<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>i</td>
</tr>
<tr>
<td>1.0 MATHEMATICS AND PROGRAMMING DESCRIPTIONS OF GEODYN SUPPORT PROGRAMS</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1 GEODYN ANALYSES AND GRAPHICS SUPPORT PROGRAMS</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1.1 DELTA</td>
<td>1-2</td>
</tr>
<tr>
<td>1.1.2 GEORGE</td>
<td>1-28</td>
</tr>
<tr>
<td>1.1.3 GROUNDTRACK</td>
<td>1-69</td>
</tr>
<tr>
<td>1.1.4 WRDC SC4020 Plot Package</td>
<td>1-113</td>
</tr>
<tr>
<td>1.2 GEODYN DATA HANDLING SUPPORT PROGRAMS</td>
<td>1-208</td>
</tr>
<tr>
<td>1.2.1 DODS Sort-Merge</td>
<td>1-209</td>
</tr>
<tr>
<td>1.2.2 GEOS Sort-Merge</td>
<td>1-224</td>
</tr>
<tr>
<td>1.2.3 Ephemeris Tape Generator</td>
<td>1-239</td>
</tr>
<tr>
<td>1.2.4 ORB1 Conversion</td>
<td>1-281</td>
</tr>
<tr>
<td>1.2.5 TDIF Table Generator</td>
<td>1-288</td>
</tr>
<tr>
<td>2.0 OPERATIONS DESCRIPTION OF GEODYN SUPPORT PROGRAMS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1 GEODYN ANALYSES AND GRAPHICS SUPPORT PROGRAMS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1.1 DELTA</td>
<td>2-2</td>
</tr>
<tr>
<td>2.1.1.1 DELTA Input Cards</td>
<td>2-3</td>
</tr>
<tr>
<td>2.1.1.2 DELTA Job Control Language and Hardware and Software Restrictions</td>
<td>2-5</td>
</tr>
<tr>
<td>2.1.1.3 DELTA Example Job</td>
<td>2-7</td>
</tr>
<tr>
<td>2.1.1.4 DELTA Error Messages</td>
<td>2-8</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>2.1.2 GEORGE</td>
<td>2-9</td>
</tr>
<tr>
<td>2.1.2.1 GEORGE Input Cards</td>
<td>2-10</td>
</tr>
<tr>
<td>2.1.2.1.1 Measurement Card</td>
<td>2-11</td>
</tr>
<tr>
<td>2.1.2.1.2 EL CUT</td>
<td>2-12</td>
</tr>
<tr>
<td>2.1.2.1.3 HISTGM</td>
<td>2-13</td>
</tr>
<tr>
<td>2.1.2.1.4 PLOT</td>
<td>2-14</td>
</tr>
<tr>
<td>2.1.2.1.5 REJECT</td>
<td>2-15</td>
</tr>
<tr>
<td>2.1.2.1.6 Option Termination Card</td>
<td>2-16</td>
</tr>
<tr>
<td>2.1.2.1.7 Case Termination Card</td>
<td>2-17</td>
</tr>
<tr>
<td>2.1.2.2 GEORGE Job Control Language and Hardware and Software Restrictions</td>
<td>2-18</td>
</tr>
<tr>
<td>2.1.2.3 GEORGE Example Job</td>
<td>2-20</td>
</tr>
<tr>
<td>2.1.2.4 GEORGE Error Messages</td>
<td>2-21</td>
</tr>
<tr>
<td>2.1.3 GROUNDTRACK</td>
<td>2-22</td>
</tr>
<tr>
<td>2.1.3.1 GROUNDTRACK Input Cards</td>
<td>2-23</td>
</tr>
<tr>
<td>2.1.3.1.1 Station Position Cards</td>
<td>2-24</td>
</tr>
<tr>
<td>2.1.3.1.2 Station Position Termination Card</td>
<td>2-25</td>
</tr>
<tr>
<td>2.1.3.1.3 GRDSET</td>
<td>2-26</td>
</tr>
<tr>
<td>2.1.3.1.4 LNDPLT</td>
<td>2-27</td>
</tr>
<tr>
<td>2.1.3.1.5 PLOTS</td>
<td>2-28</td>
</tr>
<tr>
<td>2.1.3.1.6 TIME</td>
<td>2-29</td>
</tr>
<tr>
<td>2.1.3.1.7 Option Termination Card</td>
<td>2-30</td>
</tr>
<tr>
<td>2.1.3.1.8 Case Termination</td>
<td>2-31</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (Cont.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.3.2 GROUNDTRACK Job Control Language and Hardware and Software Restrictions</td>
<td>2-32</td>
</tr>
<tr>
<td>2.1.3.3 GROUNDTRACK Example Job</td>
<td>2-34</td>
</tr>
<tr>
<td>2.1.3.4 GROUNDTRACK Error Messages</td>
<td>2-35</td>
</tr>
<tr>
<td>2.1.4 WRDC SC4020 Plot Package</td>
<td>2-36</td>
</tr>
<tr>
<td>2.2 GEODYN DATA HANDLING SUPPORT PROGRAMS</td>
<td>2-38</td>
</tr>
<tr>
<td>2.2.1 DODS Sort-Merge</td>
<td>2-39</td>
</tr>
<tr>
<td>2.2.1.1 Job Control Language and Job Submittal</td>
<td>2-40</td>
</tr>
<tr>
<td>2.2.1.2 DODS Sort-Merge Printer Output</td>
<td>2-42</td>
</tr>
<tr>
<td>2.2.2 GEOS Sort-Merge</td>
<td>2-45</td>
</tr>
<tr>
<td>2.2.2.1 Job Control Language and Job Submittal</td>
<td>2-46</td>
</tr>
<tr>
<td>2.2.2.2 GEOS Sort-Merge Printer Output</td>
<td>2-49</td>
</tr>
<tr>
<td>2.2.3 EPHEMERIS TAPE GENERATOR</td>
<td>2-51</td>
</tr>
<tr>
<td>2.2.3.1 Input Card</td>
<td>2-53</td>
</tr>
<tr>
<td>2.2.3.2 Job Control Language</td>
<td>2-54</td>
</tr>
<tr>
<td>2.2.4 ORB1 Conversion (9-7 track)</td>
<td>2-55</td>
</tr>
<tr>
<td>2.2.5 TDIF TABLE GENERATOR</td>
<td>2-56</td>
</tr>
<tr>
<td>2.2.5.1 Data Deck</td>
<td>2-57</td>
</tr>
<tr>
<td>2.2.5.2 Job Control Language</td>
<td>2-59</td>
</tr>
</tbody>
</table>
INTRODUCTION

The GEODYN Orbit Determination and Geodetic Parameter Estimation System consists of a set of computer programs designed to determine and analyze definitive satellite orbits and their associated geodetic and measurement parameters. This manual describes the Support Programs used by the GEODYN System. The mathematics and programming descriptions are detailed in the first section. The second section contains the operational procedures of each program.

GEODYN ancillary analysis programs may be grouped into three different categories:

1. Orbit Comparison - DELTA

2. Data Analysis using Reference Orbits - GEORGE

3. Pass Geometry Computations - GROUNDTRACK

All of the above three programs use one or more tapes written by the GEODYN program in either a data reduction or orbit generator run. Although it is not necessary, these programs are generally run immediately following the associated GEODYN run, thus minimizing tape handling problems. In addition all three programs use the WRDC PLOT PACKAGE and can produce a graphical depiction of their results both on printer plots and on SC4020 microfilm or hardcopy plots.
In addition to the above analysis programs, the GEODYN System contains five data management routines:

1. Sort-merge program in DODS format - DODS SORT-MERGE
2. Sort-merge program in GEOS format - GEOS SORT-MERGE
3. EPHEMERIS TAPE GENERATOR
4. 9-7 Track conversion - ORB1 CONVERSION
5. TDIF TABLE GENERATOR

The flowchart on the following page depicts the structure of the entire GEODYN System.
GEODYN-SYSTEM FLOWCHART

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

[Flowchart diagram showing the process of GEODYN system, including inputs, outputs, and various steps like tape handling, sorting, merging, and printing.]
SECTION 1.0
MATHEMATICS AND PROGRAMMING
DESCRIPTIONS OF THE GEODYN SUPPORT PROGRAMS

1.1 GEODYN ANALYSES AND GRAPHICS SUPPORT PROGRAMS

There exist three ancillary programs, DELTA, GEORGE, and GROUNDTRACK, which are used with the GEODYN program in the analysis of GEODYN determined trajectories and residuals. These programs are entirely independent of the GEODYN program. All three use as input GEODYN generated data files, thus, usually they are run as a second job step after a GEODYN run.

DELTA is used to print and/or plot along-track, cross-track and radial differences between two trajectories. It differences orbits of the same satellite for the same time period but generated with different values for certain parameters or reduced over different data spans.

GEORGE performs a regression analysis of the residuals for each pass of data about a trajectory to determine trends in possible timing and measurement biases.

GROUNDTRACK simply plots the groundtrack of the satellite over a particular tracking station or stations to provide geometric insights into data trends.

All three programs will optionally produce printer and/or SC4020 plots to illustrate the computed results. Hence the WOLF PLOT PACKAGE must be included when using these programs.
INTRODUCTION

The graphic support program DELTA prints and/or plots trajectory differences. The two trajectories enter the program from two magnetic tapes in either an R-V tape format or ORB1 tape format. If the tapes are in the ORB1 format the subroutine RDORB1 is called to obtain each trajectory point; DELTA itself can read the R-V tapes. The subroutine READER is the driver for the sequence of calls to the Plot Package, which provide the plots of the trajectory differences.

DELTA uses the DSQRT, MOD, and FLOAT system routines and approximately 250K bytes of core. The program will difference 1400 time points of two orbits in less than three minutes of CPU time.

Subroutine and common block cross reference charts appear in this section. The calling routines are at the top of the subroutine chart and the common blocks are listed down the side of the common block chart.

The routines in the Plot Package are all in G and H level FORTRAN with the exception of TIMING which is in IBM 360 Assembly Language. These routines were designed to be efficient on the IBM 360 series machines; no attempt whatever has been made to pursue the myth of compatibility.
The trajectory tapes input to DELTA consist of the satellite positions \((X_1, Y_1, Z_1)\) and velocities \((\dot{X}, \dot{Y}, \dot{Z})\) in the Cartesian system at given time intervals.

If \(X_1, Y_1, Z_1\) are the Cartesian coordinates of satellite position from tape 1 and \(X_2, Y_2, Z_2\) are the coordinates from tape 2 then the position difference vector is

\[
\Delta \vec{P} = (\Delta X = X_2 - X_1, \Delta Y = Y_2 - Y_1, \text{ and } \Delta Z = Z_2 - Z_1).
\]

The velocity difference vector \(\Delta \vec{V} = (\Delta \dot{X}, \Delta \dot{Y}, \Delta \dot{Z})\) is computed similarly.

These vectors are then resolved into a radial vector, \(R\), a cross-track vector \(C\), and an approximation to an along-track vector, \(L\) (for nearly circular orbits).

First the distance from the geocenter to the satellite, \(R\), is computed where

\[
R = \sqrt{X^2 + Y^2 + Z^2}
\]

and the square of the magnitude of the velocity vector \((\vec{V})\),

\[
V^2 = \dot{X}^2 + \dot{Y}^2 + \dot{Z}^2.
\]
Thus the unit vector, $\hat{U}$, in the radial direction is

$$\hat{U} = \left( \frac{X}{R}, \frac{Y}{R}, \frac{Z}{R} \right).$$

Then to calculate the magnitude of the vector in our along-track direction (normal to $\hat{U}$ in the orbit plane), $A$, we must compute $\hat{U} \cdot V$ because

$$A = \sqrt{V^2 - (\hat{U} \cdot V)^2}.$$

Now we compute the unit vectors in our along-track direction $\vec{a} = (a_1, a_2, a_3)$ where

$$a_1 = \left( \frac{\dot{X}_2 - (\hat{U} \cdot V) \left( \frac{X}{R} \right)}{A} \right)$$

$$a_2 = \left( \frac{\dot{Y}_2 - (\hat{U} \cdot V) \left( \frac{Y}{R} \right)}{A} \right)$$

$$a_3 = \left( \frac{\dot{Z}_2 - (\hat{U} \cdot V) \left( \frac{Z}{R} \right)}{A} \right)$$

and the cross-track direction $\vec{c} = (c_1, c_2, c_3)$ where

$$\vec{c} = \vec{a} \times \hat{U}$$

or

1-4
Finally we compute the position differences in radial, $H_p$, cross-track $C_p$, and approximation to along-track, $L_p$:

$$H_p = \hat{U} \cdot \Delta \overrightarrow{p}$$

$$C_p = \overrightarrow{C} \cdot \Delta \overrightarrow{p}$$

$$L_p = \overrightarrow{L} \cdot \Delta \overrightarrow{p}$$

and the velocity differences in the radial, $H_v$, cross-track, $C_v$, and approximation to along-track, $L_v$:

$$H_v = \hat{U} \cdot \Delta \overrightarrow{V}$$

$$C_v = \overrightarrow{C} \cdot \Delta \overrightarrow{V}$$

$$L_v = \overrightarrow{L} \cdot \Delta \overrightarrow{V}$$
DESCRIPTION

The main routine DELTA reads data from two RV tapes or receives data from the routine RDORB1, calculates and prints radial, cross-track, and along-track differences, and calls READER to make plots if requested.
NAME: MAIN-Delta

PURPOSE: DIFFERENCES GLODYN GENERATED RV ON CRIB TAPES

SOURCES USED: RDORBI READER

COMMON BLOCK: PLOTTTP

INPUT FILES: INPT  - DELTA INPUT CARDS
              RVTAP1 - RV TAPE1
              RVTAP2 - RV TAPE2

OUTPUT FILE: OUTP - PRINTER

RESTRICTIONS: NONE

REFERENCES: NONE

DOUBLE PRECISION XYZEN1(6),XYZEN2(3),DSORT,DX(3),DXOUT(3),U(3),  DELT  22
* R2,R3,V2,UDOTV,UDOTV2,AT(3),C(3),J2(3),DV(3),V30  DELT  23
REAL GAT31,DAYS2,YMO1,YMO2,HS1,HS2,EUF,SUMPOS,SUMVEL,DAY2(2),  DELT  24
   SCP(3),SVD(3),XIND,TITLE
REAL NTRVL
DIMENSION DOR(3)
LOGICAL IFLOT,ISW1,ISW2
DOUBLE PRECISION DELTAT
LOGICAL CFU1,LASTS*
INTEGER RVTAP1,RVTAP2,INTP,OUTF
DATA INF,OUTP,RVTAP1,RVTAP2/5,6,21,22/  DELT  23
DATA ECF,ISW1,ISW2/C,9550.3,2*,FALSE,*/  DELT  24
DATA URO1/FAUL*/  DELT  25
COMMON/PLOTTTP/DAYS(4),FACL(4),CTHK(4),ATHK(4),TITLE(21),  DELT  25
   EPOCH(2),INDEX,NOPT,SCALE(2),NTRVL
DATA DAY6,SOR,SDV,XIND/3.66D2,3.66D2,3.66D2,3.66D2/  DELT  26
CATA NUM/C/
C DATA INDEX=0
C C DETERMINE INFLT TAPE UNIT NUMBERS, PLOTTING OPTIONS AND SCALES, TYPE OF  DELT  45
   READ(INFT,1500) IRV1,IRV2,ILOT,NOPT,NORBI,NUM1,ILAST,SCALE,NTRVL  DELT  46
   NUM1=MAXC(1,NUM1)  DELT  47
   IF(NOPT.LT.1. OR NORBI.NOPT,GE.7) NOP1=7  DELT  48
C SET CDRU1 SWITCH  DELT  49
   CDRU1=NLRE1,GT.C  DELT  50
   LASTS=ILAST,ED*=0  DELT  51
C RSET RV TAPE UNITS IF REQUEDED  DELT  45
   IF(IRV1.LT.0) RVTAP1=IRV1  DELT  46
   IF(IRV2.LT.0) RVTAP2=IRV2  DELT  47
   RE=INO RVTAP1  DELT  48
   RE=INO RVTAP2  DELT  49
   MDU1=1  DELT  50
   IF(UMD1) U TO 49  DELT  51
C READ FIRST DATA RECORD  DELT  52
   READ(RVTAPI) GAT31,YMO1,HS1,SEC,XYZEN1

ILLUSTRATIONS LIEPO SLIDE
READ (HVTAF2) CAYS2, IYMD2, IHM2, SEC2, XYZEN2

DO 1 CONTINUE

IYMD1 = (IYMD2 * 100) + IFIX (SEC1)
IHM2 = (IY2 * 10) + IFIX (SEC2)
N = 1

C WRITE EPOCH AND ELEMENTS OF RV TAPE 1
C WRITE (OUTF, 3, C) N, IYMD1, IHMS1, XYZEN1
N = 2
C WRITE EPOCH AND ELEMENTS OF RV TAPE 2
C WRITE (OUTF, 3, C) N, IYMD2, IHMS2, XYZEN2
IF (NOT PLOT) GO TO 2
C READ TITLE IF PLOT IS REQUESTED
READ (INF, 99) TITLE
C CALCULATE EPOCH IN YEAR, MONTH, DAY, HOUR, MINUTE, SECOND FOR PLOT
IEPCH(1) = IYMD2
IEPCH(2) = IHMS2
2 N = 0
C WRITE (OUTF, 3, C) I
IY1 = IYMD1 / 10000
IY2 = IYMD2 / 10000
IF (IY1 .LT. IY2) GO TO 3
LI = MIN (400 ((IY1,4),1)+1
IS = 1, TRUE
GO TO 1
C READ READER RECORDS ON ORBI TAPES
10 IF (ORBI) GO TO 50
C READ RV DATA RECORD
READ (HVTAPI) CAYS1
READ (HVTAP) CAYS2
C READ RV DATA RECORD
READ (HVTAPI) CAYS1, IYMD1, IHM1, SEC1, XYZEN1
READ (HVTAP) CAYS2, IYMD2, IHM2, SEC2, XYZEN2
42 CONTINUE
IF (IS = 1) CAYS1 = DAYS1 + DAYR(L1)
IF (IS = 2) CAYS2 = DAYS2 + DAYR(L1)
10 IF (DAYS1 .LE. EOF OR DAYS2 .LE. EOF) GO TO 330
IF (DAYS1 .LT. DAYS2 - ED - 6) GO TO 25
IF (DAYS2 .LT. DAYS1) GO TO 15
C READ DATA
IF (NOT (CFBI)) READ (RVTAFT) DAYS1, IYMD1, IHM1, SEC1, XYZEN1
IF (DIRECT) (ALL FORDRV1(DAYS1, XYZEN1, RVTAFT1, IYMD1, IHM1, SEC1)
IF (IS = 1) CAYS1 = DAYS1 + DAYR(L1)
GO TO 10
C READ DATA
15 IF (NOT (CFBI)) READ (RVTAFT) DAYS2, IYMD2, IHM2, SEC2, XYZEN2
IF (DIRECT) (ALL FORDRV1(DAYS2, XYZEN2, RVTAFT2, IYMD2, IHM2, SEC2)
IF (IS = 2) CAYS2 = DAYS2 + DAYR(L1)
GO TO 10
25 IF (INDEX.EQ.400) GO TO 330
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR.
MAT N- DELTA
Page 5 of 9
30 September 1972

REPRODUCIIBILITY'OF TE
oRiGUi4AL PAGE IS POOR_._

SVVLDC-T(UV-LXIC
bIT L(CUT Ps J C3 )

C.:LT
r).LT

16~
I

tLLT

I?

01-zI-T

0O-LT
DELT
t EL T

1 '2
173
17A
175

CtLT

1 76

--:L T
OL T
D-

177
17-3

Ct:LT

1 i

12.1 193F12.5)

Ci:LT
DELT
CELT
0E!L T

. 5)

DThLT

D'-LT
U=LT

I ci
1!3
I 3
1e7
I3A
1 !3'
I'

2L T

1i2

SC109SLMPCS SDV SU-WiVL

hm%2
i C-WN
IFtNjT.IFL!-T) GO To 5CC
C
C

'MAKM OPTIUNAL d-LUT AND/'3PF LOT
CALL dEALER(OLLTAT. LAST1S*
Ic N6T LtN0FL[Lo.kL-:AD DATA
STOP
SCC IF (L AS T3>
= 0
N ull

TAPE

IS*1=*FALEE

CO*51C

1=1.2

CALL kDOI-,E ( DAYS.
GO TU oL.C
59', FUjkMAT(7AE)
IlC -'C FUkMAT1229~L1 .211.
1031 Fu~rN.AT (3 L 10 . 5/etD IC
3C')2 i- M AT I - 9 16*1 7 91
POCM
33001.. F U R?'A T(I1-I.

