTR0007
    2 SA(I) = DSIN(A(I))                                                DGTR0008
      CALL DGTSN (T,N,SA,CA,M)                                          DGTR0009
      RETURN                                                            DGTR0010
      END
```

Subroutine: DGTSN

Purpose: To compute the transformation matrix corresponding to: (1) a rotation about a given line, or (2) a sequence of rotations about the coordinate axes.

Calling Sequences: CALL DGTSN (T,N,SA,CA,M)
CALL DGTSN (T,O,SA,CA,W)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	N	(M)			Axes of rotations.
I	SA	d(M)			Sines of angles of rotation.
I	CA	d(M)			Cosines of angles of rotation.
I	M				Number of rotations.
I	W	d(3)			Vector along line of rotation.
Ø	T	d(3,3)			Transformation matrix.

Common storages used: 32 cells

Subroutines required: DVNØRM

```
$IBFTC MC132N  XR3,M94,NODD,LIST
      SUBROUTINE DGTSN (T,N,SA,CA,M)
      DOUBLE PRECISION  CA(10),SA(10),T(9)
     1                 ,ST( 9),WT( 5),S,C
      INTEGER           N(10),NN(4)
      COMMON            SS(18),WT,S,C
      EQUIVALENCE      (SS( 1),ST)
      DATA              NN/6,7,2,3/
C
    1 MG = 1
      ASSIGN 20 TO LL
      IF (N.EQ.0) GO TO 10
      DO 2 I=1,18
    2 SS(I) = 0.
      DO 3 I=1,18,8
    3 SS(I) = 1.
      MG = M
      ASSIGN 30 TO LL
C
   10 DO 35 I=1,MG
      C = CA(I)
      S = SA(I)
      GO TO LL, (20,30)
C
   20 CALL DVNORM(M,WT)
      L = 1
      WT(4) = 1.-C
      DO 21 J=1,3
      WT(5) = WT(4)*WT(J)
      DO 21 K=1,3
      ST(L) = WT(5)*WT(K)
   21 L = L+1
      DO 22 J=1,9,4
   22 ST(J) = ST(J)+C
      DO 23 J=1,3
      K = NN(J)
      L = NN(J+1)+1
      ST(K) = ST(K)-S*WT(J)
   23 ST(L) = ST(L)+S*WT(J)
      GO TO 35
C
   30 J = IABS(N(I))+1
      IF (N(I).LT.0) S=-S
      IF (J.GT.3) J=1
      K = J+1
      IF (K.GT.3) K=1
      DO 31 L=1,3
      WT(4) = ST(J)
      ST(J) = C*WT(4)+S*ST(K)
      ST(K) =-S*WT(4)+C*ST(K)
      J = J+3
   31 K = K+3
   35 CONTINUE
      DO 36 I=1,9
   36 T(I) = ST(I)
      RETURN
      END
```

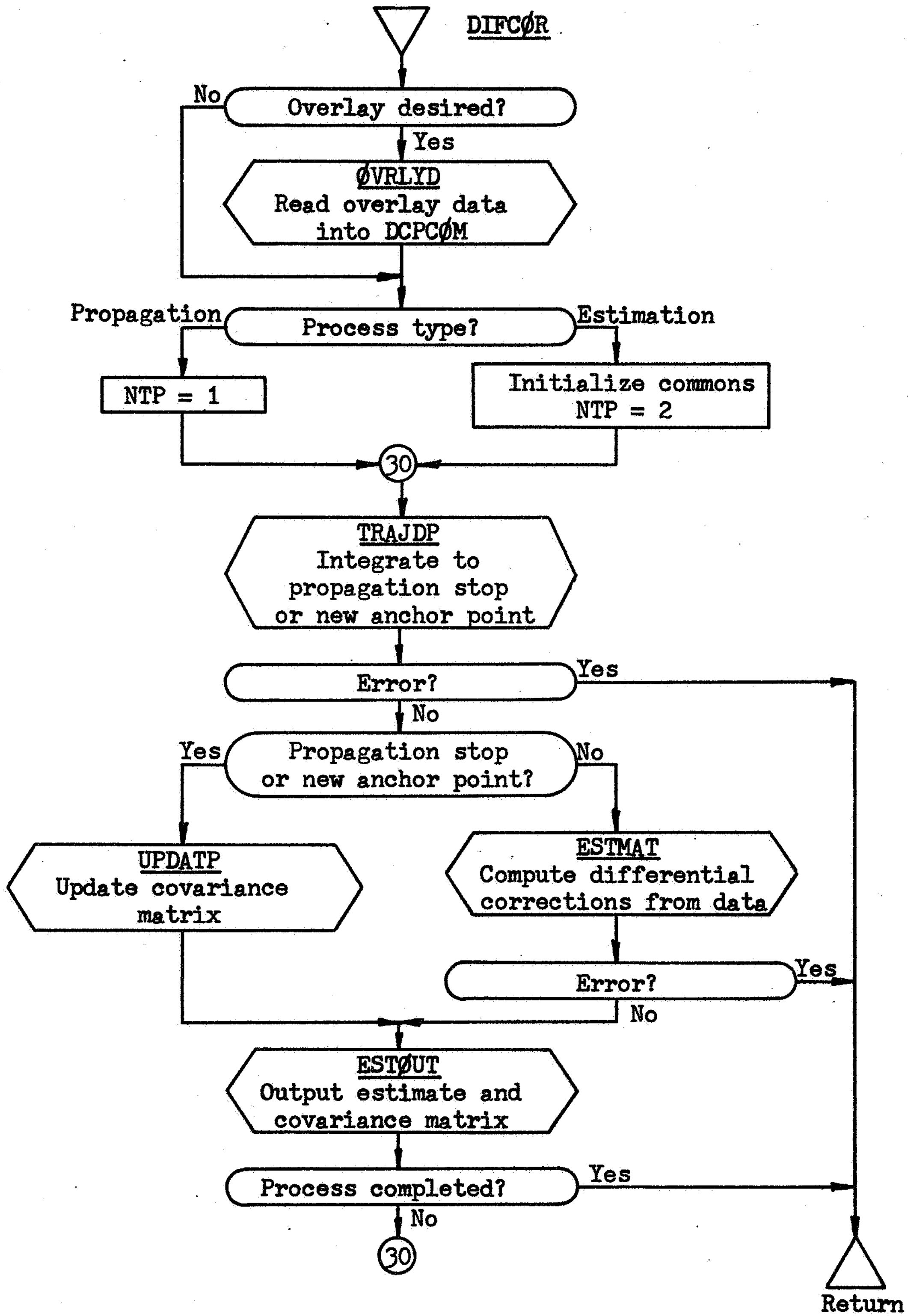
Subroutine: DIFCØR

Purpose: To control integration of the nominal trajectory and the accumulation of differential corrections to the estimate.

Calling Sequence: CALL DIFCØR

Common storages used: /DCPCØM/,/DCRCØM/,/SBFCØM/

Subroutines required: ESTMAT,ESTØUT,ØVRLYD,TRAJDP,UPDATP


DIFCØR
Overlay desired?
No
Yes
ØVRLYD
Read overlay data into DCPCØM
Process type?
Propagation
Estimation
NTP = 1
Initialize commons NTP = 2
30
TRAJDP
Integrate to propagation stop or new anchor point
Error?
Yes
No
Propagation stop or new anchor point?
Yes
No
UPDATP
Update covariance matrix
ESTMAT
Compute differential corrections from data
Error?
Yes
No
ESTØUT
Output estimate and covariance matrix
Process completed?
Yes
No
30
Return

```
$IBFTC MC133H  XR3,M94,NODD,LIST
      SUBROUTINE DIFCOR
C     CONTROL ROUTINE FOR DCP ESTIMATION/PROPAGATION LINKS
C
      COMMON             /DCPCOM/CDCP(900)
      DIMENSION           STIMNX(20)
      DOUBLE PRECISION    STIMR(2)
      EQUIVALENCE        (CDCP(111),IERR   ) ,(CDCP(740),NPREST)
     1                  ,(CDCP(758),INDSTA) ,(CDCP(732),NPROVR)
     2                  ,(CDCP(756),IPROCS) ,(CDCP(123),STIMNX)
     3                  ,(CDCP(105),MAXZ   ) ,(CDCP(781),STIMR )
C
      COMMON             /SBFCOM/CSBF(12),RBF(45),VBF(45,8)
      DOUBLE PRECISION    RBF,VBF
      EQUIVALENCE        (CSBF(2),NTP)
C
      COMMON             /DCRCOM/ZBAR(30),ZHAT(30),ZTOT(30),DCSAVE(74)
      DOUBLE PRECISION    ZBAR ,ZHAT ,ZTOT
C
      LOGICAL             LTRAJ
C
  601 FORMAT(1H0//45X,24H*** PROPAGATION LINK ***/1H )
  602 FORMAT(1H0//45X,23H*** ESTIMATION LINK ***/1H )
  603 FORMAT(1H0//43X,29H*** EXIT PROPAGATION LINK ***)
  604 FORMAT(1H0//43X,28H*** EXIT ESTIMATION LINK ***)
C
C     READ OVERLAY DATA
    1 IF (NPROVR.NE.0) CALL OVRLYD(CDCP)
      GO TO (10,20) ,IPROCS
C
C     PROPAGATE
   10 NTP = 1
      WRITE (6,601)
      GO TO 30
C
C     ESTIMATE
   20 CONTINUE
      DO 21 I=1,MAXZ
      ZBAR(I) = 0.D0
      ZHAT(I) = 0.D0
   21 ZTOT(I) = 0.D0
      DO 22 I=1,6
   22 CSBF(I) = 0.
      NTP = 2
      LTRAJ = .FALSE.
      WRITE (6,602)
C
C     INTEGRATE
   30 CALL TRAJDP
      IF (IERR.NE.0) GO TO 999
      IF (NTP.GE.3) GO TO 33
   31 CALL UPDATP
      NPREST = 0
   32 CALL ESTOUT
      IF (NTP.EQ.1) GO TO 950
      IF (NTP.EQ.4) GO TO 900
      NTP = 2
      GO TO 30
   33 CALL ESTMAT (LTRAJ)
      IF (IERR.NE.0) GO TO 999
      GO TO 32
C
  900 STIMNX(INDSTA) = STIMR(2)
      WRITE (6,604)
      GO TO 999
  950 WRITE (6,603)
  999 RETURN
      END
```

Subroutine: DINVRT

Purpose: To invert a double precision matrix of any dimension, possibly stored as a submatrix of a larger array. Replaces the matrix by its inverse.

Calling Sequence: CALL DINVRT(A,N,M)

Input and Output:

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
IØ	A	d(M,ℓ)			I = matrix to be inverted. Ø = inverse of input matrix.
I	N				Dimension of matrix to be inverted.
I	M				Dimension of storage array.

Common storages used: 7 + N

Subroutines required: None

Method:

Reference: F.B. Hildebrand, "Introduction to Numerical Analysis," McGraw-Hill, 1956.

The inverse of the n × n matrix $A = \{Aij\}$ is the solution, $X = A^{-1}$, of the matrix equation

$$AX = I \qquad (1)$$

It is obtained by premultiplying both sides of (1) by a sequence of matrices chosen to eliminate the matrix from the left-hand side.

The sequence of matrices used in this subroutine constitute a Gauss-Jordan reduction. Storage is shared between the reduced matrix and its inverse during the inversion.

We define an augmented matrix,

$$M = (A \,\vdots\, I) \qquad (2)$$

We may place 1 on the main diagonal and null the remaining elements of the first column by premultiplying by the matrix

$$E_1 = \left[\begin{array}{c|c} {}^1/a_{11} & \bar{0} \\ \hline -a_{21}/a_{11} & \\ -a_{31}/a_{11} & \\ \vdots & I \\ -a_{n1}/a_{11} & \end{array}\right] \qquad (3)$$

Similarly, the second column diagonal element is replaced by 1 and the remaining elements of the column are nulled by premultiplication by

$$E_2 = \left[\begin{array}{c|c|c} 1 & -a'_{12}/a'_{22} & 0 \\ \hline 0 & 1/a'_{22} & 0 \\ \hline & -a'_{32}/a'_{22} & \\ & \vdots & \\ \bar{0} & -a'_{n2}/a'_{22} & I \end{array}\right] \qquad (4)$$

where primes denote the elements resulting from the previous premultiplication.

We proceed columnwise across A in this fashion, obtaining finally an auxiliary matrix,

$$M' = C' (A \vdots I) = (A' \vdots C)$$
$$C' = E_n E_{n-1} \cdot \cdot \cdot E_2 E_1 \qquad (5)$$

The determinant of the matrix A is the product of the divisors of the E_j, as may be seen from

$$| E_j | = {}^1/a'_{jj}$$
$$1 = | I | = | E_n | \, | E_{n-1} | \, . \, . \, . \, | E_1 | \, | A | \qquad (6)$$

The principal disadvantage of this process as described is the use of $2n^2$ storage locations for the augmented matrix. The process is however, easily modified to share storage between the reduced matrix A' and the uncompleted inverse.

Since the columns preceding the j^{th} have non-zero elements only on the main diagonal before the j^{th} premultiplication, they are not changed during that premultiplication. Similarly, the columns after the j^{th} in the right half of the augmented matrix are not changed. We need not, then, store the columns known to contain 0's and 1's and both right and left matrices may be stored in the single n X n array, with appropriate attention to the jth row.

We obtain the process

$$
\begin{aligned}
a'_{ij} &= \begin{cases} 1/a'_{jj} & i = j \\ -a'_{ij} & i \neq j \end{cases} \\
a'_{ik} &= a'_{ik} + a'_{ij}a'_{jk} \qquad i, k \neq j \\
a'_{ji} &= a'_{ji} \cdot a'_{jj} \qquad i \neq j
\end{aligned}
\tag{7}
$$

repeated for j = 1, 2, . . ., n. Again, after n steps, $X = A^{-1}$ occupies the spaces originally occupied by A.

Finally, we note that maximum numerical accuracy is obtained if the diagonal element a_{ji} has the largest magnitude of the elements remaining in the unused rows and columns in A'. Then, if on the j^{th} step, a'_{ik} is the element of largest magnitude, we may eliminate the k^{th} column by placing a 1 in a'_{ik} by the process

$$
\begin{aligned}
a'_{\ell k} &= \begin{cases} 1/a'_{ik} & \ell = i \\ -a'_{\ell m}/a'_{ik} & \ell \neq i \end{cases} \\
a'_{\ell m} &= a'_{\ell m} + a'_{\ell k}\, a'_{ik} \qquad \ell \neq i,\ m \neq k \\
a'_{i\ell} &= a'_{i\ell} \qquad \ell \neq k
\end{aligned}
\tag{8}
$$

Although the column available for storage is the k^{th}, the column of the right matrix stored there is its i^{th}. In addition, the resulting left half is a matrix with ones on each row and column, but not necessarily on the diagonal. It is necessary to record the locations of the 1's and to use this record to reorder the rows and columns of the inverse.

```
$IBFTC MC13MH  XR3,M94,NODD,LIST
      SUBROUTINE DINVRT(A,N,M)
C     REPLACES THE N BY N MATRIX A BY ITS INVERSE.
C     A IS STORED IN THE CALLING PROGRAM AS A SUBMATRIX OF AN M BY M
C       ARRAY.
C
      DOUBLE PRECISION A,T,S
      DIMENSION  A(1)
      COMMON     T,S,ND,NM,NI,IN(20)
    1 CONTINUE
      NI=N
      NM=M
      IF(NI)999,999,2
    2 ND=NM-NI
      NN=NI*NM
      DO 3 J=1,NI
    3 IN(J)=0
   10 DO 29 L=1,NI
      T=0.0
      K=1
      DO 16 J=1,NI
      DO 12 I=1,NI
      IF(IN(I)-J)12,11,12
   11 K=K+NM
      GO TO 16
   12 CONTINUE
      DO 15 I=1,NI
      IF(IN(I))15,13,15
   13 IF( DABS(A(K))-DABS(T) ) 15,15,14
   14 ICOL=J
      IROW=I
      K1=K
      T=A(K)
   15 K=K+1
      K=K+ND
   16 CONTINUE
   20 IN(IROW)=ICOL
      A(K1)=1.0
   21 K=K1-IROW
      DO 24 I=1,NI
      K=K+1
      IF(IROW-I)22,24,22
   22 S=-A(K)/T
      A(K)=0.0
      J1=I
      DO 23 J=IROW,NN,NM
      A(J1)=A(J1)+S*A(J)
   23 J1=J1+NM
   24 CONTINUE
      DO 25 J=IROW,NN,NM
   25 A(J)=A(J)/T
   29 CONTINUE
   30 DO 39 I=1,NI
      DO 31 J=I,NI
      IF(IN(J)-I)31,32,31
   31 CONTINUE
   32 IF(J-I)33,39,33
   33 CONTINUE
      IN(J)=IN(I)
      J1 = (IN(I)-1)*NM
      K1=(I-1)*NM
      DO 34 K=I,NN,NM
      S=A(K)
      A(K)=A(J)
      A(J)=S
   34 J=J+NM
      DO 35 K=1,NI
      K1=K1+1
      J1=J1+1
      S=A(K1)
      A(K1)=A(J1)
   35 A(J1)=S
   39 CONTINUE
  999 CONTINUE
      RETURN
      END
```

Subroutine: DLUNE

Purpose: To compute the transformation from Earth's true equator equinox of date to Moon-fixed coordinates and the Moon's angular velocity.

Calling Sequence: CALL DLUNE (T,E,DE,DL,TR,ØM)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T	d	t_E	sec	Time from 1950 Jan 0.0 ET
I	E	d	$\bar{\epsilon}$	rad	Mean obliquity of the ecliptic
I	DE	d	$\delta\epsilon$	rad	Nutation in obliquity
I	DL	d	δL	rad	Nutation in longitude
Ø	TR	d(3,3)	T_{D2B}		Transformation matrix
Ø	ØM	d	ω_M	rad/sec	Moon's angular velocity

Common storages used: 72 cells

Subroutines required: DGTRN

Notation

Symbol	Quantity
L	Longitude of the Sun, from the equinox of date.
Γ	Longitude of perigee of the Sun.
ϵ	Obliquity of the ecliptic.
L'	Longitude of the Moon, measured in the ecliptic from the mean equinox of date to the mean ascending node of the lunar orbit, and then along the orbit.
Γ'	Longitude of perigee of the Moon, measured as L'.
Ω	Longitude of the ascending node of the Moon's orbit on the ecliptic.
I	Inclination of the Moon's equator with the ecliptic.
φ'	Right ascension of the Moon's prime meridian, from the ascending node of the equator on the ecliptic.
$\bar{(\ )}$	Mean ( )
$\delta(\ )$	Nutation or libration in ( ).
ω_M	Moon's angular velocity about its polar axis.

Method

The transformation from Earth's equator, equinox of date to Moon-fixed coordinates, T_{D2B}, may be written

$$T_{D2B} = T_3 (\Upsilon') \; T_1 (I) \; T_3 (\Omega + \delta L - 180) \; T_1 (\epsilon)$$

The nutations δL, $\delta\epsilon$ and the mean obliquity $\bar{\epsilon}$ are input quantities. The librations δI, δL, $\delta\Omega$ are

$$\delta I = 0.^\circ0297222 \cos g_2 - 0.^\circ00305555 \cos g_3 + 0.^\circ0102777 \cos g_4$$

$$\delta L = 0.^\circ0163888 \sin g_1 - 0.^\circ003333 \sin g_2 + 0.^\circ005 \sin g_3$$

$$\delta\Omega = \csc \bar{I} \; \{ -0.^\circ0302777 \sin g_2 - 0.^\circ00305555 \sin g_3 + 0.^\circ0102777 \sin g_4 \}$$

where

$$g_1 = \bar{L} - \bar{\Gamma}$$

$$g_2 = \bar{L}' - \bar{\Gamma}'$$

$$g_3 = 2\bar{\Gamma}' - 2\bar{\Omega}$$

$$g_4 = g_2 + g_3 = \bar{L}' + \bar{\Gamma}' - 2\bar{\Omega}$$

The right ascension Υ' is

$$\Upsilon' = L' - \Omega = (\bar{L}' - \bar{\Omega}) + \delta L' - \delta\Omega$$

and the polar component of angular velocity is

$$\omega_M = \dot{L}' - \dot{\Omega} + \dot{\Omega} \cos I$$

$$= \dot{L}' - \dot{\Omega} (1-\cos I)$$

From reference 3, pp. 98, 107, the mean ephemerides of the Sun and Moon, reduced to the epoch T_{E1}, yield

$$\bar{I} = 1^{\circ}.535$$

$$\bar{\Omega} = 12^{\circ}.1127905556 - 1934^{\circ}.1440877778 T_E + 0^{\circ}.002081 T_E^{\ 2} + 0^{\circ}.22\text{x}10^{-5} T_E^{\ 3}$$

$$g_1 = 1^{\circ}.9993186111 + 129596578''.550 T_E - 0''.559 T_E^{\ 2} - 0''.012 T_E^{\ 3}$$

$$g_2 = - 144^{\circ}.4685381556 + 477198^{\circ}.6450208333 T_E + 0^{\circ}.0092136 T_E^{\ 2} + 1^{\circ}.439\text{x}10^{-5} T_E^{\ 3}$$

$$g_3 = 33^{\circ}.4623977778 + 12006^{\circ}.7621433333 T_E - 0^{\circ}.02485 T_E^{\ 2} - 2^{\circ}.94\text{x}10^{-5} T_E^{\ 3}$$

$$g_4 = g_2 + g_3$$

$$\bar{\varphi}' = g_2 + \frac{1}{2} g_3$$

The Moon's angular velocity is computed from

$$\bar{L}' = .2661702799661347\text{x}10^{-5} - .12505386\text{x}10^{-13} T_E$$

$$\bar{\Omega} = -.1069700561503563\text{x}10^{-7} + .23018226\text{x}10^{-13} T_E$$

$$\delta L' = .569493743\text{x}10^{-10} \cos g_1 - .53527184\text{x}10^{-9} \cos g_2 + .579490975\text{x}10^{-11} \cos g_3$$

$$\delta\Omega = -.520641836\text{x}10^{-7} \cos g_2 - .132200251\text{x}10^{-9} \cos g_3 + .188410597\text{x}10^{-7} \cos g_4$$