FOPMATI It- 1.Z -=
IC-E

SIf-

. RVTAP I t-

3x . 2(

fAi

T

I

12 1 H 0

1 14HXDOT

12XIH2

IT ION

X -2rI-PJS

[)

X'Y EA F

-3X4HYDL)T

'D Fc

C
I(TTv

2t CZtt

tS

D

D

T9E

YYNIMD0)
O

7

M CNT H

9XA~iZDOT/

)LT

F;PFNJCFS.~X

:-:H
.- tv
27:(
2:2
1 w -y.
;E C 5
)/CELT
95H CATAt3X9,.,1(2HJ
2~3HCCM/SEC))/IH
*86X5HCPCSS5X5HALCNG.1-3X9
,'%

bt-CROSS95X9.:HALCN./1t'H

b
6

9IYM)1.IHMI 9SECfl

1 F I' .2 , Fq. 2
A ND E L E 4E N T
12XIHY

I -C~lIX1HX
)e43(f(i

XYZEN

k

5-

T:

-

HHV~MSsuoX92HDX%-3Xs2HDYs8X9
L94
'H<A
4
-iZU

5H7%9ACK .5X,5HT'PACK.tX.15HPADIAL,4X,5HTRACK,5X.5-HTRACK/)
3C03 FO lAT //4e'.SXs'4S OF POSI1Ut, AND) VELOCIrY DIFrERFNC:-S//1S5X,
OC ITY 0 1F-zER ENCES I
PCJS IT ICN C IFFE kE NC ES (MET = S) I,42 ,VL
I1
1XI(CI/SEC)'//4X,'P;ACIAL,97('C.-)*SS TRACK',4X0ALONG TRACK',
2
7XoT.CTAL,919X,'PADIAL'.7X.'CROS'. TRACK'v4X9'AL0tJG TRACK'97Xv
3
ITOTAL'//F1C .2,3F15.2,15X,2-F1C.*2,3F15.2)
4
END

Ic

'-JLT

!o;

DLT

1965
19)7

D1-L T

I n-

-LT
D.D

119

Ot LT

21 :l

C'zL T

;^

DEL T
CELT
CELT
CELT
CELT

2C2
2f3
21)4
2Cn
2C6

I


DESCRIPTION

The subroutine ADTIME updates the time of the measurement by the number of seconds between each data point, calling ADDYMD to recomputed the date when necessary.
NAME          ADTIME

PURPOSE       CONVERTS HOURS, MINUTES, SECONDS TO DAYS

CALLING SEQUENCE CALL ADTIME(IYMD,IM,SEC)

SYMBOL TYPE   DESCRIPTION

IYMD          INPUT - YEAR,MONTH,DAY IN FORM YTMDD

IM            INPUT - HOUR,MINUTE IN FORM HHMM

SEC           INPUT - SECOND

SUBROUTINE USED ADTNYMD

COMMON BLOCKS NONE

INPUT FILES  NONE

OUTPUT FILES NONE

RESTRICTIONS NONE

REFERENCES   NONE

SUBROUTINE ADTIME(IYMD,IM,SEC)

C CALCULATE MINUTES
20  IM=SEC/60.
   IF(SEC.LT.0.)IM=IM-1
   SEC=SEC-60.*FLOAT(IM)
   IM=IM-40.*(IM/100)+IM

C CALCULATE DAYS
   ID=IM/1440
   IF(ID.LT.1.ID=ID-1

C CALCULATE HOURS,MINUTES
   IM=IM-1440
   IM=IM-40.*(IM/60)

C ADD DAYS TO DATE
   IF(ID.NE.0) CALL ADDYMD(IYMD,IM)
   RETURN

END
ENTRY

CALCULATE MINUTES

CALCULATE DAYS

COMBINE HOURS AND MINUTES

DAYS

NUMBER OF DAYS = 0

ADDYMD ADD DAYS TO DAY

RETURN
DESCRIPTION

RDORB1 reads a record of 50 data points from one of two ORB1 tapes and stores them, returning one point to the calling program. One point is returned for each subsequent call to RDORB1 for a specific tape until it is necessary to read another record.
PURPOSE
READS ORB1 TAPES

CALLING SEQUENCE
CALL RDORB1(TIME,XYZ,ORB1,N,IYMDA,IMHA,ASEC)

SYMBOL TYPE DESCRIPTION
TIME OF OUTPUT - NUMBER OF DAYS FROM EPOCH
XYZ OF OUTPUT - COORDINATES OF POSITION AND VELOCITY
ORB1 I INPUT - UNIT NUMBER OF RV TAPE
N I INPUT - RV TAPE INDICATOR (1 OR 2)
IYMDA I OUTPUT - YEAR,MONTH,DAY OF COORDINATES
IMHA I OUTPUT - HOUR,MINUTE OF COORDINATES
ASEC R OUTPUT - SECONDS OF COORDINATES

SUBROUTINE USED ADFMTIME
COMMON BLOCKS NONE
INPUT FILE ORBI - ORB1 TAPE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES NONE

SUBROUTINE RDORB1(TIME,XYZ,ORB1,N,IYMDA,IMHA,ASEC)
READ Buf(5),BUF2(5),BUF(5,2),ELEMS(6,50),ELEMS2(6,50),
I ELEMS(5,50,2),DELTAT(2),DAYS(2),SEC,TIME,XYZ(6),EDF
INTEGER CFB1
DIMENSION M(2),NOT1ST(2),IYMD(2),IMH(2),SEC(2)
LOGICAL NOTIST
EQUIVALENCE (BUF1,BUF1),(BUF2,BUF1), (ELEMS1,ELEMS1),
 I (ELEMS2,ELEMS(1,1,2))
CATA M/2*50/
DATA NOT1ST/2*FALSE/ DATA ELF/5995999903/
IF(N-LT-0) GO TO 25
C TEST IF ARRAY IS EMPTY
IF(M(N).LT.-0) GO TO 5
C READ COORDINATES INTO AN ARRAY
IF(N.EQ.1) READ(ORB1,END-20) Buf1,ELEMS1
IF(N.EQ.2) READ(ORB1,END=20) Buf2,ELEMS2

RDOR 39
RDOR 40
RDOR 41
RDOR 42
RDOR 43
RDOR 44
RDOR 45
RDOR 46
RDOR 47
RDOR 48
RDOR 49
RDOR 50
RDOR 51
RDOR 52
RDOR 53
RDOR 54
RDOR 55
M(N)=G

C CALCULATE TIME OF FIRST SET OF COORDINATES
   IF(ULF(1,1)<ULF(4,N)) GO TO 2
   IYM(N)=ULF(1,N)
   SEC(N)=ULF(3,N)-ULF(4,N)
   IH=INT(SEC(N)/3)+3
   IM=IDINT(SEC(N)-DFLOAT(1H/100)*3+0.03)
   I=SEC(N)-DFLOAT(1H/100)*3+0.03
   IHM(N)=ITM(N)
   IF(NOTIST(N)) GO TO 5

C CALCULATE DAYS FROM EPOCH
   DAYS(N)=ULF(2,N)+ULF(3,N)/3.644

C CALCULATE TIME DIFFERENCE BETWEEN COORDINATE SETS
   DELTAT(N)=ULF(4,N)/8.644
   NOTIST(N)=.TRUE.
   S(N)=4(N)+1
   YSM(N)=ITM(N)

C FILL OUTPUT VECTORS
   CALL ACT(I,YSM(N),IHM(N),SEC(N))
   YM(N)=YSM(N)
   ASCC=SEC(N)

C TEST FOR ENDFILE
   IF(XYZ(1).EQ.EOF) TIME=555.0
   RETURN

C RETURN ENDFILE
   CALL ACT(I,YSM(N),IHM(N),SEC(N))
   YM(N)=YSM(N)
   ASCC=SEC(N)
   RETURN

C RETURN ENDFILE
   CALL ACT(I,YSM(N),IHM(N),SEC(N))
   YM(N)=YSM(N)
   ASCC=SEC(N)
   RETURN

END
 ENTRY 

 OUTPUT FILE FULL

 NO

 READ AND STORE 50 SETS OF COORDINATES

 CALCULATE TIME OF FIRST SET

 CALCULATE DAYS FROM EPOCH

 5

 CALCULATE TIME DIFFERENCES BETWEEN SETS

 5

 FILL OUTPUT VECTORS

 UPDATE TIMES

 END FILE

 YES

 RETURN

 A
DESCRIPTION

The subroutine READER controls the calls to the WRDC Plot Package routines to generate the printer plots and/or plot tape for the orbital differences computed in DELTA.
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

NAME: READER

PURPOSE: PLOTS ORBITAL DIFFERENCES

CALLING SEQUENCE: CALL READER(E,T,LASTSW)

SYMBOL TYPE DESCRIPTION
E T INPUT - TIME FROM EPOCH IN DAYS
LASTSW INPUT - LAST PLOT SWITCH

SUBROUTINES USED: ANOD, EDIT, ARMADV, MAXLIN, MAXMIN, OGRID, PLOT, PLOTT, PTYNUM, VERLIN

COMMON BLOCKS: CPlots, CPlot, PlotTP

INPUT FILES: NONE

OUTPUT FILES: NONE

RESTRICTIONS: NONE

REFERENCES: NONE

SUBROUTINE READER(E,T,LASTSW)
REAL NTVL
DOUBLE PRECISION DT
COMMON/CPlots/31(2),LOGX,LOGY,XLOLIM,YLLOLIM,XHILIM,YHILIM,
* XSCAL, YSCAL, FXLD, FYLD, G2(5)
COMMON/Plottays((GG 6),RADIAL(AGC),CRSTRK(AGC),ALGTRK(AGC)),
* ARRAY(21), IMOD2, IMOD2, INDEX, NNOPT, SCALE, SCALE2, NTVL

DOUBLE PRECISION TITLE(3), ARRAY
DATA TITLE=8=EPOCH,8H,8H
DATA NNUM=1, A3
DATA TAPE/=FALSE,
C. INITIALIZE
IF(NNOPT.NE.0) TAPE=.TRUE.
E=ABS(E)
ANON=ABS((T-E)*DT
DO 10 I=1,INDEX
10 DAYS(I)=((LAYS(I)-1)/ASF)*24.
C.DETERMINING MAXIMA AND MINIMA OF ARRAY
CALL MAXMIN(RADIAL, INDEX, XM, XMAX)
CALL MAXMIN(ALGTPA, INDEX, AMAX, ALMA)
CALL MAXMIN(CRSTRK, INDEX, XM, CHMAX)
IF(NTVL.GE.0) GO TO 17
REAL MAX=MAXI (XMAX, CHMAX, ALMA)
REAL MIN=MIND (XM, CHMIN, ALMA)
C.DETERMINING MAXIMUM AND MINIMUM OF PLOTTING GRID

READ 29
READ 30
READ 31
READ 32
READ 33
READ 34
READ 35
READ 36
READ 37
READ 38
READ 39
READ 40
READ 41
READ 42
READ 43
READ 44
READ 45
READ 46
READ 47
READ 48
READ 49
READ 50
READ 51
READ 52
READ 53
READ 54
READ 55
REPRODUCIBILITY OF THE
ORIGNAL PAGE IS POOR.

CALL PTKN(NX,XY,XY,XY,XY)
G0 T0 1
17 CONTINUE
REALX = SCALE1
REALM = SCALE2
N=NTXL+1,E
16 CONTINUE
ISTART=1
XX=0.
XMIN=512.
C START PLOT
PLT3=THL3.
CALL PLOT(3,NOCT,PLOTS)
CALL HRCLIN(22HTRAJECTORY DIFFERENCES,22,512,520)
CALL LSIT(IYM02,11C),TITLE(2),P)
CALL LSIT(IHMS2,11G),TITLE(3),N)
CALL HRCLIN(ARRAY(1),52,512,57)
CALL HRCLIN(ARRAY(3),56,512,45)
CALL HRCLIN(ARRAY(15),55,512,43)
CALL HRCLIN(TITLE,24,512,47)
C INITIALIZE GRID ON NEXT PAGE
25 CALL FRMACV
CALL HRCLIN(22HTRAJECTORY DIFFERENCES,22,512,1053)
CALL HRCLIN(16HOURS FROM EPOCH,11,512,57)
CALL HRCLIN(25H----- RACIAL DIFFERENCES,25,512,503)
CALL HRCLIN(25H----- CROSS TRACK DIFFERENCES,25,512,587)
CALL HRCLIN(25H----- ALONG TRACK DIFFERENCES,29,512,471)
25 X=X/X+1(XX+/2.)
YULIM=32.+57.*XX
YHILIM=432.+57.*XX
NUM=MAX(INX-ISTART,N1)+1
XMAX=XMAX+NUM
CALL OGRIC(XH1XMAX,12,F5,1)*1.1.REALXREALX,XY,XYF7.1)+1.0
XMIN=XMAX
CALL VERLIN(6FMETFS,6.05XYH1YH1+YULIM/2)
C PLOT DATA POINTS
CALL PLOT(DAYS(START),RADIAL(START),NUM(4)
CALL PLOT(DAYS(START),CHRSTEK(START),NUM(5)
CALL PLOT(DAYS(START),ALGTHRK(START),NUM(4)
ISTART=START(NUM)
IF(INDEX=LT,ISTART) GO TO 50
IF(INX+57.1) GO TO 25
G0 T0 20
5) CALL FRMACV
IF(TAPE.1,AND.LASTS) CALL PLOTST(7,TRUE)
C END CF PLOT
IF(LASTS) CALL ENDPLT
RETURN
END
ENTRY

DETERMINE MAXIMA AND MINIMA OF ARRAYS

DETERMINE APPROPRIATE GRID SCALE

FIXED GRID

NO

YES

SET GRID SCALE

20

CALL PLOTTING ROUTINES

LAST PLOT

NO

YES

RETURN
DESCRIPTION

(See GEODYN)
1.1.2 GEORGE

INTRODUCTION

The support program GEORGE analyzes GEODYN measurement residuals. The residuals enter GEORGE from a tape generated by GEODYN and are analyzed on a pass by pass basis for either the station and/or measurement type specified by card input to GEORGE.

The main routine GEORGE selects the residuals to be analyzed and breaks them up into individual passes. GEORGE also controls which types of plots are to be made, if any.

REGANL performs the regression analysis and can edit data points on the basis of their standard deviations from the mean.

The subroutines HISTO and PLOTER provide visual aids in analyzing the residuals. HISTO plots a histogram of either the residuals or the ratios to sigma for each pass and a grand summation histogram for all the passes analyzed. PLOTER plots either residuals versus time or measurement rate versus residuals for each pass of data. Both subroutines are driver routines for the Plot Package.

The subroutine DIFF computes the difference in days between any two dates, and the subroutine RYMDI resolves a date in one word into three words: the year, the month, and the day.

GEORGE requires approximately 525K bytes of core and the IBM 360 system routines DSQRT and .MOD. GEORGE will analyze about 1000 residuals in less than three minutes.
The subroutine REGANL determines measurement biases
(or zero-set errors) and timing errors in each pass of
data and then performs a regression and analysis of the
residuals.

The zero-set error, $A$, and timing error, $B$, are determined by using a least squares method of solving the
following equation:

$$Y = A + BX$$

where

$Y$ is the residual and

$X$ is the measurement rate.

Taking the partials of (1) with respect to $B$ and
then with respect to $A$ and setting them to zero, we get

$$\sum_{i=1}^{N} X_i Y_i - B \sum_{i=1}^{N} X_i^2 - A \sum_{i=1}^{N} X_i = 0$$

(2)
\[
\sum_{i=1}^{N} Y_i - B \sum_{i=1}^{N} X_i - NA = 0 \quad (3)
\]

where \( N \) is the number of points in the pass.

The two equations are solved simultaneously for \( A \) and \( B \).

First REGANL computes the sums of the rates,

\[
\sum_{i=1}^{N} X_i,
\]

and residuals,

\[
\sum_{i=1}^{N} Y_i,
\]

the products of \( X_i \) and \( Y_i \),

\[
\sum_{i=1}^{N} X_i Y_i,
\]

the squares of the rates,

\[
\sum_{i=1}^{N} X_i^2
\]
and finally, the squares of the residuals,

\[ \sum_{i=1}^{N} Y_i^2. \]

Then the corrected sum of the products, CSXY, and the corrected sums of the squares, CSX^2 and CSY^2, are computed as follows:

\[
CSXY = \sum_{i=1}^{N} X_i Y_i - \sum_{i=1}^{N} X_i \sum_{i=1}^{N} Y_i \div N
\]

\[
CSX^2 = \sum_{i=1}^{N} X_i^2 - \left( \sum_{i=1}^{N} X_i \right)^2 \div N
\]

\[
CSY^2 = \sum_{i=1}^{N} Y_i^2 - \left( \sum_{i=1}^{N} Y_i \right)^2 \div N
\]

Now, solving for B we get

\[ B = \frac{CSXY}{CSX^2}, \]

and solving for A using B we get

\[ A = \left( \sum_{i=1}^{N} Y_i - B \sum_{i=1}^{N} X_i \right) \div N. \]

The regression sum of squares, RSS, is

\[ RSS = \frac{CSXY^2}{CSX^2} \]

and the regression mean, RM, is

\[ RM = \frac{(CSY^2 - RSS)}{(N - 1)} \]

which is nothing more than the square of the standard deviation of the residuals about the trajectory.

The standard deviations of the zero-set error, SDZ, and timing error, SDT, are

\[ SDZ = \sqrt{RM \sum_{i=1}^{N} x_i^2 / NCSX^2} \]

and

\[ SDT = \sqrt{RM / (N-1)} \]
The noise about the fitted line, $D$, is

$$ D = \sqrt{RM} $$

The residual mean square, $RMSQ$, is computed as

$$ RMSQ = \frac{CSY^2 - RSS}{N - 1} $$

To test the randomness of the result, we compute the residuals corrected for zero-set and timing error biases, $CR_i$, as

$$ CR_i = RESID_i - A_i - B_i X_i $$

where $RESID_i$ is the residual.

Then we compute difference sum of squares between subsequent residuals, $DSQ$, as

$$ DSQ = \sum_{i=1}^{N} \left( CR_{i+1} - CR_i \right)^2 $$

The random normal deviate, $RND$, is then

$$ RND = \frac{(DSQ/2RM) - 1}{\sqrt{(N-2)/(N^2-1)}} $$

1-33
The noise is random if

\[ |\text{RND}| < 2.58 \]

and non-random if

\[ |\text{RND}| > 2.58. \]
## SUBROUTINE CROSS REFERENCE CHART

### CALLING ROUTINES

<table>
<thead>
<tr>
<th></th>
<th>MAIN</th>
<th>DIFF</th>
<th>HISTO</th>
<th>NEWMM</th>
<th>PLOTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMAX1</td>
<td></td>
<td></td>
<td>⬜</td>
<td></td>
<td>⬜</td>
</tr>
<tr>
<td>AMIN1</td>
<td></td>
<td></td>
<td>⬜</td>
<td></td>
<td>⬜</td>
</tr>
<tr>
<td>DIFF</td>
<td>⬜</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>ENDPLT</td>
<td>⬜</td>
<td></td>
<td></td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>FRMADV</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>HISTO</td>
<td>⬜</td>
<td></td>
<td></td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>HORLIN</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>MAXMIN</td>
<td>⬜</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>MINT</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>NEWMM</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>OGRID</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>PLOT</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>PLOTER</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>PLOTST</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>PTYNUM</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>REGANL</td>
<td>⬜</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>RYMDI</td>
<td></td>
<td>⬜</td>
<td></td>
<td>⬜</td>
<td>⬜</td>
</tr>
<tr>
<td>VERLIN</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
<td>⬜</td>
</tr>
</tbody>
</table>

1-35
## COMMON BLOCK CROSS REFERENCE CHART

<table>
<thead>
<tr>
<th>COMMON BLOCKS</th>
<th>ROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>ARRAY</td>
<td>![Circle]</td>
</tr>
<tr>
<td>COONST</td>
<td>![Circle]</td>
</tr>
<tr>
<td>LOGIC</td>
<td>![Circle]</td>
</tr>
</tbody>
</table>
DESCRIPTION

The main routine GEORGE reads the GEORGE INPUT CARDS and sets the switches for the type, station number and network of the data to be analyzed. It also sets the switches for the type of analysis (residual or ratio) and type or types of plots desired. GEORGE then reads the residual tape and separates the data into passes. Once a pass is established, GEORGE calls REGANL to compute the zero set and timing errors and perform the regression analysis. If plots of the residuals or ratios are desired, PLOTER is called. If histograms are desired, HISTO is called. This procedure is followed until all the data specified is analyzed.
The tracking networks acceptable to GEORGE and the code abbreviations are given below:

<table>
<thead>
<tr>
<th>Network</th>
<th>Code Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>STADAN</td>
<td>STADAN</td>
</tr>
<tr>
<td>DOPPLER</td>
<td>DOPLER</td>
</tr>
<tr>
<td>U.S.A.F.</td>
<td>USAF</td>
</tr>
<tr>
<td>C-BAND</td>
<td>C BAND</td>
</tr>
<tr>
<td>SECOR</td>
<td>SECOR</td>
</tr>
<tr>
<td>U.S.C.&amp;GS.</td>
<td>USC+GS</td>
</tr>
<tr>
<td>SPEOPT</td>
<td>SPEOPT</td>
</tr>
<tr>
<td>INTERNATIONAL</td>
<td>INTERL</td>
</tr>
<tr>
<td>SAO</td>
<td>SAO</td>
</tr>
</tbody>
</table>

The types of measurements and the code names acceptable to GEORGE are listed below:

<table>
<thead>
<tr>
<th>Measurement Type</th>
<th>Code Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>right ascension</td>
<td>RT ASC</td>
</tr>
<tr>
<td>declination</td>
<td>DECLIN</td>
</tr>
<tr>
<td>range</td>
<td>RANGE</td>
</tr>
<tr>
<td>range rate</td>
<td>R RATE</td>
</tr>
<tr>
<td>alpha</td>
<td>ALPHA</td>
</tr>
<tr>
<td>beta</td>
<td>BETA</td>
</tr>
<tr>
<td>x angle</td>
<td>X ANGL</td>
</tr>
<tr>
<td>y angle</td>
<td>Y ANGL</td>
</tr>
<tr>
<td>azimuth</td>
<td>AZMUTH</td>
</tr>
<tr>
<td>elevation</td>
<td>ELEV</td>
</tr>
</tbody>
</table>
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

MAIN - GEORGE

NAME: MAIN - GEORGE

PURPOSE: PERFORMS A REGRESSION ANALYSIS OF RESIDUALS OR RATIOS AND PLOTS THE RESULTS

SUBROUTINES USED: DATE, DIFF, HISTO, REGANL, RYNDI, EDIT, ENDUCT, FHADY, HIRLIN, MAXMIN, MINT, NEWAM, GRID, PLOTST, PTVNUM, VERLIN, PLUTER

COMMON BLOCKS: ARRAY, LOGIC, CONST

INPUT FILES: GEORGE INPUT CARDS

OUTPUT FILE: NONAME RESIDUAL TAPE

RESTRICTIONS: A MAXIMUM OF 4000 POINTS PER PASS WILL BE ANALYZED

REFERENCES: NONE

10001 FORMAT(3(A6,4X))

10002 FORMAT(A6,4X,5(F10.0))

20001 FORMAT(1M1,10X,18 * ANALYSIS OF 'A5, * RESIDUALS ***)

20002 FORMAT(1H,14X,NETWORK = - 'A5)

20003 FORMAT(1H,14X,STATION = - 'A5)

20004 FORMAT(1H,14X,ILLEGAL MEASUREMENT TYPE = SKIPPING TO NEXT CASE )

20005 FORMAT(1H,14X,ILLEGAL NETWORK NAME = SKIPPING TO NEXT CASE )

20006 FORMAT(1H,14X,ILLEGAL OPTION CARD = 'A5, 46H REMAINING OPTION, GEOR 27)

20020 FORMAT(1H,14X,NO DATA OF THE TYPE SPECIFIED FOUND -- SKIPPING TO NEXT CASE)

20021 FORMAT(1H,14X,20H OBSERVATIONS BELOW *51,29H DEGREES WILL NOT BE ANALYZED)

20022 FORMAT(1H,14X,20H RESIDUALS DEVIATING *61,52H UNITS OR MORE FROM THE FITTED LINE WILL BE REJECTED)

20023 FORMAT(1H,14X,20H TOO MANY OBSERVATIONS -- REMAINDER IGNORED)

DOUBLE PRECISION ACHAN(3,50), ATYPE, CHAN, CCHAN(3,50), DATAGEOR 41

DOUBLE PRECISION BL, ELEV, FNEY(9), FTYPE

DOUBLE PRECISION FTYPE(14), GRANL, IBLANK, ISTA, KSTA

DOUBLE PRECISION LASER, LAST, KTYPE, NAME, NET

DOUBLE PRECISION OBS01, OBS02, OPT, OPTION(10), SAOLAS

DOUBLE PRECISION STRAK(100), TEST, TYPE

DIMENSION LNGCH(100), LSTART(100), VALUE(5)

COMMON /ARRAY/ LYLA(4000), LMP(4000), SCC(4000), ELEV(4000), RESID(4000), OHOST(4000), ICOUNT(4000), NAMEST(4000), GEOR 49