```
$IBFTC MC1320  XR3,M94,NODD,LIST
      SUBROUTINE DLUNE (T,E,DE,DL,TR,OM)
      DOUBLE PRECISION  DE,DL,E,OM,T,TR
     1                  ,A(6,4),B(15),C,DSIN,DCOS
      INTEGER           N(4)
      COMMON            C(36)
      DATA  A/-.58  D-7,0.D0, .251 D-7,0.D0,-.513 D-7, .38  D-7
     1        ,-.271 D-5,0.D0, .1608D-4,0.D0,-.4337D-4, .3632D-4
     2        , .6283019430602663OD+03,-.12505386           D-13
     3        , .8328687541669l855D+04, .23018226           D-13
     4        , .2095575310497740 9D+03,-.3375718243970482 5D+02
     5        ,-.34894692560095     D-01, .2661702799661347 D-05
     6        ,-.2521451656358276 D+01,-.1069700561503563 D-07
     7        , .5840290l6834644        ,-.3353000730389487 D+01    /
      DATA  B/ .569493743D-10, .28603852 D-03,-.520641836D-07
     1        ,-.19727246 D-01,-.53329408 D-04,-.153527184D-09
     2        ,-.58171824 D-04,-.132200251D-09,-.19908245 D-02
     3        , .17937970 D-03, .579490975D-11, .87266463 D-04
     4        , .188410597D-07, .66963710 D-02,-.51875025 D-03    /
      DATA  N/-3,-1,-3,-1/
C
    1 C(1) = T/3.15576D+09
      DO 2 I=1,6
    2 C(I+1) = A(I,1)
      DO 3 I=1,3
      DO 3 J=1,6
    3 C(J+1) = A(J,I+1)+C(1)*C(J+1)
      C(8) = C(4)+C(6)
      DO 4 I=1,7,2
      C(1) = C(I+1)
      C(I+ 8) = DCOS(C(1))
    4 C(I+12) = DSIN(C(1))
      C(17) = C(11)
      DO 5 I=1,5
    5 C(I+17) = B(I)*C(I+8)+B(I+5)*C(I+10)+B(I+10)*C(I+12)
      C(6) = C(4)+C(19)-C(21)+C(6)/2.D0
      C(3) = C(3)+C(18)
      C(5) = C(5)+C(20)
      C(8) = C(7)+C(21)+DL
      C(9) = E+DE
      C(7) = .0267908040181l2958+C(22)
      OM = C(3)-C(5)+C(5)*DCOS(C(7))
      CALL DGTRN (TR,N,C(6),4)
  999 RETURN
      END
```

Subroutine: DMPLY

Purpose: To form the product of any two double precision matrices for which the product is defined.

Calling Sequence: CALL DMPLY(A,B,C,L,M,N,NA,NB,NC,IP)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d(NA,ℓ)			Pre-matrix.
I	B	d(NB,m)			Post-matrix.
Ø	C	d(NC,n)			Product matrix.
I	L,M,N				Product dimensions.
I	NA,NB,NC*				Number of rows in storage arrays, A,B,C.
I	IP				Option index.

Common storages used: None

Subroutines required: None

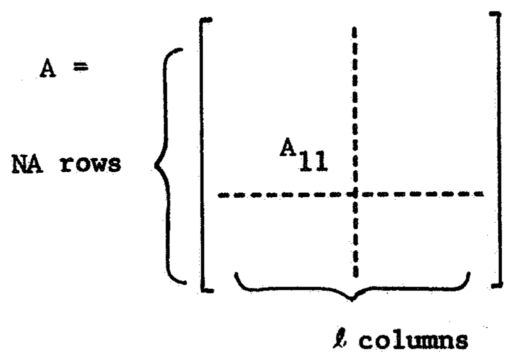
*Restrictions. The number of rows of storage (NA,NB,NC) must be not less than the number of rows used in the product (L,M,N, respectively).

The subroutine allows multiplication of matrices stored as submatrices of large arrays. This allows operation with variable-dimensioned matrices within fixed dimensioned programs, and operations with partitioned matrices.

The options index, IP, allows transposing either or both of the factor matrices before multiplication. The subroutine output is

IP = 0: $C = AB$
IP = 1: $C = A^T B$
IP = 2: $C = AB^T$
IP = 3: $C = A^T B^T$

The storage arrays are assumed to be partitioned in the form



and C_{11} is the product of A_{11} and B_{11}. Other submatrices in the arrays are ignored. The product dimensions are L × N, with M terms in each summation. That is, the pre-matrix, A_{11} or its transpose, is L × M, and the post-matrix B_{11} or its transpose, is M × N.

For example, the call

```
CALL MTMPLY (A,B,C,5,3,2,7,5,6,0)
```

forms the product $C_{11} = A_{11} B_{11}$, where A_{11}, B_{11}, C_{11} are 5 X 3, 3 X 2, 5 X 2 matrices stored in 7 X ℓ, 5 X m, 6 X n arrays. Similarly

CALL MTMPLY (A,B,C,5,3,2,7,5,6,1)

forms the product $C_{11} = A_{11}^{T} B_{11}$, where A_{11}, B_{11}, C_{11} are 3 X 5, 3 X 2, 5 X 2 matrices stored in 7 X ℓ, 5 X m, 6 X n arrays.

```
$IBFTC MC13MO  XR3,M94,NODD,LIST
      SUBROUTINE DMPLY(A,B,C,NRA,NRB,NCB,NA,NB,NC,IP)
      DIMENSION  A(1),B(1),C(1)
      DOUBLE PRECISION A,B,C
    1 IF(IP-1)2,2,3
    2 ND=1
      ND1=NB
      IF(IP)4,4,5
    3 ND=NB
      ND1=1
      IF(IP-2)4,4,5
    4 MD=NA
      MD1=1
      GO TO 6
    5 MD=1
      MD1=NA
    6 N1=1
      L1=1
      DO 9 J=1,NCB
      L=L1
      M1=1
      DO 8 I=1,NRA
      C(L)=0.0
      M=M1
      N=N1
      DO 7 K=1,NRB
      C(L)=C(L)+A(M)*B(N)
      M=M+MD
    7 N=N+ND
      M1=M1+MD1
    8 L=L+1
      N1=N1+ND1
    9 L1=L1+NC
 9999 RETURN
      END
```

Subroutine: DMVTRN

Purpose: To compute the matrix product of a 3 x 3 matrix and a 3 x N matrix in double precision.

Calling Sequence: CALL DMVTRN (A, B, C, M, N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d(3, 3)			Pre-matrix.
I	B	d(3, N)			Post-matrix.
Ø	C	d(3, N)			Matrix product, AB or A^TB.
I	M				=1: Pre-matrix is A. =2: Pre-matrix is A^T.
I	N				Number of columns of B, C.

Common storages used: None

Subroutines required: None

```
$IBFTC MC132J  XR3,M94,NODD,LIST
      SUBROUTINE DMVTRN (A,B,C,M,N)
      DOUBLE PRECISION   A(9),B(9),C(9)
      INTEGER            NN(4)
      DATA               NN/3,1,8,0/
C
    1 I1 = 1
      J1 = 1
      K1 = 1
      DO 4 I=1,N
      DO 3 J=1,3
      C(I1) = 0.D0
      DO 2 K=1,3
      C(I1) = C(I1)+A(J1)*B(K1)
      J1 = J1+NN(M)
    2 K1 = K1+1
      I1 = I1+1
      J1 = J1-NN(M+2)
    3 K1 = K1-3
      K1 = K1+3
    4 J1 = 1
      RETURN
      END
```

Subroutine: DNØRM

Purpose: Compute the magnitude of a vector.

Calling Sequence: Z = DNØRM (X)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	d(3)			Vector
Ø	DNØRM	d			Magnitude of X.

Common storages used: None

Subroutines required: DDØT

```
$IBFTC MC132F  XR3,M94,NODD,LIST
      DOUBLE PRECISION FUNCTION DNORM (X)                               DNRM0001
      DOUBLE PRECISION          DDOT,DSQRT                              DNRM0002
    1 DNORM = DSQRT(DDOT(X,X))                                          DNRM0003
      RETURN                                                            DNRM0004
      END
```

Subroutine: DØT

Purpose: Computes the inner (dot) product of two 3-vectors.

Calling Sequence: Z = DØT (X,Y)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X, Y	(3)			Input vectors
Ø	Z				X • Y

Common storages used: None

Subroutines required: None

```
$IBFTC DOT      XR3,M94,NODD
      FUNCTION DOT  (X,Y)                                    DOT 0001
      DIMENSION  X(3) ,Y(3)                                  DOT 0002
      SUM = 0.                                               DOT 0003
      DO 1 I=1,3                                             DOT 0004
    1 SUM = SUM+X(I)*Y(I)                                    DOT 0005
      DOT = SUM                                              DOT 0006
      RETURN                                                 DOT 0007
      END                                                    DOT 0008
```

Subroutine: DPFMRS

Purpose: To compute planetary positions and velocities.

Calling Sequence: CALL DPFMRS (ET,UCB,ICB,NERR,T)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ET	d	t_E	sec	Ephemeris time, seconds from 1950 January 0.0.
I	UCB	d(14)			Gravitation constants and ephemeris scale factors.
I	ICB				Central body number
Ø	NERR				Error flag.
IØ	T	(4)			Ephemeris tape position indicators.

Common storages used: //22 cells, /DFMCØM/

Subroutines required: None

Reference: "User's Description of JPL Ephemeris Tapes," JPL-TR32-580, P.R. Peabody, J.F. Scott, E.G. Orozco, March 2, 1964.

1. /DFMCØM/CFM(2232) (See paragraph 3, below.)

Location	Symbol	Dimension	Description
CFM(1)	IFM	14	Output options (input) IFM(i) = 0 ignore i^{th} body 1 output position only 2 output position and velocity
(15)	RFM	d(6,12)	RFM(1-3,i) = i^{th} body position wrt central body RFM(4-6,i) = i^{th} body velocity wrt central body
(159)	DFM	d(4)	DFM(1-2) = nutations in longitude and obliquity DFM(3-4) = nutation rates
(167)	BFM	d(829)	Position-velocity buffer
(1825)	SNT	2,204	Nutation buffer

2. Usage:

The planets are numbered outward from the sun. That is

ICB = 1	Mercury	6	Saturn
2	Venus	7	Uranus
3	Earth	8	Neptune
4	Mars	9	Pluto
5	Jupiter		

In addition,

ICB = 10 Sun
11 Moon
12 Earth-Moon Barycenter

Output of the nutations is keyed by IFM(13). The array IFM must be loaded prior to the first call of DPFMRS. If either the Earth or Moon are keyed for output, it is necessary that both IFM(11),IFM(12) be not less than the maximum of IFM(3),IFM(11),IFM(12). In addition, IFM(ICB) must be not less than the maximum of IFM(i), i = 1,2, . . . , 12.

The array UCB contains, in components 1-11, the gravitational constants of the bodies. UCB(12) is computed from

$$\mu_{12} = \mu_3 + \mu_{11}$$

The heliocentric positions and velocities of the planets, except Earth, and the Barycenter are computed in a.u. and a.u./day, and the geocentric position and velocity of the Moon are computed in earth radii and earth radii/day. For output, they are scaled into km and km/sec using

UCB(13) = km/a.u.

UCB(14) = km/earth radius

The interpolation formulas are described in the reference.

The array T is used for maintaining knowledge of the ephemeris tape position. Overlay of T will cause unnecessary rewinding of the ephemeris tape. For the initial call of DPFMRS, T(3) must be not greater than 0.

3. Reduced Ephemeris

The dimensions in paragraph 1 are used in decks MC1322(DPFMRS) and MC1324(DFMCØM) is also supplied. The reduced ephemeris is interchangeable with the full ephemeris with the following exceptions:

1. Positions and velocities of Mercury, Uranus, Neptune and Pluto are not computed. The 1, 7, 8, 9 components of IFM are ignored.

2. IFM(13) <u>must</u> <u>not</u> <u>be</u> 2; i.e., no nutation rates may be requested.

3. The dimensions of CFM are

CFM	(1524)
BFM	d(577)
SNT	(2,102)

```
$IBFTC MC1322  XR3,M94,NODD,LIST
      SUBROUTINE DPFMRS (ET,UCB,ICB,NERR,T)
C     PLANETARY EPHEMERIS
C     USES JPL EPHEMERIS TAPE SYSTEM
C
      DOUBLE PRECISION   ET,UCB(14)
      DIMENSION          T(4)
C
      DOUBLE PRECISION   BFM,DFM,RFM
      DIMENSION          BUFEFM(1658)
      COMMON            /DFMCOM/IFM(14),RFM(6,12)
     1                         ,DFM (4),BFM (829),SNT(2,204)
      EQUIVALENCE       (BFM(1),BUFEFM,TBUF)
C
      DOUBLE PRECISION   D(11)
      COMMON             S(22)
      EQUIVALENCE       (S(1),D)
C
      DIMENSION          DEL(3),IDF(3),IDL(13,2)
      DATA  DEL/ 2., 4., .5
     1    /,IDF/ 45, 27,153
     2    /,IDL/  1,  2,  2,  2,  2,  2,  2,  2,  2,  0,  3,  0,  3
     3          ,  2, 92,146,200,254,308,362,416,470,  0,524,  0,830
     4    /
C
C     INITIALIZATION AND ERROR CHECK
    1 D(1) = ET/86400.D0+2433282.5D0
      JCB  = ICB
      IF (T(3).GT.0.)  GO TO 4
    3 REWIND 8
      READ (8) (BUFEFM(I),I=1,24)
      READ (8) T
      T(1) = T(3)-8.
      T(2) = T(3)
    4 IERR = 2
      IF (S(1).GT.T(4)) GO TO 999
      IERR = 1
      IF (S(1).LT.T(3)) GO TO 999
      IERR = 0
C
C     CHECK BUFFER AND READ IN CORRECT 8-DAY RECORD
   10 S(3) = S(1)-T(1)
      IF (S(3).GE.0.) GO TO 12
      IF (S(1).LT.T(2)) GO TO 3
      IKK = DEL(1)-S(3)/8.
      DO 11 I=1,IKK
   11 BACKSPACE 8
      GO TO 15
   12 IF (S(3).LT.8.) GO TO 14
      IKK  = S(3)/8.-1.
      IF (IKK.EQ.0) GO TO 15
      DO 13 I=1,IKK
   13 READ (8) TBUF
      GO TO 15
   14 IF (TBUF.EQ.T(1)) GO TO 16
      BACKSPACE 8
   15 READ (8) BUFEFM,(SNT(1,I),I=1,204)
      T(1) = TBUF
      T(2) = TBUF+(T(3)-TBUF)/4.
   16 CONTINUE
      ILL = 1
   17 IKK = IFM(3)
      IFM( 3) = IFM(12)
      IFM(12) = IKK
      DO 18 I=1,6
      D(2) = RFM(I,3)
      RFM(I, 3) = RFM(I,12)
   18 RFM(I,12) = D(2)
      GO TO (20,45) ,ILL
C
C     INTERPOLATION CONTROL
   20 D(2) = D(1)-BFM(1)
      ILL = 4
      NNN = 3
      D(10) = UCB(13)
```

```
      D(11) = D(10)/86400.D0
      DO 39 I=1,13
      IKK = IFM(I)
      IF (IKK.EQ.0) GO TO 39
      IF (I.EQ.13)  GO TO 22
      IF (I.EQ.11)  GO TO 21
      IF (ILL.EQ.IDL(I,1)) GO TO 30
      GO TO 25
   21 D(10) = UCB(14)
      D(11) = D(10)/86400.D0
      GO TO 25
   22 D(10) = 1.D0
      NNN = 2
      IF (ILL.EQ.3) GO TO 30
C
C     COMPUTE INTERPOLATION COEFFICIENTS
   25 ILL = IDL(I,1)
      IF (ILL.EQ.0) GO TO 39
      ILC = S(3)/DEL(ILL)
      D(6) = D(2)-DEL(ILL)*FLOAT(ILC)
      D(3) = 1.D0-D(6)
      DO 26 J=3,6,3
      D(8) = D(J)*D(J)-1.D0
      D(J+1) = D(J)*D(8)/6.D0
   26 D(J+2) = D(J+1)*(D(8)-3.D0)/20.D0
C
C     INTERPOLATE
   30 LOC  = IDL(I,2)+3*NNN*ILC
      IMM = LOC
      INN = IMM+3*NNN
      DO 32 K=1,NNN
      D(9) = 0.D0
      DO 31 J=3,5
      D(9) = D(9)+D(J)*BFM(IMM)+D(J+3)*BFM(INN)
      IMM = IMM+1
   31 INN = INN+1
   32 RFM(K,I) = D(10)*D(9)
      IF (IKK.EQ.1) GO TO 39
   35 IMM = LOC+IDF(ILL)
      IF (I.EQ.13) IMM=IMM-51
      INN = IMM+3*NNN
      DO 38 K=1,NNN
      D(9) = 0.D0
      DO 36 J=3,5
      D(9) = D(9)+D(J)*BFM(IMM)+D(J+3)*BFM(INN)
      IMM = IMM+1
   36 INN = INN+1
      IF (I.EQ.13) GO TO 37
      RFM(K+3,I) = D(11)*D(9)
      GO TO 38
   37 RFM(K+2,I) = D(9)
   38 CONTINUE
   39 CONTINUE
C
C     EARTH/EARTH-MOON BARYCENTER
   40 UCB(12) = UCB(3)+UCB(11)
      IF (IFM(11).EQ.0) GO TO 42
      D(3) = UCB( 3)/UCB(12)
      D(4) = UCB(11)/UCB(12)
      ILL = 3*IFM(11)
      DO 41 I=1,ILL
      RFM(I,12) = RFM(I,3)-D(4)*RFM(I,11)
   41 RFM(I,11) = RFM(I,3)+D(3)*RFM(I,11)
   42 ILL = 2
      GO TO 17
C
C     SUN
   45 IF (IFM(10).EQ.0) GO TO 50
      ILL = 3*IFM(10)
      DO 46 I=1,ILL
   46 RFM(I,10) = 0.D0
C
C     SHIFT BODY CENTERS
   50 IF (JCB.EQ.10) GO TO 999
      DO 51 I=1,6
```

```
   51 D(I) = RFM(I,JCB)                                              FMRS0150
      DO 53 I=1,12                                                   FMRS0151
      IF (IFM(I).EQ.0) GO TO 53                                      FMRS0152
      ILL = 3*IFM(I)                                                 FMRS0153
      DO 52 J=1,ILL                                                  FMRS0154
   52 RFM(J,I) = RFM(J,I)-D(J)                                       FMRS0155
   53 CONTINUE                                                       FMRS0156
C                                                                    FMRS0157
  999 NERR = IERR                                                    FMRS0158
      RETURN                                                         FMRS0159
      END
```

```
$IBFTC MC1342  XR3,M94,NODD,LIST
      SUBROUTINE DPFMRS (ET,UCB,ICB,NERR,T)
C     REDUCED PLANETARY EPHEMERIS
C     USES JPL EPHEMERIS TAPE SYSTEM
C
      DOUBLE PRECISION   ET,UCB(14)
      DIMENSION          T(4)
C
      DOUBLE PRECISION   BFM,DFM,RFM
      DIMENSION          BUFEFM(1154)
      COMMON            /DFMCOM/IFM(14),RFM(6,12)
     1                      ,DFM (4),BFM (577),SNT(2,102)
      EQUIVALENCE       (BFM(1),BUFEFM,TBUF)
C
      DOUBLE PRECISION   D(11)
      COMMON             S(22)
      EQUIVALENCE       (S(1),D)
C
      DIMENSION          DEL(3),IDF(3),IDL(13,2)
      DATA  DEL/ 2., 4., .5
     1    /,IDF/ 45, 27,153
     2    /,IDL/  0,  2,  2,  2,  2,  2,  0,  0,  0,  0,  3,  0,  3
     3          ,  0,  2, 56,110,164,218,  0,  0,  0,  0,272,  0,578
     4    /
C
C     INITIALIZATION AND ERROR CHECK
    1 D(1) = ET/86400.D0+2433282.5D0
      JCB  = ICB
      IF (T(3).GT.0.)  GO TO 4
    3 REWIND 8
      READ (8) (BUFEFM(I),I=1,24)
      READ (8) T
      T(1) = T(3)-8.
      T(2) = T(3)
    4 IERR = 2
      IF (S(1).GT.T(4)) GO TO 999
      IERR = 1
      IF (S(1).LT.T(3)) GO TO 999
      IERR = 0
C
C     CHECK BUFFER AND READ IN CORRECT 8-DAY RECORD
   10 S(3) = S(1)-T(1)
      IF (S(3).GE.0.) GO TO 12
      IF (S(1).LT.T(2)) GO TO 3
      IKK = DEL(1)-S(3)/8.
      DO 11 I=1,IKK
   11 BACKSPACE 8
      GO TO 15
   12 IF (S(3).LT.8.) GO TO 14
      IKK  = S(3)/8.-1.
      IF (IKK.EQ.0) GO TO 15
      DO 13 I=1,IKK
   13 READ (8) TBUF
      GO TO 15
   14 IF (TBUF.EQ.T(1)) GO TO 16
      BACKSPACE 8
   15 READ (8)  BUFEFM(1) ,BUFEFM(2) ,(BUFEFM(I),I=543, 722)
     1         ,(BUFEFM(I),I=  3, 542),(BUFEFM(I),I=543, 866)
     2         ,(BUFEFM(I),I=543,1154),(SNT (1,I),I=  1, 102)
      T(1) = TBUF
      T(2) = TBUF+(T(3)-TBUF)/4.
   16 CONTINUE
      ILL = 1
   17 IKK = IFM(3)
      IFM( 3) = IFM(12)
      IFM(12) = IKK
      DO 18 I=1,6
      D(2) = RFM(I,3)
      RFM(I, 3) = RFM(I,12)
   18 RFM(I,12) = D(2)
      GO TO (20,45) ,ILL
C
C     INTERPOLATION CONTROL
   20 D(2) = D(1)-BFM(1)
      ILL = 4
```

```
      NNN = 3
      D(10) = UCB(13)
      D(11) = D(10)/86400.D0
      DO 39 I=1,13
      IKK = IFM(I)
      IF (IKK.EQ.0) GO TO 39
      IF (I.EQ.13)  GO TO 22
      IF (I.EQ.11)  GO TO 21
      IF (ILL.EQ.IDL(I,1)) GO TO 30
      GO TO 25
   21 D(10) = UCB(14)
      D(11) = D(10)/86400.D0
      GO TO 25
   22 D(10) = 1.D0
      NNN = 2
      IF (ILL.EQ.3) GO TO 30
C
C     COMPUTE INTERPOLATION COEFFICIENTS
   25 ILL = IDL(I,1)
      IF (ILL.EQ.0) GO TO 39
      ILC = S(3)/DEL(ILL)
      D(6) = D(2)-DEL(ILL)*FLOAT(ILC)
      D(3) = 1.D0-D(6)
      DO 26 J=3,6,3
      D(8) = D(J)*D(J)-1.D0
      D(J+1) = D(J)*D(8)/6.D0
   26 D(J+2) = D(J+1)*(D(8)-3.D0)/20.D0
C
C     INTERPOLATE
   30 LOC  = IDL(I,2)+3*NNN*ILC
      IMM = LOC
      INN = IMM+3*NNN
      DO 32 K=1,NNN
      D(9) = 0.D0
      DO 31 J=3,5
      D(9) = D(9)+D(J)*BFM(IMM)+D(J+3)*BFM(INN)
      IMM = IMM+1
   31 INN = INN+1
   32 RFM(K,I) = D(10)*D(9)
      IF (IKK.EQ.1) GO TO 39
   35 IMM = LOC+IDF(ILL)
      INN = IMM+3*NNN
      DO 38 K=1,NNN
      D(9) = 0.D0
      DO 36 J=3,5
      D(9) = D(9)+D(J)*BFM(IMM)+D(J+3)*BFM(INN)
      IMM = IMM+1
   36 INN = INN+1
      RFM(K+3,I) = D(11)*D(9)
   38 CONTINUE
   39 CONTINUE
C
C     EARTH/EARTH-MOON BARYCENTER
   40 UCB(12) = UCB(3)+UCB(11)
      IF (IFM(11).EQ.0) GO TO 42
      D(3) = UCB( 3)/UCB(12)
      D(4) = UCB(11)/UCB(12)
      ILL = 3*IFM(11)
      DO 41 I=1,ILL
      RFM(I,12) = RFM(I,3)-D(4)*RFM(I,11)
   41 RFM(I,11) = RFM(I,3)+D(3)*RFM(I,11)
   42 ILL = 2
      GO TO 17
C     SUN
   45 IF (IFM(10).EQ.0) GO TO 50
      ILL = 3*IFM(10)
      DO 46 I=1,ILL
   46 RFM(I,10) = 0.D0
C
C     SHIFT BODY CENTERS
   50 IF (JCB.EQ.10) GO TO 999
      DO 51 J=1,6
   51 D(J) = RFM(J,JCB)
      DO 53 I=1,12
      IF (IFM(I).EQ.0) GO TO 53
```

```
      ILL = 3*IFM(I)                                                    FMRS0150
      DO 52 J=1,ILL                                                     FMRS0151
   52 RFM(J,I) = RFM(J,I)-D(J)                                          FMRS0152
   53 CONTINUE                                                          FMRS0153
C                                                                       FMRS0154
  999 NERR = IERR                                                       FMRS0155
      RETURN                                                            FMRS0156
      END
```

Subroutine: DRAGD

Purpose: To compute the acceleration due to atmospheric drag.

Calling Sequence: CALL DRAGD (R,V,TR,WE,RE,C,A)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	R	d(3)	R	km	Vehicle position relative to central body.
I	V	d(3)	V	km/sec	Vehicle velocity relative to central body.
I	TR	d(3,3)	T		Transformation from EE of date or body-fixed system to base coordinate system.
I	WE	d	ω_E	rad/sec	Central body rotation rate.
I	RE	d	r_E	km	Central body's radius.
I	C	d(2)	C_1, C_2		Drag coefficients.
IØ	A	d(3)	A	km/sec^2	Acceleration.

Common storages used: 12 cells

Subroutines required: DDØT, DCRØSS

```
$IBFTC MC132T  XR3,M94,NODD,LIST
      SUBROUTINE DRAGD (R,V,TR,WE,RE,C,A)                               DRAG0001
C     ATMOSPHERIC DRAG ACCELERATION                                     DRAG0002
      DOUBLE PRECISION  A(3),C(2),R(3),TR(3,3),V(3),WE,RE               DRAG0003
     1                 ,D,DDOT,DEXP,DSQRT                               DRAG0004
      COMMON            D(6)                                            DRAG0005
C                                                                       DRAG0006
    1 CALL DCROSS (TR(1,3),R,D)                                         DRAG0007
      DO 2 I=1,3                                                        DRAG0008
    2 D(I) = V(I)-WE*D(I)                                               DRAG0009
      D(4) = DDOT(R,R)                                                  DRAG0010
      D(4) = DSQRT(D(4))                                                DRAG0011
      D(5) = RE-D(4)                                                    DRAG0012
      D(6) = DDOT(D,D)                                                  DRAG0013
      D(6) = DSQRT(D(6))                                                DRAG0014
      D(5) = C(2)*D(5)                                                  DRAG0015
      D(5) =-C(1)*D(6)*DEXP(D(5))                                       DRAG0016
      DO 3 I=1,3                                                        DRAG0017
    3 A(I) = A(I)+D(5)*D(I)                                             DRAG0018
  999 RETURN                                                            DRAG0019
      END
```

Subroutine: DRAGDP

Purpose: To compute the acceleration due to atmospheric drag, its gradient and partial derivatives with respect to the drag coefficients.

Calling Sequence: CALL DRAGDP (R,V,TR,WE,RE,C,A,AP,G,N)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	R	d(3)	R	km	Vehicle position relative to the central body.
I	V	d(3)	V	km/sec	Vehicle velocity relative to the central body.
I	TR	d(3,3)	T		Transformation from EE of date or body-fixed system to base coordinate system.
I	WE	d	ω_E	rad/sec	Central body rotation rate.
I	RE	d	r_E	km	Central body radius.
I	C	d(2)	C_1, C_2		Drag coefficients.
IØ	A	d(3)	A	km/sec^2	Acceleration.
Ø	AP	d(6)			$\partial A/\partial c_1$ $\partial A/\partial c_2$
IØ	G	d(3,6)	∇A		Gradients $\nabla_R A$, $\nabla_V A$
I	N				Option switch =o: acceleration only. ≠o: $\partial A/\partial c_i$ and gradient.

Common storages used: 54 cells

Subroutines required: DDOT, DCROSS, DMVTRN

Method:

The expression assumed for atmospheric drag is

$$A = - c_1 e^{-c_2 h} \left| V_a \right| V_a$$

where c_1, c_2 are constants and

h = altitude $(= |R| - r_E)$
V_a = velocity relative to the atmosphere.

Implied assumptions include an exponential atomosphere rotating rigidly with the central body. The first coefficient, c_1, is

$$c_1 = \frac{1}{2} \rho_o \ c_D \ S/M$$

where

ρ_o = "sea level" atmospheric density (Earth ≈ 1.225 kg/m^3)
c_D = drag coefficient (≈ .15 - .20 for hypersonic velocities)
S = cross-sectional area of vehicle
M = vehicle mass

The second coefficient, c_2, is

$$c_2 \approx \frac{g_E M_o}{R^* T_M}$$

where

g_E = gravity at central body surface
M_o = mean molecular weight of atmosphere

R^* = universal gas constant

T_M = molecular temperature of the atmosphere

For Earth, at altitudes above 100 km,

$$C_2 \approx .14$$

The velocity of the vehicle relative to the atmosphere, V_a, is

$$V_a = V - \omega_E \hat{k} \times R$$

where $\hat{k}$ is a unit vector along the central body's north polar axis.

The partial derivatives of A are

$$\frac{\partial A}{\partial C_1} = \frac{A}{C_1} \quad , \quad \text{computed if } N = -1, -3$$

$$\frac{\partial A}{\partial C_2} = -hA \quad , \quad \text{computed if } N = -2, -3$$

The gradients of A are

$$\nabla_R A = -\frac{C_2}{r} AR^T - \omega_E \nabla_V A \ (\hat{k}x)$$

$$\nabla_V A = \frac{1}{V_a^2} AV_a^T - C_1 e^{-c_2 h} |V_a| \ I$$

Both A and its gradients are summed with input values.

```
$IBFTC MC132U  XR3,M94,NODD,LIST
      SUBROUTINE DRAGDP (R,V,TR,WE,RE,C,A,AP,G,N)
C     ATMOSPHERIC DRAG ACCELERATION
      DOUBLE PRECISION   A(3),AP(3,2),C(2),G(3,6),R(3),TR(3,3),V(3)
     1                   ,WE,RF,U(5),D,DDOT,DEXP,DSQRT
      COMMON             D(3,9)
      DATA               U/ 0.D0, 0.D0,-1.D0, 0.D0, 0.D0/
C
    1 DO 2 I=1,3
    2 D(I,2) = WE*TR(I,3)
      CALL DCROSS (D(1,2),R,D(1,1))
      DO 3 I=1,3
    3 D(I,1) = V(I)-D(I,1)
      D(1,7) = DDOT(R,R)
      D(1,7) = DSQRT(D(1,7))
      D(2,7) = RE-D(1,7)
      D(3,7) = DDOT(D,D)
      D(1,8) = DSQRT(D(3,7))
      D(2,8) = C(2)*D(2,7)
      D(2,8) =-D(1,8)*DFXP(D(2,8))
    4 ASSIGN 10 TO LL
      IF (N.GE.0) GO TO 7
      ASSIGN  8 TO LL
      D(1,9) = 1.D0
      D(2,9) = C(1)*D(2,7)
      IF (N+2) 7,5,6
    5 D(1,9) = D(2,9)
    6 ASSIGN  9 TO LL
    7 CONTINUE
      DO 10 I=1,3
      D(I,3) = D(2,8)*D(I,1)
      A(I) = A(I)+C(1)*D(I,3)
      GO TO LL, (8,9,10)
    8 AP(I,2) = D(2,9)*D(I,3)
    9 AP(I,1) = D(1,9)*D(I,3)
   10 CONTINUE
      IF (N.EQ.0) GO TO 999
   11 D(2,7) = C(1)*D(2,8)
      D(1,7) =-C(1)*C(2)/D(1,7)
      DO 13 I=1,3
      D(1,8) = D(I,3)*D(1,7)
      D(2,8) = D(I,3)/D(3,7)
      DO 12 J=1,3
      D(I,J+3) = D(2,8)*D(J,1)
   12 G(I,J) = G(I,I)+D(1,8)*R(J)
   13 D(I,I+3) = D(I,I+3)+D(2,7)
      J = 9
      DO 14 I=1,3
      CALL DCROSS (D(1,2),U(I),D(1,J))
   14 J = J-1
      CALL DMVTRN (D(1,4),D(1,7),D(1,1),1,3)
      DO 15 I=1,3
      DO 15 J=1,6
   15 G(I,J) = G(I,J)+D(I,J)
  999 RETURN
      END
```

Subroutine: DSDAT

Purpose: Computes three DSIF (JPL) measurements (hour angle, declination, doppler).

Calling Sequence: CALL DSDAT(GHA)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	GHA	d	♈	rad	Greenwich hour angle at time of measurements.

Common storages used: /DATCØM/

Subroutines required: DCRØSS,DDØT,DMVTRN,DVNØRM

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 position in B-frame (km). STA (4-5) Refraction constants. STA(6-10), Same as above, but for transmitting station if doppler is three-way.
	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 when in three-way doppler mode.
	C(269)	XV	d(12)	XV(1-6), Spacecraft position and velocity at end of doppler interval as signal leaves spacecraft. XV(7-12), Same as above, but at beginning of count interval.
	C(294)	MØDE		Doppler mode key +2, two way +3, three way
	C(298)	NANG		Angle inclusion key*.
	C(299)	NFRAC		Refraction correction key*. *+1, include. +2, omit.
	C(5)	ØBS	d(64)	Measurements. See Table 1, description of CBDAT.

```
$IBFTC MC13DM  M94,NODD,XR3
CMC13DM    DSIF MEASUREMENTS
      SUBROUTINE DSDAT(GHA)
      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   DCXSOB(3),    DCXSTR(3),    DELTE(2),     DNOB(3)
     1,                  DNTR(3),      DRDT(2),      DUM1(3),      DUM2(3)
     2,                  E(2),         EDOT(2),      EK(3),        OMG(3)
     3,                  RM(2),        RTMG(2),      RTOB(6),      RTOBX(6)
     4,                  RTTR(6),      RTTRX(6),     SCEK(3),      SXOB(3)
     5,                  SXTR(3),      TT2CO(9),     TT2CT(9),     XSOB(6)
     6,                  XSOBX(6),     XSTR(6),      XSTRX(6),     Z1(3)
     7,                  Z2(3)
      DOUBLE PRECISION   BIAS,      CF,        CONST,     CX,        CY
     1,                  OBS,                  DDOT,      DEN1,      DEN2
     2,                  DNORM,     F,         FTR,       GHA,       OMEGA
     3,                  RATOB,     RATOBX,    RATTR,     RATTRX,    RSPLT
     4,                  RSPLT2,    R2OB,      R2TR,      SA,        SB
     5,                  SE,        SLON,      SPDLT,     STA,       S1
     6,                  S2,        T,         TAU,       TB2CO,     TB2CT
     7,                  TT2BO,     TT2BT,     XTDOB,     XTDTR,     XTEOB
     8,                  XTETR,     XTNOB,     XTNTR,     XV
      DOUBLE PRECISION   AZ,        CPHI,      SPHI
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION   DATAN2,DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE   (SXOB(1),XTNOB),   (SXOB(2),XTEOB),   (SXOB(3),XTDOB)
     1,             (SXTR(1),XTNTR),   (SXTR(2),XTETR),   (SXTR(3),XTDTR)
     2,             (TT2CO(7),DNOB),   (TT2CT(7),DNTR),   (RM(1),RATOB)
     3,             (RM(2),RATTR)
      IF(MODE.EQ.3) GO TO 30
      DO 10 I=1,5
   10 STA(I+5)=STA(I)
      DO 20 I=1,9
   20 TT2BT(I)=TT2BO(I)
   30 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 COUNT 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 40 I=1,3
   40 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 50 I=1,6
      XSOB(I)=XV(I)-RTOB(I)
      XSTR(I)=XV(I)-RTTR(I)
      XSOBX(I)=XV(I+6)-RTOBX(I)
   50 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 (60,70),NANG
C     HOUR ANGLE AND DECLINATION
```

```
   60 SLON=DATAN2(STA(2),STA(1))                                          DSDT0075
      OBS(1)=GHA+SLON-DATAN2(XSOB(2),XSOB(1))                             DSDT0076
      CALL DVNORM(TB2CO(7),EK)                                            DSDT0077
      CALL DCROSS(XSOB,EK,SCEK)                                           DSDT0078
      S2=DDOT(SCEK,SCEK)                                                  DSDT0079
      S1=DSQRT(S2)                                                        DSDT0080
      OBS(17)=DATAN(XSOB(3)/S1)                                           DSDT0081
C     DIFFERENCED DOPPLER OBSERVABLE                                      DSDT0082
   70 CONST=BIAS(2)*FTR*RSPLT                                             DSDT0083
      OBS(49)=BIAS(1)*TAU+CONST*(RATOB+RATTR-RATOBX-RATTRX)               DSDT0084
      GO TO (80,170),NFRAC                                                DSDT0085
C     REFRACTION CORRECTIONS                                              DSDT0086
C     FIRST, ELEVATIONS AND ELEVATION CORRECTIONS                         DSDT0087
   80 E(1)=DATAN(-XTDOB/DSQRT(XTNOB*XTNOB+XTEOB*XTEOB))                   DSDT0088
      E(2)=DATAN(-XTDTR/DSQRT(XTNTR*XTNTR+XTETR*XTETR))                   DSDT0089
      DO 120 I=1,2                                                        DSDT0090
      DELTE(I)=0.                                                         DSDT0091
      IF(E(I).LT..01) GO TO 120                                           DSDT0092
      SE=DSIN(E(I))                                                       DSDT0093
      CE=DCOS(E(I))                                                       DSDT0094
      IF(E(I).GE..17453293D0) GO TO 110                                   DSDT0095
      T=(1.03585796D0-(.01072014D0-(.1279119D-7-.1227363D-7/E(I))/E(I))/DSDT0096
     1E(I))*STA(5*I-1)*CE/SE                                              DSDT0097
   90 F=RTMG(I)/RM(I)                                                     DSDT0098
  100 DELTE(I)=T-F*((STA(5*I-1)+T*T/2.)*CE-T*SE)                          DSDT0099
      GO TO 120                                                           DSDT0100
  110 DELTE(I)=STA(5*I-1)*CE/SE                                           DSDT0101
  120 CONTINUE                                                            DSDT0102
      CALL DCROSS(DNOB,XSOB,DCXSOB)                                       DSDT0103
      GO TO (130,140),NANG                                                DSDT0104
C     ANGLE CORRECTIONS                                                   DSDT0105
  130 SPHI=RTOB(3)/RTMG(1)                                                DSDT0106
      CPHI=DSQRT(RTOB(2)**2+RTOB(3)**2)/RTMG(1)                           DSDT0107
      AZ=DATAN2(XTEOB,XTNOB)                                              DSDT0108
      OBS(1)=OBS(1)+DSIN(OBS(1))**2*CPHI/(DSIN(AZ)*DCOS(E(1))**2)*DELTE(DSDT0109
     11)                                                                  DSDT0110
      OBS(17)=OBS(17)+(DCOS(E(1))*SPHI-DCOS(AZ)*DSIN(E(1))*CPHI)/DCOS(OBDSDT0111
     1S(17))*DELTE(1)                                                     DSDT0112
C     DOPPLER CORRECTION                                                  DSDT0113
  140 CALL DCROSS(DNTR,XSTR,DCXSTR)                                       DSDT0114
      CALL DCROSS(DCXSOB,XSOB,DUM1)                                       DSDT0115
      CALL DCROSS(DCXSTR,XSTR,DUM2)                                       DSDT0116
      DEN1=R2OB*DNORM(DCXSOB)                                             DSDT0117
      DEN2=R2TR*DNORM(DCXSTR)                                             DSDT0118
      CALL DCROSS(XSOB,OMG,Z1)                                            DSDT0119
      CALL DCROSS(XSTR,OMG,Z2)                                            DSDT0120
      DO 150 I=1,3                                                        DSDT0121
      Z1(I)=Z1(I)+XSOB(I+3)                                               DSDT0122
      Z2(I)=Z2(I)+XSTR(I+3)                                               DSDT0123
      DUM1(I)=DUM1(I)/DEN1                                                DSDT0124
  150 DUM2(I)=DUM2(I)/DEN2                                                DSDT0125
      EDOT(1)=DDOT(DUM1,Z1)                                               DSDT0126
      EDOT(2)=DDOT(DUM2,Z2)                                               DSDT0127
      DO 160 I=1,2                                                        DSDT0128
      SA=DSIN(E(I)+DELTE(I))                                              DSDT0129
      SB=DSIN(E(I)+DELTE(I)-EDOT(I)*TAU)                                  DSDT0130
  160 DRDT(I)=STA(5*I-1)/STA(5*I)*(1.D-3/SA-1.D-3/SB)                     DSDT0131
      OBS(49)=OBS(49)+CONST*(DRDT(1)+DRDT(2))                             DSDT0132
  170 GO TO (180,190),NANG                                                DSDT0133
  180 CONTINUE                                                            DSDT0134
      IF(OBS(1).LT.0.D0) OBS(1)=OBS(1)+6.2831853071795864                 DSDT0135
      IF(OBS(17).LT.0.D0) OBS(17)=OBS(17)+6.2831853071795864              DSDT0136
      IF(OBS(1).GT.6.2831853071795864) OBS(1)=OBS(1)-6.2831853071795864 DSDT0137
  190 RETURN                                                              DSDT0138
      END
```

Subroutine: DSDATP

Purpose: Computes three DSIF measurements and their partials (hour angle, declination, doppler).

Calling Sequence: CALL DSDATP(GHA,R2).

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	GHA	d	♈	rad	Greenwich hour angle at time of measurements.
Ø	R2	d	S^2	km^2	Slant range squared.

Common storages used: /DATCØM/

Subroutines required: DCRØSS,DDØT,DMVTRN,DNØRM,DVNØRM

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 position in B-frame (km). STA(4-5), Refraction constants. STA(6-10), As above, but for transmitting station if doppler mode is three-way.
	C(159)	TAU	d	Doppler count interval.
	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 at receiving station in B-frame.
	C(251)	TT2BT	d(9)	Same as TT2BØ, but for transmitting station if doppler mode is three-way.
	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), as above, but at beginning of doppler count.
	C(293)	MLT		Speed of light partial option key*.
	C(294)	MODE		Doppler mode +2, two-way +3, three-way
	C(295)	MSTA		Station location partial option key*.
	C(296)	MTIM		Station clock partial option key*.

COMMON LOCATIONS

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