COMMON /LOGIC/ LAST1, MSWTCH

COMMON /CONST/ NET, TYPE, KSTA, ABO, ISAVE, REJCT, REJST

LOGICAL FRESH, ICST, ICST1, LAST1, PLOTST, PLIST, PRESH, REJST, Switch

LOGICAL, COND

DATA HS1=1, PLST / 2*, FALSE, /

30 September 1972
PROGRAMmE: DATA

30 September 1972

Page 4 of 10

REPRODUCTION OF THE
ORIGINAL PAGE IS POOR

READ DATA INPUT CARDS AND SET SWITCHES FOR NETWORK AND STATION

READ 10001 TYPE NET KSTA

IF (NET.EQ.GRAPH) NET(1) = GRAPH

IF (NET.EQ.LASER) NET(1) = LASER

IF (NET.EQ.SASLAM) NET(1) = SASLAM

PRINT 20001 TYPE

GROUM = FALSE

LASTK = FALSE

SWITCH = FALSE

IF TYPE.EQ.FTYPE(1) SAVE = 1

CONTINUE

IF (SAVE.EQ.0) GO TO 95

PRINT 20004

GO TO 27

CONTINUE

IF (SAVE.EQ.0) GO TO 1

PRINT 20002 NET

DO 5 = 1, 9

IF (NET.EQ.FTYPE(1)) SAVE = 1

CONTINUE

IF (SAVE.EQ.0) GO TO 1

PRINT 20005

GO TO 27

CONTINUE

IF (NET.EQ.FTYPE(1)) PRINT 20003 KSTA

C INITIALIZE CONSTANTS AND SWITCHES

DO 20 = 1, 100

C NAME(I) = IBLANK

CUT = 0.0

10020 = 4

AMINS = FALSE

FITS = FALSE

FLOPS = FALSE

FROS = FALSE

RSQ = FALSE

C READ GEORGE INPUT CARDS AND SET ANALYSIS AND PLOTTING OPTIONS

GO TO 1005
DO 7 1=1,10
IF (OPT.NE.OPTION(1))GO TO 7
GO TO (6,5,1C,11,12,13,14,15,16,17,1)
7 CONTINUE
PRINT 20006,OPT
GO TO 17
8 JACHAN = JACHAN + 1
ACHAN(1,JACHAN) = VALUE(1)
ACHAN(2,JACHAN) = VALUE(2)
ACHAN(3,JACHAN) = VALUE(4)+1
GO TO 6
9 CUT = VALUE(1)
PRINT 20001,CUT
GO TO 6
10 JCCHAN = JCCHAN + 1
CCCHAN(1,JCCCHAN) = VALUE(1)
CCCHAN(2,JCCCHAN) = VALUE(2)
CCCHAN(3,JCCCHAN) = VALUE(4)+1
GO TO 6
11 ANB56 = .TRUE.
GO TO 6
12 PHISW = .TRUE.
GO TO 6
13 PISTK = .TRUE.
SWITCH = VALUE(1).EQ.1+OR.VALUE(1).EQ.3,
GROSSUM = VALUE(1).EQ.2+OR.VALUE(1).EQ.3
GO TO 6
14 FLGLOS = .TRUE.
IF (VALUE(1).EQ.1) 14026=6
GO TO 6
15 CONTINUE
GO TO 6
16 CONTINUE
REJSW = .TRUE.
REJECT = VALUE(1)
PRINT 20022,REJECT
GO TO 6
C READ NONAME RESIDUAL TAPE
READ(15,END=16,ERR=16) IYMD1,IPH1,SEC1,ISTA,TYPE,OBSO1,RESID1,
* RATIO1,DECOT1,OBSS2,RESID2,RATIO2,OBSS2,EL,INET
C TEST FOR END OF DATA
IF (ISTA.EQ.ELANK)GO TO 10
IF (INET.EQ.ELNET)INET = 2
C TEST FOR REQUESTED DATA
IF (EL.LT.CUT)GO TO 17
IF (ISTA.NE.KSTA.AND.KSTA.NE.ELANK)GO TO 17
IF (NET.NE.INET.AND.NET.NE.ELANK)GO TO 17
IF (TYPE.NE.TYPE.E1GO TO 17
C STORE RESIDUALS, TIME, ELEVATION, AND STATION NAME
MMEAS = MMEAS + 1
IF (MMEAS.LT.4000)GO TO 164
PRINT 20023
GO TO 16
164 IYMD(MMEAS) = IYMD1
INCH(MMEAS) = INCH1
SEC(MMEAS) = SEC1
ELLY(NMFM) = EL
NAMEST(NMEAS) = ISTA
IF(KSTA.EQ.1)GO TO 101
IST=1
NAME(1)=KSTA
GO TO 22
101 CONTINUE
GO 21 I=1,100
IF (STNAME(I).EQ.ISTA) CC TO 22
IF (STNAME(I).NE.IBLANK) GO TO 21
IST=1
NAME(1)=ISTA
GO TO 22
21 CONTINUE
22 CONTINUE
RESID(NMEAS) = RESID1
RATIO(NMEAS) = PATIO1
CDOOT(NMEAS) = CDOOT1
C STORE PAIRED RESIDUALS IN UPPER HALF OF ARRAY
IF(ISAVE.NE.1.AND.ISAVE.NE.6.AND.ISAVE.NE.7) GO TO 17
KASIN=NMEAS+200
MEDID(KASIN) = RESID2
RATIO(KASIN) = PATIO2
CDOOT(KASIN) = CDOOT2
IYM(KASIN)=IYMD1
IHM(KASIN)=IHM1
SEC(KASIN)=SEC1
FVYK(KASIN)=FVY1
NAMEST(KASIN)=ISTA
GO TO 17
18 REWIND 15
IF (NMEAS.NE.6) GO TO 50
PRINT 20020
GO TO 102
50 CONTINUE
C MATCH MEASUREMENTS WITH STATION NAME
DO 23 J =1,1ST
JJ=0
DO 19 I=1,NMEAS
IF (NAMEST(I).NE.STNAME(J)) GO TO 19
JJ = JJ + 1
ICOUNT(JJ)=I
19 CONTINUE
ISIART(1) = ICOUNT(1)
NPASS = 1
KK = JJ - 1
DO 24 I=1,22
N = ICOUNT(1)
M = ICOUNT(1+1)
24 CONTINUE
C TEST FOR A NEW PASS
IM7S1 = IIM7+100
IM7S2 = IIM7+100
CALL DIFF(IYMDN),IM7S1,IYMDM,IM7S2,IDAY,ISFC1
IM7 = IDAY+450+SEC300
IF (IIM7.LT.1) GO TO 24
NPASS = NPASS + 1
GO TO 50
ISTART(NPASS) = M
IEND(NPASS-1) = N
24 CONTINUE
IF((NOT(FLOTST) OR PLSTST) AND (NOT(HSTST) OR HSTST)) GO TO 106
C INITIALIZE PLOT PACKAGE
CALL PLSTST(ISC20), (TRUE)
CALL FRAEV
PLSTST = TRUE,
HSTST = TRUE,
106 IEND(NPASS) = N
DO 25 I = 1, NPASS
M = ISTART(I)
C PERFORM THE REGRESSION ANALYSIS
CALL REGREL(ISSTART(I), IEND(I), STNAME(J), MM)
IF((NOT(FLOTST)) GO TO 107
C MAKE THE PLOTS
CALL PLOTREL(ISSTART(I), IEND(I), PM, STNAME(J), ISAVE)
IF(ISAVE = EQ.0 OR ISAVE = EQ.6 OR ISAVE = EQ.7)
CALL PLOTREL(ISSTART(I) + 2000, IEND(I) + 2000, MM, STNAME(J), ISAVE = 7)
107 IF (GRIDST) GO TO 25
IF ((NOT(HSTST)) GO TO 25
C MAKE THE HISTOGRAMS
CALL HISTC(ISSTART(I), IEND(I), MM, STNAME(J), ISAVE)
IF(ISAVE = NE.1 AND ISAVE = NE.6 AND ISAVE = NE.7) GO TO 25
CALL HISTC(ISSTART(I) + 2000, IEND(I) + 2000, MM, STNAME(J), ISAVE = 7)
25 CONTINUE
23 CONTINUE
LAST = TRUE,
IF ((NOT(HSTST)) GO TO 102
CALL HISTC(ISSTART(I), IEND(I), MM, STNAME(J), ISAVE)
IF(ISAVE = EQ.1 OR ISAVE = EQ.6 OR ISAVE = EQ.7)
CALL HISTC(ISSTART(I) + 2000, IEND(I) + 2000, MM, STNAME(J), ISAVE = 7)
C TEST FOR LAST DATA CARD
102 READ 10002, TEST
IF (TEST = EO. TEST) GO TO 4
IF (TEST = EO. LAST) GO TO 27
GO TO 102
27 CONTINUE
IF(PLSTST, CR, HSTST) CALL ENDPST
STOP
END
DESCRIPTION

(See EPHemeris Tape Generator)
DESCRIPTION

HISTO determines the grid size and labels to produce histograms of each pass and a grand summation histogram if requested. It is basically a drive program for the WOLF PLOT PACKAGE; hence all routines called by HISTO are members of the PLOT PACKAGE.
NAME HISTO

PURPOSE PLOTS HISTOGRAMS OF ANALYZED DATA

CALLING SEQUENCE CALL HISTO(I_START,I_END,MM,NAME,ISAVE)

SYMBOL TYPE DESCRIPTION
I_START I INPUT - INDEX OF START OF PASS IN ARRAY
I_END I INPUT - INDEX OF END OF PASS IN ARRAY
MM I INPUT - NUMBER OF POINTS IN PASS
ISAVE I INPUT - INDEX OF TYPE OF PLOT REQUESTED

SUBROUTINES USED EDIT MADCY MURLIN MAXMIN MINT
NEWMEM DGPID PLOT PTYNUM VERLIN

COMMON BLOCKS ARRAY LOGIC

INPUT FILES NONE

OUTPUT FILE PRINTER

RESTRICTIONS NONE

REFERENCES NONE

SUBROUTINE HISTO(I_START,I_END,MM,NAME,ISAVE)
DOUBLE PRECISION ELEV,FTYPE,NAMEST,TITLE(6),ISAV
DOUBLE PRECISION TYPE,XTITLE(6),YTITLE(6)
COMMON /ARRAY/ YMD(4000),HMM(4000),SEC(4000),ELEV(4000),
* RESID(4000),GROWTH(4000),NAME(4000),NAMEST(4000),TITLE(6)
* FTYPE(14),RATIC(4000)
COMMON /LOGIC/ LASTIM,SWITCH
LOGICAL LSTIM,SWITCH
DIMENSION SIZE(2),X(2),Y(2)
DATA XTITLE/8HISTOGRAM OF RHSATIO/9FREQUENCY/6H
* DATA YTITLE/8HISTOGRAM OF RHSRESIDUAL/4H FREQUEN
* 8H RESIDUAL
C GENERATE TITLE FOR TYPE OF HISTOGRAM
IF (MM.LT.5) GO TO 70
TYPE=FTYPE(ISAVE)
TITLE(6)=TYPE
DO 5 I=1,60
5 SIZE(I)=6
IF(SWITCH) GO TO 15
C SET UP HISTOGRAM GRID SIZE FOR RESIDUAL ANALYSIS
CALL NEWMEM(NAME,RESID,I_START,I_END,MM,MAX,MIN)
CALL PTYNUM(MIN,MAX,MIN,MAX,NX)
HISTO
Page 3 of 6
30 September 1972

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR.

C SIZE=(RMAX-RMIN)/NX
C DO IC=1,START,END
IF(NOT=LAST I,AND,NAME,NE,NAMEST(I)) GO TO 10
J=INT(SICL-I-RMIN)/C SIZE+1.
SIZE(J)=SIZE(J)+1.
10 CONTINUE
GO TO 40
C DO 20 IC=1,START,END
IF(NOT=LAST I,AND,NAME,NE,NAMEST(I)) GO TO 23
RATIO(IC)=SIGN(RATIO(IC),RESID(IC))
20 CONTINUE
C SET UP HISTOGRAM GRID SIZE FOR RATIO ANALYSIS
CALL MAXMIN(NAME,RATIO,START,END,VMAX,RMIN)
IF(VMIN>0,0) GO TO 65
CALL PTKAIF(VMAX,RMIN,MAX,RMIN,NX)
C SIZE=(ABS(RMAX-RMIN)/NX)
ED 30 IC=1,START,END
IF(NOT=LAST I,AND,NAME,NE,NAMEST(I)) GO TO 30
IF((RATIO(IC)+RATIO(IC)+EQ+C,)) GO TO 33
J=INT(SICL-I-RMIN)/C SIZE+1.
SIZE(J)=SIZE(J)+1.
30 CONTINUE
C FIND THE HIGHEST COUNT AND DETERMINE INCREMENTS
40 CALL MAXMIN(SIZE,NX,VMIN,VMAX)
VMAX=VMAX+1.
CALL PTKAIF(000,VMAX,VMIN,VMAX,000)
IF(VMIN>0,0) GO TO 65
C SET INITIAL LIMIT VALUES
CALL UGRIC(RMIN,RMAX,NX,SHF0,1,1,000,000,000,000)
IF(SWITCH) GO TO 45
41 IC=1,5
42 IF(NOT=LAST I,GO TO 56
43 ED 41 IC=1,5
44 IF(NOT=LAST I,GO TO 56
45 ED 41 IC=1,5
46 IC=1,5
47 IF(NOT=LAST I,GO TO 56
48 ED 41 IC=1,5
49 IF(NOT=LAST I,GO TO 56
50 IC=1,5
51 IC=1,5
52 IF(NOT=LAST I,GO TO 56
53 ED 51 IC=1,5
54 IF(NOT=LAST I,GO TO 56
55 CONTINUE
WRITE(6,100) IC
56 CONTINUE
C PRINT DATA
IF(NOT=LAST I,GO TO 56
WRITE(6,100) IC
57 CONTINUE
WRITE(6,100) IC
58 CONTINUE
C PRINT DATA
X(I)=RMIN
SIZE(NX+1)=0.
LU 60  I=1,NX
X(I)=X(1)+I*SIZE
Y(1)=SIZE(1)
Y(2)=SIZE(1)
IF(SIZE(1).GT.1)CALL PLCT(X,Y,2,4H
Y(1)=MAXI(SIZE(1),SIZE(1+1))
Y(2)=0.
X(1)=X(2)
60 IF(Y(1).LE.0.) CALL PLCT(X,Y,2,4H
CALL FNAME
RETURN
65 WRITE(6,103)
RETURN
72 PRINT 100
RETURN
100 FORMAT(1X,'INSUFFICIENT DATA FOR A MEANINGFUL HISTOGRAM*')
101 FORMAT(1X,'JSTATION NAME',6X,'RESIDUAL VALUE',6X,'COUNT'/*/)
102 FORMAT(1X,'JSTATION NAME',6X,'RESIDUAL VALUE',6X,'COUNT'*5X)
'*CONTINUE'/*/)
103 FORMAT(1X,'JSTATION NAME',6X,'RATIO VALUE',6X,'COUNT'/*/)
104 FORMAT(1X,'JSTATION NAME',6X,'RATIO VALUE',6X,'COUNT'6X)
'*CONTINUE'/*/)
105 FORMAT(1X,'HISTOGRAM DATA',20X,'INTERVAL',3X,'FREQUENCY'/*/)
106 FORMAT(1X,'TAL TO',2X,'TOTAL NO. OF WT. POINTS = ',13A)
107 FORMAT(1X,'THE RATIO TO SIGMA VALUES ARE ALL ZERO. NO PLOHIST 14?
'T CAN BE MADE.*/)
END
START

GENERATE TITLE

RATIO ANALYSIS

NEUMM

PEND MAXIMUM AND MINIMUM OF RESIDUALS

ueva

PDYNOM

ROUND OFF MAXIMUM AND MINIMUM

CALCULATE CELL SIZE

15

DETERMINE SIGN OF RATIO VALUE

NEUMM

FIND MINIMUM AND MINIMUM OF RATIOS

PDYNOM

ROUND OFF MAXIMUM AND MINIMUM

CALCULATE CELL SIZE

40

15

40

30 September 1972
40

FIND LARGEST CELL AND FIX INCREMENTS

PRINT TITLE AND LABELS ON PLOT

PRINT DATA

PLOT DATA

RETURN
DESCRIPTION

NEWMM is a simple program utilizing WOLF PLOT PACKAGE routines to determine maximum and minimum values for either all or part of a specified array.
NAME

PURPOSE

FIND THE MAXIMUM AND MINIMUM VALUES IN AN ARRAY FOR SPECIFIED STATIONS

CALLING SEQUENCE

CALL NEWMM(NAME, ARRAY, ISTART, IEND, RMAX, RMIN)

SYMBOL TYPE DESCRIPTION

NAME OF INPUT - NAME OF STATION

ARRAY R INPUT - ARRAY TO BE SEARCHED

(4CO)

ISTART I INPUT - INDEX OF STARTING VALUE IN ARRAY

IEND I INPUT - INDEX OF ENDING VALUE IN ARRAY

RMAX R OUTPUT - MAXIMUM VALUE

RMIN R OUTPUT - MINIMUM VALUE

SUBROUTINES USED

NONE

COMMON BLOCKS

ARRAY LOGIC

INPUT FILES

NONE

OUTPUT FILES

NONE

RESTRICTIONS

NONE

REFERENCES

NONE

SUBROUTINE NEWMM(NAME, ARRAY, ISTART, IEND, RMAX, RMIN)

DOUBLE PRECISION ELEV, FTYPE, NAME, NAMEST

COMMON ARRAY / IYMD(4CO), IMH(4CO), SEC(4CO), ELEV(4CO), RESID(4CO), ORCOT(4CO), ICOUNT(4CO), NAMEST(4CO), RMAX(14), RATIC(4CO)

COMMON /LOGIC / LASTIM, SWITCH

LOGICAL LASTIM

DIMENSION ARRAY(4CO)

RMAX=ARRAY(ISTART)

RMIN=RMAX

C TEST IF BEGINNING AND ENDING INDICES ARE DIFFERENT

IF(ISTART.EQ.IEND)RETURN

DO 10 I=ISTART, IEND

C SEARCH ARRAY FOR STATION NAME

IF(.NOT.LASTIM.AND.NAMEST(I).NE.NAME) GO TO 10

C FIND MAXIMUM

RMAX=MAX(RMAX, ARRAY(I))

C FIND MINIMUM
10 CONTINUE
RETURN
END
DESCRIPTION

PLOTER is the drive program for the WOLF PLOT PACKAGE which produces the plots of residuals vs. time or measurement rate vs. residuals if either are requested.
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

NAME
PLOTER

PURPOSE
PLOTS RESIDUALS VS. TIME AND MEASUREMENT RATE VS.
RESIDUALS FOR SPECIFIED PASSES

CALLING SEQUENCE
CALL PLOTER(IPASS,ISTOP,IM,NAME,ISAVE)

SYMBOL TYPE DESCRIPTION
IPASS I INPUT - INDEX OF STARTING POINT
ISTOP I INPUT - INDEX OF ENDING POINT
IM I INPUT - NUMBER OF POINTS IN PASS
ISAVE I INPUT - INDEX OF TYPE OF DATA

SELECTED LISTS
EDIT EMPLT FMAOV MHLIN MAXMIN
MINT OGRID PLOTST PTVNUM VERLIN
NEWMM

COMMON BLOCK
ARRAY

INPUT FILES
NONE

OUTPUT FILES
NONE

RESTRICTIONS
NONE

REFERENCES
NONE

SUBROUTINE PLOTER(IPASS,ISTOP,IM,NAME,ISAVE)
DOUBLE PRECISION ELEV, FTYPE, NAMEST, TITLE(8), TYPE
REAL NAME, NAME
COMMON/ARRAY/IYMC(4000), IYMC(4000), SEC(4000), ELEV(4000),
* RESID(4000), OBCOT(4000), IDCOUNT(4000), NAMEST(4000), PLOTST(4000),
* FTYPE(14), PTVNUM(4000)
DIMENSION FMIN(4000)
REAL WMMAX, WMMIN

DATA TITLE/EH ,EH RESIDUALS, 3HLS /
* 6H DATE ,6H 6H TIME ,3H /

50 FORMAT(3F14.8) INSUFFICIENT DATA IN THIS PASS FOR A MEANINGFUL PLOT

C TEST IF ENOUGH DATA
IF(AW.LT.2) GO TO 40

TYPE=FTYPE(ISAVE)

C FORM TIME ARRAY AND FIND MAXIMUM AND MINIMUM TIMES
IYMDI=IYMC(IPASS)
IHMI=IHMC(IPASS)

LMIN=LMC(IPASS)-LMC(IPASS)/100*40
DO 10 IC=IPASS+1,ISTOP
FM(1)=FCAT(IHMC(1)-IHMC(1)/100*40)-2MIN+SEC(1)/60.
10 IF(FM(1).LT.* C) FM(1)=FM(1)+1440

C PLOT:

PLOT 35
PLOT 36
PLOT 37
PLOT 38
PLOT 39
PLOT 40
PLOT 41
PLOT 42
PLOT 43
PLOT 44
PLOT 45
PLOT 46
PLOT 47
PLOT 48
PLOT 49
PLOT 50
PLOT 51
PLOT 52
PLOT 53
PLOT 54
PLOT 55

1-58
PLOTER
Page 3 of 5
30 September 1972

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

PLT 56
CALL MAXIN(FVIN(IPASS),I,MIN,SMAX)
PLT 57
CALL PTYNM(SMIN,SMAX,MIN,SMAX,NX)
PLT 58
KSAVE=IPASS
PLT 59
C FIND MAXIMUM AND MINIMUM VALUES AND ROUND
PLT 60
CALL NEWX(NAME,RESID,IPASS,IST0P,MAX,NMIN)
PLT 61
CALL PTYNM(RMIN,SMAX,MIN,SMAX,NX)
PLT 62
C PRINT TITLE AND GRID FOR RESIDUAL VS. TIME PLOTS
PLT 63
TITLE(1)=NAME
PLT 64
TITLE(2)=TYPE
PLT 65
CALL UGR((SMIN,SMAX,NX,SHF7,1),1,1,MNIN,MRMAX,MY,SHF6,1),1,0)
PLT 66
CALL EDIT((YMC1,3H16),TITLE(3),P)
PLT 67
CALL EDIT((HMC1,3H16),TITLE(4),F)
PLT 68
CALL HCRLIN(TITLE,64,512,1000)
PLT 69
CALL HCRLIN(15,TIME IN MINUTES,15,312,0)
PLT 70
IF(MOD(15,SAV)=7,LC=1)
PLT 71
CALL VERLIN(24,RESIDUALS IN ARC SECONDS,24,7,512)
PLT 72
IF(ISAVE.EQ.2) CALL VERLIN(19,RESIDUALS IN METERS,19,0,512)
PLT 73
IF(ISAVE.EQ.3) CALL VERLIN(31,RESIDUALS IN CENTIMETERS/SECOND,PLT 74
PLOT 75
.)S(L,512)
PLT 76
C PLOT RESIDUAL VS. TIME
PLT 77
DO 5 IS=IPASS,IST0P
PLT 78
IF(NNAME ..NE..NAME) GO TO 5
PLT 79
CALL PLUT(IFMIN(I),RESIC(1),1,1,4+)
PLT 80
5 CONTINUE
PLT 81
CALL FMINAV
PLT 82
C FIND MAXIMUM AND MINIMUM VALUES AND ROUND
PLT 83
CALL NEWX(NAME,RESID,IPASS,IST0P,MRMAX,MMIN)
PLT 84
CALL PTYNM(MMIN,MRMAX,MMIN,MRMAX,NX)
PLT 85
C PRINT TITLE AND GRID FOR MEASUREMENT RATE VS. RESIDUAL PLOT
PLT 86
CALL UGR((MMIN,MRMAX,MY,SHF7,0),1,MNIN,MMAX,MY,SHF6,1),1,0)
PLT 87
CALL HGCRLIN(TITLE,64,512,1000)
PLT 88
GO TO (11,12,13,14,15,16,17,18,19,22,23,24),ISAVE
PLT 89
11 CALL HGCRLIN(14,15,ARC RATE,11,512,1)
PLT 90
GO TO 20
PLT 91
12 CALL HGCRLIN(1CHARGE RATE,10,512,0)
PLT 92
GO TO 20
PLT 93
13 CALL HGCRLIN(11HR RATE,11,512,0)
PLT 94
GO TO 20
PLT 95
14 CALL HGCRLIN(14,FREQUENCY RATE,14,512,0)
PLT 96
GO TO 20
PLT 97
15 CALL HGCRLIN(10,ALPHA RATE,10,512,0)
PLT 98
GO TO 20
PLT 99
16 CALL HGCRLIN(12,ANGLE RATE,12,512,0)
PLT 100
GO TO 20
PLT 101
17 CALL HGCRLIN(11,HAZMUTH RATE,11,512,0)
PLT 102
GO TO 20
PLT 103
18 CALL HGCRLIN(16,DECLINATION RATE,16,512,0)
PLT 104
GO TO 20
PLT 105
22 CALL HGCRLIN(9,PHIETA RATE,9,512,0)
PLT 106
GO TO 20
PLT 107
23 CALL HGCRLIN(12,ANGLE RATE,12,512,0)
PLT 108
GO TO 20
PLT 109
24 CALL HGCRLIN(14,HELEVATION RATE,14,512,0)
PLT 110
25 IF(MOD(ISAVE,7).LE.1)
PLT 111
CALL VERLIN(24,RESIDUALS IN ARC SECONDS,24,0,512)
IF(ISAVE<.EQ.2) CALL VERLIN(13HRESIDUALS IN METERS,19,.512) PLOT 112
IF(ISAVE<.EQ.3) CALL VERLIN(31HRESIDUALS IN CENTIMETERS/SECOND,PLOT 113
C 31(6,.12)
C 00 1=IFASSIST,STOP
C PLOT RESIDUALS VS. MEASUREMENT RATE
C IF(NAME.NE.NAKEST(1)) GOTO 30
C CALL PLOT(OODOT(I),RESID(I),1,4H)
30 CONTINUE
C CALL FRMALV
C GU TO GU
55 PRINT 50
50 RETURN
END
DESCRIPTION

REGANL is the analysis subroutine of the GEORGE PROGRAM. It uses a least squares method to determine zero set measurement biases and timing errors for each pass of data. REGANL also computes standard deviations of the errors and the noise about the fitted line. Finally it performs a randomness test of the results.

If data is to be edited, REGANL uses the results of its computations to eliminate points above a specified rejection criterion and re-computes all of the results.
**NAME**
REGANL

**PURPOSE**
PERFORMS THE REGRESSION ANALYSIS
MAKES A RANDOMNESS TEST AND COMPUTES
ZERO SET AND TIMING ERRORS, IN PASSES OF DATA.

**CALLING SEQUENCE**
CALL REGANL(IPASS, IPASS2, NAME, MM)

**SYMBOL TYPE DESCRIPTION**

<table>
<thead>
<tr>
<th>IPASS</th>
<th>INPUT - INDEX OF BEGINNING POINT IN ARRAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPASS2</td>
<td>INPUT - INDEX OF END POINT IN ARRAY</td>
</tr>
<tr>
<td>NAME</td>
<td>INPUT - STATION NAME</td>
</tr>
<tr>
<td>MM</td>
<td>INPUT - NUMBER OF POINTS IN PASS</td>
</tr>
</tbody>
</table>

**SUBROUTINES USED**
NONE

**COMMON BLOCKS**
ARRAY: CCONST

**INPUT FILES**
NONE

**OUTPUT FILE**
PRINTER

**RESTRICTIONS**
NONE

**REFERENCES**
NONE

**SUBROUTINE REGANL (IPASS, IPASS2, NAME, MM)**

1. **FORMAT**(1F1.16)
   - STATION NAME = A6/

2. **FORMAT**(1F16.16)
   - DATE OF PASS = 16/
   - TIME OF PASS = 16/

3. **FORMAT**(1F6.5)
   - TIME OF DATA = 16H

4. **FORMAT**(1F14.16)
   - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

5. **FORMAT**(1F14.16)
   - TIME OF DATA = 16H

6. **FORMAT**(1F4.10)
   - STATION NAME = 16/

7. **FORMAT**(1F14.16)
   - DATE OF PASS = 16/

8. **FORMAT**(1F16.16)
   - TIME OF PASS = 16/

9. **FORMAT**(1F16.16)
   - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

10. **FORMAT**(1F14.16)
    - TIME OF DATA = 16H

11. **FORMAT**(1F14.16)
    - STATION NAME = 16/

12. **FORMAT**(1F4.10)
    - DATE OF PASS = 16/

13. **FORMAT**(1F16.16)
    - TIME OF PASS = 16/

14. **FORMAT**(1F16.16)
    - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

15. **FORMAT**(1F14.16)
    - TIME OF DATA = 16H

16. **FORMAT**(1F14.16)
    - STATION NAME = 16/

17. **FORMAT**(1F4.10)
    - DATE OF PASS = 16/

18. **FORMAT**(1F16.16)
    - TIME OF PASS = 16/

19. **FORMAT**(1F16.16)
    - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

20. **FORMAT**(1F14.16)
    - TIME OF DATA = 16H

21. **FORMAT**(1F14.16)
    - STATION NAME = 16/

22. **FORMAT**(1F4.10)
    - DATE OF PASS = 16/

23. **FORMAT**(1F16.16)
    - TIME OF PASS = 16/

24. **FORMAT**(1F16.16)
    - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

25. **FORMAT**(1F14.16)
    - TIME OF DATA = 16H

26. **FORMAT**(1F14.16)
    - STATION NAME = 16/

27. **FORMAT**(1F4.10)
    - DATE OF PASS = 16/

28. **FORMAT**(1F16.16)
    - TIME OF PASS = 16/

29. **FORMAT**(1F16.16)
    - MEAS. RATE = RESIDUAL ELEVATION = RGA 30

30. **FORMAT**(1F4.10)
    - DATE OF PASS = 16/
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

REGANL
Page 3 of 6
30 September 1972

2.10 FORMAT(1PE3.11) NOISE ABOUT THE LINE IS RANDOM
* 25M RANDOM NORMAL DEVIATE = G13.3)
2.17 FORMAT(1PE3.11) NOISE ABOUT THE LINE IS SIGNIFICANTLY NON RANDOM
* 25M RANDOM NORMAL DEVIATE = F3.11)
* 2.11 FORMAT(1PE3.11) REJECTED POINTS
DOUBLE PRECISION, ELEV, TYPE, NAME, NAMEST, TYPE
DOUBLE PRECISION KSTANA, .UISAVE, .REJECT, REJSW
COMMON /ARRAY / IYMD(14), IHM(4000), SFC(4000), ELEV(4000)
* RESID(400), OCOT(4000), ICOUNT(4000), NAMEST(4000), .UISAVE, .REJECT, REJSW
* .TYPE(11), KATC(4000)
COMMON/CONSTNET, TYPE, KSTANA, .UISAVE, .REJECT, REJSW
DINION>ICHECK(5+)
LOGICAL .REJSW

C INITIALIZE VARIABLES
IFLAG=0
31 SIGX=0.0
DO 10 I=1,5
1 SIGX=1.0
SIGY=0.0
SIGX50 = (-)
SIGY50 = 0.0
MM = 0
MMIEC
SIGX1=0.0
SIGY1=0.0
SIGXY1=0.0
SIGX51=0.0
SIGY51=0.0
K= 1
K=ISAVE+IFLAG/7
C PRINT HEADING
PRINT 20CC7,NAME,1YMD(IFPASS),1HM(IFPASS)
PRINT 20CC3,TYPE(K)
PRINT 20CC9
GO TO (1Z,1,2,4,4,1,1),ISAVE
1 PRINT 20CC12
GO TO 4
2 PRINT 20CC10
GO TO 4
3 PRINT 20CC11
4 CONTINUE
DO 20 L=IFPASS,IPASS2
IF(NAMEST(L),NC,NAME) GO TO 26
LL=L
IF(IFLAG.EQ.1) LL=LL+2000
IF ((ISAVE.EQ.0) OR (ISAVE.EQ.7)) ORDOT(LL)=ORDOT(LL)+10.0
IF ((ISAVE.EQ.0) OR (ISAVE.EQ.7)) ORDOT(LL)=ORDOT(LL)+6.1296E7/6.283165
MM = MM + 1
C SUM THE RATES, RESIDUALS, THEIR PRODUCTS AND THEIR SQUARES
SIGX = SIGX + (ORDOT(LL))
SIGY = SIGY + RESID(LL)
SIGXY = SIGXY + (ORDOT(LL)*RESID(LL))
SIGX50 = SIGX50 + (ORDOT(LL)+2)
SIGY50 = SIGY50 + (RESID(LL)+2)
PRINT 20CC9,1YMD(LL),1HM(LL),SIG(L),SIGY(LL),RESID(LL),ELEV(LL)
26 CONTINUE

1-64
Reproducibility of the original page is poor.

34 C=FLOAT(MM+MM1)
IFLAG2 = C
C TEST FOR MORE THAN TWO POINTS
JH((MM+MM1)*LT+1) PRINT P*214
IF ((MM+MM1)*LT+3) RETURN
C COMPUTE TIMING FACTOR
VAY=SIGAY+SIGXY1-(SIGX+SIGXI)*SIGY1/C
VX=SIGX+S1XSI-(SIGX+SIGXI)*2/C
VY=SIGYSI-S1YI-(SIGX+SIGXI)*2/C
D=VAY/VX
C COMPUTE ZEROU SIT
A=((SIGX+SIGXI)*B*(SIGX+SIGXI))/C
REGSS=VXY+VXY/VX
C COMPUTE THE REGRESSION MEAN
ALS=VY-REGSS
VVAR=RES/(C-1)**1
C COMPUTE STANDARD DEVIATIONS
B1ASU=SQR(RVAR+(SIGX+S1XSI)/(C+VX))
ENTIM = SQR(RVAR/VX)
I1 = 1
JJ = MM+MM1-2
NN = MM+MM1-1
C COMPUTE THE COVARIANCE ABOUT THE FITTED LINE
D = SQR(VVAR)
KESM3 = RES/(C-1)**1
C PRINT SUMMARY TABLE AT END OF EACH ANALYSIS
PRINT 2C15.3,ANAMES,H,ERTID,0,REGE:3,11,REGSS,RES,JJ,REMS,SY,NNREGA
IF((MM+MM1)*LT+10)GO TO 3E
DSQ=DSO/1
K=IPASS2+1
DO 32 L=IPASS,K
IF(NAMEST(L),NAME) GO TO 32
LL=L
IF(IFLAG+EQ.1)LL=LL+2C
L1=KEST1(LL)-A*O*O00T(L1)
D2=RES1(LL+1)-A*O*O00T(LL)
CSO=DSO+(C1-D2)**2
32 CONTINUE
DSO=DSO/(-1)**1
C PERFORM RANJVC=SS TEST
RND = (CSO/(2*O*OVAR)-1)**3/(SQR((C-2.0)/(C*C-1.0)))
IF (AAS(POD).LT.2.5) PRINT 2C016, RND
IF (AJS(POD).GT.2.53) PRINT 2C017, RND
38 CONTINUE
C TEST REJECTION VALUE
IF (*NJ+REJSW) GO TO 36
SIGX1=0.0
SIGY1=0.0
SIGXY1=0.0
SIGXSI=0.0
SIGYSI=0.0
MM1=C.0
K=0
DO 33 L=IPASS1,IPASS2
IF(NAMEST(L),NAME) GO TO 33
LL=L
IF (IFLAG. EQ. 1) LL=LL*2
DIRECTIONS = DIRECTIONS - 1

C TEST ACCEPTABILITY OF POINT
IF (A3 < JIERF) OR (REJECT) GO TO 23
K=K+1
IF (K.GT.1) GO TO 14

C PRINT INCREMENT
PRINT C1C19
PRINT C1C23
GO TO (11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23)

11 PRINT C1C12
GO TO 14
12 PRINT C1C10
GO TO 14
13 PRINT C1C11
14 CONTINUE

IF (ICHECK(K).NE.LL) IFLAG2 = 1
ICHECK(K) = LL

C PRINT TABLE OF REJECTED VALUES
PRINT C1C8, CTAB(L), CTAB(L), SEC(L), NHDOT(LL), RESID(LL), ELEV(L)
SIGX1=SIGX1-NHDOT(LL)
SIGX1=SIGX1-RESID(LL)
SIGX1=SIGX1-CONDOT(LL)*RESID(LL)
SIGX1=SIGX1-CONDOT(LL)**2
SIGX1=SIGX1-CONDOT(LL)**P
P=1=MA1-1

33 CONTINUE

76 CONTINUE
IF (K.GT.1) AND (IFLAG2.EQ.1) GO TO 34
IF (ISAVE.EQ.1) AND (ISAVE.EQ.7) RETURN
IF (IFLAG.EQ.1) RETURN
IFLAG=1
GO TO 31
END
DESCRIPTION

(See EPHemeris Tape Generator)
1.1.3 GROUNDTRACK

INTRODUCTION

GROUNDTRACK provides geometric insights into GEODYN results by plotting the satellite groundtrack for each pass over a particular station.

The main routine GROUNDTRACK controls the type of plot (groundtrack only or groundtrack with land plots), fixes the size of the grid, reads the data required for the groundtrack requested, and makes the required calls to the Plot Package.

The subroutine CENTER centers the station position on the plotting grid. The subroutine LAND finds the required data in the WRLMAP block data to plot the land masses on the grid. WRLMAP is part of the Plot Package.

The subroutine DATIME converts minutes into days, hours, and minutes. The subroutine ADDYMD is a member of GEODYN; DIFTIM is the same as subroutine DIFF in GEODYN; RYHDI is in GEORGE.

This program requires a minimum of 500K bytes of memory and uses as input one 9-track tape.
### Subroutine Cross Reference Chart

**Calling Routines**

<table>
<thead>
<tr>
<th></th>
<th>MAIN</th>
<th>CENTER</th>
<th>LAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDYMD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COORD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIFTIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENDPLT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HORLIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXMIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTYNUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RYMDI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERLIN</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DESCRIPTION

The main program GROUNDTRACK reads and separates satellite ephemeris data into passes by station and determines from the GROUNDTRACK INPUT CARDS which data is to be plotted.

GROUNDTRACK calls CENTER to center the station position on the grid. If requested it calls LAND to determine the land masses on the grid. Finally it calls the PLOT PACKAGE routines to make the plots.
NAME MAIN - GROUNDTRACK

PURPOSE PLOTS SATELLITE GROUNDTRACKS FOR A SPECIFIED STATION

SUBROUTINES USED ACDFMC CENTER DATIME DIFTIM LAND RYDI

COMMON BLOCKS NJNE

INPUT FILES GROUNDTRACK INPUT CARDS
             GROUNDTRACK TAPE

OUTPUT FILE S - PRINTER

RESTRICTIONS MAXIMUM OF ONE STATION PER PLOT

REFERENCES NJNE

DIMENSION ISTAN0(10), LAL(50), ITIME(50), IPDAY(100)
DIMENSION IPIN(720), LYPD(720), LH(720)
DIMENSION STAL(10), STALON(10), VALUE(6), SATLAT(720)
    SATLON(720), SATEH(720)
REAL*6 UPT, OPTION(5), LAST, END, BLANK, STANAM(10), NAME(730), EXTRAS
DATA OPTION/PLOTS' , TIME ' , 'GROSET' , 'LNDOPLT' , 'DATA '/
DATA LAST, BLANK/LAST '/
DATA END/END/END/
LOGICAL*1 FT(21)/FONE/
LOGICAL*1 FIXGFD, PLOTIN, TIMEIN, LANOPT, PRIME
CONTINUE

C READ IN STATION POSITION CARDS
10 READ(5,100) STANAM(NSTA+1), ISTAN0(NSTA+1), STALAT(NSTA+1).
       STALON(NSTA+1)
       IF(STANAM(NSTA+1).EQ.0) GO TO 40
       NSTA=NSTA+1
       GO TO 10
C READ UPTILNAL GROUNDTRACK INPUT CARDS
40 READ(5,100) OPT, VALUE
       DD 32 1=1+5
       IF(UPT.NE.OPTION(1)) GO TO 32
       GO TO (33,33,33,37,46), 1
JZ CONTINUE
C ERRORIOUS INPUT CARD
       WRITE(6,IC03) OPT
       GO TO 46
C SET SWITCH FOR SC620 TAPE
33 CONTINUE
FLUTIN=.TRUE.,
142C=.TRUE.,
IF(VALUE(1)*142C=.TRUE.)
GO TO 40
C SET TIME PARAMETERS FOR A DATA PERIOD REQUESTED
34 CONTINUE
TIMEIN=.TRUE.,
IYMDST=VALUE(1)+.5,
NHMST=VALUE(2)+.5,
IYMDEN=VALUE(2)+.5,
NHMN=VALUE(3)+.5,
GO TO 40
C SET GRID PARAMETERS FOR A FIXED GRID
36 CONTINUE
LANDPT=.TRUE.,
CALL HFLMAF
GO TO 40
C CONTINUE
CALL PLOIIT(14023,.TRUE.)
CALL FRMACV
WRITE(6,1017) STANAM(NSTA)
IF(LANUPT) WRITE(6,1018)
IF(=IXGK(4)) WRITE(6,1019)
IF(TIMEIN) WRITE(6,1020)
IF(PLCI?, AND.4 14023.EQ.4) WRITE(6,1021)
READ(11,1022) LHM(IDATA+1),LHM(IDATA+1),NM=(IDATA+1),
SATLAT(IDATA+1),SATLCN(IDATA+1),SATH(IDATA+1)
IF(NAME(ICATA+1).NE.STANAM(NSTA)) GO TO 30
NPASS=0
JHM=-64
C STORE STATION DATA AND TEST FOR GREENWICH MERIDIAN(Prime)
35 REIN 11
CALL DIFTIW(LYMO(1),0,LYMO(IDATA),LHM(IDATA),LOAY,LMIN)
IEN=40,LHAY*1440+LMIN
GO TO 30
C SEPARATE STATION DATA
DG SC (P=1,IDATA
ITF(MIN(IF),LT,1STNT) GO TO SC
ITF(MIN(IF),GT,1ENC) GC TO SC
NPINT=NFCINT+1
IF(PINF,MO,SATLUN(NPINT)+GT,13C) SATLUN(NPINT)=SATLUN(NPINT)
C TEST FOR NEW PASS
ITF(MIN(IF),GT,1STNT) GO TO 45
NPASS=NINT+1
C SAVE LAST DATA POINT
IFM(NPASS)=LH(IP)
ITIME(NPASS)=NPINT
AH=LYM(IP)
NYM=NYMC(IP)
IYDN=NYNO-NYRN*1CC
IPLDAY(NPASS)=IYDNJ
IF(NPASS.EQ.1) WRITE(6,1202) STANA(ISTA),IYMDST,NHMST,IYDEN.

PP:PN
JMM=IMIN(IP)
CALL LATE(IFMIN,IDAY)
JMM=INMIN
IYMD=LYM(IP)
JMM=IYMD(IDAY)
LINES=0
GO TO 43
43 IF(MIC(LINES,E=1,G=0) WRITE(6,1102)
WRITE(6,1102) LHAM(IP),SATLUN(IP),SATLUN(IP),SATH(IP)
IF(LINES.EQ.40) GO TO 42
LINES=LINES+1
GO TO 50
42 WRITE(6,1102) STANA(ISTA)
43 LINES=0
WRITE(6,1102) NPASS,IYMD,JMM
50 CONTINUE
C PLACE STATION IN MIDLE OF GRID
55 CONTINUE
ITF(NPOINT,EQ.6) GO TO 7C
ITIME(NPASS+1)=NPOINT+1
IF(FLUX1) GO TO 56
CALL CENTER(ISTALON(ISTA),SATLUN,NPINT,SATMIN,SATMAX)
CALL PTYLM(SATN,NPINT,SATM,SMIN,NMAX)
CALL CENTER(ISTALON(ISTA),SATLAT,NPINT,SATLM,SATLX)
CALL PTYLM(SATLN,SATLM,SATLX,FMIN,FMAX,NX)
CALL GRIC(SMIN,SMAX,NY,*3)*1,PMIN,PMA3,NX,*13)*1,i)
IF(LANDPT) CALL LAND(SWMIN,SMAX,PMIN,PMA1)
WRITE(6,1102) SWMIN,SMAX,NY,PMIN,PMA1
GO TO 57
C USING FIXED GRID METHOD
56 CONTINUE
CALL GRIC(SWMIN,SWMAX,NY,*3)*1,PMIN,PMA1
IF(LANDPT) CALL LAND(SWMIN,SWMAX,PMIN,PMA1)
GO TO 57
C CENTER AND LABEL EПИD
CALL EDIT(IYMDST,3M16),ICATE,P)
CALL HCBLIN(1CATE,6,512,1C16)
CALL HCLIN(STANAM(ISTA),6,F12.1CC0)
CALL HURLIN(I7XPLotted PERIOD 15.1,70C(4176))
CALL HCLIN(STATION POSITION,2C3;1CC0)
CALL EUIT(IYX,3H1c,FT(E),P)
CALL EUIT(IYXEN,JH(4),FT(11),F)
CALL HCLIN(IF3T,1.5551,1.1G)
CALL HCLIN(SPONTITUDE,5.512,1)
CALL VEHJNCL(PLATITUDE,5.512)
CALL PLOT(STALON(ISTA),STALAT(ISTA),1.4H *)

C PLUT PASS
CO CO IP=1,1NPASS
IFPT=1111EI(P)
CALL EUIT(INH(IP),3HI,JHM,GARO)
CALL EUIT(IPC,LAY(IP),3H2),NCA,NOTE)
CALL PLOT(SATLON(IFPT),STALAT(IFPT),1TIME(IP+1)=IFPT,4H S)
CALL PLOT(SATLON(IFPT),STALAT(IFPT),1,4H *)
CALL CLURE(SATLON(IFPT),STALAT(IFPT),IIX,IXY)
CALL CLURE(SATLON(IFPT+1),STALAT(IFPT+1),IIX,IXY)
IX=IX+1
DL=DQRT(0X*2+CY*2)
IX=2X*0X/2+IX
IV=2X*0Y/1Y
CALL HCLIN(IFXHM,4,IX,IXY-6)
CALL HCLIN(INCD,2,IX,IX+6)
60 CONTINUE
CALL SHARE
70 CONTINUE
C TEST FOR ENO CF DATA
72 READ(3,1(CS) EXTRAS
IF(READY.EQ.0) GO TO 5
IF(READY.EQ.LAST) GO TO 75
GO TO 72
75 CONTINUE
CALL ENDFLT
STOP
100 FORMAT(A4,F3.0,12X,F3.0)
1001 FORMAT(A6,2X,E11.1X,A6X,2X,F15,6,2X,F15,9,2X,F15,5)
1002 FORMAT(I1,19X,'STATION NAME',A6,19X,'START DATE',16,19X
'END DATE',16,19X
'END TIME',14,1X)
1003 FORMAT(I1,19X,'PASS NUMBER',13,19X,'DATE OF PASS',16,19X
'TIME UF PASS 14X,6X,'TIME',17X,'SATELLITE',6X,'HOUR',5X,'MINUTE',5X,'DEGREE',5X
'LATITUDE',4X,'LONGITUDE',5X,'HEIGHT',8X,'MINUTE',4X,'METERS')
1004 FORMAT(IH,5X,1X,5X,F9.3,2X,F9.3,3X,F15,2)
1005 FORMAT(A6,4X,6(F10.0))
1006 FORMAT(I15X,A9,3X,6CONT,3X)
1007 FORMAT(I1,19X,'OPTIONS REQUESTED ARE AS FOLLOWS',16,19X
'15X',EMAIL NAME 4X,'A6',1X)
1008 FORMAT(I1,1X,3A1,3X,'UNCORED REMAINING OPTIONS EXECUTION CONTINUING')
1009 FORMAT(I1,1X,4X,'BASIC GRID SIZE',6X,'LONGITUDE VALUE',3X
'LATITUDE VALUE',6X,'HIGH LOW INT HIGH LOW INT',16,19X
'5X,F5.1X,F5.1X,F5.1X,F5.1X,F5.1X,F5.1X,F5.1X,F5.1X')
1010 FORMAT(I1X)