```
$IBFTC MC13DN  XR3,M94,NODD,LIST
CMC13DN    DSIF MEASUREMENTS AND PARTIALS
      SUBROUTINE DSDATP(GHA,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  DCXSOB(3),   DCXSTR(3),   DELTE(2),    DNOB(3)
     1,                 DNTR(3),     DRDT(2),     DUM1(3),     DUM2(3)
     2,                 E(2),        EDOT(2),     EK(3),       OMG(3)
     3,                 RM(2),       RTMG(2),     RTOB(6),     RTOBX(6)
     4,                 RTTR(6),     RTTRX(6),    SCEK(3),     SXOB(3)
     5,                 SXTR(3),     TT2CO(9),    TT2CT(9),    XSOB(6)
     6,                 XSOBX(6),    XSTR(6),     XSTRX(6),    Z1(3)
     7,                 Z2(3)
      DOUBLE PRECISION  BIAS,      CE,        CONST,     CX,        CY
     1,                 OBS,                  DDOT,      DEN1,      DEN2
     2,                 DNORM,     F,         FTR,       GHA,       OMEGA
     3,                 RATOB,     RATOBX,    RATTR,     RATTRX,    RSPLT
     4,                 RSPLT2,    R2OB,      R2TR,      SA,        SB
     5,                 SE,        SLON,      SPDLT,     STA,       S1
     6,                 S2,        T,         TAU,       TB2CO,     TB2CT
     7,                 TT2BO,     TT2BT,     XTDOB,     XTDTR,     XTEOB
     8,                 XTETR,     XTNOB,     XTNTR,     XV
      DOUBLE PRECISION  AZ,        CPHI,      SPHI
C
C     DECLARE LIBRARY FUNCTIONS DOUBLE PRECISION TO SATISFY UNIVAC
      DOUBLE PRECISION  DATAN2,DATAN,DCOS,DSIN,DSQRT
C
      EQUIVALENCE  (SXOB(1),XTNOB),  (SXOB(2),XTEOB),  (SXOB(3),XTDOB)
     1,            (SXTR(1),XTNTR),  (SXTR(2),XTETR),  (SXTR(3),XTDTR)
     2,            (TT2CO(7),DNOB),  (TT2CT(7),DNTR),  (RM(1),RATOB)
     3,            (RM(2),RATTR)
      IF(MODE.EQ.3) GO TO 30
      DO 10 I=1,5
   10 STA(I+5)=STA(I)
      DO 20 I=1,9
   20 TT2BT(I)=TT2BO(I)
   30 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 COUNT 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 40 I=1,3
   40 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 50 I=1,6
      XSOB(I)=XV(I)-RTOB(I)
      XSTR(I)=XV(I)-RTTR(I)
      XSOBX(I)=XV(I+6)-RTOBX(I)
   50 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 (60,70),NANG
C     HOUR ANGLE AND DECLINATION
```

```
   60 SLON=DATAN2(STA(2),STA(1))
      OBS(1)=GHA+SLON-DATAN2(XSOB(2),XSOB(1))
      CALL DVNORM(TB2CO(7),EK)
      CALL DCROSS(XSOB,EK,SCEK)
      S2=DDOT(SCEK,SCEK)
      S1=DSQRT(S2)
      OBS(17)=DATAN(XSOB(3)/S1)
C     DIFFERENCED DOPPLER OBSERVABLE
   70 CONST=BIAS(2)*FTR*RSPLT
      OBS(49)=BIAS(1)*TAU+CONST*(RATOB+RATTR-RATOBX-RATTRX)
      GO TO (80,170),NFRAC
C     REFRACTION CORRECTIONS
C     FIRST, ELEVATIONS AND ELEVATION CORRECTIONS
   80 E(1)=DATAN(-XTDOB/DSQRT(XTNOB*XTNOB+XTEOB*XTEOB))
      E(2)=DATAN(-XTDTR/DSQRT(XTNTR*XTNTR+XTETR*XTETR))
      DO 120 I=1,2
      DELTE(I)=0.
      IF(E(I).LT..01) GO TO 120
      SE=DSIN(E(I))
      CE=DCOS(E(I))
      IF(E(I).GE..17453293D0) GO TO 110
      T=(1.03585796D0-(.01072014D0-(.1279119D-7-.1227363D-7/E(I))/E(I))/
     1E(I))*STA(5*I-1)*CE/SE
   90 F=RTMG(I)/RM(I)
  100 DELTE(I)=T-F*((STA(5*I-1)+T*T/2.)*CE-T*SE)
      GO TO 120
  110 DELTE(I)=STA(5*I-1)*CE/SE
  120 CONTINUE
      CALL DCROSS(DNOB,XSOB,DCXSOB)
      GO TO (130,140),NANG
C     ANGLE CORRECTIONS
  130 SPHI=RTOB(3)/RTMG(1)
      CPHI=DSQRT(RTOB(2)**2+RTOB(3)**2)/RTMG(1)
      AZ=DATAN2(XTEOB,XTNOB)
      OBS(1)=OBS(1)+DSIN(OBS(1))**2*CPHI/(DSIN(AZ)*DCOS(E(1))**2)*DELTE(
     11)
      OBS(17)=OBS(17)+(DCOS(E(1))*SPHI-DCOS(AZ)*DSIN(E(1))*CPHI)/DCOS(OB
     1S(17))*DELTE(1)
C     DOPPLER CORRECTION
  140 CALL DCROSS(DNTR,XSTR,DCXSTR)
      CALL DCROSS(DCXSOB,XSOB,DUM1)
      CALL DCROSS(DCXSTR,XSTR,DUM2)
      DEN1=R2OB*DNORM(DCXSOB)
      DEN2=R2TR*DNORM(DCXSTR)
      CALL DCROSS(XSOB,OMG,Z1)
      CALL DCROSS(XSTR,OMG,Z2)
      DO 150 I=1,3
      Z1(I)=Z1(I)+XSOB(I+3)
      Z2(I)=Z2(I)+XSTR(I+3)
      DUM1(I)=DUM1(I)/DEN1
  150 DUM2(I)=DUM2(I)/DEN2
      EDOT(1)=DDOT(DUM1,Z1)
      EDOT(2)=DDOT(DUM2,Z2)
      DO 160 I=1,2
      SA=DSIN(E(I)+DELTE(I))
      SB=DSIN(E(I)+DELTE(I)-EDOT(I)*TAU)
  160 DRDT(I)=STA(5*I-1)/STA(5*I)*(1.D-3/SA-1.D-3/SB)
      OBS(49)=OBS(49)+CONST*(DRDT(1)+DRDT(2))
C     PARTIAL DERIVATIVES
  170 GO TO (180,260),NANG
C     ANGLE PARTIALS
C     PARTIALS WRT VEHICLE STATE
  180 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
      IF(OBS(1).GT.6.2831853071795864) OBS(1)=OBS(1)-6.2831853071795864
      CALL DCROSS(SCEK,XSOB,OBS(18))
      DEN1=R2OB*S1
      DO 190 I=1,3
      OBS(I+1)=SCEK(I)/S2
      OBS(I+4)=0.D0
      OBS(I+17)=OBS(I+17)/DEN1
  190 OBS(I+20)=0.D0
C     PARTIALS WRT MEASUREMENT BIAS
      OBS(8)=1.D0
```

```
      OBS(24)=1.D0                                                      DSDP0150
      GO TO (200,210),MTIM                                              DSDP0151
C     PARTIALS WRT OBSERVING STATION CLOCK BIAS                         DSDP0152
  200 OBS(9)=DDOT(OBS(2),XSOB(4))+OMEGA                                 DSDP0153
      OBS(25)=DDOT(OBS(18),XSOB(4))                                     DSDP0154
  210 GO TO (220,230),MLT                                               DSDP0155
C     PARTIALS WRT SPEED OF LIGHT                                       DSDP0156
  220 OBS(10)=DDOT(OBS(2),XV(4))*RATOB*RSPLT2                           DSDP0157
      OBS(26)=DDOT(OBS(18),XV(4))*RATOB*RSPLT2                          DSDP0158
  230 GO TO (240,260),MSTA                                              DSDP0159
C     PARTIALS WRT STATION LOCATION ERRORS                              DSDP0160
  240 CALL DCROSS(EK,RTOB,DUM1)                                         DSDP0161
      DEN1=DDOT(DUM1,DUM1)                                              DSDP0162
      CALL DMVTRN(TB2CO,OBS(18),OBS(27),2,1)                            DSDP0163
      DO 250 I=1,3                                                      DSDP0164
      OBS(I+13)=0.D0                                                    DSDP0165
      OBS(I+29)=0.D0                                                    DSDP0166
      DUM2(I)=DUM1(I)/DEN1-OBS(I+1)                                     DSDP0167
  250 OBS(I+26)=-OBS(I+26)                                              DSDP0168
      CALL DMVTRN(TB2CO,DUM2,OBS(11),2,1)                               DSDP0169
C     DOPPLER OBSERVABLE PARTIALS                                       DSDP0170
C     PARTIALS WRT VEHICLE STATE                                        DSDP0171
C     NORMALIZE RANGE VECTORS                                           DSDP0172
  260 DO 270 I=1,3                                                      DSDP0173
      XSOB(I)=XSOB(I)/RATOB                                             DSDP0174
      XSTR(I)=XSTR(I)/RATTR                                             DSDP0175
      XSOBX(I)=XSOBX(I)/RATOBX                                          DSDP0176
  270 XSTRX(I)=XSTRX(I)/RATTRX                                          DSDP0177
C     PARTIALS WRT TRANSMITTING STATION LOCATION ARE USED IN THE VEHICLEDSDP0178
C     STATE PARTIALS AND WILL BE CALCULATED HERE                        DSDP0179
      CALL DCROSS(OMG,XSTRX,OBS(62))                                    DSDP0180
      CALL DCROSS(OMG,OBS(62),DUM1)                                     DSDP0181
      CALL DCROSS(OMG,DUM1,DUM2)                                        DSDP0182
      DO 280 I=1,3                                                      DSDP0183
  280 DUM2(I)=(TAU*(OBS(I+61)+.5D0*TAU*(DUM1(I)+TAU*DUM2(I)/3.D0))+XSTRXDSDP0184
     1(I)-XSTR(I))*CONST                                                DSDP0185
      CX=DDOT(DUM2,RTTR(4))*RSPLT                                       DSDP0186
      CY=DDOT(DUM2,RTTRX(4))*RSPLT                                      DSDP0187
      DO 290 I=1,3                                                      DSDP0188
      OBS(I+49)=(XSOB(I)+XSTR(I))*(CONST+CX)-(XSOBX(I)+XSTRX(I))*(CONST+DSDP0189
     1CY)                                                               DSDP0190
  290 OBS(I+52)=CONST*TAU*(XSOBX(I)+XSTRX(I))                           DSDP0191
C     PARTIALS WRT MEASUREMENT BIAS                                     DSDP0192
      OBS(56)=TAU                                                       DSDP0193
      GO TO (300,310),MTIM                                              DSDP0194
C     PARTIAL WRT OBSERVING STATION CLOCK BIAS                          DSDP0195
  300 OBS(57)=CONST*(DDOT(XSOB,XSOB(4))+DDOT(XSTR,XSTR(4))-DDOT(XSOBX,XSDSDP0196
     1OBX(4))-DDOT(XSTRX,XSTRX(4)))                                     DSDP0197
  310 GO TO (320,330),MLT                                               DSDP0198
C     PARTIAL WRT SPEED OF LIGHT                                        DSDP0199
  320 OBS(58)=RSPLT2*(DDOT(OBS(50),XV(4))*RATOB-DDOT(OBS(50),XV(10))*RATDSDP0200
     1OBX+DDOT(DUM2,RTTR(4))*(RATOB+RATTR)-DDOT(DUM2,RTTRX(4))*(RATOBX+RDSDP0201
     2ATTRX))-CONST*RSPLT*(RATOB+RATTR-RATOBX-RATTRX)                   DSDP0202
  330 GO TO (340,370),MSTA                                              DSDP0203
C     PARTIALS WRT STATION LOCATION ERRORS                              DSDP0204
  340 CALL DMVTRN(TB2CT,DUM2,OBS(62),2,1)                               DSDP0205
      CALL DCROSS(OMG,XSOBX,OBS(59))                                    DSDP0206
      CALL DCROSS(OMG,OBS(59),DUM1)                                     DSDP0207
      CALL DCROSS(OMG,DUM1,DUM2)                                        DSDP0208
      DO 350 I=1,3                                                      DSDP0209
  350 DUM2(I)=(TAU*(OBS(I+58)+.5D0*TAU*(DUM1(I)+TAU*DUM2(I)/3.D0))+XSOBXDSDP0210
     1(I)-XSOB(I))*CONST                                                DSDP0211
      CALL DMVTRN(TB2CO,DUM2,OBS(59),2,1)                               DSDP0212
      IF (MODE.EQ.3) GO TO 370                                          DSDP0213
C     ADD STATION LOCATION PARTIALS FOR TWO-WAY DOPPLER                 DSDP0214
      DO 360 I=59,61                                                    DSDP0215
  360 OBS(I)=OBS(I)+OBS(I+3)                                            DSDP0216
  370 RETURN                                                            DSDP0217
      END
```

Subroutine: DSTAT

Purpose: Finds tracking-station position vector in body-fixed coordinates and finds the orthogonal topocentric to body-frame transformation.

Calling Sequence: CALL DSTAT(A,B,DTR,STA,SN)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	A	d	a	km	Body semi-major axis.
I	B	d	b	km	Body semi-minor axis.
I	DTR	d		rad/deg	Degree-to-radian conversion factor.
I	STA	d(3)		deg deg meters	STA(1) = Station latitude (geodetic) STA(2) = Station longitude STA(3) = Station altitude
Ø	SN	d(12)		km, dimension-less	SN(1-3) Station position in body-fixed coordinates. SN(4-12) Topocentric to body-fixed orthogonal transformation.

Common storages used: None

Subroutines required: None

Coordinate Frames:

The body-fixed coordinate frame is a right-handed Cartesian (dextral orthogonal) system with the first axis lying in the body equator plane in the direction of the prime meridian, the third axis lying along a line from the body center through the north pole, and the second axis chosen so as to complete the right-handed system.

The topocentric coordinate frame is a dextral orthogonal system. It may be called a local tangent plane system. The first axis lies along the local North vector, the second along the local East vector, and the third along the local Down vector. (The local Down vector at a point on the surface of the body is parallel to the radius vector from the body center to that point if and only if the body is spherical, except for the two poles and all points in the equator.)

Tracker Position Vector:

Let:

- θ be the longitude of the tracker
- ϕ be the latitude of the tracker (geodetic)
- h be the altitude of the tracker above the reference spheroid
- a be the body semi-major axis
- b be the body semi-minor axis

Let $c = \sqrt{\cos^2 \phi + \frac{b^2}{a^2} \sin^2 \phi}$

Then

$$SN(1) = \left(\frac{a}{c} + h\right) \cos \theta \cos \phi$$

$$SN(2) = \left(\frac{a}{c} + h\right) \sin \theta \cos \phi$$

$$SN(3) = \left(\frac{b^2}{ac} + h\right) \sin \phi$$

Transformation

The columns of the topocentric to body-fixed transformation matrix (stored by columns in SN(4) to SN(12)) are the local unit North, East, and Down vectors:

$$
\begin{array}{ccc}
N & E & D \\
\end{array}
$$

$$
\begin{bmatrix}
-\cos\theta\sin\phi & -\sin\theta & -\cos\theta\cos\phi \\
-\sin\theta\sin\phi & \cos\theta & -\sin\theta\cos\phi \\
\cos\phi & 0 & -\sin\phi
\end{bmatrix}
$$

```
$IBFTC MC13DW  XR3,M94,NODD,LIST
      SUBROUTINE DSTAT(A,B,DTR,STA,SN)
C     FINDS STATION POSITION VECTORS AND TRANSFORMATIONS
      DOUBLE PRECISION  A,         B,         STA(3),    SN(12)
     1,         SL,        CL,        ST,        CT,        C,         DTR
      DOUBLE PRECISION  DSIN,DCOS,DSQRT
      SL=DSIN(STA(1)*DTR)
      CL=DCOS(STA(1)*DTR)
      ST=DSIN(STA(2)*DTR)
      CT=DCOS(STA(2)*DTR)
      C=DSQRT(CL*CL+B*B*SL*SL/(A*A))
      SN(4)=-SL*CT
      SN(5)=-SL*ST
      SN(6)=CL
      SN(7)=-ST
      SN(8)=CT
      SN(9)=0.D0
      SN(10)=-CL*CT
      SN(11)=-CL*ST
      SN(12)=-SL
      SN(3) = STA(3)/1.D3
      SN(1)=-(A/C+SN(3))*SN(10)
      SN(2)=-(A/C+SN(3))*SN(11)
      SN(3)=-(B*B/(A*C)+SN(3))*SN(12)
      RETURN
      END
```

Subroutine: DSTEST

Purpose: To decode data and convert units for DSIF(JPL) data.

Calling Sequence: CALL DSTEST(IERR)

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
Ø	IERR				Error flag. Set = 2 by DSTEST if total number of station names exceeds 20. Otherwise set = 0.

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

Subroutines required: DATINP, PØLYFT

Discussion:

Subroutine DSTEST is presently a dummy subroutine. If it is called, it writes out an error message and returns control to the main program for processing of the next data tape.

Subroutine DSTEST was intended to parallel the operation of the similar subroutines CBTEST,GRTEST, and SBTEST, but could not be written because neither descriptions nor samples of DSIF (JPL) data were made available.

```
$IBFTC MC134V  M94,NODD,XR3
CMC134V   DSIF RAW DATA PROCESSOR                                        DSTS0001
      SUBROUTINE DSTEST(IERR)                                            DSTS0002
  601 FORMAT(48H0***  DSIF DATA PROCESSING NOT PERMITTED BECAUSE/        DSTS0003
     16X,41HFORMATTING INFORMATION WAS NOT AVAILABLE./                   DSTS0004
     26X,34HPROCESSING OF THIS TAPE ABANDONED./                          DSTS0005
     36X,38HPROGRAM PROCEEDS TO NEXT TAPE, IF ANY.)                      DSTS0006
      IERR=0                                                             DSTS0007
      WRITE(6,601)                                                       DSTS0008
      RETURN                                                             DSTS0009
      END                                                                DSTS0010
```

Subroutine: DTRANP

Purpose: To transform the covariance matrix from one inertial frame to another (given the transformation matrix).

Calling Sequence: CALL DTRANP(P,NR,T,S,Q,KK)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	P	d(NR,NR)			Input covariance matrix
I	NR				Dimension of P
I	T	d(3,3)			Position transformation matrix
I	S	d(3,3)			Velocity transformation matrix
Ø	Q	d(6,6)			Output covariance matrix
I	KK				Option key; see description following.

Common storages used: 54 cells.

Subroutine required: DMVTRN

DTRANP Usage

Consider the upper left 6×6 of P as being partitioned:

$$
\begin{array}{c c}
 & \begin{array}{cc} 3 & 3 \end{array} \\
\begin{array}{c} 3 \\ 3 \end{array} & \left[\begin{array}{c|c} P_1 & P_2 \\ \hline P_3 & P_4 \end{array} \right]
\end{array}
$$

If KK = 0

$$
Q = \begin{bmatrix} T & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix} \begin{bmatrix} T^T & 0 \\ 0 & S^T \end{bmatrix}
$$

If KK = 1

$$
Q = \begin{bmatrix} T P_1 T^T & Q' \\ Q' & S p_4 S^T \end{bmatrix}
$$

where Q' is left as it was on input.

```
$IBFTC MC13DJ  XR3,M94,NODD,LIST
      SUBROUTINE DTRANP(P,NR,T,S,Q,KK)
C     TRANSFORMS COVARIANCE MATRIX FROM ONE INERTIAL FRAME TO ANOTHER
      DOUBLE PRECISION  T(3,3),    P(NR,NR),  Q(6,6),    S(3,3)
      COMMON            A,B,D
      DOUBLE PRECISION  A(3,3),    B(3,3),    D(3,3)
C     Q = (T 0 ) (P1 P2) (TT 0)  WHERE P1, P2, P3, P4 ARE
C         (0 S ) (P3 P4) (0 ST)  3X3 MATRICES, UPPER LEFT OF P
C     KK=0 GIVES FULL 6X6 IN Q
C     KK=1 GIVES T P1 TT AND S P4 ST IN Q1, Q4
      DO 1 I=1,3
      DO 1 J=1,3
      A(I,J)=T(J,I)
    1 B(I,J)=S(J,I)
      DO 2 I=1,3
    2 CALL DMVTRN(T,P(1,I),D(1,I),1,1)
      DO 3 I=1,3
    3 CALL DMVTRN(D,A(1,I),Q(1,I),1,1)
      DO 4 I=1,3
    4 CALL DMVTRN(S,P(4,I+3),D(1,I),1,1)
      DO 5 I=1,3
    5 CALL DMVTRN(D,B(1,I),Q(4,I+3),1,1)
      IF(KK.GT.0) GO TO 9
      DO 6 I=1,3
    6 CALL DMVTRN(T,P(1,I+3),D(1,I),1,1)
      DO 7 I=1,3
    7 CALL DMVTRN(D,B(1,I),Q(1,I+3),1,1)
      DO 8 I=4,6
      DO 8 J=1,3
    8 Q(I,J)=Q(J,I)
    9 RETURN
      END
```

Subroutine: DTRDB

Purpose: To compute either the Darboux or the local tangent plane transformation.

Calling Sequence: CALL DTRDB(R, V, T, K)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	R	d(3)			Position, body centered coordinates
I	V	d(3)			Velocity, body centered coordinates
Ø	T	d(3,3)			Transformation matrix
I	K				K = 1, T is Darboux K = 2, T is local tangent plane.

Common storages used: None

Subroutines required: DCRØSS, DNØRM

DTRDB Equations:

1. Darboux

The columns of T are the vectors

VX(RXV), (RXV)x [VX(RXV)], RXV : all normalized

2. Local tangent plane

R, (RXV)XR, RXV: all normalized.

Subroutine: DTRDB

Purpose: To compute either the Darboux or the local tangent plane transformation.

Calling Sequence: CALL DTRDB(R, V, T, K)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	R	d(3)			Position, body centered coordinates
I	V	d(3)			Velocity, body centered coordinates
Ø	T	d(3,3)			Transformation matrix
I	K				K = 1, T is Darboux K = 2, T is local tangent plane.

Common storages used: None

Subroutines required: DCRØSS, DNØRM

DTRDB Equations:

1. Darboux

The columns of T are the vectors

VX(RXV), (RXV)X [VX(RXV)], RXV : all normalized

2. Local tangent plane

R, (RXV)XR, RXV: all normalized.