1-75
**30th September 1972**

```
**FORMAT** (1F*10X,*WORLD-MAP OVERLAY OUTPUT) GANT 224
**FORMAT** (1F*10X,*GRID SET WITH THE VALUES*), 1X,*MINIMUM LATITUDE GANT 225
   *VALUE* ,*,F9.1/, 1X,*MAXIMUM LATITUDE VALUE* ,*,F9.1/, 1X,*NUMBER GANT 226
   *LATITUDE INTERVALS* ,*,13/, 1X,*MINIMUM LONGITUDE VALUE* ,*,F9.1/, 1X,*NUMBER OF LONITUDE GANT 227
   *INTERVALS* ,*,13/, 1X:
**FORMAT** (1F*10X,*TIME INTERVAL TO BE PLOTTED*), 1X:
   *10X,*START DATE* ,*,F9.1X,*START TIME* ,*,14/, 1X:
   *10X,*END DATE* ,*,F9.1X,*END TIME* ,*,14/, 1X:
**FORMAT** (1F*10X,*PRINT PLOT ONLY*), 1X:
**FORMAT** (1F*10X,*PRINT AND SCARG PLOT*), 1X:
END.
```
DESCRIPTION

CENTER calls MAXMIN to determine the center of the grid at which point it places the station.
## NAME
- CENTER

## PURPOSE
- PLACES A GIVEN PCINT IN THE CENTER OF A GRID

## CALLING SEQUENCE
- CALL CENTER(CENPT,ARRAY,N,CMIN,CMAX)

## SYMBOL TYPE DESCRIPTION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENPT</td>
<td>INPUT</td>
<td>POINT TO BE CENTERED</td>
</tr>
<tr>
<td>ARRAY</td>
<td>INPUT</td>
<td>ARRAY OF POINTS TO BE PLotted</td>
</tr>
<tr>
<td>N</td>
<td>INPUT</td>
<td>NUMBER OF ENTRIES IN THE ARRAY</td>
</tr>
<tr>
<td>CMIN</td>
<td>OUTPUT</td>
<td>MINIMUM VALUE OF THE PLOTTING SCALE</td>
</tr>
<tr>
<td>CMAX</td>
<td>OUTPUT</td>
<td>MAXIMUM VALUE OF THE PLOTTING SCALE</td>
</tr>
</tbody>
</table>

## SUBROUTINE USED
- MAXMIN

## COMMON BLOCKS
- NONE

## INPUT FILES
- NONE

## OUTPUT FILES
- NONE

## RESTRICTIONS
- NONE

## REFERENCES
- NONE

```fortran
SUBROUTINE CENTER(CENPT,ARRAY,N,CMIN,CMAX)  
DIMENSION ARRAY(N)  
C COMPUTE MAXIMA AND MINIMA  
CALL MAXMIN(ARRAY,N,CMIN,CMAX)  
FLNTH=AMAX1(CMAX-CENPT,CENPT-CMIN)  
C CENTER THE POINT  
CMIN=CENPT-FLNTH  
CMAX=CENPT+FLNTH  
RETURN  
END
```
DESCRIPTION

Subroutine DATIME converts a given number of minutes to days, hours and minutes.
NAME       DATIME
PURPOSE    CONVERTS MINUTES TO DAYS AND HOURS AND MINUTES
CALLING SEQUENCE  CALL DATIME(MIN, IHRMIN, IDAY)

SYMBOL   TYPE  DESCRIPTION
MIN     R     INPUT - MINUTES TO BE CONVERTED
IHRMIN  R     OUTPUT - NUMBER OF HOURS AND MINUTES AFTER CONVERSION
IDAY    R     OUTPUT - NUMBER OF DAYS AFTER CONVERSION

SUBROUTINES USED   NONE
COMMON BLOCKS    NONE
INPUT FILES      NONE
OUTPUT FILES     NONE
RESTRICTIONS    NONE
REFERENCES       NONE

SUBROUTINE DATIME(MIN, IHRMIN, IDAY)

C CONVERT MINUTES TO DAYS
IDAY=MIN/1440
IF(IDAY.(E+1) GO TO 20
MIN=MIN
GO TO 10
20 MIN=MIN-IDAY*1440
10 CONTINUE

C EXTRACT HOURS
IHR=MIN/60
LHR=MIN-IHR*60

C COMBINE HOURS AND MINUTES
IHRMIN=IHR+100*LHR
RETURN
END
DESCRIPTION

LAND determines the points which make up the land masses on the grid. It references the block data routine WRLMAP through the entry to EARTH to obtain the data.
NAME LAND
ENTRY POINT PURPOSE
LAND PLOTS LAND MASSES ON GRID
EARTH INITIALIZATION OF ARGUMENTS IN CALLING SEQUENCE

CALLING SEQUENCE CALL LAND(LONG1, LONG2, LAT1, LAT2)

SYMBOL TYPE DESCRIPTION
LONG1 R INPUT - STARTING LONGITUDINAL BOUNDARY OF THE REGION
LONG2 R INPUT - STOPPING LONGITUDINAL BOUNDARY OF THE REGION
LAT1 R INPUT - STARTING LATITUDINAL BOUNDARY OF THE REGION
LAT2 R INPUT - STOPPING LATITUDINAL BOUNDARY OF THE REGION

CALLING SEQUENCE CALL EARTH(LATL, LAT2, A3)

SYMBOL TYPE DESCRIPTION
NE I INPUT-OUTPUT - NUMBER OF BODIES OF LAND
NPOD I INPUT-OUTPUT - NUMBER OF VECTOR POINTS IN EACH BODY
A R INPUT-OUTPUT - LONGITUDE OF THE VECTOR POINTS
B R INPUT-OUTPUT - LATITUDE OF THE VECTOR POINTS

SUBROUTINE USED PLOT
CCMCH BLOCKS NONE
INPUT FILES NONE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES NONE

SUBROUTINE LAND(LONG1, LONG2, LAT1, LAT2)
REAL LONG1, LONG2, LAT1, LAT2, A(1), B(1)
LOGICAL LKG
INTEGER NPOD(1)
LKG, LONG1, LAT1, LAT2

C FIND FIRST POINT IN DATA TO BE ON THE GRID
CU 30  I=1,NE
  K=0
  NP=NBOD(1)
  DO 20  J=1,MP
    IP=IP+1
  C IF NEGATIVE INTERVAL, SUBTRACT FROM 360
    IF(A(IP).AND.(A(IP).GT.150.).AND.(-A(IP)).GT.360.)
      A(IP)=360.-A(IP)
  C TEST IF POINTS ARE ON GRID
  C PLOT POINTS
    IF(N.GT,0) CALL PLOT(A(IP-N),B(IP-N),N,
      A=IP-N+1
      GO TO 20
    C TEST IF PLOT POINTS EXTEND BEYOND INTERVAL, PLOT TO EXTREMITIES
      IF(N.GT,0) CALL PLOT(A(IP-N+1),B(IP-N+1),N,
        A=IP-N+1
        GO TO 30
      C REVERSE PLOTTING
        IP=0
        GC 60  I=1,NE
        K=NBOD(1)
        DO 40  J=1,K
          IP=IP+1
          IF(ATA(IP).LT.0.) A(IP)=360.-A(IF)
          CONTINUE
          ENTRY EARTHNE,NBOD,A,E
          RETURN
        CONTINUE
        END
DESCRIPTION

WRLMAP contains all the data needed to plot land masses anywhere on the earth.
### NAME
WRLMAP

### PURPOSE
TO CALL EARTH WITH THE WORLD MAP DATA

### CALLING SEQUENCE
CALL WRLMAP

### SUBROUTINE USED
EARTH

### COMMON BLOCKS
NONE

### INPUT FILES
NONE

### OUTPUT FILES
NONE

### RESTRICTIONS
NONE

### REFERENCES
NONE

---

**SUBROUTINE WRLMAP**

**REAL**
- 4: A 1(490), A 2(135), A 3(139), A 4(139), A 5(139), A 6(139),
- A 7(135), A 8(135), A 9(139), A 10(139), A 11(139), A 12(139),
- A 13(135), A 14(135), A 15(139), A 16(139), A 17(139), A 18(139),
- A 19(139), A 20(139), A 21(139), A 22(139), A 23(139), A 24(139),
- A 25(139), A 26(139), A 27(139), A 28(139), A 29(139), A 30(139),
- A 31(139), A 32(139), A 33(139), A 34(139), A 35(139), A 36(139),
- B 1(490), B 2(135), B 3(139), B 4(139), B 5(139), B 6(139),
- B 7(135), B 8(135), B 9(139), B 10(139), B 11(139), B 12(139),
- B 13(139), B 14(139), B 15(139), B 16(139), B 17(139), B 18(139),
- B 19(139), B 20(139), B 21(139), B 22(139), B 23(139), B 24(139),
- B 25(139), B 26(139), B 27(139), B 28(139), B 29(139), B 30(139),
- B 31(139), B 32(139), B 33(139), B 34(139), B 35(139), B 36(139), B 37(139)

**INTEGER**
- C(110)

**EQUIVALENCE**
- (A 2(1), A 1(140)), (A 3(1), A 1(279)), (A 4(1), A 1(418)),
- (A 5(1), A 1(557)), (A 6(1), A 1(696)), (A 7(1), A 1(835)),
- (A 8(1), A 1(974)), (A 9(1), A 1(1113)), (A 10(1), A 1(1252)),
- (A 11(1), A 1(1391)), (A 12(1), A 1(1530)), (A 13(1), A 1(1669)),
- (A 14(1), A 1(1809)), (A 15(1), A 1(1947)), (A 16(1), A 1(2066)),
- (A 17(1), A 1(2225)), (A 18(1), A 1(2364)), (A 19(1), A 1(2503)),
- (A 20(1), A 1(2642)), (A 21(1), A 1(2781)), (A 22(1), A 1(2920)),
- (A 23(1), A 1(355)), (A 24(1), A 1(3198)), (A 25(1), A 1(3337)),
- (A 26(1), A 1(3476)), (A 27(1), A 1(3513)), (A 28(1), A 1(3754)),
- (A 29(1), A 1(3930)), (A 30(1), A 1(4037)), (A 31(1), A 1(4171)),
- (A 32(1), A 1(4319)), (A 33(1), A 1(4447)), (A 34(1), A 1(4580)),
- (A 35(1), A 1(4727)), (A 36(1), A 1(4866))

**EQUIVALENCE**
- (B 2(1), A 1(420)), (B 3(1), A 1(279)), (B 4(1), A 1(1252)),
- (B 5(1), A 1(557)), (B 6(1), A 1(696)), (B 7(1), A 1(835)),
- (B 8(1), A 1(1113)), (B 9(1), A 1(1252)), (B 10(1), A 1(1391)),
- (B 11(1), A 1(1530)), (B 12(1), A 1(1669)), (B 13(1), A 1(1809)),
- (B 14(1), A 1(1947)), (B 15(1), A 1(2066)), (B 16(1), A 1(2225)),
- (B 17(1), A 1(2364)), (B 18(1), A 1(2503)), (B 19(1), A 1(2642)),
- (B 20(1), A 1(2781)), (B 21(1), A 1(2920)), (B 22(1), A 1(355)),
- (B 23(1), A 1(3198)), (B 24(1), A 1(3337)), (B 25(1), A 1(3476)),
- (B 26(1), A 1(3513)), (B 27(1), A 1(3754)), (B 28(1), A 1(3930)),
- (B 29(1), A 1(4037)), (B 30(1), A 1(4171)), (B 31(1), A 1(4319)),
- (B 32(1), A 1(4447)), (B 33(1), A 1(4580)), (B 34(1), A 1(4727)),
- (B 35(1), A 1(4866))
<table>
<thead>
<tr>
<th>DATA A 1/</th>
<th>275.41</th>
<th>275.71</th>
<th>278.34</th>
<th>277.97</th>
<th>278.34</th>
<th>277.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>276.71</td>
<td>275.53</td>
<td>274.65</td>
<td>274.79</td>
<td>275.16</td>
<td>275.23</td>
<td>276.41</td>
</tr>
<tr>
<td>276.71</td>
<td>277.67</td>
<td>278.12</td>
<td>273.23</td>
<td>279.45</td>
<td>270.71</td>
<td>277.62</td>
</tr>
<tr>
<td>277.78</td>
<td>276.05</td>
<td>274.86</td>
<td>274.05</td>
<td>273.97</td>
<td>273.97</td>
<td>272.47</td>
</tr>
<tr>
<td>274.19</td>
<td>273.97</td>
<td>274.56</td>
<td>273.69</td>
<td>272.94</td>
<td>273.31</td>
<td>273.01</td>
</tr>
<tr>
<td>272.27</td>
<td>271.31</td>
<td>273.57</td>
<td>269.97</td>
<td>258.94</td>
<td>268.13</td>
<td>268.87</td>
</tr>
<tr>
<td>269.66</td>
<td>267.90</td>
<td>272.42</td>
<td>271.04</td>
<td>271.97</td>
<td>272.34</td>
<td>271.75</td>
</tr>
<tr>
<td>270.80</td>
<td>270.12</td>
<td>265.05</td>
<td>260.27</td>
<td>257.75</td>
<td>257.45</td>
<td>266.67</td>
</tr>
<tr>
<td>266.50</td>
<td>266.04</td>
<td>266.72</td>
<td>264.50</td>
<td>255.76</td>
<td>265.91</td>
<td>264.94</td>
</tr>
<tr>
<td>264.16</td>
<td>264.28</td>
<td>264.28</td>
<td>264.91</td>
<td>255.24</td>
<td>266.27</td>
<td>267.24</td>
</tr>
<tr>
<td>267.10</td>
<td>266.20</td>
<td>265.30</td>
<td>270.49</td>
<td>271.31</td>
<td>272.71</td>
<td>274.27</td>
</tr>
<tr>
<td>275.08</td>
<td>274.62</td>
<td>273.69</td>
<td>276.93</td>
<td>276.86</td>
<td>277.75</td>
<td>277.67</td>
</tr>
<tr>
<td>276.63</td>
<td>275.96</td>
<td>278.66</td>
<td>275.00</td>
<td>278.95</td>
<td>266.12</td>
<td>280.71</td>
</tr>
<tr>
<td>260.12</td>
<td>262.26</td>
<td>279.67</td>
<td>279.69</td>
<td>280.70</td>
<td>282.41</td>
<td>282.46</td>
</tr>
<tr>
<td>251.67</td>
<td>281.89</td>
<td>291.41</td>
<td>281.05</td>
<td>281.45</td>
<td>281.45</td>
<td>281.23</td>
</tr>
<tr>
<td>281.00</td>
<td>281.82</td>
<td>283.00</td>
<td>283.59</td>
<td>284.70</td>
<td>285.37</td>
<td>286.13</td>
</tr>
<tr>
<td>287.52</td>
<td>267.59</td>
<td>285.38</td>
<td>263.66</td>
<td>289.07</td>
<td>285.91</td>
<td>289.59</td>
</tr>
<tr>
<td>290.26</td>
<td>290.63</td>
<td>251.25</td>
<td>252.77</td>
<td>253.73</td>
<td>292.95</td>
<td>293.73</td>
</tr>
<tr>
<td>294.47</td>
<td>295.30</td>
<td>295.25</td>
<td>295.16</td>
<td>295.14</td>
<td>295.98</td>
<td>296.60</td>
</tr>
<tr>
<td>296.92</td>
<td>296.62</td>
<td>275.78</td>
<td>275.21</td>
<td>297.66</td>
<td>296.47</td>
<td>295.77</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA B 1/</th>
<th>68.96</th>
<th>68.56</th>
<th>68.27</th>
<th>67.57</th>
<th>66.79</th>
<th>66.35</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.01</td>
<td>66.54</td>
<td>66.35</td>
<td>65.81</td>
<td>65.71</td>
<td>65.18</td>
<td>65.13</td>
</tr>
<tr>
<td>64.74</td>
<td>64.64</td>
<td>64.01</td>
<td>63.81</td>
<td>63.18</td>
<td>63.16</td>
<td>63.23</td>
</tr>
<tr>
<td>63.71</td>
<td>63.32</td>
<td>62.75</td>
<td>62.13</td>
<td>63.28</td>
<td>63.52</td>
<td>64.15</td>
</tr>
<tr>
<td>64.44</td>
<td>64.04</td>
<td>65.42</td>
<td>66.10</td>
<td>65.96</td>
<td>65.76</td>
<td>65.27</td>
</tr>
<tr>
<td>65.67</td>
<td>65.67</td>
<td>65.71</td>
<td>65.57</td>
<td>65.96</td>
<td>65.62</td>
<td>65.47</td>
</tr>
<tr>
<td>65.37</td>
<td>65.13</td>
<td>65.32</td>
<td>65.06</td>
<td>65.08</td>
<td>64.40</td>
<td>63.37</td>
</tr>
<tr>
<td>63.86</td>
<td>63.57</td>
<td>63.76</td>
<td>62.93</td>
<td>62.84</td>
<td>63.26</td>
<td>63.57</td>
</tr>
<tr>
<td>63.13</td>
<td>62.69</td>
<td>63.30</td>
<td>61.71</td>
<td>61.32</td>
<td>60.93</td>
<td>60.93</td>
</tr>
<tr>
<td>60.15</td>
<td>58.18</td>
<td>58.04</td>
<td>58.79</td>
<td>58.45</td>
<td>58.74</td>
<td>59.01</td>
</tr>
<tr>
<td>56.59</td>
<td>56.74</td>
<td>56.66</td>
<td>55.47</td>
<td>55.33</td>
<td>54.69</td>
<td>54.79</td>
</tr>
<tr>
<td>54.99</td>
<td>54.45</td>
<td>53.62</td>
<td>52.99</td>
<td>52.30</td>
<td>51.67</td>
<td>51.23</td>
</tr>
<tr>
<td>50.69</td>
<td>50.99</td>
<td>51.13</td>
<td>51.52</td>
<td>51.23</td>
<td>51.52</td>
<td>51.06</td>
</tr>
<tr>
<td>52.69</td>
<td>53.30</td>
<td>53.96</td>
<td>54.66</td>
<td>54.74</td>
<td>55.86</td>
<td>56.30</td>
</tr>
<tr>
<td>56.98</td>
<td>57.72</td>
<td>56.55</td>
<td>59.13</td>
<td>59.67</td>
<td>60.11</td>
<td>61.15</td>
</tr>
<tr>
<td>62.15</td>
<td>62.56</td>
<td>62.50</td>
<td>62.20</td>
<td>62.35</td>
<td>62.01</td>
<td>61.96</td>
</tr>
<tr>
<td>61.96</td>
<td>61.63</td>
<td>60.86</td>
<td>60.45</td>
<td>60.15</td>
<td>59.42</td>
<td>56.55</td>
</tr>
<tr>
<td>58.35</td>
<td>57.42</td>
<td>57.76</td>
<td>57.67</td>
<td>57.67</td>
<td>57.47</td>
<td>57.72</td>
</tr>
<tr>
<td>59.81</td>
<td>59.72</td>
<td>59.37</td>
<td>58.55</td>
<td>57.67</td>
<td>57.47</td>
<td>57.72</td>
</tr>
<tr>
<td>57.18</td>
<td>56.90</td>
<td>56.31</td>
<td>56.30</td>
<td>55.89</td>
<td>55.62</td>
<td>55.23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA C 1/</th>
<th>295.56</th>
<th>295.76</th>
<th>301.50</th>
<th>299.88</th>
<th>300.32</th>
<th>301.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>302.17</td>
<td>303.13</td>
<td>303.65</td>
<td>303.13</td>
<td>303.66</td>
<td>301.21</td>
<td>300.02</td>
</tr>
<tr>
<td>295.51</td>
<td>295.91</td>
<td>297.51</td>
<td>297.51</td>
<td>297.65</td>
<td>295.16</td>
<td>296.19</td>
</tr>
<tr>
<td>297.51</td>
<td>296.55</td>
<td>295.56</td>
<td>295.56</td>
<td>294.97</td>
<td>294.58</td>
<td>293.95</td>
</tr>
<tr>
<td>293.07</td>
<td>294.37</td>
<td>294.10</td>
<td>295.14</td>
<td>295.21</td>
<td>294.23</td>
<td>296.43</td>
</tr>
<tr>
<td>295.11</td>
<td>295.56</td>
<td>294.77</td>
<td>297.60</td>
<td>296.99</td>
<td>295.14</td>
<td>294.25</td>
</tr>
<tr>
<td>292.77</td>
<td>293.86</td>
<td>294.70</td>
<td>293.36</td>
<td>292.10</td>
<td>292.11</td>
<td>290.52</td>
</tr>
<tr>
<td>DATA A12/</td>
<td>DATA A13/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64.49</td>
<td>58.70</td>
<td>67.71</td>
<td>67.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.58</td>
<td>37.76</td>
<td>25.21</td>
<td>25.13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.72</td>
<td>36.17</td>
<td>35.63</td>
<td>35.13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.17</td>
<td>34.93</td>
<td>35.63</td>
<td>35.13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43.64</td>
<td>44.31</td>
<td>43.62</td>
<td>43.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43.64</td>
<td>45.83</td>
<td>45.62</td>
<td>45.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48.23</td>
<td>45.75</td>
<td>50.92</td>
<td>52.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54.23</td>
<td>56.37</td>
<td>57.88</td>
<td>57.88</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61.18</td>
<td>62.90</td>
<td>63.77</td>
<td>64.29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67.23</td>
<td>67.70</td>
<td>68.68</td>
<td>69.61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71.77</td>
<td>71.64</td>
<td>72.20</td>
<td>72.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72.51</td>
<td>73.10</td>
<td>73.43</td>
<td>74.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74.43</td>
<td>74.21</td>
<td>74.65</td>
<td>75.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75.39</td>
<td>76.13</td>
<td>75.97</td>
<td>76.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77.76</td>
<td>78.58</td>
<td>79.67</td>
<td>80.13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79.44</td>
<td>79.93</td>
<td>81.03</td>
<td>81.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84.35</td>
<td>84.72</td>
<td>86.13</td>
<td>86.37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.61</td>
<td>90.01</td>
<td>91.34</td>
<td>91.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92.12</td>
<td>92.64</td>
<td>93.63</td>
<td>93.38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94.18</td>
<td>93.30</td>
<td>93.90</td>
<td>93.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66.15</td>
<td>66.26</td>
<td>66.47</td>
<td>66.80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63.03</td>
<td>62.98</td>
<td>63.57</td>
<td>64.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63.86</td>
<td>64.15</td>
<td>64.83</td>
<td>65.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65.37</td>
<td>66.15</td>
<td>67.42</td>
<td>68.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68.15</td>
<td>68.71</td>
<td>66.93</td>
<td>66.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66.59</td>
<td>67.59</td>
<td>67.11</td>
<td>67.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67.23</td>
<td>67.76</td>
<td>67.96</td>
<td>67.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.58</td>
<td>28.72</td>
<td>25.43</td>
<td>25.72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.16</td>
<td>22.65</td>
<td>22.51</td>
<td>21.97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.43</td>
<td>21.87</td>
<td>21.36</td>
<td>20.99</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.48</td>
<td>19.92</td>
<td>17.92</td>
<td>16.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.61</td>
<td>13.78</td>
<td>13.34</td>
<td>13.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.58</td>
<td>10.75</td>
<td>10.17</td>
<td>9.73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.31</td>
<td>6.41</td>
<td>5.44</td>
<td>9.92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.41</td>
<td>14.81</td>
<td>15.53</td>
<td>16.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.80</td>
<td>18.58</td>
<td>20.17</td>
<td>20.51</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.92</td>
<td>22.75</td>
<td>23.19</td>
<td>22.95</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.49</td>
<td>20.17</td>
<td>19.97</td>
<td>19.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.97</td>
<td>17.00</td>
<td>16.70</td>
<td>16.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97.67</td>
<td>97.36</td>
<td>97.07</td>
<td>97.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97.67</td>
<td>97.52</td>
<td>97.45</td>
<td>97.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59.45</td>
<td>59.23</td>
<td>59.75</td>
<td>59.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.66</td>
<td>104.54</td>
<td>104.25</td>
<td>104.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106.49</td>
<td>102.71</td>
<td>102.19</td>
<td>101.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109.12</td>
<td>95.65</td>
<td>95.82</td>
<td>97.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.66</td>
<td>102.26</td>
<td>103.52</td>
<td>103.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104.42</td>
<td>106.32</td>
<td>106.57</td>
<td>107.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106.92</td>
<td>106.57</td>
<td>106.57</td>
<td>106.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107.90</td>
<td>106.77</td>
<td>107.14</td>
<td>107.14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111.51</td>
<td>112.67</td>
<td>113.44</td>
<td>114.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110.61</td>
<td>115.51</td>
<td>120.75</td>
<td>120.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121.00</td>
<td>120.44</td>
<td>121.91</td>
<td>122.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121.50</td>
<td>120.62</td>
<td>119.66</td>
<td>119.61</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**WRLMAP**

Page 11 of 28
30 September 1972

1-95
| DATA A11/  | 106.34 | 185.08 | 184.34 | 182.27 | 180.79 | WRLM 616 |
| DATA B15/  | 66.74  | 66.93  | 66.90  | 66.93  | 66.92  | WRLM 616 |
| DATA A16/  | 85.28  | 85.28  | 85.28  | 85.28  | 85.28  | WRLM 617 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 618 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 619 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 620 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 621 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 622 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 623 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 624 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 625 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 626 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 627 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 628 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 629 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 630 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 631 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 632 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 633 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 634 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 635 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 636 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 637 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 638 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 639 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 640 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 641 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 642 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 643 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 644 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 645 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 646 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 647 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 648 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 649 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 650 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 651 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 652 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 653 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 654 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 655 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 656 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 657 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 658 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 659 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 660 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 661 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 662 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 663 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 664 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 665 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 666 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 667 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 668 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 669 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 670 |
| 52.97  | 52.97  | 52.97  | 52.97  | 52.97  | WRLM 671 |
|----------|------|-------|------|---------|---------|---------|---------|---------|
| -83.12,  | 174.90,  | 175.90,  | 176.90,  | 177.90,  | 178.90,  | 179.90,  | 180.90,  | 181.90,  |
| 167.70,  | 168.70,  | 169.70,  | 170.70,  | 171.70,  | 172.70,  | 173.70,  | 174.70,  | 175.70,  |
| 163.90,  | 164.90,  | 165.90,  | 166.90,  | 167.90,  | 168.90,  | 169.90,  | 170.90,  | 171.90,  |
| 157.10,  | 158.10,  | 159.10,  | 160.10,  | 161.10,  | 162.10,  | 163.10,  | 164.10,  | 165.10,  |
| 164.70,  | 165.70,  | 166.70,  | 167.70,  | 168.70,  | 169.70,  | 170.70,  | 171.70,  | 172.70,  |
| 162.70,  | 163.70,  | 164.70,  | 165.70,  | 166.70,  | 167.70,  | 168.70,  | 169.70,  | 170.70,  |
| 161.70,  | 162.70,  | 163.70,  | 164.70,  | 165.70,  | 166.70,  | 167.70,  | 168.70,  | 169.70,  |

**Data**

Page 16 of 28

30 September 1970
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>292.77</td>
<td>293.73</td>
</tr>
<tr>
<td>294.79</td>
<td>295.58</td>
</tr>
<tr>
<td>296.80</td>
<td>297.59</td>
</tr>
<tr>
<td>298.61</td>
<td>299.30</td>
</tr>
<tr>
<td>300.71</td>
<td>301.50</td>
</tr>
<tr>
<td>302.72</td>
<td>303.51</td>
</tr>
<tr>
<td>304.73</td>
<td>305.52</td>
</tr>
<tr>
<td>306.80</td>
<td>307.69</td>
</tr>
<tr>
<td>308.72</td>
<td>309.51</td>
</tr>
<tr>
<td>310.73</td>
<td>311.52</td>
</tr>
<tr>
<td>312.74</td>
<td>313.53</td>
</tr>
<tr>
<td>314.75</td>
<td>315.54</td>
</tr>
<tr>
<td>316.76</td>
<td>317.55</td>
</tr>
<tr>
<td>318.77</td>
<td>319.56</td>
</tr>
<tr>
<td>320.78</td>
<td>321.57</td>
</tr>
<tr>
<td>322.79</td>
<td>323.58</td>
</tr>
<tr>
<td>324.70</td>
<td>325.59</td>
</tr>
<tr>
<td>326.81</td>
<td>327.60</td>
</tr>
<tr>
<td>328.72</td>
<td>329.51</td>
</tr>
<tr>
<td>330.73</td>
<td>331.52</td>
</tr>
<tr>
<td>332.74</td>
<td>333.53</td>
</tr>
<tr>
<td>334.75</td>
<td>335.54</td>
</tr>
<tr>
<td>336.76</td>
<td>337.55</td>
</tr>
<tr>
<td>338.77</td>
<td>339.56</td>
</tr>
<tr>
<td>340.78</td>
<td>341.57</td>
</tr>
<tr>
<td>342.79</td>
<td>343.58</td>
</tr>
<tr>
<td>344.80</td>
<td>345.59</td>
</tr>
<tr>
<td>DATA</td>
<td>70.60</td>
</tr>
<tr>
<td>75.55</td>
<td>76.54</td>
</tr>
<tr>
<td>80.50</td>
<td>81.49</td>
</tr>
<tr>
<td>85.45</td>
<td>86.44</td>
</tr>
<tr>
<td>90.40</td>
<td>91.39</td>
</tr>
<tr>
<td>95.35</td>
<td>96.34</td>
</tr>
<tr>
<td>100.30</td>
<td></td>
</tr>
</tbody>
</table>

WRLMAP
Page 19 of 28
30 September 1972
| 135.12 | 134.61 | 132.54 | 132.97 | 137.25 | 132.65 | 132.01 | WFLM1120 |
| 131.72 | 121.32 | 140.90 | 131.27 | 131.72 | 131.72 | 132.64 | WFLM1121 |
| 133.77 | 132.10 | 133.13 | 134.15 | 134.54 | 134.92 | 135.50 | WFLM1122 |
| 136.03 | 135.41 | 135.59 | 140.37 | 142.16 | 142.75 | 143.04 | WFLM1123 |
| 144.52 | 145.49 | 146.25 | 148.45 | 148.33 | 147.63 | 148.64 | WFLM1124 |

**DATA U27**

| 76.25 | 78.04 | 78.34 | 78.32 | 77.91 | 77.65 | 75.44 | WFLM1125 |
| 79.03 | 76.93 | 77.52 | 77.91 | 77.95 | 78.10 | 76.35 | WFLM1126 |
| 78.34 | 76.54 | 77.26 | 77.17 | 73.33 | 72.73 | 70.03 | WFLM1127 |
| 79.71 | 75.52 | 76.37 | 75.93 | 73.91 | 73.66 | 73.76 | WFLM1128 |
| 73.68 | 74.15 | 74.69 | 74.74 | 74.93 | 74.59 | 74.59 | WFLM1129 |
| 73.96 | 77.17 | 76.76 | 76.85 | 77.27 | 77.67 | 77.61 | WFLM1130 |
| 77.27 | 76.20 | 75.81 | 75.91 | 76.35 | 76.28 | 76.56 | WFLM1131 |
| 76.30 | 68.40 | 68.40 | 68.40 | 68.40 | 68.69 | 69.98 | WFLM1132 |

**DATA A28**

| 148.00 | 148.36 | 148.67 | 149.04 | 147.63 | 150.67 | WFLM1133 |
| 151.76 | 151.04 | 150.47 | 150.47 | 151.43 | 151.43 | 151.93 | WFLM1134 |
| 119.80 | 119.51 | 119.65 | 122.67 | 123.68 | 123.66 | 123.66 | WFLM1135 |
| 119.45 | 119.95 | 120.51 | 121.65 | 122.47 | 123.29 | 123.29 | WFLM1136 |
| 124.47 | 124.69 | 124.69 | 125.43 | 125.65 | 125.65 | 125.65 | WFLM1137 |
| 124.24 | 122.84 | 122.92 | 122.79 | 122.10 | 122.09 | 122.09 | WFLM1138 |
| 123.00 | 124.76 | 124.69 | 123.87 | 123.53 | 123.53 | 123.53 | WFLM1139 |
| 122.69 | 122.62 | 122.91 | 123.50 | 123.50 | 123.50 | 123.50 | WFLM1140 |
| 122.69 | 122.47 | 121.95 | 121.95 | 121.95 | 121.95 | 121.95 | WFLM1141 |
| 120.69 | 121.29 | 120.95 | 120.99 | 120.99 | 120.99 | 120.99 | WFLM1142 |
| 120.91 | 120.62 | 118.62 | 118.62 | 116.63 | 116.63 | 116.63 | WFLM1143 |
| 111.37 | 105.50 | 105.48 | 105.50 | 105.49 | 105.14 | 105.14 | WFLM1144 |
| 110.37 | 105.07 | 105.33 | 110.79 | 110.79 | 110.79 | 110.79 | WFLM1145 |
| 113.36 | 113.51 | 113.59 | 114.92 | 115.35 | 115.35 | 115.35 | WFLM1146 |
| 116.18 | 116.77 | 117.43 | 118.60 | 118.60 | 118.60 | 118.60 | WFLM1147 |
| 118.32 | 118.17 | 117.73 | 119.17 | 119.17 | 119.17 | 119.17 | WFLM1148 |
| 118.69 | 118.17 | 118.03 | 118.03 | 118.03 | 118.03 | 118.03 | WFLM1149 |
| 116.29 | 116.25 | 115.95 | 114.92 | 114.92 | 114.92 | 114.92 | WFLM1150 |
| 95.60 | 97.30 | 97.45 | 97.67 | 97.67 | 97.67 | 97.67 | WFLM1151 |
| 96.50 | 98.86 | 150.24 | 100.19 | 100.34 | 100.34 | 100.34 | WFLM1152 |

**DATA 628**

| -6.10 | -10.02 | -10.22 | -5.49 | -5.14 | -4.51 | -4.12 | WFLM1153 |
| -3.39 | -2.76 | -2.22 | -1.53 | -1.19 | -0.37 | -0.22 | WFLM1154 |
| 0.51 | 0.90 | 1.10 | 1.34 | 1.10 | 0.65 | 0.76 | WFLM1155 |
| 0.71 | 1.50 | 1.73 | 1.73 | 1.15 | 0.70 | 0.32 | WFLM1156 |
| 0.32 | 0.32 | 0.37 | -0.22 | -0.32 | -0.17 | -0.56 | WFLM1157 |
| -0.41 | -1.24 | -1.19 | -1.10 | -1.10 | -1.25 | -1.34 | WFLM1158 |
| -1.24 | -3.32 | -3.19 | -3.44 | -4.12 | -4.51 | -5.10 | WFLM1159 |
| -2.61 | -3.19 | -3.05 | -3.49 | -2.17 | -2.22 | -4.55 | WFLM1161 |
| -5.00 | -5.63 | -5.97 | -3.69 | -3.58 | -3.10 | -3.19 | WFLM1162 |
| DATA A297 | 105.50 | 105.55 | 105.67 | 105.74 | 105.86 | 106.00 | 106.04 | 106.08 | 106.20 | 106.29 | 106.32 | 106.35 | 106.39 | 106.53 |
| 106.56 | 106.64 | 106.68 | 106.80 | 106.84 | 106.89 | 106.93 | 107.00 | 107.04 | 107.08 | 107.13 | 107.16 | 107.20 | 107.24 | 107.29 |
| 107.33 | 107.37 | 107.41 | 107.45 | 107.49 | 107.53 | 107.57 | 107.60 | 107.64 | 107.68 | 107.73 | 107.76 | 107.80 | 107.84 | 107.88 |
| 108.51 | 108.55 | 108.59 | 108.63 | 108.67 | 108.71 | 108.75 | 108.79 | 108.83 | 108.87 | 108.91 | 108.95 | 108.99 | 109.03 | 109.07 |
| 110.01 | 110.05 | 110.09 | 110.13 | 110.17 | 110.21 | 110.25 | 110.29 | 110.33 | 110.37 | 110.41 | 110.45 | 110.49 | 110.53 | 110.57 |
| 110.61 | 110.65 | 110.69 | 110.73 | 110.77 | 110.81 | 110.85 | 110.89 | 110.93 | 110.97 | 111.01 | 111.05 | 111.09 | 111.13 | 111.17 |
| 111.21 | 111.25 | 111.29 | 111.33 | 111.37 | 111.41 | 111.45 | 111.49 | 111.53 | 111.57 | 111.61 | 111.65 | 111.69 | 111.73 | 111.77 |
| 111.81 | 111.85 | 111.89 | 111.93 | 111.97 | 112.01 | 112.05 | 112.09 | 112.13 | 112.17 | 112.21 | 112.25 | 112.29 | 112.33 | 112.37 |
| 112.41 | 112.45 | 112.49 | 112.53 | 112.57 | 112.61 | 112.65 | 112.69 | 112.73 | 112.77 | 112.81 | 112.85 | 112.89 | 112.93 | 112.97 |
| 113.01 | 113.05 | 113.09 | 113.13 | 113.17 | 113.21 | 113.25 | 113.29 | 113.33 | 113.37 | 113.41 | 113.45 | 113.49 | 113.53 | 113.57 |
| 113.61 | 113.65 | 113.69 | 113.73 | 113.77 | 113.81 | 113.85 | 113.89 | 113.93 | 113.97 | 114.01 | 114.05 | 114.09 | 114.13 | 114.17 |
| 114.21 | 114.25 | 114.29 | 114.33 | 114.37 | 114.41 | 114.45 | 114.49 | 114.53 | 114.57 | 114.61 | 114.65 | 114.69 | 114.73 | 114.77 |
| 114.81 | 114.85 | 114.89 | 114.93 | 114.97 | 115.01 | 115.05 | 115.09 | 115.13 | 115.17 | 115.21 | 115.25 | 115.29 | 115.33 | 115.37 |
| 115.41 | 115.45 | 115.49 | 115.53 | 115.57 | 115.61 | 115.65 | 115.69 | 115.73 | 115.77 | 115.81 | 115.85 | 115.89 | 115.93 | 115.97 |
| 116.61 | 116.65 | 116.69 | 116.73 | 116.77 | 116.81 | 116.85 | 116.89 | 116.93 | 116.97 | 117.01 | 117.05 | 117.09 | 117.13 | 117.17 |
| 117.21 | 117.25 | 117.29 | 117.33 | 117.37 | 117.41 | 117.45 | 117.49 | 117.53 | 117.57 | 117.61 | 117.65 | 117.69 | 117.73 | 117.77 |
| 117.81 | 117.85 | 117.89 | 117.93 | 117.97 | 118.01 | 118.05 | 118.09 | 118.13 | 118.17 | 118.21 | 118.25 | 118.29 | 118.33 | 118.37 |
| 118.41 | 118.45 | 118.49 | 118.53 | 118.57 | 118.61 | 118.65 | 118.69 | 118.73 | 118.77 | 118.81 | 118.85 | 118.89 | 118.93 | 118.97 |
| 119.01 | 119.05 | 119.09 | 119.13 | 119.17 | 119.21 | 119.25 | 119.29 | 119.33 | 119.37 | 119.41 | 119.45 | 119.49 | 119.53 | 119.57 |
| 119.61 | 119.65 | 119.69 | 119.73 | 119.77 | 119.81 | 119.85 | 119.89 | 119.93 | 119.97 | 200.01 | 200.05 | 200.09 | 200.13 | 200.17 |

WRLMAP
Page 23 of 28
30 September 1972
| DATA 0301 | 0320 | 0338 | 0346 | 0354 | 0362 | 0370 | 0378 | 0386 | 0394 | 0402 | 0410 | 0418 | 0426 | 0434 | 0442 | 0450 | 0458 | 0466 | 0474 | 0482 | 0490 | 0498 | 0506 | 0514 | 0522 | 0530 | 0538 | 0546 | 0554 | 0562 | 0570 |
|-----------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0.56      | 0.88  | 1.20  | 1.52  | 1.84  | 2.16  | 2.48  | 2.80  | 3.12  | 3.44  | 3.76  | 4.08  | 4.40  | 4.72  | 5.04  | 5.36  | 5.68  | 6.00  | 6.42  | 6.84  | 7.26  | 7.68  | 8.10  | 8.52  | 8.94  | 9.36  | 9.78  | 10.20 | 10.62 |

WRLMAP
Page 24 of 28
30 September 1972

1-108
**Reproducibility of the Original Page is Poor**

<table>
<thead>
<tr>
<th>DATA A.34/</th>
<th>286.33</th>
<th>286.92</th>
<th>285.96</th>
<th>287.15</th>
<th>288.33</th>
</tr>
</thead>
<tbody>
<tr>
<td>287.52</td>
<td>286.92</td>
<td>286.11</td>
<td>286.11</td>
<td>284.76</td>
<td>284.41</td>
</tr>
<tr>
<td>285.37</td>
<td>286.37</td>
<td>286.03</td>
<td>286.29</td>
<td>289.66</td>
<td>285.68</td>
</tr>
<tr>
<td>289.29</td>
<td>286.11</td>
<td>287.58</td>
<td>287.58</td>
<td>289.77</td>
<td>298.65</td>
</tr>
<tr>
<td>301.65</td>
<td>302.17</td>
<td>304.65</td>
<td>303.05</td>
<td>302.98</td>
<td>301.26</td>
</tr>
<tr>
<td>302.25</td>
<td>302.91</td>
<td>303.73</td>
<td>303.87</td>
<td>302.91</td>
<td>302.02</td>
</tr>
<tr>
<td>301.13</td>
<td>301.95</td>
<td>296.63</td>
<td>290.88</td>
<td>300.99</td>
<td>301.55</td>
</tr>
<tr>
<td>300.10</td>
<td>295.65</td>
<td>291.14</td>
<td>291.03</td>
<td>298.32</td>
<td>295.35</td>
</tr>
<tr>
<td>292.70</td>
<td>285.25</td>
<td>286.25</td>
<td>289.65</td>
<td>299.66</td>
<td>296.66</td>
</tr>
<tr>
<td>289.44</td>
<td>290.25</td>
<td>291.65</td>
<td>292.43</td>
<td>293.27</td>
<td>291.07</td>
</tr>
<tr>
<td>295.10</td>
<td>292.45</td>
<td>293.87</td>
<td>294.87</td>
<td>303.63</td>
<td>305.67</td>
</tr>
<tr>
<td>286.54</td>
<td>287.27</td>
<td>287.27</td>
<td>291.27</td>
<td>295.93</td>
<td>297.42</td>
</tr>
<tr>
<td>287.88</td>
<td>291.08</td>
<td>291.08</td>
<td>291.08</td>
<td>291.08</td>
<td>291.08</td>
</tr>
<tr>
<td>270.12</td>
<td>256.51</td>
<td>257.46</td>
<td>303.55</td>
<td>301.68</td>
<td>301.24</td>
</tr>
<tr>
<td>300.17</td>
<td>300.62</td>
<td>296.73</td>
<td>304.32</td>
<td>302.42</td>
<td>301.73</td>
</tr>
</tbody>
</table>

**WRLMAP**

Page 26 of 28
30 September 1972
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR.

-36.73, -36.60, -36.26, -35.65, -34.92, -34.76, -34.76
-35.77, -35.52, -35.32, -34.81, -34.81, -34.76
-45.13, -45.11, -44.62, -44.57, -44.57, -44.57
-45.97, -45.76, -45.51, -45.31, -45.31, -45.31
-46.33, -46.20, -45.83, -45.73, -45.73, -45.73
-46.89, -46.71, -46.59, -46.46, -46.46, -46.46
-52.31, -51.97, -51.65, -51.39, -51.39, -51.39
-53.62, -53.39, -53.11, -52.86, -52.86, -52.86
-56.22, -55.99, -55.70, -55.45, -55.45, -55.45
-61.66, -61.43, -61.21, -60.99, -60.99, -60.99
-62.96, -62.74, -62.52, -62.29, -62.29, -62.29
-65.35, -65.13, -64.91, -64.67, -64.67, -64.67
-67.00, -66.77, -66.54, -66.31, -66.31, -66.31
-67.54, -67.31, -67.08, -66.85, -66.85, -66.85
-67.50, -67.27, -67.04, -66.81, -66.81, -66.81
-70.25, -69.99, -69.74, -69.50, -69.50, -69.50
-76.91, -76.67, -76.43, -76.19, -76.19, -76.19
-76.65, -76.41, -76.17, -75.93, -75.93, -75.93
-91.48, -91.24, -90.99, -90.75, -90.75, -90.75

THE CATA IN THE "C" ARRAY ARE THE NUMBER OF VECTOR SEGMENTS IN EACH
DISTINCT BODY STORED IN ORDER IN THE "A" AND "U" ARRAYS.

DATA C / 637, 746, 576, 454, 130, 279, 46, 35, 122, 83, 55
12, 60, 62, 51, 43, 39, 22, 33, 20, 17, 12
6, 7, 9, 21, 11, 8, 15, 14, 13, 10, 10, 11
10, 7, 6, 5, 4, 5, 4, 5, 10, 8, 6
40, 42, 22, 9, 6, 6, 5, 9, 9, 11, 11, 11, 9
22, 13, 15, 12, 9, 11, 13, 5, 5, 16, 29, 8, 6
6, 7, 6, 5, 11, 7, 4, 6, 6, 29, 23, 14, 6
8, 6, 7, 6, 7, 4, 7, 9

CALL EARTH(N, C, A1, U1)
RETURN
END
1.1.4 WOLF SC4020 PLOT PACKAGE

INTRODUCTION

The WOLF Plot Package is a complete system for producing SC4020 and/or printer plots. The package has been designed to be highly flexible and easy to use. Any plot from a quick simple plot (which requires only one call to the package) to highly sophisticated plots (including motion picture plots) can be easily generated with only a basic knowledge of FORTRAN being necessary.

The SC4020 (Stromberg Carlson 4020) is a cathode ray plotter whose outstanding feature is its plotting speed. As such, any user who is producing series of plots should use this plotter. Film (35 mm and 16 mm) and hardcopy are available and the WOLF Plot Package also allows for printer plots which can be used as a quick look for the SC4020 output.

A typewriter mode is available which conveniently allows plotting of character information on the SC4020. This is especially useful as a printer substitute for large amounts of output.

The routines in the Plot Package are all in G and H level FORTRAN with the exception of TIMING which is in IBM 360 Assembly Language. These routines were designed to be efficient on the IBM 360 series machines; no attempt whatever has been made to pursue the myth of compatibility.

SYSTEM REQUIREMENTS

The system requirements for this package are:
- An IBM 360 which supports G or H level FORTRAN. The 360 Assembler must also be available.

- This IBM 360 must use O.S. (360 Operating System)

- The Plot Package requires 45K bytes of core storage.

- An IBM 2400 series 7 track tape drive must be available for the SC4020 Plotter Driver Tape.

In addition, the WOLF Plot Package requires the FORTRAN library routines ALOG10, SIN, and COS.
The WOLF Plot Package is a system of FORTRAN callable subroutines which are used to create plots. It is structured into four major levels as follows:

1. **Basic Level** - The basic level routines perform the primary functions of the plot package. Except for a few auxiliary routines, the basic level routines are necessary for all other routines. However, few of the basic routines are user called.