```
$IBFTC MC13DF  XR3,M94,NODD,LIST
      SUBROUTINE DTRDB(R,V,T,K)                                         TRDB0001
C     COMPUTES DARBOUX OR LOCAL TANGENT PLANE TRANSFORMATION            TRDB0002
      DOUBLE PRECISION  T(3,3),   R(3),     V(3),     A,        B       TRDB0003
      DOUBLE PRECISION  DNORM                                           TRDB0004
C     K=1 GIVES DARBOUX                                                 TRDB0005
C     K=2 GIVES LOCAL TP                                                TRDB0006
      CALL DCROSS(R,V,T(1,3))                                           TRDB0007
      IF(K.EQ.1) GO TO 2                                                TRDB0008
      DO 1 I=1,3                                                        TRDB0009
    1 T(I,1)=R(I)                                                       TRDB0010
      GO TO 3                                                           TRDB0011
    2 CONTINUE                                                          TRDB0012
      CALL DCROSS(V,T(1,3),T(1,1))                                      TRDB0013
    3 A=DNORM(T(1,1))                                                   TRDB0014
      B=DNORM(T(1,3))                                                   TRDB0015
      DO 4 I=1,3                                                        TRDB0016
      T(I,3)=T(I,3)/B                                                   TRDB0017
    4 T(I,1)=T(I,1)/A                                                   TRDB0018
      CALL DCROSS(T(1,3),T(1,1),T(1,2))                                 TRDB0019
      RETURN                                                            TRDB0020
      END
```

Subroutine: DVNØRM

Purpose: To normalize a vector.

Calling Sequence: X = DVNØRM (Z, W)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	Z	d(3)			Input vector.
Ø	W	d(3)			Unit vector along Z.
Ø	DVNØRM	d			Magnitude of Z.

Common storages used: None

Subroutines required: DDØT

```
$IBFTC MC132H  XR3,M94,NODD,LIST
      DOUBLE PRECISION FUNCTION DVNORM (Z,W)                            DVNM0001
      DOUBLE PRECISION          Z(3),W(3),SCL,DDOT,DSQRT                DVNM0002
    1 SCL = DSQRT(DDOT(Z,Z))                                            DVNM0003
      DO 2 I=1,3                                                        DVNM0004
    2 W(I) = Z(I)/SCL                                                   DVNM0005
      DVNORM = SCL                                                      DVNM0006
      RETURN                                                            DVNM0007
      END
```

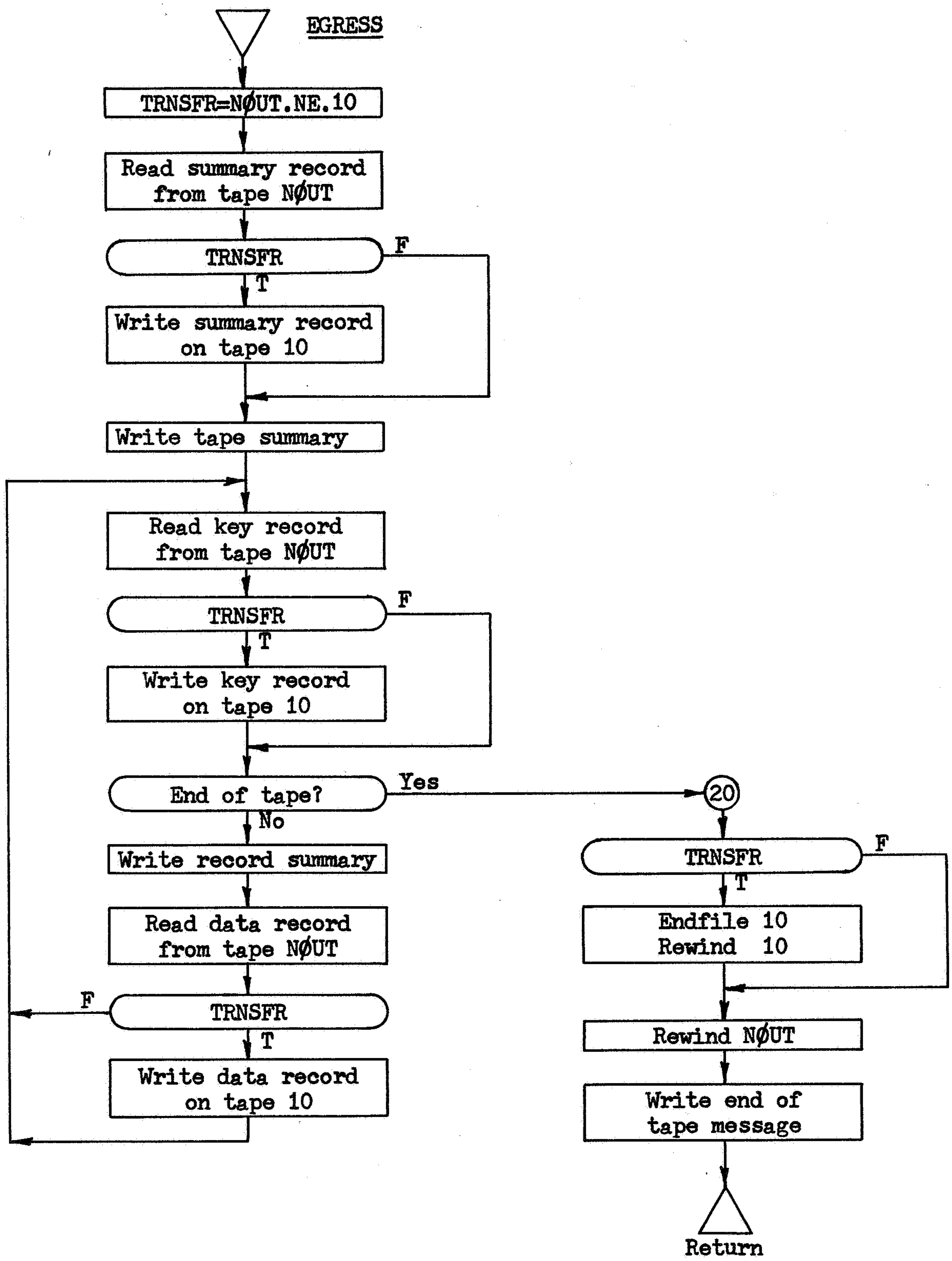
Subroutine: EGRESS

Purpose: To transfer the final edited data to tape 10 if necessary, and to write the output summary.

Calling Sequence: CALL EGRESS

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

Subroutines required: DATØUP


EGRESS
TRNSFR=NØUT.NE.10
Read summary record from tape NØUT
TRNSFR
F
T
Write summary record on tape 10
Write tape summary
Read key record from tape NØUT
TRNSFR
F
T
Write key record on tape 10
End of tape?
Yes
No
20
Write record summary
Read data record from tape NØUT
TRNSFR
F
T
Write data record on tape 10
TRNSFR
F
T
Endfile 10
Rewind 10
Rewind NØUT
Write end of tape message
Return

```
$IBFTC MC134U  M94,NODD,XR3
CMC134U   WRITES SUMMARY AND INSURES OUTPUT TO BE ON TAPE 10              EGRS0001
      SUBROUTINE EGRESS                                                   EGRS0002
      COMMON  /MIXCOM/IMIX(4)                                             EGRS0003
      EQUIVALENCE             (IMIX(1),LONNIN),    (IMIX(2),LON12)        EGRS0004
     1,  (IMIX(3),NIN),       (IMIX(4),NOUT)                              EGRS0005
C                                                                         EGRS0006
      COMMON  /SUMCOM/SUMM(56)                                            EGRS0007
      DOUBLE PRECISION  TSTART,   TSTOP                                   EGRS0008
      DIMENSION   HEADER(11),   STIMNX(20)                                EGRS0009
      REAL        NAMSTA(20)                                              EGRS0010
      EQUIVALENCE  (SUMM(1),HEADER), (SUMM(12),NUMSTA),(SUMM(13),NAMSTA)EGRS0011
     1,           (SUMM(33),TSTART),(SUMM(35),TSTOP), (SUMM(37),STIMNX)EGRS0012
C                                                                         EGRS0013
      COMMON  /OUTCOM/BUFF(40)                                            EGRS0014
      DIMENSION   SSD(4)                                                  EGRS0015
      DOUBLE PRECISION  TFIRST,   TLAST                                   EGRS0016
      EQUIVALENCE  (BUFF(1),NREC),    (BUFF(2),NEOT),    (BUFF(3),NAMOBS) EGRS0017
     1,           (BUFF(4),NAMTRA), (BUFF(6),NPTS),    (BUFF(7),KONT)     EGRS0018
     2,           (BUFF(8),MTYPE),   (BUFF(15),TFIRST),(BUFF(17),TLAST)   EGRS0019
     3,           (BUFF(33),NB1),    (BUFF(34),NB2),    (BUFF(35),NB3)    EGRS0020
     4,           (BUFF(36),NB4),    (BUFF(37),SSD)                       EGRS0021
C                                                                         EGRS0022
      COMMON  /MESCOM/DATA(510)                                           EGRS0023
C                                                                         EGRS0024
      DIMENSION   TYPE(4),      DUM(3)                                    EGRS0025
      LOGICAL TRNSFR                                                      EGRS0026
C                                                                         EGRS0027
  601 FORMAT(1H1,46X,26H***  EDITED DATA TAPE  ***/24X,7HHEADER,11A6)    EGRS0028
  602 FORMAT(19H0FIRST DATA ON TAPE,D23.16,20H SECONDS FROM 1950.0)      EGRS0029
  603 FORMAT(19H0 LAST DATA ON TAPE,D23.16,20H SECONDS FROM 1950.0)      EGRS0030
  604 FORMAT(1H0,51X,16HSTATIONS ON TAPE/49X,4HNAME,5X,13HFIRST ON TIME/EGRS0031
     1/(48X,A6,E18.8))                                                   EGRS0032
  605 FORMAT(/////24H ***  RECORD PAIR NUMBER,I5//75H RECEIVING STATIONEGRS0033
     1    TRANSMITTING STATION   MEASUREMENT TYPE   CONTINUATION/7X,A6,15EGRS0034
     2X,A6,15X,A6,12X,A3//12H DATA BEGINS,D24.16,20H SECONDS FROM 1950.0EGRS0035
     3)                                                                  EGRS0036
  606 FORMAT(12H0 DATA ENDS ,D24.16,20H SECONDS FROM 1950.0)             EGRS0037
  607 FORMAT(1H0,3X,5HTOTAL,25X,10HBAD POINTS/12H TIME POINTS,9X,7HANGLEEGRS0038
     1 1,3X,7HANGLE 2,3X,5HRANGE,3X,7HDOPPLER/I7,17X,I2,8X,I2,6X,I2,7X,IEGRS0039
     22//30X,19HSTANDARD DEVIATIONS/5X,7HANGLE 1,13X,7HANGLE 2,14X,5HRANEGRS0040
     3GE,14X,7HDOPPLER/E16.8,3E20.8)                                     EGRS0041
  608 FORMAT(//44X,32H *** END OF EDITED DATA TAPE ***)                  EGRS0042
      DATA TYPE/6HC-BAND,6HGODDRD,6HS-BAND,6H DSIF /                     EGRS0043
      DATA QYES,QNO/3HYES,3H NO/                                         EGRS0044
      TRNSFR=NOUT.NE.10                                                  EGRS0045
C     READ TAPE SUMMARY RECORD                                           EGRS0046
      READ(NOUT) SUMM                                                    EGRS0047
C     IF NOUT .NE. 10 WRITE IT ON 10                                     EGRS0048
      IF(TRNSFR) WRITE(10) SUMM                                          EGRS0049
C     WRITE OUT SUMMARY RECORD DATA ON SYSOU1                            EGRS0050
      WRITE(6,601) HEADER                                                EGRS0051
      WRITE(6,602) TSTART                                                EGRS0052
      CALL DATOUP(TSTART,DUM,0)                                          EGRS0053
      WRITE(6,603) TSTOP                                                 EGRS0054
      CALL DATOUP(TSTOP,DUM,0)                                           EGRS0055
      WRITE(6,604) (NAMSTA(I),STIMNX(I),I=1,NUMSTA)                      EGRS0056
C     READ A KEY RECORD FROM NOUT                                        EGRS0057
   10 READ(NOUT) BUFF                                                    EGRS0058
C     IF NOUT . NE. 10 WRITE IT ON 10                                    EGRS0059
      IF(TRNSFR) WRITE(10) BUFF                                          EGRS0060
C     TEST FOR END OF TAPE                                               EGRS0061
      IF(NEOT.EQ.1) GO TO 20                                             EGRS0062
      QOUT=QNO                                                           EGRS0063
      IF(KONT.EQ.1) QOUT=QYES                                            EGRS0064
C     WRITE SUMMARY OF KEY RECORD ON SYSOU1                              EGRS0065
      WRITE(6,605) NREC,NAMOBS,NAMTRA,TYPE(MTYPE),QOUT,TFIRST            EGRS0066
      CALL DATOUP(TFIRST,DUM,0)                                          EGRS0067
      WRITE(6,606) TLAST                                                 EGRS0068
      CALL DATOUP(TLAST,DUM,0)                                           EGRS0069
      WRITE(6,607) NPTS,NB1,NB2,NB3,NB4,SSD                              EGRS0070
C     READ DATA RECORD                                                   EGRS0071
      READ(NOUT) DATA                                                    EGRS0072
C     IF NOUT .NE. 10 WRITE IT ON 10                                     EGRS0073
      IF(TRNSFR) WRITE(10) DATA                                          EGRS0074
```

```
      GO TO 10                                                          EGRS0075
   20 CONTINUE                                                          EGRS0076
      IF(TRNSFR) ENDFILE 40                                             EGRS0077
      IF(TRNSFR) REWIND 40                                              EGRS0078
      REWIND NOUT                                                       EGRS0079
      WRITE(6,608)                                                      EGRS0080
      RETURN                                                            EGRS0081
      END                                                               EGRS0082
```

Subroutine: ENCKED

Purpose: To compute the inverse-square perturbing acceleration due to a central body (the so-called "Encke acceleration") in double precision.

Calling Sequence: CALL ENCKED(U,R,D,A)

Input and **Output**

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	U	d	μ	km^3/sec^2	Central body gravitational constant.
I	R	d(3)	R_o	km	Reference conic Position vector
I	D	d(3)	δR	km	Displacement from the reference conic position.
Ø	A	d(3)	$\delta \ddot{R}$	km/sec^2	Encke acceleration.

Common storages used: 6 cells

Subroutines required: DDØT

Description:

In using Encke's method for integration of trajectories, we set

$$R = R_o + \delta R$$

where R is the position vector, R_o is the position vector on a reference conic, and δR is the displacement from the conic to the true position.

$$\ddot{R} = -\frac{\mu}{r^3} R$$

$$\ddot{R}_o = -\frac{\mu}{r_o^3} R_o$$

where $r = |R|$, etc., then

$$\delta\ddot{R} = -\frac{\mu}{r^3} R + \frac{\mu}{r_o^3} R_o$$

where $\delta\ddot{R}$ is the acceleration of R relative to R_o due to the inverse-square attraction of the central body. The terms in $\delta\ddot{R}$ are large compared to $\delta\ddot{R}$, and to save significance, we compute $\delta\ddot{R}$ from the expansion

$$\delta\ddot{R} = \frac{\mu b}{r_o^3}(cR_o - \delta R)$$

where

$$q = (2R_o + \delta R) \cdot \delta R / r_o^2$$

$$p = -1 + (1 + q)^3 = q(3 + 3q + q^2)$$

$$b = 1/\sqrt{1 + p}$$

$$c = p/(1 + \sqrt{1 + p})$$

```
$IBFTC MC1320  XR3,M94,NODD,LIST
      SUBROUTINE ENCKED (U,R,D,A)                                       ENKE0001
C     ENCKE ACCELERATION                                                ENKE0002
      DOUBLE PRECISION   A(3),D(3),R(3),U                               ENKE0003
     1                   ,C,DDOT,DSQRT                                  ENKE0004
      COMMON             C(3)                                           ENKE0005
C                                                                       ENKE0006
    1 C(1) = DDOT(R,R)                                                  ENKE0007
      C(2) = DDOT(D,D)                                                  ENKE0008
      C(3) = (2.D0*DDOT(R,D)+C(2))/C(1)                                 ENKE0009
      C(2) = C(3)*(3.D0+C(3)*(3.D0+C(3)))                               ENKE0010
      C(3) = 1.D0+C(2)                                                  ENKE0011
      C(3) = DSQRT(C(3))                                                ENKE0012
      C(1) = C(3)*(C(1)*DSQRT(C(1)))/U                                  ENKE0013
      C(3) = 1.D0+C(3)                                                  ENKE0014
      C(2) = C(2)/C(3)                                                  ENKE0015
      DO 2 I=1,3                                                        ENKE0016
    2 A(I) = (C(2)*R(I)-D(I))/C(1)                                      ENKE0017
  999 RETURN                                                            ENKE0018
      END
```

Surboutine: ERRØUT (Dummy)

Purpose: Provides a dummy ERRØUT in FØRTRAN IV to satisfy the needs of several subroutines that call ERRØUT. (previously, a MAP-coded error trace routine).

Calling Sequence: CALL ERRØUT (List ignored)

Common storages used: None

Subroutines required: None

```
$IBFTC MC134A  M94,NODD,NOREF,XR3
      SUBROUTINE ERROUT                                                 ERDM0001
C            THIS IS A DUMMY REPLACEMENT FOR THE MAP-CODED              ERDM0002
C            ERROR TRACE ROUTINE ERROUT                                 ERDM0003
      WRITE(6,601)                                                      ERDM0004
  601 FORMAT(29H0ERROR STOP FROM DUMMY ERROUT)                          ERDM0005
      STOP                                                              ERDM0006
      END                                                               ERDM0007
```

Subroutine: ESTMAT

Purpose: Controls accumulation of differential corrections to a state estimate using the data from a specified tracking station.

Calling Sequence: CALL ESTMAT (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: //270 cells, /DATCØM/,/DCPCØM/,/DCRCØM/,/EDTCØM/, /ESTCØM/,/ES1CØM/,/SBFCØM/

Subroutines required: CBDATP, DDØT, DEHA, DEQTR, DGTRN, DGTSN, DMPLY, DMVTRN, DSDATP, ESTØUT, GRDATP, SBDATP

Method

ESTMAT uses the modified Kalman filter described in Reference 1, Appendix B, to compute differential corrections to an a priori state estimate and its covariance matrix using the data from a single tracking station. Let

$$Z = \begin{bmatrix} X(t) \\ U \\ V \end{bmatrix}$$

denote the true state, where

$X(t)$ = vehicle state,

U = a set of parameters affecting the motion, and

V = a set of parameters affecting the measurements.

We assume that a "nominal" state, $\overline{Z}$, is stored and define the deviations

$$\overline{Z} = Z_o + \overline{z}$$
$$\hat{Z} = Z_o + \hat{z} = \text{current state estimate}$$

where Z_o is the a priori estimate of state.

Interpolation on the Nominal Trajectory

TRAJDP integrates a nominal trajectory, $\overline{X}(t_v)$, from the initial conditions stored in ESTCØM. The nominal trajectory, transition matrix, parameter sensitivity matrix, Earth nutations, and down-leg transmission delay time are provided to ESTMAT as a set of interpolation coefficients in SBFCØM. Let

$R(t_v)$ = vehicle position on the nominal trajectory (3x1),

$\phi(t_v;t_{v1})$ = state transition matrix, integrated along the nominal trajectory (3x6),

$\phi_u(t_v;t_{v1})$ = parameter sensitivity matrix, integrated along the

$\delta\psi$, $\delta\epsilon$ = nutations in longitude, obliquity,

$\delta t(t_r)$ = down-leg transmission delay

$= t_r - t_v(t_r)$,

and let A be the vector

$$A = \begin{bmatrix} R(t_v) \\ \phi(t_v;t_{v1}) \\ \phi\ (t_v;t_{v1}) \\ a_n \\ a_{n+1} \\ a_{n+2} \end{bmatrix} \qquad \dot{A} = \begin{bmatrix} \dot{R}(t_v) \\ \dot{\phi}(t_v;t_{v1}) \\ \dot{\phi}_u(t_v;t_{v1}) \\ \delta\psi \\ \delta\epsilon \\ \delta t(t_r) \end{bmatrix}$$

where the components of ϕ, ϕ_u are stored in column order in sequential components of A.

SBFCØM contains

TBF = t_o = the initial time for the interpolation interval,

TFF = t_f = the final time for the interpolation interval,

$= t_o + 7\ h$,

HBF = h = the interval size used in constructing the interpolation tables,

RBF = $A(t_o)$

VBF = $\{\alpha_{ij}\}$ = interpolation coefficients for $\dot{A}(t)$.

For any t, we set

$$d = (t - t_o)/h$$

and compute

$$\dot{a}_i = \sum_{j=1}^{8} \alpha_{ij} \, d^{j-1}/(j-1)!$$

$$a_i = a_i(t_o) + \sum_{j=1}^{8} \alpha_{ij} \, d^j/j!$$

The measurement data is given at known values of reception time, t_{Sr}. In ephemeris time,

$$t_{Ur} = t_{Sr} + \gamma_1 + \gamma_2 t_{Sr}$$

$$t_r = t_{Ur} + \gamma_3 + \gamma_4 t_{Ur}$$

We first interpolate for $\delta t(t_r)$ and compute

$$t_v = t_r - \delta t(t_r)$$

The remaining components of A and $\dot{A}$ are then computed as functions of t_v.

Computation of Residuals

Let

$$y = G(X(t), V, t) + q$$

be the measurement value, where q is a random error. We compute the observable on the nominal

$$y = G(\overline{X}(t), \overline{V}, t)$$

and the residual from the nominal, $y-\overline{y}$, using the measurement model described in Reference 1, Appendix C. Since

$$\hat{Z} = (\overline{Z} - \overline{z}) + \hat{z}$$

the residual from the current estimate is

$$y - \hat{y} = y - \bar{y} - H\hat{z} + H\bar{z}$$

where

$$H = \frac{\partial y}{\partial Z}$$

<u>Differential Corrections</u>

Let

$$P = E((Z - \hat{Z})(Z - \hat{Z})^T)$$

$$Q = E(qq^T)$$

$$E(q) = 0$$

$$E(qZ^T) = 0$$

The differential corrections are computed from

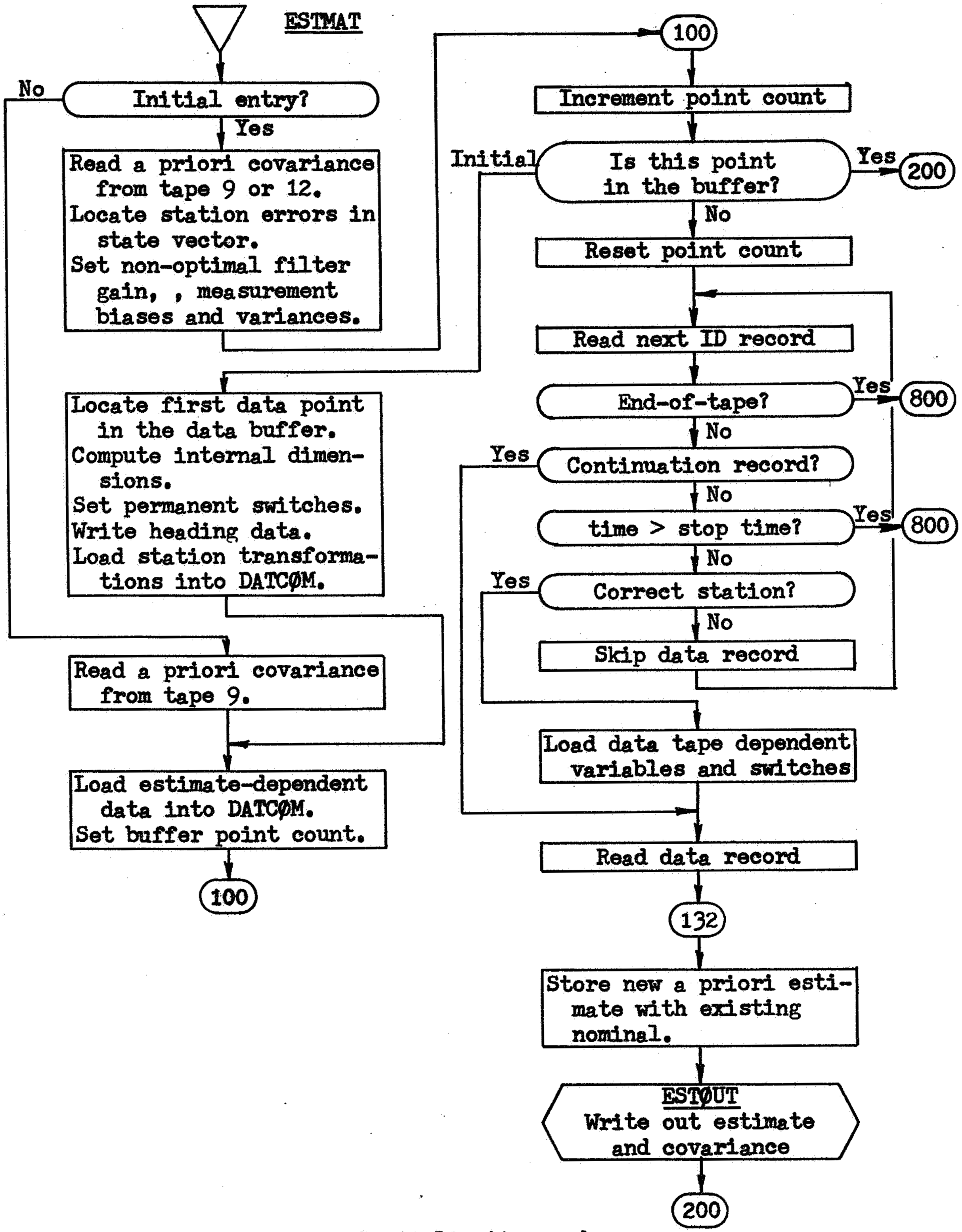
$$\hat{z} \longleftarrow \hat{z} + J(PH^T + \epsilon H^T)(HPH^T + Q)^{-1}(y - y)$$

$$P \longleftarrow P - J(PH^T + \epsilon H^T)(HPH^T + Q)^{-1}(PH^T + \epsilon H^T)^T$$

where J is a diagonal matrix with

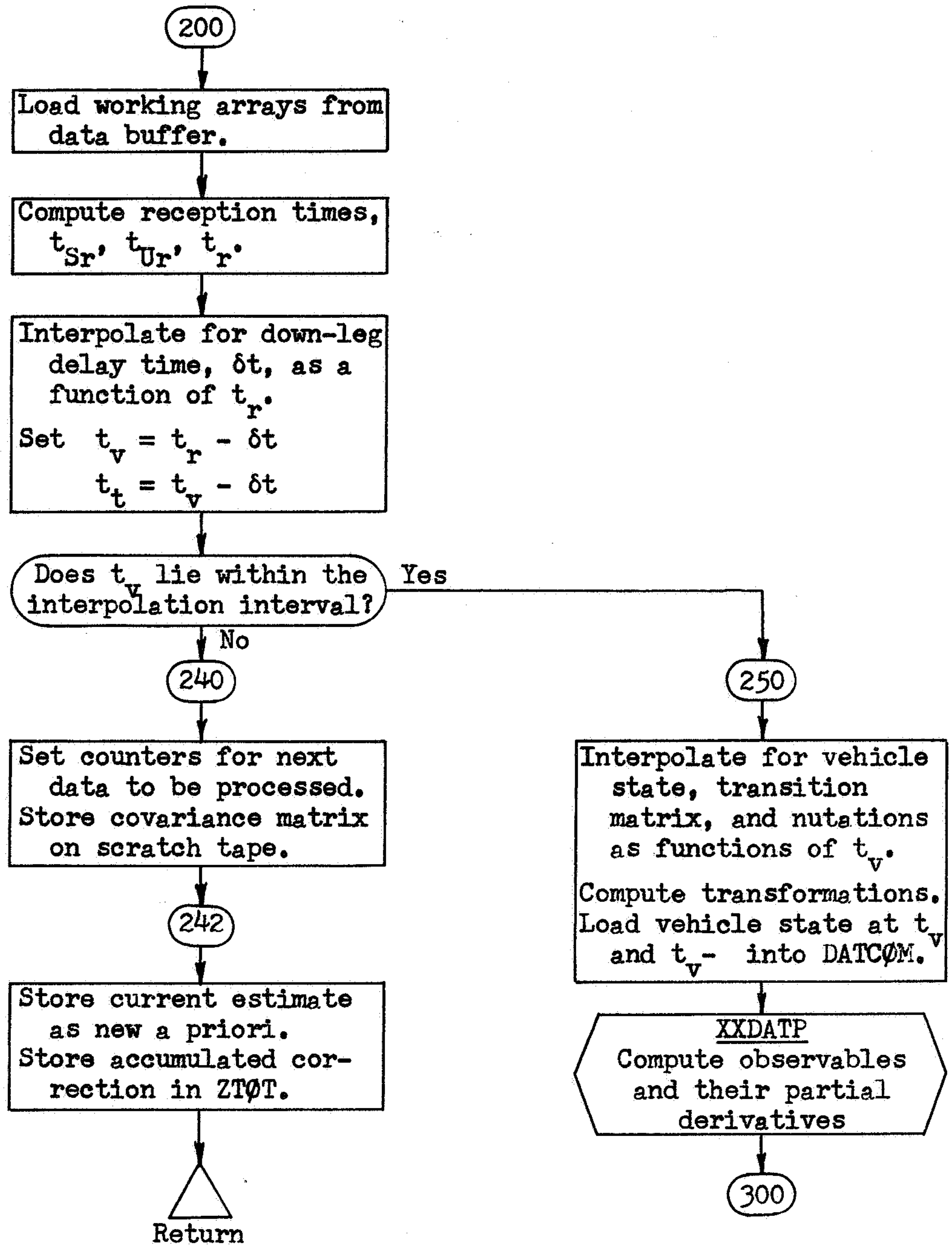
$J_{ii} = 1$ if Z_i is to be estimated

$J_{ii} = 0$ if Z_i is to be included as uncertain, but not solved for.

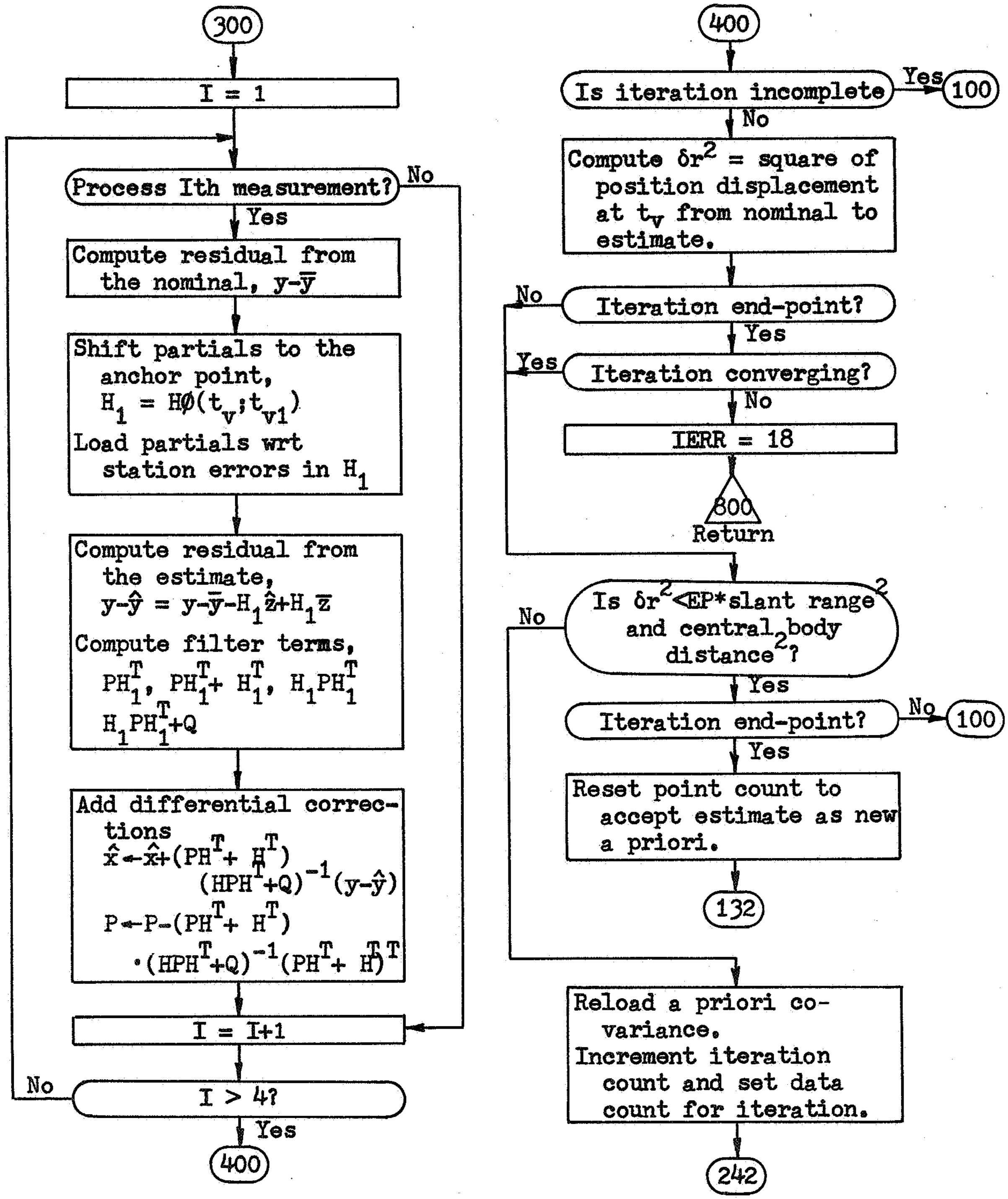


Initialization and
Data Input Control

ESTMAT-6



Computation of Observables
and Partial Derivatives


300
I = 1
Process Ith measurement?
No
Yes
Compute residual from the nominal, y-ȳ
Shift partials to the anchor point,
H1 = HØ(tv;tv1)
Load partials wrt station errors in H1
Compute residual from the estimate,
y-ŷ = y-ȳ-H1ẑ+H1z̄
Compute filter terms,
PH1T, PH1T+ H1T, H1PH1T
H1PH1T+Q
Add differential corrections
x̂←x̂+(PHT+ HT)
(HPHT+Q)-1(y-ŷ)
P←P-(PHT+ HT)
·(HPHT+Q)-1(PHT+ HT)T
I = I+1
I > 4?
No
Yes
400
400
Is iteration incomplete
Yes
100
No
Compute δr2 = square of position displacement at tv from nominal to estimate.
Iteration end-point?
No
Yes
Iteration converging?
Yes
No
IERR = 18
800
Return
Is δr2<EP*slant range2 and central body distance2?
No
Yes
Iteration end-point?
No
100
Yes
Reset point count to accept estimate as new a priori.
132
Reload a priori covariance.
Increment iteration count and set data count for iteration.
242

```
$IBFTC MC133J  XR3,M94,NODD,LIST
      SUBROUTINE ESTMAT (LTRAJ)
C     CONTROL ROUTINE FOR DIFFERENTIAL CORRECTION OF ESTIMATE
C
      LOGICAL          LTRAJ
      DOUBLE PRECISION  CDOT ,DCOS ,DSIN
C
      COMMON           /DCPCOM/CDCP(900)
      DOUBLE PRECISION  CBRAN2 ,EPSDC ,SN(13,20)
     1                 ,CON(8)        ,STIMR (2)
      DIMENSION         IFSKIP(4) ,IPROC(5) ,MSKIP(4,4) ,NPROC(22)
      EQUIVALENCE             (CDCP(811),EPSDC ) ,(CDCP(700),NPEND )
     1                       ,(CDCP(111),IERR  ) ,(CDCP(731),NPROC )
     2                       ,(CDCP(756),IPROC ) ,(CDCP(143),SN    )
     3    ,(CDCP(899),CBRAN2) ,(CDCP(881),MSKIP ) ,(CDCP(781),STIMR )
     4    ,(CDCP(  1),CON   ) ,(CDCP(702),NEMPS ) ,(CDCP(110),YTEST )
      EQUIVALENCE       (IPROC( 5),IFTSTP) ,(IPROC( 2),ITRSTA)
     1                 ,(IPROC( 3),INDSTA)
      EQUIVALENCE       (NPROC(18),IFBADQ) ,(NPROC(10),NPREST)
     1                 ,(NPROC(19),IFOUTL) ,(NPROC( 3),NPRSTA)
     2                 ,(NPROC(14),IFSKIP) ,(NPROC(13),NPSKIP)
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( 8),MTYPE  ) ,(INDDAT( 4),NTSTA  )
     3   ,(INDDAT( 1),DATIND) ,(INDDAT( 2),NEOT   ) ,(INDDAT(13),ONTIME)
C
      COMMON           /ESTCOM/CEST(804)
      DOUBLE PRECISION  DELDAN(2) ,SE(14,20) ,SPCDAN(6)
      EQUIVALENCE       (CEST( 97),DELDAN) ,(CEST(245),SE     )
     1                 ,(CEST( 27),KOUNTN) ,(CEST( 49),SPCDAN)
C
      COMMON           /ESICOM/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 ,HBF ,TBF ,TFF
      EQUIVALENCE             (CSBF( 6),ITER   ) ,(CSBF( 2),NTP    )
     1                       ,(CSBF( 3),NPT1   ) ,(CSBF( 9),TBF    )
     2    ,(CSBF( 7),HBF    ) ,(CSBF( 4),NPT2   ) ,(CSBF(11),TFF    )
C
      COMMON           /DCRCOM/ZBAR(30),ZHAT(30),ZTOT(30),DCSAVE(74)
      DOUBLE PRECISION  ZBAR ,ZHAT ,ZTOT
      DOUBLE PRECISION  DPREV    ,OBIAS(4) ,TFIRST
     1                 ,EPSK(4) ,Q2    (4) ,YMOD(4)
      LOGICAL          LSKIP(4) ,LSTA ,KOUTL
      DIMENSION         NASGN(9) ,LABEL(12) ,SYMBOL(4)
      EQUIVALENCE             (DCSAVE(39),LABEL  ) ,(DCSAVE( 1),OBIAS )
     1                       ,(DCSAVE(55),LSKIP  ) ,(DCSAVE(25),Q2    )
     2   ,(DCSAVE(33),DPREV ) ,(DCSAVE(59),LSTA   ) ,(DCSAVE(37),STPTIM)
     3   ,(DCSAVE(17),EPSK  ) ,(DCSAVE(66),NASGN  ) ,(DCSAVE(51),SYMBOL)
     4   ,(DCSAVE(64),ISTA1 ) ,(DCSAVE(61),NDELT  ) ,(DCSAVE(35),TFIRST)
     5   ,(DCSAVE(65),ISTA2 ) ,(DCSAVE(63),NHU    ) ,(DCSAVE(38),TREF  )
     6   ,(DCSAVE(60),KOUTL ) ,(DCSAVE(62),NINT   ) ,(DCSAVE( 9),YMOD  )
      EQUIVALENCE             (NASGN( 2),N129  ) ,(NASGN( 6),N204  )
     1                       ,(NASGN( 3),N130  ) ,(NASGN( 7),N290  )
     2                       ,(NASGN( 4),N200  ) ,(NASGN( 8),N300  )
     3    ,(NASGN( 1),N122  ) ,(NASGN( 5),N203  ) ,(NASGN( 9),N302  )
C
      COMMON           /CATCOM/CDAT(299)
      DIMENSION         IDATA(7)
      DOUBLE PRECISION  BIAS  (2) ,OMEGA   ,TAU      ,TT2BT(9)
     1                 ,FTR       ,SPDLT   ,TB2C(9,4) ,XV  (12)
     2                 ,OBS(16,4) ,STA(10) ,TT2BO (9)
      EQUIVALENCE             (CDAT(293),IDATA ) ,(CDAT(159),TAU   )
     1                       ,(CDAT(  5),OBS   ) ,(CDAT(161),TB2C  )
     2                       ,(CDAT(135),OMEGA ) ,(CDAT(233),TT2BO )
     3    ,(CDAT(  1),BIAS  ) ,(CDAT(137),SPDLT ) ,(CDAT(251),TT2BT )
     4    ,(CDAT(133),FTR   ) ,(CDAT(139),STA   ) ,(CDAT(269),XV    )
```