   The primary basic routine assembles the instructions for the SC4020 tape. There is a printer simulation (of the SC4020) in this routine. This allows for SC4020 plots, printer plot or both simultaneously. The other major basic level routine is used for initialization and termination of the Plot Package.

2. **Intermediate Level** - The intermediate level contains the major user called routine. Some of the functions of this level are:

   a. Grid Overlays (both Cartesian and Polar) with labels

   b. Scaling functions

   c. Plotting of vectors or characters in any of the following coordinate systems:

      Linear
      Semi-Log
      Log-Log
      Polar
3. **High Level** - This level is for quick plots with a minimum of programming effort. At this level, all of the other levels are called upon. Only one FORTRAN statement is necessary to produce a plot of any array of data complete with a labeled grid overlay.

4. **Independent Level** - These routines perform functions that are independent of all other levels except the basic level. The following are among the functions of this level:

   a. **Labels**: A string of characters can be plotted horizontally, vertically or diagonally (at any inclination and direction).

   b. **Graphic Letters**: Letters can be output in any size and in any font design (i.e., standard block letters, mathematical symbols or even old English script).

   c. **Typewriter Mode**: The typewriter function in the SC4020 plotter can be used by calling the various typewriter routines. These allow for information to be typed (strings of characters output in page format) on either the SC4020 or printer.

In addition to these four levels, there are also a number of auxiliary routines. These perform such functions as conversion of decimal (binary) numbers to EBCDIC equivalents and dump of the SC4020 plot tape.

The functional structure of the Plot Package is illustrated in Figure 1.
The flowchart of the subroutine structure is presented below. The entry points associated with each subroutine are presented with their respective control section names ("subroutine names").

It should be noted that the flow chart is divided according to the four major levels of the Plot Package:

- **Basic Level** - The basic level routines perform the primary functions of the Plot Package.

- **Intermediate Level** - The intermediate level contains the major user-called routines.

- **High Level** - This level is for quick plots with a minimum of programming effort. At this level, all of the other levels are invoked.

- **Independent Level** - These routines perform functions which are independent of all levels other than the basic level.
SUMMARY OF SUBROUTINE ENTRIES
IN THE WOLF PLOT PACKAGE

BLKLET  Draws any set of characters on the SC4020 to any size.

CONDNS  For one page printer plots.

COORD   Recovers the raster coordinates of a point.

CSET    Initializes the character font in BLKLET.

DATE    Returns the current date (in alphanumeric).

DIAGLN  Generate a diagonal label.

EDIT    Converts and edits binary numbers to EBCDIC.

EMPTY   Terminates the ploter tape output.

ENDPLT  Terminates the Plot Package.

FORMAT  Generates a format code for use with EDIT.

FRAMES  Returns frame count.

FRMADV  Advances the frame.

GRDNUM  Computes arguments for GRID or OGRID.
HORLIN Generate horizontal label.

IDFRME Generates the identification frame for the Plot Package.

INTENS Sets the intensity for PLOT or PPLOT.

MAXMIN Finds maximum and minimum of an array.

MINT Truncates to the next algebraically smaller number.

NEGLOG Enables plotting of negative arguments logarithimically.

NOW Obtains the current date and time from the system.

NWUNIT Sets the output unit numbers.

OGRID Computes the necessary scaling for PLOT; plots and labels an open grid.

PLOT Plots a set of points or a series of contiguous vectors.

PLOTST Initializes the Plot Package.

POLAR Computes the necessary scaling for PPLOT or PLOT; draws and labels a polar grid.
PLOT  Plots a set of points or a series of contiguous vectors in polar coordinates.

PTYNUM  Computes esthetic plotting limits on data.

QUICKY  Plots X-Y values on an appropriate grid.

SCALE  Computes the scaling for PLOT.

SCHAR  Function value is EBDIC character value corresponding to the input SC4020 character value.

SC4020  Translates plot commands into SC4020 instructions and/or printer plots.

SETGRD  Sets the raster grid limits.

SETPAG  Sets the line count and starting column for TYPLIN.

TYPLIN  Type a line of information on the SC4020.

UCS  Calls CSET with a standard character font.

VBAR  Set use of vertical bar "|" instead of "I" for vertical lines on printer plots.

VERLIN  Generate vertical label.
The cross reference chart for the WOLF Plot Package is given below. The called routines are listed across the top; the calling routines are listed down the left side. It should be noted that this chart is by subroutine and function entry rather than by control section name. The appropriate control sections have been designated in the flowchart of Subroutine Structure given previously.
## CALLED ROUTINES

<table>
<thead>
<tr>
<th>CALLED ROUTINES</th>
<th>CSET</th>
<th>EDIT</th>
<th>EMPTY</th>
<th>GRIDNUM</th>
<th>GRID</th>
<th>HORLIN</th>
<th>MINT</th>
<th>PLOT</th>
<th>SC4020</th>
<th>SCHAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLKLET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAGLN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENDPLT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRMADV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDFRME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OGRID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOTST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POLAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPLOT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUICKY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC4020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPLIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERLIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMMON BLOCK</td>
<td>ROUTINES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPLOTS</td>
<td>PLOTST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GRID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HORLIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC4020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DESCRIPTION

BLKLET is primarily a user routine which is used to produce block letters (letters that are drawn as vectors instead of being printed).

Before BLKLET can be used, a character set (which consists of the character description arrays) must be input through the entry CSET.

BLKLET first determines the position of the first character and then for each character, using the character description arrays, determines the vectors that make up each character and calls SC4020 to plot these characters.
NAME       BLKLET
ENTRY POINT PURPOSE
BLKLET       TO DRAW ANY SET OF CHARACTERS ON THE SC4020 TO ANY SIZE
CSET       TO INITIALIZE A CHARACTER FONT IN BLKLET

CALLING SEQUENCE CALL BLKLET(CHAR,N,IXX,IYY,IDLTX,IDLTY,JSIZE)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>L*1</td>
<td>INPUT - CHARACTER STRING TO BE PLOTTED</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>INPUT - NUMBER OF CHARACTERS</td>
</tr>
<tr>
<td>IXX</td>
<td>I</td>
<td>INPUT - X RASTER COUNT OF CENTER OF LINE</td>
</tr>
<tr>
<td>IYY</td>
<td>I</td>
<td>INPUT - Y RASTER COUNT OF CENTER OF LINE</td>
</tr>
<tr>
<td>IDLTX</td>
<td>I</td>
<td>INPUT - X INCREMENT BETWEEN CHARACTERS</td>
</tr>
<tr>
<td>IDLTY</td>
<td>I</td>
<td>INPUT - Y INCREMENT BETWEEN CHARACTERS</td>
</tr>
<tr>
<td>JSIZE</td>
<td>I</td>
<td>INPUT - THE ABSOLUTE VALUE OF JSIZE IS THE SIZE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FACTOR: =1 NORMAL PRINTED SIZE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*GE1 NORMAL UPRIGHT CHARACTERS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*LT1 CHARACTERS ROTATED 90 DEGREES COUNTER CLOCKWISE</td>
</tr>
</tbody>
</table>

CALLING SEQUENCE CALL CSET(NC,ICHAR,IPOS,IVEC)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>I</td>
<td>INPUT - NUMBER OF CHARACTER IN SET</td>
</tr>
<tr>
<td>ICHAR</td>
<td>L</td>
<td>INPUT - CHARACTERS</td>
</tr>
<tr>
<td>IPOS</td>
<td>I</td>
<td>INPUT - ARRAY OF POINTERS TO IVEC</td>
</tr>
<tr>
<td>IVEC</td>
<td>I</td>
<td>INPUT - ARRAY OF CHARACTERS DESCRIPTION</td>
</tr>
</tbody>
</table>

SUBROUTINE USED SC4020
COMMON BLOCKS NONE
INPUT FILES NONE
OUTPUT FILES NONE
RESTRICTIONS MASTER CHARACTER SET MUST BE INPUT THROUGH ONE OF THE FOLLOWING METHODS:

1. CALL UCS WILL INPUT A STANDARD FONT
2. SPECIAL CHARACTER FONTS MAY BE INPUT VIA CSET

IPOS AND IVEC CANNOT BE CHANGED AFTER THE CALL TO CSET BECAUSE THEIR LOCATION AND NOT THEIR VALUE ARE SAVED.

REFERENCES
NONE

SUBROUTINE BLKLET(CHAR,N,IXX,IYY,IDLTX,IDLTY,JSIZE)

LOGICAL*1 CHAR(1),ICHAR(1),LLI(4)
EQUIVALENCE(LL,LLI(1))
INTEGER IX(4),SHIFT(4)/Z1000,Z0100,Z0010,Z0001/
INTEGER IPOS(1),IVEC(1)

C COMPUTE CENTER OF FIRST CHARACTER
ISIZE=IABS(JSIZE)
IX1=IXX-(N-1)*IDLTX-6*ISIZE/2
IY3=IYY-(N-1)*IDLTY-9*ISIZE/2

C LOOP ON ALL CHARACTERS
DO 50 I=1,N
LLl(4)=CHAR(I)
IC=LL
CONTINUE
GO TO 45

C FIND CHARACTER
DO 10 J=1,NC
LLl(4)=ICHAR(J)
IF (IC.EQ*LL) GO TO 20
10 CONTINUE
GO TO 45

C DRAW CHARACTER
C NOTE THAT IVEC IS PACKED X,Y,DX-6,CY

20 IS=IPCS(J)
IE=IPCS(J+1)-1
DO 40 K=IS,IE
DO 30 L=1,4
30 IX(L)=MOD(IVEC(K)/SHIFT(L),SHIFT(3))+ISIZE
IX(3)=IX(3)-6*ISIZE
IF(JSIZE.GT.0) GO TO 35
DO 32 L=1,3,2
IT=IX(L)
IT=IT-6
IX(L)=IX(L)+IT
32 IX(L+1)=IT
35 IX(1)=IX(1)+IXB
IX(2)=IX(2)+1YB
40 CALL SC4020 (II,IX(1),IX(2),IX(3),IX(4))
45 IXB=1X8+IDLTX
IYB=1Y8+IDLTY
CONTINUE

C CSET ENTRY
ENTRY CSET (NC,CHAR,IPOS,IVEC)
C INITIALIZE CHARACTER FONT
RETURN
END
DESCRIPTION

DATE produces an array describing the current date in alphanumerics. This routine is used by PLOTST to produce the ID frames, but can also be used by the user.

DATA calls NOW in order to determine the current date in the integer form YYDDD where YY is the year and DDD is the number of the day in the year. The year is determined by division by 1000 and the day is the modulo 1000. Then using the array IDAYS (which give the relationship of days versus month), the month and day of the month are determined (leap years are taken into consideration).

Finally the year, month and day of the month are put into character coding of the form year/month/day.
NAME DATE
PURPOSE TO RETURN THE CURRENT DATE (IN ALPHANUMERIC)
CALLING SEQUENCE CALL DATE(CHAR)
SYMBOL TYPE DESCRIPTION
CHAR L#1 OUTPUT - CURRENT DATE IN THE FORM MM/DD/YY (MONTH, DAY, YEAR)

SUBROUTINE DATE (CHAR)

DIMENSION CHAR(B), ICAYS(12), IDATE(3)
LOGICAL CHAR, LCHAR1, LCHAR2, LSLASH, DUM(2)
INTEGER ICHAR

EQUIVALENCE (ICHAR, CUM(1), LCHAR1), (DUM(2), LCHAR2), (IDATE(2), IDATE)

C EBCDIC

DATA (Z, LSLASH, ISHIFT /ZF0,1H/; 236 /
C ELAPSED DAYS OF YEAR FOR EACH MONTH
DATA IDAYS /31,59,90,120,151,181,212,243,273,304,334,365 /
C RECOVER DAY OF YEAR IN IBM FORMAT (YYDDD)
CALL NOW (YYDDD, DUMMY)
C GET YEAR, MONTH, DAY
IDATE(3) = YYYY/1000
ID = MOD(YYYY, 1000)
ICOR = 0
IF (MOD(IDATE(3), 4) .EQ. 0 .AND. ID .GT. 31) ICOR = 1
DO 10 I = 1, 12
10 CONTINUE

I = ID - ICOR
IDAYS(I) = 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365

C CONVERT DATE TO ALPHANUMERIC - REMEMBER EQUIVALENCE
J = I
DO 60 J = 1, 13
ICHAR = (IDATE(I) / 10 + 12) * ISHIFT(MOD(IDATE(I), 10) + 12)
CHAR(J) = ICHAR
CHAR(J+1) = LCHAR2

1-130
DESCRIPTION

EDIT is used to convert any single precision number (integer or floating point) to an equivalent character array. EDIT is used for labeling values. Such routines as QUICKY and GRID use EDIT although the user often has the need to call EDIT.

EDIT first determines the characteristics of the format. The type (E, F or I), the width, the decimal point position and the power factor must be determined.

Then by using divisions by 10 to determine the least significant digits and modulo 10 to determine the remaining higher order digits, the number is converted to character codes.
NAME  
EDIT  

PURPOSE  
TO CONVERT AND EDIT BINARY NUMBERS TO EEDCIC  

CALLING SEQUENCE  
CALL EDIT(A*FCRi ,OUT,N)  

SYMBOL  TYPE  DESCRIPTION  
A  R  INPUT - NUMBER TO BE CONVERTED  
FCRM  A*I  INPUT - EDITING FORMAT (K IS THE NUMBER OF BYTES IN THE FORMAT)  
OUT  A*I  OUTPUT - EDITED EEDCIC NUMBER (W IS THE REQUESTED FIELD WIDTH)  
N  I  OUTPUT - NUMBER OF PRINTABLE CHARACTERS  

SUBROUTINES USED  NONE  
COMMON BLOCKS  NONE  
INPUT FILES  NONE  
OUTPUT FILES  NONE  
RESTRICTIONS  NONE  
REFERENCES  NONE  

SUBROUTINE EDIT (A,FCRi,OUT,N)  
LOGICAL SETDIG,LDIG,LMODE,OUT,FORM,NEG,ZSW  
INTEGER SPECS,W,DEE,PER,BLANK,PLUS,P,Z  
DIMENSION SPECS(3),SETDIG(36),OUT(I),FORM(I)  
EQUIVALENCE (SPECS(1),W),(SPECS(2),D),(SPECS(3),P),(M,BI)  
EQUIVALENCE (SETDIG(1),PLUS),(SETDIG(5),MINUS),(SETDIG(9),Z)  
EQUIVALENCE (SETDIG(13),PER),(SETDIG(17),),,(SETDIG(21),EE)  
EQUIVALENCE (SETDIG(25),BLANK),(SETDIG(29),IDIG),(SETDIG(32),LDIG)  
EQUIVALENCE (SETDIG(33),MODE),(SETDIG(36),LMODE)  
DATA SETDIG /  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HO,  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HE,  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HE,  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HE,  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HE,  
ZOO,ZOC,ZOO,1H+,ZCO,ZOO,ZCO,1H-,ZOO,ZCO,ZOO,1HE,  
C CLEAR FORMAT SCAN BUFFER  
DO 5 1=1,3  
5 SPECS(11)=0  
C SCAN FORMAT  
J=1  
LDIG=FCRi(1)  
MODE=IDIG  

1-133
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

EDIT 56
EDIT 57
EDIT 58
EDIT 59
EDIT 60
EDIT 61
EDIT 62
EDIT 63
EDIT 64
EDIT 65
EDIT 66
EDIT 67
EDIT 68
EDIT 69
EDIT 70
EDIT 71
EDIT 72
EDIT 73
EDIT 74
EDIT 75
EDIT 76
EDIT 77
EDIT 78
EDIT 79
EDIT 80
EDIT 81
EDIT 82
EDIT 83
EDIT 84
EDIT 85
EDIT 86
EDIT 87
EDIT 88
EDIT 89
EDIT 90
EDIT 91
EDIT 92
EDIT 93
EDIT 94
EDIT 95
EDIT 96
EDIT 97
EDIT 98
EDIT 99
EDIT 100
EDIT 101
EDIT 102
EDIT 103
EDIT 104
EDIT 105
EDIT 106
EDIT 107
EDIT 108
EDIT 109
EDIT 110
EDIT 111

NSIGN=1
I=1
I=I+1
LOIG=FLRM(1)
IF (IDIG.LT.20.OR.ICIG.GT.22+9) GO TO 10
SPECS(J)=SPECS(J)*10+IDIG-ZZ
GO TO 7
10 IF (ICIG.NE.PER) GO TO 15
SPECS(J)=SPECS(J)*NSIGN
NSIGN=1
J=J+1
GO TO 7
15 IF (ICIG.EQ.PLUS) GO TO 7
IF (IDIG.NE.MINUS) GO TO 10
NSIGN=-1
GO TO 7
18 N=W
C GET ABSOLUTE VALUE OF NUMBER IN FLOATING POINT
NEG=A.LT.0.
B=ABS(A)
B1=A
W=ABS(N)
IF (W.LT.15728641) B=W
IF(MODE.NE.11) GO TO 20
C INTEGER PROCESSING
D=-1
B=I+1
GO TO 35
20 IF (MODE.NE.EE) GO TO 30
C E FORMAT PROCESSING
D=MIND(D,N-4)
W=MAXO(D+1,N-4)
IF (W.LT.0.) IPOW=-INT(-.ALCG10(B))-P
B=B+10**(-IPOW)+.5*10.0**(-D)
IF (B.LT.10.0**P) GO TO 35
B=B/10.0
IPOW=IFOW+1
GO TO 35
C ROUND E AND F FORMAT NUMBERS
30 B=B+5.510.0**(-D)
C EDIT WORD INTL output AREA
35 IPER=W-D
POW=10.**IPER-2
I1=1
IWM=0
ZSW=*F*LSE.
I=0
38 I=I+1
IF(I.NE.IPER) GO TO 40
IDIG=PER
GO TO 60
40 IF(ZW.OR.IDIG.NE.ZZ.OR.I.EC.IPER-1.OR.I.EQ.V) GO TO 60
IWM=I1
IDIG=ELank

1-134
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

```
GO TO 70
25W=TRUE
70 IRTN=0
GO TO 200
80 IF(1*LT.10) GO TO 38
IF(WUP.EQ.EE) GO TO 130
IF(W*E1*N-4) GO TO 50
DIG=EE
GO TO 200
90 IRTN=IRTN+1
100 DIG=PLUS
IF(IPCX*LT.0) DIG=MINUS
IPOW=ABS(IPOW)
GO TO 200
110 DIG=IFOW/10+ZZ
GO TO 200
120 DIG=MOD(IPOW,10)+ZZ
GO TO 200
130 IF(WM*EO.0 OR . NOT. NEG) RETURN
DIG=MINUS
W=W
IRTN=5
200 OUT(W)=LDIG
W=W+1
IRTN=IRTN+1
GO TO (80,160,110,120,130,140),IRTN
140 RETURN
END
```
ENTRY

SCAN FORMAT

GET ABSOLUTE VALUE OF NUMBER IN FLOATING POINT

INTEGER OUTPUT

INTEGER PROCESSING

20

E FORMAT OUTPUT

YES

E FORMAT PROCESSING

30

ROUND E AND F FORMAT NUMBERS

35

EDIT WORD INTO OUTPUT AREA

RETURN

20

NO
DESCRIPTION

GRDNUM is a routine which computes certain characteristics of an array of data values (usually either x values or y values). These characters are used in calls to GRID or OGRID. Each characteristic can also be had separately by a separate entry and calls to GRDNUM go through each of these entries.

MAXMIN determines FMIN and FMAX (the minimum and maximum values of the array, respectively).

PTYNUM, given FMIN and FMAX, determine rounded values PMIN and PMAX of FMIN and FMAX, resp., such that [FMIN, FMAX] lies in [PMIN, PMAX] and PMIN and PMAX are esthetically nice boundaries. NINT, the suggested number of intervals in [PMIN, PMAX] is also determined.

Finally, FORMAT determines a good format for numbers in [PMIN, PMAX]. An F format is usually produced unless the values are either too large or too small in which case E9.2.1 is used.
GRDNUM

NAME

GRIDNUM

ENTRY POINT
PURPOSE

GRIDNUM
TO COMPUTE ARGUMENTS TO GRID AND OGRID

MAXMIN
TO FIND ARRAY MAXIMUM AND MINIMUM VALUES

PTYNUM
TO COMPUTE ESTHETIC PLOTTING LIMITS ON DATA

FORMAT
TO GENERATE A FORMAT CODE TO LABEL NUMBERS WHOSE
VALUES LIE BETWEEN PHIN AND PHAX (FOR USE WITH
"EDIT")

CALLING SEQUENCE
CALL GRIDNUM(ARRAY,N,PHIN,PHAX,NINT,FNT)

SYMBOL TYPE DESCRIPTION

ARRAY R INPUT - PLOTTING ARRAY

N I INPUT - NUMBER OF ITEMS IN ARRAY

PHIN R OUTPUT - SUGGESTED PLOTTING MINIMA

PHAX R OUTPUT - SUGGESTED PLOTTING MAXIMA

NINT I OUTPUT - SUGGESTED NUMBER OF INTERVALS

FNT A OUTPUT - SUGGESTED LABELING FORMAT

CALLING SEQUENCE
CALL MAXMIN(ARRAY,N,PHIN,PHAX)

SYMBOL TYPE DESCRIPTION

ARRAY R INPUT - THE ARRAY

N I INPUT - NUMBER OF ITEMS IN THE ARRAY

PHIN R OUTPUT - ARRAY MINIMA

PHAX R OUTPUT - ARRAY MAXIMA

CALLING SEQUENCE
CALL PTYNUM(FHIN,FMAX,PHIN,PHAX,NINT)

SYMBOL TYPE DESCRIPTION

FHIN R INPUT - ARRAY MINIMA

FMAX R INPUT - ARRAY MAXIMA

PHIN R OUTPUT - SUGGESTED PLOTTING MINIMA

PHAX R OUTPUT - SUGGESTED PLOTTING MAXIMA

NINT I OUTPUT - SUGGESTED NUMBER OF INTERVALS
### Calling Sequence

**CALLING SEQUENCE**

**CALL FORTAM(PMIN, PMAX, FMT)**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIN</td>
<td>R</td>
<td>INPUT - SUGGESTED PLOTTING MINIMA</td>
</tr>
<tr>
<td>PMAX</td>
<td>R</td>
<td>INPUT - SUGGESTED PLOTTING MAXIMA</td>
</tr>
<tr>
<td>FMT</td>
<td>A</td>
<td>OUTPUT - SUGGESTED LABELING FORMAT</td>
</tr>
</tbody>
</table>

### Subroutines Used

**NONE**

### Common Blocks

**NONE**

### Input Files

**NONE**

### Output Files

**NONE**

### Restrictions

**NONE**

### References

**NONE**

---

**SUBROUTINE** GRDNUM(ARRAY, N, PMIN, PMAX, NINT, FMT)  
**DIMENSION** ANKNEM(1)  
**LOGICAL** NUMS(10), EFMT(7)  
* FMT(1), PAREN, POINT, GROSW, TRUE, ANI, I  
**EQUIVALENCE** (PAREN, EFMT(7)), (PEINT, EFMT(5))  
**GROSW** = .FALSE.