```
      EQUIVALENCE        (IDATA( 2),MODE  ) ,(IDATA( 5),NALIGN)
C
      COMMON              SAVE(90),STT(45,2)
      DOUBLE PRECISION    STT
C
      DOUBLE PRECISION    D(36) ,ETIMR ,GHA     ,PEH(30) ,TNPT1
     1                   ,DELT  ,ETIMT ,FLT(8) ,PH (36) ,UTIMR
     2                   ,DELTP ,ETIMV ,H1(30) ,SLRAN2  ,W (7)
     3                   ,EP
      DIMENSION           DATA(4) ,IQL    (4) ,LBTYP(4) ,PRCD (5)
     1                   ,FIL(12) ,JDATA (7) ,NM     (4) ,SYM(4,4)
     2                   ,FMT (8) ,LBM(12,4) ,NN     (8)
      LOGICAL             L ,LIN ,LOUTL
      EQUIVALENCE        (D(1),PEH) ,(W(1),TIME) ,(QUAL,IQUAL)
      DATA    EP    /3.D-4/
      DATA    FLT   /1.D0,2.D0,3.D0,4.D0,5.D0,6.D0,7.D0,8.D0/
      DATA    JDATA/ 1, 0, 1, 1, 0, 1, 1/
      DATA    NM  /-3,1,3,-1/
      DATA    NN    / 0, 6,20,24,26,50,74,84/
      DATA    FIL   /6H     9,6H 1X,E1,6H14X,E1,6H27X,E1,6H40X,E1
     1              ,6HF7.1,2,6H3.5,40,6H3.5,27,6H3.5,14,6H3.5, 1
     2              ,6H0     ,6H      /
      DATA    FMT   /6H(A1,   ,6HF7.1,2,6HX,A3,E,6H17.8, ,6H   X,E1
     1              ,6H3.5,   ,6HX,3E12,6H.4)   /
      DATA    LBM   /6HAZIMUT,6HELEVAT,6HRANGE ,6H      ,6HH      ,6HION
     1              ,6H      ,6H      ,6HRAD    ,6HRAD   ,6HKM     ,6H
     2              ,6HX     ,6HY     ,6HRANGE ,6HDOPPLE,6H      ,6H
     3              ,6H      ,6HR     ,6HRAD    ,6HRAD   ,6HSEC    ,6HCYCLES
     3              ,6HX     ,6HY     ,6HRANGE ,6HDOPPLE,6H      ,6H
     2              ,6H      ,6HR     ,6HRAD    ,6HRAD   ,6HKM     ,6HCYCLES
     4              ,6HHOUR A,6HDECLIN,6H       ,6HDOPPLE,6HNGLE   ,6HATION
     4              ,6H      ,6HR     ,6HRAD    ,6HRAD   ,6H       ,6HCYCLES
     5              /
      DATA    LBTYP/6HC-BAND,6HG-RR  ,6HS-BAND,6HDSIF  /
      DATA    PRCD /6H, PROC,6HEED TO,6H THE N,6HEXT PR,6HOCESS /
      DATA    START ,STOP  /6HOSTART,6H STOP /
      DATA    SYM   /3HAZM,3HELV,3HRGE,3H    ,3HX  ,3HY  ,3HRGE,3HDOP
     1              ,3HX  ,3HY  ,3HRGE,3HDOP ,3HHRA,3HDEC,3H   ,3HDOP /
  601 FORMAT(//23HORECEIVING STATION      ,A6/23H TRANSMITTING STATION  ,
     1A6/10H DATA TYPE,13X,A6,2H (,3(2A6,2H, ),2A6/32X,4(A6,8X),1H))
  602 FORMAT(15H MEAS VARIANCES,13X,4E14.4)
  603 FORMAT(/41HOTIME TAGS ARE REFERRED TO FIRST ONTIME =,D24.16,23H SE
     1CONDS (ST) FROM 1950)
  604 FORMAT(A6,14H PROCESSING AT,F12.2,13H SECONDS (ST))
  605 FORMAT(35H STOP PROCESSING AT END OF DATA ARC)
  606 FORMAT(6HOSKIP ,I3,31H DATA PTS BETWEEN PROCESSED PTS)
  609 FORMAT(/26HO    TIME  OBS      MEASURED,28X,9HRESIDUALS,24X,5HHZHAT,
     17X,5HHZBAR,7X,4HHPHT)
  612 FORMAT(31HO** ALL MEASUREMENTS SUPPRESSED,5A6)
  613 FORMAT(21HONEW DATA RECORD READ)
  641 FORMAT(/30HOLINEARITY TEST FAILED, ITER =,I2)
  642 FORMAT(/30HOLINEARITY TEST PASSED, ITER =,I2)
C
C**** INITIALIZATION
C
    1 IF (LTRAJ)               GO TO 60
C
C     HERE FOR INITIAL ENTRY FOR EACH PROCESS
C     LOAD A PRIORI COVARIANCE MATRIX AND STATE INDICES
   10 CONTINUE
      IF (NPREST.EQ.0)         GO TO 11
      READ   (12) DEST
      WRITE  (9) DEST
      NPREST = 0
      GO TO 12
   11 READ    (9) DEST
   12 BACKSPACE 9
      ISTA1 = 1
      LSTA   = .TRUE.
      J = 100+INDSTA
      DO 13 I=1,53,2
      IF (KLOCN(I).EQ.0)       GO TO 15
      IF (KLOCN(I)-J)          13,14,15
   13 ISTA1 = ISTA1+KLOCN(I+1)
      GO TO 15
```

```
   14 ISTA2 = ISTA1+KLOCN(I+1)-1
      LSTA  = .FALSE.
   15 CONTINUE
      DO 16 I=1,4
      EPSK(I) = EPSDC*SE(I+10,INDSTA)
   16 Q2(I)    = SE(I+10,INDSTA)**2
C
C     SET UP CCUNTERS 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 TC 110
   22 ASSIGN 800 TO N122
      ASSIGN 130 TO N129
      ASSIGN  23 TO N130
   23 CONTINUE
      STP = STIMR(1)-CNTIME
      DO 24 NPT=1,NPTS
   24 IF (STP.LE.BUFDAT(NPT,2)) GO TO 25
      NPT = NPTS+1
      GO TO 100
   25 ASSIGN 131 TC N130
      NPT1  = NPT-NPSKIP
      NPT2  = NPT1
      NDELT = NPEND+2
      NINT  = NPEND+1
      NHU   = NEMPS+6
      YMOD(1) = CON(3)
      YMOD(2) = CON(3)
      YMOD(4) = YTEST
      OBIAS(4) = 0.D0
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 (6,602) Q2
      WRITE (6,603) TFIRST
      CALL DATOUP (TFIRST,D,0)
      WRITE (6,604) START ,BUFDAT(NPT,2)
      GO TO (31,32) ,IFTSTP
   31 WRITE (6,604) STOP ,STPTIM
      ASSIGN 201 TO N200
      GO TC 33
   32 WRITE (6,605)
      ASSIGN 202 TC N200
   33 CONTINUE
      IF (III.NE.0)           WRITE (6,606) III
C
C     SET SWITCHES
   40 CONTINUE
      IF (IFOUTL.NE.0)        GO TO 41
      ASSIGN 203 TO N203
      ASSIGN 349 TO N302
      KOUTL = .TRUE.
      GO TO 42
   41 ASSIGN 204 TO N203
      ASSIGN 303 TO N302
      KOUTL = .FALSE.
   42 IF (IFBADD.NE.0)        GO TO 43
      ASSIGN 100 TC N204
      ASSIGN 301 TC N300
      GO TO 44
   43 ASSIGN 205 TO N204
      ASSIGN 302 TC N300
   44 CONTINUE
C
C     LOAD FIXED DATA
   50 CONTINUE
```

```
      DO 51 I=1,9
      TT2BO(I) = SN(I+4,INDSTA)
   51 TT2BT(I) = SN(I+4,ITRSTA)
      GO TO 70
C
C     HERE FOR REPEATED ENTRIES
   60 CONTINUE
      READ   (9) DEST
      BACKSPACE 9
C
C     LOAD STATE-DEPENDENT DATA
   70 CONTINUE
      DO 71 I=1,KOUNTN
   71 H1(I) = 0.D0
      CALL DMVTRN (TT2BO,SE(1,INDSTA),STA   ,1,1)
      CALL DMVTRN (TT2BT,SE(1,ITRSTA),STA(6),1,1)
      DO 72 I=1,3
      STA(I)    = SN(I+1,INDSTA)+STA(I)
   72 STA(I+5) = SN(I+1,ITRSTA)+STA(I+5)
      BIAS(1) = DATIND(15)+SE(10,INDSTA)/TAU
      SPDLT = CON(8)+SE(6,INDSTA)
      DO 73 I=1,3
   73 OBIAS(I) = SE(I+6,INDSTA)
C
C     CONTINUE PROCESSING DATA
   90 CONTINUE
      LTRAJ = .FALSE.
      NPT = NPT1
      TNPT1 = TBF
      DELTP = VBF(NDELT,1)
      WRITE (6,609)
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 (KONT.NE.0)            GO TO 130
C     NEW STATION PASS
      ASSIGN 103 TO N101
      NPT = 1
      GO TO (103,800) ,IFTSTP
  103 IF (STIMR(2).LT.ONTIME) 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
      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)+SE(10,INDSTA)/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.
        DO 121 J=I,12,4
    121 LABEL(J)  = LBM(J,MTYPE)
        SYMBOL(I) = SYM(I,MTYPE)
        IF (MSKIP(I,MTYPE)+IFSKIP(I).GT.0) GO TO 122
        LSKIP(I) = .FALSE.
        L = .TRUE.
    122 CONTINUE
        IF (L) GO TO 123
        WRITE (6,612) PRCD
        GO TO N122, (800,801)
    123 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, ( 23,131)
C
C       STORE NEW A PRIORI ESTIMATE WITH EXISTING NOMINAL
    131 CONTINUE
        NPT1 = NPT-NPSKIP
        NPT2 = NPT1
    132 WRITE  (9) DEST
        BACKSPACE 9
        DO 133 I=1,KOUNTN
        ZBAR(I) = ZBAR(I)-ZHAT(I)
        ZTOT(I) = ZTOT(I)+ZHAT(I)
    133 ZHAT(I) = 0.D0
        IF (ITER.NE.0)            GO TO 425
        WRITE (6,613)
        CALL ESTOUT
C
C**** COMPUTE OBSERVABLES AND PARTIALS
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)+SE(10,INDSTA)/TAU
    207 CONTINUE
C
C       COMPUTE VEHICLE TIME AND STATE
C
C       RECEPTION TIME
    210 CONTINUE
        STIMR = TFIRST+TIME
        UTIMR = STIMR+SE(4,INDSTA)+SE(5,INDSTA)*STIMR
        ETIMR = UTIMR+DELDAN(1)+DELDAN(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) ,III
C
C     COMPUTE DOWN-LEG DELAY TIME
  230 DELT = 0.D0
      DO 231 I=1,8
  231 DELT = DELT+D(I)*VBF(NDELT,I)
      ETIMV = ETIMR-DELT
      ETIMT = ETIMV-DELT
      IF (ETIMV.LE.TFF)       GO TO 250
C
C     SET ICS FOR NEW INTERPOLATION TABLES
  240 CONTINUE
      NPT1 = NPT-NPSKIP
      NPT2 = NPT1
      WRITE  (9) DEST
      BACKSPACE 9
      TBF = ETIMV
      VBF(NDELT,1) = DELT
  242 LTRAJ = .TRUE.
      N = 0
      DO 244 I=1,53,2
      J = KLOCN(I)
      K = KLOCN(I+1)
      IF (J.EQ.0)             GO TO 245
      JJ = 0
      IF (J.LT.100)           GO TO 243
      JJ = J-100
      J  = 8
  243 DO 244 III=1,K
      N = N+1
      M = ILOCN(N)+NN(J)+14*JJ
      SPCDAN(M) = SPCDAN(M)-ZBAR(N)+ZHAT(N)
      ZBAR(N) = ZHAT(N)
      IF (ITER.NE.0)          GO TO 244
      ZTOT(N) = ZTOT(N)+ZHAT(N)
      ZBAR(N) = 0.D0
  244 ZHAT(N) = 0.D0
  245 CONTINUE
      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,NINT,8,2,45,1,45,0)
      DO 252 I=1,NINT
  252 STT(I,2) = RBF(I)+HBF*STT(I,2)
      CALL DEGTR (ETIMV,D(10))
      D(2) = STT(NPEND+1,1)+D(12)
      D(3) = STT(NPEND,1)
      D(4) = D(12)
      D(5) = D(3)*DCOS(D(2))
      D(6) = ETIMV-DELDAN(1)-DELDAN(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)                                                     ESTM0450
      D(14) = D(10)                                                     ESTM0451
  256 CALL DMVTRN (D,D(10),TB2C(1,I),1,3)                               ESTM0452
C                                                                       ESTM0453
C     LOAD SPACECRAFT STATE                                             ESTM0454
  260 CONTINUE                                                          ESTM0455
      DO 261 I=1,3                                                      ESTM0456
      XV(I)   = STT(I,2)                                                ESTM0457
  261 XV(I+3) = STT(I,1)                                                ESTM0458
      D(2) = (ETIMV-TBF-TAU)/HBF                                        ESTM0459
      III = 3                                                           ESTM0460
      GO TO 220                                                         ESTM0461
  262 CONTINUE                                                          ESTM0462
      CALL DMPLY (VBF,D(2),XV(7),3,8,2,45,-1,3,0)                       ESTM0463
      DO 263 I=1,3                                                      ESTM0464
  263 XV(I+6) = XV(I+6)*HBF+RBF(I)                                      ESTM0465
C                                                                       ESTM0466
C     COMPUTE OBSERVABLES AND PARTIALS                                  ESTM0467
  290 CONTINUE                                                          ESTM0468
      GO TO N290, (291,292,293,294)                                     ESTM0469
C     C-BAND                                                            ESTM0470
  291 CALL CBDATP (SLRAN2)                                              ESTM0471
      GO TO 300                                                         ESTM0472
C     GODDARD                                                           ESTM0473
  292 CALL GRDATP (SLRAN2)                                              ESTM0474
      GO TO 300                                                         ESTM0475
C     S-BAND                                                            ESTM0476
  293 CALL SBDATP (SLRAN2)                                              ESTM0477
      GO TO 300                                                         ESTM0478
C     DSIF                                                              ESTM0479
  294 CALL DSDATP (GHA,SLRAN2)                                          ESTM0480
      GO TO 300                                                         ESTM0481
C                                                                       ESTM0482
C**** COMPUTE RESIDUALS AND DIFFERENTIAL CORRECTIONS                    ESTM0483
C                                                                       ESTM0484
C     RESIDUALS FROM THE NOMINAL                                        ESTM0485
  300 CONTINUE                                                          ESTM0486
      III = 1                                                           ESTM0487
      FMT(2) = FIL(6)                                                   ESTM0488
      DO 349 I=1,4                                                      ESTM0489
      IF (LSKIP(I))              GO TO 349                              ESTM0490
      IF (DATA(I).EQ.YTEST)      GO TO 349                              ESTM0491
      GO TO N300, (301,302)                                             ESTM0492
  301 IF (IQL(I).GT.0)           GO TO 349                              ESTM0493
  302 W(3) = DATA(I)                                                    ESTM0494
      IF (DATA(I).GE.0.)         GO TO 304                              ESTM0495
      GO TO N302, (303,349)                                             ESTM0496
  303 W(3) =-W(3)                                                       ESTM0497
  304 W(4) = W(3)-OBS(1,I)-OBIAS(I)                                     ESTM0498
      W(2) = SYMBOL(I)                                                  ESTM0499
C                                                                       ESTM0500
C     COMPUTE PARTIALS WRT STATE                                        ESTM0501
  310 CONTINUE                                                          ESTM0502
      IF (NHU.EQ.0)              GO TO 320                              ESTM0503
      CALL DMPLY (OBS(2,I),STT(4,2),PH ,1,3,NHU,1,3,1,0)                ESTM0504
      CALL DMPLY (OBS(2,I),STT(4,1),PEH,1,3,NHU,1,3,1,0)                ESTM0505
      J = 1                                                             ESTM0506
      N = NHU                                                           ESTM0507
      IF (KLOCN(1).EQ.1)         GO TO 312                              ESTM0508
      J = 7                                                             ESTM0509
      N = NEMPS                                                         ESTM0510
  312 CONTINUE                                                          ESTM0511
      DO 313 K=1,N                                                      ESTM0512
      H1(K) = PH(J)+PEH(J)                                              ESTM0513
  313 J = J+1                                                           ESTM0514
C                                                                       ESTM0515
C     COMPUTE PARTIALS WRT STATION ERRORS                               ESTM0516
  320 CONTINUE                                                          ESTM0517
      IF (LSTA)                  GO TO 330                              ESTM0518
      DO 329 M=ISTA1,ISTA2                                              ESTM0519
      K = ILOCN(M)                                                      ESTM0520
      GO TO (321,321,321,322,323,324,325,325,325,325) ,K                ESTM0521
  321 H1(M) = OBS(K+10,I)                                               ESTM0522
      GO TO 329                                                         ESTM0523
  322 H1(M) = OBS(9,I)                                                  ESTM0524
```

```
      GO TO 329
  323 H1(M) = OBS(9,I)*STIMR
      GO TO 329
  324 H1(M) = OBS(10,I)
      GO TC 329
  325 IF (K.NE.I+6)          GO TO 329
      H1(M) = OBS(8,I)
  329 CONTINUE
C
C     RESIDUALS FROM THE CURRENT ESTIMATE
  330 W(5) = 0.D0
      W(6) = 0.D0
      W(7) = 0.D0
      DO 332 M=1,KOUNTN
      W(5)    = W(5)+H1(M)*ZHAT(M)
      W(6)    = W(6)+H1(M)*ZBAR(M)
      PH(M)   = 0.D0
      DO 331 K=1,KOUNTN
  331 PH(M)   = PH(M)+PNEW(M,K)*H1(K)
      PEH(M)  = PH(M)+EPSK(I)*H1(M)
  332 W(7)    = W(7)+PH(M)*H1(M)
      D(33)   = W(7)+Q2(I)
      W(4)    = W(4)-W(5)+W(6)
      STP     = W(4)/YMOD(I)+0.5D0
      IF (STP.LT.0.)         STP = STP-1.
      STP     = AINT(STP)
      W(4)    = W(4)-STP*YMOD(I)
C
C     COMPUTE DIFFERENTIAL CORRECTION
  340 CONTINUE
      D(32) = W(4)/D(33)
      D(31) = EPSK(I)**2
      DO 347 J=1,KOUNTN
      IF (ITRETN(J).EQ.0)    GO TO 341
      ASSIGN 344 TO N342
      ASSIGN 343 TO N343
      ZHAT(J) = ZHAT(J)+D(32)*PEH(J)
      GO TC 342
  341 ASSIGN 346 TO N342
      ASSIGN 344 TO N343
  342 DO 346 K=J,KOUNTN
      IF (ITRETN(K).EQ.0)    GO TO N342, (344,346)
      GO TO N343, (343,344)
  343 PNEW(J,K) = PNEW(J,K)-(PH(J)*PH(K)-D(31)*H1(J)*H1(K))/D(33)
      GO TO 345
  344 PNEW(J,K) = PNEW(J,K)-PH(J)*PEH(K)/D(33)
  345 PNEW(K,J) = PNEW(J,K)
  346 CONTINUE
  347 CONTINUE
      FMT(5) = FIL(I+1)
      FMT(6) = FIL(I+6)
      WRITE (6,FMT) FIL(III+10),(W(K),K=III,7)
      III = 2
      FMT(2) = FIL(1)
  349 CONTINUE
C
C**** LINEARITY CONTROL
C
C     LINEARITY TEST
  400 CONTINUE
      IF (NPT.LT.NPT2)       GO TO 100
      DO 401 I=1,KOUNTN
  401 D(I) = ZBAR(I)-ZHAT(I)
      CALL DMPLY (STT(4,2),D,PH,3,NHU,1,3,1,1,0)
      D(4) = DDOT(PH,PH)
      LIN = .TRUE.
      III = 1
      IF (NPT.NE.NPT2)       GO TO 402
      III = 2
      IF (D(4).GT.DPREV)     GO TO 998
  402 IF (D(4).GT.SLRAN2*EP) LIN = .FALSE.
      IF (D(4).GT.CBRAN2*EP) LIN = .FALSE.
      IF (LIN)               GO TO (100,420) ,III
C
C     ITERATION CONTROL
```

```
C       RESET A PRICRI AND NOMINAL FCR NEW ITERATION
  410 DPREV = D(4)
      TBF    = TNPT1
      VBF(NDELT,1) = DELTP
      READ    (9) DEST
      BACKSPACE 9
      WRITE (6,641) ITER
      ITER = ITER+1
      IF (ITER.GT.6)           GC TO 998
      NPT2 = NPT
      GO TC 242
C
C       ACCEPT ESTIMATE AS NEW A PRIORI
  420 NPT1 = NPT2
      TNPT1 = ETIMV
      DELTP = DELT
      GO TC 132
  425 WRITE (6,642) ITER
      ITER = 0
      GC TC 100
C
C**** POSTLCGUE
C
  800 BACKSPACE 10
  801 NTP = 4
      STIMR(2) = STIMR(1)*1.0000001D0
      GO TC 240
C
C       EXIT
  998 IERR = 18
  999 RETURN
      END
```

Subroutine: ESTOUT

Purpose: To write the state estimate and covariance matrix on the system output tape and on the estimate tape.

Calling Sequence: CALL ESTOUT

Common storages used: // 20 cells, /DCPCØM/,/DCRCØM/,/ESTCØM/, /ES1CØM/,/SBFCØM/

Subroutines required: CØVØUT, DATØUP

```
$IBFTC MC1330  XR3,M94,NODD,LIST
      SUBROUTINE ESTOUT
C     OUTPUTS ESTIMATE AND COVARIANCE MATRIX
C
      COMMON           /DCPCOM/CDCP(900)
      EQUIVALENCE      (CDCP(777),ICB    ) ,(CDCP(117),NESEND)
     1                ,(CDCP(678),NCVOUT) ,(CDCP(742),NPRSTO)
C
      COMMON           /ESTCOM/CEST(804)
      DOUBLE PRECISION  ETIMVA,SPCDAN(6)
      EQUIVALENCE             (CEST( 27),KOUNTN) ,(CEST(  1),NESPOS)
     1    ,(CEST( 29),ETIMVA) ,(CEST(  2),NBY    ) ,(CEST( 49),SPCDAN)
C
      COMMON           /ESICOM/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,TBF
      EQUIVALENCE      (CSBF(2),NTP),(CSBF(6),ITER),(CSBF(9),TBF)
C
      COMMON           /DCRCOM/ZBAR(30),ZHAT(30),ZTOT(30),DCSAVE(74)
      DOUBLE PRECISION  ZBAR ,ZHAT ,ZTOT
C
      COMMON            D(30)
      DOUBLE PRECISION  D
C
      DIMENSION         NST(8),STN(6)
      DATA  NST/0,6,20,24,26,50,74,84/
      DATA  STN/36HX     Y     Z     XD    YD    ZD    /
  601 FORMAT(1H /18HOA PRIORI ESTIMATE/10H ETIMVA = ,D23.16)
  602 FORMAT(14HOITERATION NR ,I2/10H ETIMVA = ,D23.16)
  603 FORMAT(29HOESTIMATE NOT STORED ON TAPE /10H ETIMVA = ,D23.16)
  604 FORMAT(35HOCUMULATIVE DIFFERENTIAL CORRECTION)
  605 FORMAT(3(3X,A6,D24.16))
  606 FORMAT(15HOSTATE ESTIMATE)
  607 FORMAT(1H /13HOESTIMATE NR ,I3,15H STORED ON TAPE/10H ETIMVA = ,D2
     13.16)
C
C     IDENTIFY OUTPUT REQUIREMENTS
    1 GO TO (10,20,25,30) ,NTP
C
C     PROPAGATION END-POINT
   10 IF (NCVOUT.LE.0) GO TO 999
      GO TO 51
C
C     ESTABLISH ANCHOR POINT
   20 NBY = ICB+100
      ETIMVA = TBF
      DO 21 I=1,3
      SPCDAN(I)    = RBF(I)
   21 SPCDAN(I+3) = VBF(I,1)
      WRITE (6,601) ETIMVA
      CALL DATOUP (ETIMVA,D(6),0)
      GO TO 37
   25 WRITE (6,602) ITER,ETIMVA
      GO TO 35
C
C     STORE ESTIMATE ON TAPE
   30 IF (NPRSTO.NE.0) GO TO 31
      WRITE (6,603) ETIMVA
      GO TO 34
   31 NN = NESEND-NESPOS
      IF (NN.EQ.0) GO TO 33
      DO 32 I=1,NN
      READ (12) SKIP
   32 READ (12) SKIP
   33 NESPOS = NESEND+1
      NESEND = NESPOS
      WRITE (12) CEST
      WRITE (12) DEST
      WRITE (6,607) NESPOS,ETIMVA
```

```
   34 CALL DATCUP (ETIMVA,D(6),0)
   35 WRITE (6,604)
      DO 36 I=1,KOUNTN
   36 D(I) = ZTOT(I)+ZHAT(I)
      WRITE (6,605) (STNAMN(I),D(I),I=1,KOUNTN)
      IF (NTP.EQ.0) GO TO 999
      WRITE (6,606)
   37 WRITE (6,605) (STN(I),SPCDAN(I),I=1,6)
      NN = 1
      IF (KLOCN(1).EQ.1) NN=KLOCN(2)+1
      IF (KOUNTN.LT.NN) GO TO 50
      MM = NN
      DO 41 I=1,53,2
      K = KLOCN(I)
      L = KLOCN(I+1)
      IF (K-1) 42,41,38
   38 KK = 0
      IF (K.LT.100) GO TO 39
      KK = K-100
      K = 8
   39 JJ = NST(K)+14*KK
      DO 40 J=1,L
      M = JJ+ILOCN(MM)
      D(MM) = SPCDAN(M)
   40 MM = MM+1
   41 CONTINUE
   42 CONTINUE
      WRITE (6,605) (STNAMN(I),D(I),I=NN,KOUNTN)
C
C     OUTPUT COVARIANCE MATRIX
   50 WRITE (9) DEST
      BACKSPACE 9
      IF (NCVOUT.LE.0) GO TO 999
   51 DO 52 I=1,3
      D(I)   = SPCDAN(I)   +RBF(I+42)
   52 D(I+3) = SPCDAN(I+3)+VBF(I+42,1)
      CALL COVOUT (KOUNTN,D,NCVOUT)
C
  999 RETURN
      END
```

Subroutine: EXINST

Purpose: To read interpolation coefficients from a binary tape (as written by XØUT) and to reconstruct an N-dimensional vector X (and its derivative, if any) at time T.

Calling Sequence: CALL EXINST(T,M,N,X,XD,W,C,NB,ITRIG,KØUNT)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	T	d			Time, or any independent variable.
I/Ø	M				Vector number and indicator.
I/Ø	N				Vector dimension and indicator.
Ø	X	(N)			Vector array.
Ø	XD	(N)			Derivative array.
Ø	W	6K*			Internal storage.
Ø	C	As required			Critical information array.
I	NB				Binary tape number.
I	ITRIG				Trigger for KØUNT. KØUNT is altered only when ITRIG = 1.
I/Ø	KØUNT				Logical record counter. Reading forward increments KØUNT, backspacing decrements KØUNT.

Common storages used: None

Subroutines required: BACK

*K is the total of all vector dimensions (not including derivatives).

Method:

EXINST is designed to reconstruct vectors from information written on a binary tape by subroutine XØUT. The user should consult the XØUT write-up for details of method, format, etc.

X, XD, W, and C are arrays. X and XD must be large enough to hold the vector(s) generated. W is used internally by EXINST to store the interpolation coefficients read from tape. It must be dimensioned at least 6K where K equals the total of the dimensions of all vectors on the tape (exclusive of derivatives). C is the cirtical information array and must be dimensioned at least as large as the corresponding array as used by XØUT.

Usage

There are three distinct calls of EXINST:

1) Initialization
2) Normal
3) Limit shift.

The calls are illustrated in chart form below, followed by a detailed discussion of each.

IN				OUT
	Initial	Normal	Limit Shift	
T	Ignored	Time	New Limit	
M	Zero	Vector Number	Non-Zero	M > 0, normal, vector number M = 0, end of data M < 0, vector M not available
N	NC, the dimension of the C array	> 0	< 0	N > 0, normal, vector dimension N = 0, an error condition N < 0, the C array has just been filled (N = -1 tape read backward; N = -2 tape read forward)
X				Vector (if normal return)
XD				Derivative (if available and normal return)
W	Buffer	Buffer	Buffer	
C				Critical information if N < 0
NB	Logical Tape Number	Logical Tape Number	Logical Tape Number	

1) Initialization Call: This call must be the first call of EXINST for each case. It initializes certain parameters and reads in the critical information with which each case is started (or terminated if the case is being read in reverse). No vectors are generated by the initialization call.

2) Normal Call: This call will normally regenerate the specified vector at the specified time (and its derivative if available). If M and N are returned positive a vector has been generated, otherwise not. If M = 0 an end of the stored data has been reached.

If M is negative, the subroutine has been unable to find coefficients for the vector M. If N is negative a critical record has been read, i.e., the C array contains new information and no vector has been generated.

3) <u>Limit Shift</u>: Whenever a critical record is read by EXINST (N returned negative), the time associated with this record is stored internally in EXINST. This time is designated TC. Subsequent to a critical event the user may make a limit shift call of EXINST with time argument TN which will then be regarded by EXINST on subsequent normal calls as a new critical time. Essentially this device allows extrapolation at discontinuities occurring at critical points. Thus, if a critical event was preceded by a set of coefficients C1 applying over the interval (T1, TC) and followed by a set of coefficients C2 applying over the interval (TC, T2), a limit shift call of EXINST with time argument TN, where TN is in the interval (T1, TN) and set C2 to apply on the interval (TN, T2). TN was required to be in the interval (T1, T2) merely to make the illustration definite. It is in no way restricted. However, the user should bear in mind that EXINST is being forced to extrapolate, thus values of TN not in the interval (T1, T2) may cause EXINST to give poor results.

For certain applications the user may wish to use directly the information stored on the binary tape. In that instance, the W array may be regarded as output of EXINST. Its format is illustrated below.

	W(1, K)	W(2, K)	W(3, K)	W(4, K)	W(5, K)	W(6, K)
Vector W/Deriv.	M	N	x_1	$\dot{x}_1$	$\ddot{x}_{12}$	$\ddot{x}_{12}$
Vector W/O Deriv.	M	-N	x_1	--	x_2	--

```
$IBFTC MC13JY  NOREF,M94,NODD,XR3
CMC13JY  GETS VECTORS FROM TAPE  SAME AS EXIN EXCEPT FOR COUNTER
      SUBROUTINE EXINST(TTT,MMM,NNN,X,XD,W,C,NNB,ITRIG,KOUNT)
C T=TIME        M=VECTOR NUMBER          N=DIMENSION
C X=VECTOR      XD=DERIVATIVE            W=W(6,N1+...+NM)=STORAGE
C NB=BINARY TAPE NO.     C=C(NCRIT)=CRITICAL INFO BUFFER
C     KOUNT IS A LOGICAL RECORD COUNTER WHICH OPERATES WHEN ITRIG = 1
C     AND IS USED EXTERNALLY TO CONTROL BACKSPACING FOR TAPES ON WHICH
C     THE INDEPENDENT VARIABLE IS INCREASING
      DIMENSION X(3),XD(3),W(6,3),C(10),TAY(5)
      EQUIVALENCE (EM,MM),(EN,NN)
      DOUBLE PRECISION TTT,T,TL,TR,T1,T2,TC
      DATA TAY(1),TAY(2)/0.,1./
      T=TTT
      M=MMM
      N=NNN
      NB=NNB
  100 IF(M.EQ.0) GO TO 180
      IF(N.LE.0) GO TO 200
      ICRIT=0
  110 IF(T.LT.TL) GO TO 160
      IF(T.GT.TR) GO TO 170
      I=1
  115 EM=W(1,I)
      EN=W(2,I)
      N=IABS(NN)
      IF(MM.EQ.M) GO TO 120
      I=I+N
      IF(I.LE.KMAX) GO TO 115
      M=-M
      GO TO 900
C COEFFICIENTS FOR VECTOR M HAVE BEEN FOUND
  120 IF(NN) 125,900,140
C INTERPOLATE LINEARLY
  125 R=(T-T1)/(T2-T1)
      DO 130 J=1,N
      X(J)=W(3,I)+R*(W(5,I)-W(3,I))
      I=I+1
  130 CONTINUE
      GO TO 900
C USE FULL INTERPOLATION
  140 TAY(3)=T-T1
      TAY(4)=TAY(3)*TAY(3)/2.
      TAY(5)=TAY(4)*TAY(3)/3.
      DO 150 K=1,N
      X(K)=0.
      XD(K)=0.
      DO 145 J=1,4
      X(K)=X(K)+TAY(J+1)*W(J+2,I)
      XD(K)=XD(K)+TAY(J)*W(J+2,I)
  145 CONTINUE
      I=I+1
  150 CONTINUE
      GO TO 900
C INITIALIZATION
  180 M=1
      IDIR=0
      NCRIT=N
      GO TO 300
C TIME REQUESTED IS BELOW RANGE IN CORE
  160 IDIR=-INC
      GO TO 300
C TIME REQUESTED IS ABOVE RANGE IN CORE
  170 IDIR=INC
      GO TO 300
C THIS PATH SHOULD BE TAKEN ONLY WHEN RANGE IS BEING AJUSTED
  200 N=1
      ICRIT=1
C DETERMINE RANGE WITH NEW CRITICAL TIME
  210 IF(TC-T) 220,900,240
  220 IF(IDIR) 300,230,230
  230 IF(INC) 270,899,260
  240 IF(IDIR) 250,250,300
  250 IF(INC) 260,899,270
  260 TR=T
```

```
      GO TO 900
 270  TL=T
      GO TO 900
C READ IN A SET OF INFORMATION
 300  CONTINUE
      IF(IDIR) 305,308,308
 305  CONTINUE
      CALL BACK(NB,4)
      IF(ITRIG.EQ.1) KOUNT=KOUNT-4
 308  CONTINUE
      READ(NB) KEY,IKY,TIME
      IF(ITRIG.EQ.1) KOUNT=KOUNT+1
      IF(KEY.NE.3) GO TO 400
      KMAX=IKY
      READ (NB) T1,T2,((W(I,J),I=1,6),J=1,KMAX)
      IF(ITRIG.EQ.1) KOUNT=KOUNT+1
 995  FORMAT(2018,1P4E18.6)
      IF(T1.GT.T2) GO TO 310
      TL=T1
      TR=T2
      INC=1
      GO TO 320
C INDEPENDENT VARIABLE IS DECREASING
 310  TL=T2
      TR=T1
      INC=-1
 320  CONTINUE
      IF(IDIR) 350,900,350
 350  IF(ICRIT.LE.0) GO TO 110
      IDIR=-IDIR
      GO TO 210
C CRITICAL EVENT PROCEDURE
 400  IF(KEY-2) 999,405,999
 405  N=-1
      READ (NB) (C(I),I=1,NCRIT)
      IF(ITRIG.EQ.1) KOUNT=KOUNT+1
C IKY=+1 FIRST RECORD OF CASE, IKY=-1 LAST RECORD OF CASE
      IF(IKY) 410,430,420
 410  CONTINUE
      IF(IDIR) 305,305,415
 415  M=0
      CALL BACK(NB,2)
      IF(ITRIG.EQ.1) KOUNT=KOUNT-2
      GO TO 890
 420  CONTINUE
      IF(IDIR) 425,308,308
 425  M=0
      CALL BACK(NB,3)
      IF(ITRIG.EQ.1) KOUNT=KOUNT-3
      GO TO 900
C NORMAL CRITICAL EVENT
 430  CONTINUE
      IF(IDIR) 435,308,440
 435  TC=T1
      GO TO 900
 440  TC=T2
 890  CONTINUE
      N=-2
 900  MMM=M
      NNN=N
 899  CONTINUE
      RETURN
 999  CONTINUE
      N=0
      GO TO 900
      END
```

Subroutine: FIEF

Purpose: To space forward over N files, or backward over |N| + 1 files.

Calling Sequence: CALL FIEF(N,IER,IB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	N				If N > 0, FIEF spaces forward over N end-of-file marks. If N < 0, FIEF spaces backward and over |N| + 1 EØF marks then forward over the |N|+1 st.
Ø	IER				IER = 0, success exit. = - 1, end-of-tape encountered. = - 2, read check error.
I	IB				Logical tape number.

Common storages used: None

Subroutines required: None

```
$IBMAP MC1398
        ENTRY    FIEF
        REM      FIEF(N,IER,IB)
        REM      FIEF SPACES OVER N FILES, N + OR -
        REM      IGNORES IB, USES ONLY UNIT 10 FOR TRK DAT SIM
        REM      IER=0, SUCCESS EXIT
        REM      IER=2, READ ERROR (CHECK SUM)
FIEF    TXI      FFF,,1
FFF     SAVE     (1,2)I
        STZ*     4,4
        SXA      ER1,4
        CLA*     FIL
        STA      FILE
        CLA*     3,4
        PAX      0,2
        TZE      BCKUP
        TMI      BCKUP
FLE     TSX      .READ,4         SKIP N FILES
FILE    PZE      **
        PZE      EOF,,ERR
        IORPN    0,,0
        TCH      *-1
ERR     CLS      =2              IER=-2, READ ERROR
ER1     AXT      *,4
        STO*     4,4
        RETURN   FFF
EOF     TIX      FLE,2,1
        RETURN   FFF             SUCCESS EXIT, FILE SKIP
BCKUP   TXI      *+1,2,1         BACKSPACE N+1 FILES
        PXD      0,2
        ORA      FILE
        SLW      BKP
        TSX      .BSF,4
BKP     PZE      0,,0
        RETURN   FFF
FIL     PZE      .UN10.
        END
```

Subroutine: FIFL

Purpose: To locate a specified case on the binary tape, and to read record 1 of that case.

Calling Sequence: CALL FIFL (ICASE,J,REM,IER,IB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	ICASE				Case number. If ICASE = 0, FIFL initializes and does not search.
Ø	J				J = REM(2) = number of logical records per set.
Ø	REM	(255)			Vector written by TPFL.
Ø	IER				IER is set ≠ 0 if case cannot be found.
I	IB				Logical tape number.

Common storages used: None

Subroutines required: FIEF, ERRØUT

```
$IBFTC MC1397  XR3,LIST
      SUBROUTINE FIFL(ICASE,J,IREM,IER,IB)
      DIMENSION IREM(255)
C     FIFL      FINDS AND READS SET 0 OF CASE ICASE.
C               ICASE = 0 INITIALIZES AND DOES NOT READ.
C               CALLS FIEF TO SPACE OVER FILES TO READING POSITION.
C               CALLS ERROUT IF ERROR FROM FIEF,OR IF 2 TRIES TO LOCATE
C                    ICASE FAIL.
C               ASSUMES CASE NUMBERS SEQUENTIAL AND INCREASING ON TAPE.
C               (SEARCHES FORWARD OR BACKWARDS).
      IF(ICASE.NE.0) GO TO 50
      ICSV =0
      RETURN
 50   IER=0
      ISWT=0
75    CALL   FIEF(ICASE-ICSV,IER,IB)
      IF(IER.NE.0)GO TO 150
      READ(IB)IREM
      ICSV= IREM(1)
      J=IREM(2)
      IF(ICASE.EQ.ICSV) RETURN
      ISWT =ISWT+1
      GO TO(75,100),ISWT
 100  CALL  ERROUT(1,29H(5X,4HCASE,I5,11HNOT ON TAPE),ICASE)
      RETURN
 150  CALL  ERROUT(1,38H(5X,27HERROR EXIT FROM READ, IER = I2),IER)
      RETURN
      END
```

Subroutine: FIST

Purpose: To rewind the binary input tape and to read and check the header (first file). Also initializes subroutine FIFL.

Calling Sequence: CALL FIST (HEAD,IER,IB)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	HEAD	(12)			Header to be checked against the first file of tape. The header from the tape will be in blank common on exit.
Ø	IER				IER=2 if HEAD does not agree. Otherwise it is returned unchanged.
I	IB				Logical tape number.

Common storages used: 12 cells

Subroutines required: FIFL

```
$IBFTC MC1396  XR3
      SUBROUTINE FIST(IHED,IER,IB)                                      FIST000
      COMMON IR                                                         FIST001
      DIMENSION IHED(12),IR(12)                                         FIST002
C     FIST     REWINDS TAPE IB                                          FIST003
C              READS HEADER  AND CHECKS                                 FIST004
C              RETURNS IER =2  IF HEADER DOES NOT CHECK                 FIST005
C              RETURNS IER UNTOUCHED IF HEADER CHECKS                   FIST006
C              INITIALIZES FIFL.                                        FIST007
C               TAPE HEADER IS IN FIRST 12 CELLS OF BLANK COMMON        FIST008
      REWIND IB                                                         FIST009
      READ(IB) IR                                                       FIST010
      DO  10 I =1,12                                                    FIST011
      IF( IHED(I).NE.IR(I))  GO TO 25                                   FIST012
 10   CONTINUE                                                          FIST013
20    CALL  FIFL(0,IR,IR,IR,IB)                                         FIST014
      RETURN                                                            FIST015
 25   IER =2                                                            FIST016
      GO TO 20                                                          FIST017
      END                                                               FIST018
```

Subroutine: FNØRM

Purpose: Computes the magnitude of a 3-vector.

Calling Sequence: Z = FNØRM (X)

Input and Output

I/Ø	Symbolic Name or Location	Program Dimensions	Math Symbol	Data Dimensions or Units	Definition
I	X	(3)			Input vector
Ø	Z				X

Common storages used: None

Subroutines required: None

```
$IBFTC FNORM    XR3,M94,NODD
      FUNCTION FNORM(X)                                                 FNRM0001
      DIMENSION  X(3)                                                   FNRM0002
      SUM = 0.                                                          FNRM0003
      DO 1 I=1,3                                                        FNRM0004
    1 SUM = SUM+X(I)*X(I)                                               FNRM0005
      FNORM = SQRT(SUM)                                                 FNRM0006
      RETURN                                                            FNRM0007
      END                                                               FNRM0008
```

Subroutine: .FVIØ.

Purpose: To obtain IØCS pointer word address for variable tape units. This is a replacement for the IBSYS library routine.

Calling Sequence: None

Common storages used: None

Subroutines required: None

For read or write statements using variable tape logical numbers, such as

WRITE(N,FØRMAT)LIST

the IØCS pointer word is obtained by reference to a table in the library routine .FVIØ.. The library routine contains pointers (.UNOn.) for all logical units from 1 through (usually) 12, and this results in the attachment of buffers for each unit. By removing entries for all units which are not required from .FVIØ., we release the associated buffer space for use by the program.

In the furnished deck, each unused unit in the table is replaced by the dummy name FILE. To add unit n to the table, the symbol .UNOn. replaces the symbol FILE in the nth entry.

```
$IBMAP MC13IO  25
       TTL     FVIO - SET FILE ADDRESS FROM LOGICAL UNIT NO                  FVIO
.FVIO. SAVE                                                                  FVIO0001
       CLA*    3,4             LOGICAL NO                                    FVIO0002
       TZE     ERR             NO.EQ.0                                       FVIO0003
       PAX     ,4                                                            FVIO0004
       SXD     ARG,4                                                         FVIO0005
       TXL     OUT,4,MAX                                                     FVIO0006
ERR    CALL    .FXEM.(ARG)     -LOGICAL UNIT NOT DEFINED FOR VALUE XX-       FVIO0007
       CALL    EXIT                                                          FVIO0008
OUT    PAC     ,4                                                            FVIO0009
       CLA*    FILE,4                                                        FVIO0010
       TZE     ERR             FILE NOT DEFINED                              FVIO0011
       LXA     SYSLOC,4                                                      FVIO0012
       STO*    4,4             FILE ADDRESS                                  FVIO0013
       RETURN  .FVIO.                                                        FVIO0014
FILE   PZE                     FILE NAMES FOLLOW                             FVIO0015
       PZE     FILE            UNIT 1                                        FVIO0016
       PZE     FILE                                                          FVIO0017
       PZE     FILE                                                          FVIO0018
       PZE     FILE                                                          FVIO0019
       PZE     .UN05.                                                        FVIO0020
       PZE     .UN06.                                                        FVIO0021
       PZE     FILE                                                          FVIO0022
       PZE     FILE                                                          FVIO0023
       PZE     FILE                                                          FVIO0024
       PZE     .UN10.                                                        FVIO0025
       PZE     .UN11.                                                        FVIO0026
       PZE     .UN12.                                                        FVIO0027
ARG    PZE     47                                                            FVIO0028
MAX    SYN     ARG-FILE-1                                                    FVIO0029
       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.