**C MAXMIN ENTRY**

**ENTRY** MAXMIN(ARRAY, N, PMIN, PMAX)

**C FIND ARRAY MAXIMUM AND MINIMUM**

**FMIN = ARRAY(I)**  
**FMAX = FMIN**  
**IF(N.LT.2) GO TO 20**  
**DO 10 I = 2, N**  
**IF(ARRAY(I).LT.FMIN) FMIN = ARRAY(I)**  
**10 IF(ARRAY(I).GT.FMAX) FMAX = ARRAY(I)**  
**IF(GROSW) RETURN**

**C PTYNUM ENTRY**

**ENTRY** PTYNUM(FMIN, FMAX, PMIN, PMAX, NINT)  
**NINT = 0**  
**IF(FMIN.EQ.FMAX) GO TO 50**

**C COMPUTE ESTHETIC PLOTTING LIMITS**

**DMAG = 10**  
**NL = INT(-ALOG10(ABS(FMAX-FMIN)))-1**  
**NL = INT(FMIN/LMAG)**  
**PHIN = FLOAT(NL)*DMAG**  
**NH = INT(-FMAX/LMAG)**  
**PHAX = FLOAT(NH)*DMAG**  
**NINT = NH-4L**  
**MIN = 11/FINT*INT**  
**IF(FMIN=0.3) NINT=16**  
**IF(GROSW) RETURN**
C FORMAT ENTRY
ENTRY FORMAT(FLN,PHAX,FMT)
C GENERATE FORMAT TO BE USED WITH FLN AND PHAX
MAG=INT(1+ALOG10(MAX(ABS(PHAX),ABS(FLN))))
IF(MAG.GT.1)MAG=1
MAG=MIN(ALOG10(ABS(FMAX-FLN)))+1
IF(MAG.GT.10)GO TO 25
IF(MAG.GT.17)GO TO 30
FMT(1)=AN1
FMT(2)=NUMS(MAG+2)
FMT(3)=PCRNEN
GO TO 50
25 I= MAG+IAD+2
IF(1W.GT.C)GO TO 30
FMT(1)=F
FMT(2)=NUMS(IW+1)
FMT(3)=PCRNEN
FMT(4)=NUMS(MAGD+1)
FMT(5)=PCRNEN
GO TO 50
30 GO 40 I=1,7
40 FMT(I)=E:FMT(I)
50 GROSW=.TRUE.
RETURN
END
DESCRIPTION

GRID is the major user routine in the plot package. It is used to set up the scaling factors between subject space and object space, draw grid overlays, to plot vectors.

The object space (the space in the raster world on the SC4020 screen) is defined by SETGRD. If SETGRD is not called, a default object space is used.

The subject space (the space in which the user units exist) is defined by GRID, OGRID, POLAR or SCALE. SCALE defines this space and also sets up the scaling factors and initializations for the routine GRID. GRID and OGRID use the same coding as SCALE (and therefore perform the same function) but also plot a grid overlay. GRID plots a full grid while OGRID plots a partial grid with tick marks. POLAR is essentially the same as GRID except it is used for polar plots.
PLOT is used to plot vectors or characters. Given an array of coordinates in subject space and using the scaling factors from SCALE, PLOT determines corresponding coordinates in the object space and then either prints characters at the points or draws vectors connecting these points. PPLOT is the same as PLOT except it is used for polar data.

COORD is used to recovery the raster coordinate corresponding to any coordinate in the subject space. NEGLOG specifies the error procedure for negative logs and INTENS is used to set character intensity.
NAME

GRID

ENTRY POINT

PURPOSE

GRID

TO COMPUTE THE NECESSARY SCALING FOR SUBROUTINE
*PLOT* AND TO PLOT AND LABEL GRID LINES

OGRID

TO COMPUTE THE NECESSARY SCALING FOR SUBROUTINE
*PLOT* AND LABEL AN OPEN GRID

POLAR

TO COMPUTE THE NECESSARY SCALING FOR 'PLOT' & 'PPLT'
AND DRAW AND LABEL A POLAR GRID

SCALE

TO RECOMPUTE THE SCALING FOR SUBROUTINE 'PLOT'

SETGRID

TO SET THE LIMITS FOR THE GRID

INTENS

TO SET THE INTENSITY FOR SUBROUTINE 'PLOT'

NEGLOG

TO ENABLE THE PLOTTING OF NEGATIVE ARGUMENTS
LOGARITHMICALLY IN 'PLOT' WITH EITHER A DIFFERENT
SYMBOL OR SUPERPOSITION OF SYMBOLS

PLOT

TO PLOT A SET OF POINTS OR A SERIES OF CONTIGUOUS
VECTORS

PPLT

TO PLOT A SET OF POINTS OR A SERIES OF CONTIGUOUS
VECTORS IN POLAR COORDINATES

CCORD

TO RECOVER THE RASTER COORDINATES OF A POINT

CALLING SEQUENCE

CALL GRID(XLO,XHI,NX,A,NXS,YLO,YHI,NY,B,NYS,LOG)

SYMBOL  TYPE  DESCRIPTION

XLO   R   INPUT - LOWEST VALUE OF ABSCISSA (LOWEST VALUE OF X) AT LEFT SIDE OF GRID

XHI   R   INPUT - HIGHEST VALUE OF ABSCISSA (HIGHEST VALUE OF X) AT RIGHT SIDE OF GRID

NX    I   INPUT - NUMBER OF INTERVALS ON ABSCISSA.

A     A   INPUT - LABELING FORMAT FOR ABSCISSA

NXS   I   INPUT - NUMBER OF INTERVALS AT WHICH TO LABEL THE X-AXIS

YLO   R   INPUT - VALUE OF ORDINATE (LOWEST VALUE OF Y) AT BOTTOM OF GRID

YHI   R   INPUT - VALUE OF ORDINATE (HIGHEST VALUE OF Y) AT TOP OF GRID

NY    I   INPUT - NUMBER OF INTERVALS ON ORDINATE
B A INPUT - LABELING FORMAT FOR Y-AXIS
NYS I INPUT - NUMBER OF INTERVALS AT WHICH TO LABEL THE Y-AXIS
LOG I INPUT - I=0 LINEAR GRID
I=1 ABSCISSA IS LOGARITHMIC ORDI NATE IS LINEAR
I=2 ABSCISSA IS LINEAR ORDI NATE IS LOGARITHMIC
I=3 LOGARITHMIC GRID

CALLING SEQUENCE CALL GRID(XLO,XHI,NX,A,NXS,YLO,YHI,NY,B,NYS,LOG)

SYMBOL TYPE DESCRIPTION
(SEE CALLING SEQUENCE DESCRIPTION FOR ENTRY POINT GRID)

CALLING SEQUENCE CALL POLAR(RADIUS,NX,A,NXS,IRDH)

SYMBOL TYPE DESCRIPTION
RADIUS R INPUT - VALUE OF OUTER CIRCLE
NX I INPUT - NUMBER OF CONCENTRIC CIRCLES
A A INPUT - LABELING FORMAT FOR CONCENTRIC CIRCLES
NXS I INPUT - NUMBER OF CIRCLES AT WHICH TO LABEL
IRDH I INPUT - INDICATOR FOR LABELING RADIALS:
=1 LABEL RADIALS ARE IN DEGREES
=2 LABEL RADIALS ARE IN HOURS

CALLING SEQUENCE CALL SCALE(XLO,XHI,YLO,YHI,LOG)

SYMBOL TYPE DESCRIPTION
(SEE CALLING SEQUENCE DESCRIPTION FOR ENTRY POINT GRID)

CALLING SEQUENCE CALL SETGRID(XLOLIM,YLOLIM,XHILIM,YHILIM)

SYMBOL TYPE DESCRIPTION
XLOLIM R INPUT - LOWEST ABSCISSA POSITION (IN RASTERS)
YLOLIM R INPUT - LOWEST ORDI NATE POSITION (IN RASTERS)
XHILIM R INPUT - HIGHEST ABSCISSA POSITION (IN RASTERS)
YHILIM R INPUT - HIGHEST ORDI NATE POSITION (IN RASTERS)

CALLING SEQUENCE CALL INTENS(IT)

SYMBOL TYPE DESCRIPTION
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALLING SEQUENCE CALL PLOT(X,Y,N,CHAR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYMBOL</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>X</td>
<td>R</td>
<td>INPUT - ARRAY OF 'X' VALUES TO BE PLOTTED</td>
</tr>
<tr>
<td>Y</td>
<td>R</td>
<td>INPUT - ARRAY OF 'Y' VALUES TO BE PLOTTED</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>INPUT - NUMBERED OF ORDERED PAIRS TO BE PLOTTED</td>
</tr>
<tr>
<td>CHAR</td>
<td>I</td>
<td>INPUT - PLOT CHARACTER, RIGHT ADJUSTED</td>
</tr>
</tbody>
</table>

| CALLING SEQUENCE CALL PPLOT(X,Y,N,CHAR,IRDH) |
| SYMBOL | TYPE | DESCRIPTION |
| X      | R    | INPUT - ARRAY OF RADIUS VALUES TO BE PLOTTED |
| Y      | R    | INPUT - ARRAY OF ANGULAR VALUES TO BE PLOTTED |
| N      | R    | INPUT - NUMBER OF ORDERED PAIRS TO BE PLOTTED |
| CHAR   | I    | INPUT - PLOT CHARACTER, RIGHT ADJUSTED |
| IRDH   | I    | INPUT - INDICATOR OF UNITS OF ANGULAR INPUT |
|        |      | 0 ANGLES ARE IN RADIANS |
|        |      | 1 ANGLES ARE IN DEGREES |
|        |      | 2 ANGLES ARE IN HOURS |

| CALLING SEQUENCE CALL COORD(X,Y,KX,KY) |
| SYMBOL | TYPE | DESCRIPTION |
| X      | R    | INPUT - X VALUE |
| Y      | R    | INPUT - Y VALUE |
| KK     | I    | OUTPUT - X RASTER VALUE |
| KY     | I    | OUTPUT - Y RASTER VALUE |
SUBLTUTIONS USED

EDIT         HORLIN          SCAPOO

COMMON BLOCK
CPLOTS

INPUT FILES          NONE

OUTPUT FILES         NONE

RESTRICTIONS
XLO.NE.XHI AND YLO.NE.YHI

IF USED, SETGRID MUST BE CALLED BEFORE CORRESPONDING CALLS TO GRID, SCALE, OR OGRID

REFERENCES

SUBROUTINE GRID (XLO,XHI,NX,A,NXS,YLO,YHI, NY,B,NYS,LOG)

COMMON /CPLOTS/ G1(2),XLOG(2),XULIM(2),XHILIM(2),XSCAL(2),
     XLOG(2),TY,6Z(4)

LOGICAL LOGX,LOGY,OPEN,SCALX,XLOG,OVERPL,OVRPLT,NEG,NEGSW
ANGLE
LOGICAL1 CHAR(4), ARRAY(4), NCHAR(4), LCHAR

INTEGER XV,YV,BLANK

REAL SCALS,OPEN

DIMENSION X(1),Y(1)

DATA FLANK/CHAR/ /','4H

DATA CF-.10,9/

DATA L"/0/

DATA SCALS,OPEN,OVERPL,NEGSW,ANGLE/5*,FALSE*/

DATA KXY/2,2/

GO TO 5

C GRID ENTRY
ENTRY GRID (XLJ,XHI,NX,A,NXS,YLO,YHI, NY,B,NYS,LOG)
OPEN=.TRUE.
GO TO 5

C SCALE ENTRY
ENTRY SCALE (XLJ,XHI,YLO,YHI,LOG)
SCALX=1,TRUE.

C INITIALIZE ARRAYS AND ENTRY SWITCHES

5
FXLO(1)=XLJ
FXLC(1)=XLC
FXHI(1)=XHI
FXH1(I)=Y1
I(1)=X
I(2)=Y
NLABEL(I)=NXS
NLABEL(2)=NYS
C LOOP FOR X THEN Y AXIS PROCESSING
DO 50 IA=1,2
XLOG(IA)=LOG.EC*IA
GRID 224
C COMPUTE SCALE FACTORS AND LINE INCREMENTS
IF (XLOG(IA)) GO TO 10
XSCALE(IA)=XHILIM(IA)-XLOLIM(IA)/(FXHI(IA)-FXLO(IA))
IF (SCALE) GO TO 50
J=I(IA)
XINC=(FXHI(IA)-FXLO(IA))/FLOAT(J)
GO TO 15
10 IF (*LTSCALSW) GO TO 12
FXLO(IA)=ALOGIC(FXLO(IA))
FXHI(IA)=ALOGIC(FXHI(IA))
XSCALE(IA)=(XHILIM(IA)-XLOLIM(IA))/(FXHI(IA)-FXLO(IA))
GO TO 10
C COMPUTE LOG SPACING INTERVAL ALONG AXIS
12 NMIN=I(IA)/9
NINC=I(IA)*NMIN
XINC=1/GFLOAT(NINC)
NLABEL(IA)=NLABEL(IA)+NLABEL(IA)/9
I=MINI(ALOGIC(XINC))
NUM=XI/(XINC+10.**I)
XI=1
IF (NINC.GT.NMIN) XI=XI+ALOGIC(FLOAT(NUM)*XINC)
IF (NINC.LE.NMIN) NUM=0
FXLO(IA)=XI
ISTART=NINC+1+NUM
XI=FXHI(IA)
I=MINI(ALOGIC(XINC))
NUM=MIN(-XI/(XINC+10.**I))
XI=1
IF (NINC.GT.NMIN) XI=XI+ALOGIC(FLOAT(NUM)*XINC)
IF (NINC.LE.NMIN) NUM=0
FXHI(IA)=XI
XSCALE(IA)=(XHILIM(IA)-XLOLIM(IA))/(FXHI(IA)-FXLO(IA))
J=NINC+1+NUM-ISTART
15 IB=KNC(I,2)+1
IX(I,J)=XLOLIM(10)
C LOOP FOR EACH GRID LINE ON THIS AXIS
DO 40 J=1Z,J
C COMPUTE RASTER VALUES FOR GRID LINE
IF (XLOG(IA)) GO TO 20
XI=XINC*FLOAT(I)
IX(IA)=XI*XSCALE(IA)+XLOLIM(IA)
XI=XI+FXLO(IA)
K=1
GO TO 20
20 L=ISTART
K=KNC(L,NINC)
IF (K.LT.I) K=K+NINC
GRID 269
XI=(L-K)/NINC  
IF (K.EQ.C) GO TO 25  
IF (K.LT.KFIN) GO TO 40  
XI=XI+LOG10(1+FLOAT(K)*XINC)  
.25  
1X(I)=X(I)+XLOLIM(I)+XSCAL(I)+XLOLIM(I)  
XI=1X(I)+TEN  
C PLCT GRID LINE  
30  
STOP=XHILIM(ID)  
IF (OPEN/NC, NOT.*1.E0,17,OR.1.EQ,1) STOP=XLOLIM(ID)+8  
IV(1)=1STOP-1X(I)  
IF (IA(EIV(I)) GE.C2) GO TO 33  
IV(1)=0  
CALL SC4020 (1,1,1,1,1,1,1,1,1,1,1) GO TO 34  
GO TO 34  
33  
CALL SC402C(0P(1A),1X(1),1X(1),1X(1),1X(1))  
C Label GRID LINE  
34  
IF (NLABEL(IA),EQ.C,CR.,MOD(K,NLABEL(IA)),NE.C) GO TO 40  
IF (IP,NE.1) GO TO 25  
CALL ECIT(XI,OUTN)  
CALL FCRLIN (OUTN,1X(1),1X(2)-16)  
GO TO 40  
35  
CALL ECIT(XI,OUTN)  
CALL FCRLIN (OUTN,1X(1),1X(1)-4+1,1X(2))  
40  
CONTINUE  
50  
CONTINUE  
DPLC=.FALSE.  
ECLAS=.FALSE.  
RETURN  
C PLCT ENTRY  
ENTRY FOLAR(RADIUSA.XA.NX.AIRDH)  
RAD=AERN((XHILIM(1)-XLOLIM(1))/FLOAT(KXY(1)))+  
(XHILIM(2)-XLOLIM(2))/FLOAT(KXY(2)))  
RAD=AERN(RAD,632/SINS)  
RINC=RAD/FLOAT(NX)  
C INITIALIZE RADIUS ARRAYS  
DO 510 K=1,2  
GRAK=1XLOLIM(1)+RAD+FLOAT(KXY(1)-1)  
FXLO(1)=KXITY(1)-1  
XSCAL(1)=RAD/RADIUS  
510  
XLOG(1)=.FALSE.  
CS(1)=1  
CS(2)=0  
C DRAW ARCS IN 5 DEGREE SEGMENTS  
DO 550 K=1,16  
C COMPLETE UNIT CHORC  
DCS(1)=CS(1)*COS-5*CS(2)*SINE-CS(1)  
DCS(2)=CS(2)*COS5*CS(1)*SINE-CS(2)  
C DEFINE RADIAL COMPONENTS  
OFFC=0  
IF (MUT(1,1),NE.1) OFFC=16  
IF (KUL(1,1),NE.1) OFFC=22  
NVEC=(NVEA1(XS(1,1)*CS(2)))**(RAD-OFFC)-1/63  
XYV=(RAD-OFFC)/FLOAT(NVEA+1)  
C LOOP ON QUAD/NTS  
DO 560 K=1,KKI  
DO 560 K=1,KYI
GRID
Page 9 of 14
30 September 1972

\[ K = I A U S (K_1 - K_2)^+ \]
\[ K_T = - ( K_2 )^+ \]
\[ R_X = C S ( K_1 + P H ( K_1 ) ) \]
\[ R_Y = C S ( K_1 + P H ( K_1 ) ) \]
\[ R_X = O C S ( K_1 ) + P H ( K_1 ) \]
\[ R_Y = O C S ( K_1 ) + P H ( K_1 ) \]

C PLOT CONCENTRIC CIRCLES
DO 520 J = 1, NX
RJ = RX + ORG(J)
DO 530 K = 1, KY
IX(1) = RX + ORG(K)
IX(2) = RJ + ORG(K)

520 CALL SC4020 (11, IX(1), IX(2), IV(1), IV(2))

C PLOT RADIALS
DO 530 J = 1, NVEC
RJ = RX + ORG(J)
DO 540 K = 1, KY
IX(1) = RX + ORG(K)

530 CALL SC4020 (11, IX(1), IX(2), IV(1), IV(2))

540 CONTINUE

C INCREMENT ANGLES
CS(1) = CS(1) + DCS(1)
CS(2) = CS(2) + DCS(2)

C LABEL CIRCLES
DO 550 J = 1, NX
FNUM = RX + ORG(J)
DO 560 K = 1, KY
IX(1) = RX + ORG(K)
IX(2) = RX + ORG(K)

550 IF (FNE(1, 3), NE, 1) GO TO 540

560 CALL SC4020 (17, DO, D, D)

570 CALL SC4020 (17, DO, D, D)

C SETGRD ENTRY
ENTRY SETGRD (LOLIMX, LOLIMY, HILIMX, HILIMY)
IF (LOLIMX, LT, HILIMX, LT, HILIMY, LT, HILIMY, LT, HILIMY, LT, HILIMY, LT)

CSETGRD OUT OF RANGE -- LIMITS NOT RESPECTED

570 CALL SC4020 (17, DO, D, D)

RETURN
XHILIM(1)=HILIMX
XHILIM(2)=HILIMY
RETURN
C INTENS ENTRY
        ENTRY INTENS(IT1)
        IT1=IT1
        RETURN
C NEGLOG ENTRY
        ENTRY NEGLOG(OVERPL,NCHAR)
        OVRPL=OVERPL
        LCHAR=LCHAR(4)
        NEG6=1.CHAR.*NE.BLANK
        RETURN
C PLOT ENTRY
        ENTRY PLOT(X,Y,N,CHAR,IRDH)
        ANGLE=.TRUE.
        RAD=1.*RCH(IRDH+1)
C PLOT ALL CHARACTERS - SCALING FROM "GRID" ROUTINE
        ITX=5
        IF (ITX.0) ITX=6
        DO 60 I=1,N
        NEG=.FALSE.
        KK=X(I)
        IF (ANGLE) KK=X(I)*CLS(Y(I))*RAD)
        IF (*.ACT.NEGX) GO TO 54
        IF (XX) 51,54,53
51 IF (*.ACT.NEGSw) GO TO 54
        XX=-XX
        NEG=.TRUE.
        53 XX=ALOG10(XX)
        54 XX=(XX-SXLO)*SCALEX*XLCLIM(1)
        YY=Y(I)
        IF (ANGLE) YY=X(I)*SIN(Y(I))*RAD)
        IF (*.ACT.LG Dy) GO TO 58
        IF (YY) 55,58,57
55 IF (*.ACT.NEGSw) GO TO 58
        YY=-YY
        NEG=.TRUE.
45 YY=ALOG10(YY)
46 KY=(YY-SYLO)*SCALEY*XCLIM(2)
C PLOT INDIVIDUAL POINTS
        IF (*.ACT.NEG0r,OVERPL)
        * CALL SC4020(ITX,KX,KY,LCHAR,IT)
        IF (NEC)
        * CALL SC4020(ITX,KX,KY,NCHAR(4),IT)
60 CONTINUE
44 ANGLE=.FALSE.
        RETURN
C COORD ENTRY
        ENTRY COORD(X,Y,KX,KY)
        N=1
C LOOP ON ALL VECTORS

1-150
DO 90 I=1,N
X1=X2
Y1=Y2
C COMPUTE VECTOR COORDINATES IN RASTERS
C SCALING TAKEN FROM GRID ROUTINE
XX=X(I)
  IF (ANG7E.0) XX=X(I)*COS(Y(I)*RAD)
  IF (ANG7E.180.0) XX=ALOG10(ABS(XX))
  YY=Y(I)
  IF (ANG7E.0) YY=X(I)*SIN(Y(I)*RAD)
  IF (ANG7E.180.0) YY=ALOG10(ABS(YY))
X2=(XX-SXLO)*SCALEX*XLOLIM(1)
Y2=(YY-SYLO)*SCALEY*XLOLIM(2)
  IF (I.LE.1) GO TO 90
RXV=X2-X1
RYV=Y2-Y1
C LOOP TO PLOT VECTOR IN SEGMENTS NOT GREATER THAN 64 RASTERS
J=MAX1(ABS(RXV),ABS(RYV))/63
RXV=RXV/FLOAT(J+1)
RYV=RYV/FLOAT(J+1)
C PLOT VECTOR SEGMENTS
DO 80 K=12,J
  R=K
  KK=X1+R*RXV
  KI=Y1+R*RYV
  CALL SC420 (1),KK,KI,RXV,RYV)
80 CONTINUE
90 CONTINUE
XX=X2
KY=Y2
ANGLE=.FALSE.
RETURN
END
DESCRIPTION

HORLIN and its two entries VERLIN and DIAGLN are used to plot an array of characters. DIAGLN is used to output a label in which the horizontal and vertical spacing or increments between characters is user specified. HORLIN (horizontal label) assumes there will be no vertical increment and a standard horizontal increment. VERLIN (vertical label) assumes there will be no horizontal increment and a standard vertical increment. Each uses the same coding.

Since the center coordinates are input, the coordinates for the first character must be computed. Then each character is output via a call to SC4020 and after each character the coordinates are incremented for the position of the next character.
NAME       HORLIN
ENTRY POINT PURPOSE
HORLIN   TO PRINT HORIZONTAL LABELS ON THE SC4020 PLOTTER
VERLIN   TO PRINT VERTICAL LABELS ON THE SC4020 PLOTTER
DIAGLIN  TO PRINT DIAGONAL LABELS ON THE SC4020 PLOTTER

CALLING SEQUENCE CALL HORLIN(A,N,X,Y)

SYMBOL  TYPE  DESCRIPTION
A       L      INPUT - ALPHANUMERIC INFORMATION TO BE PRINTED
N       I      INPUT - NUMBER OF CHARACTERS TO BE PRINTED
X       I      INPUT - RASTER COUNT OF X-COORDINATE OF CENTER OF LINE (RASTER COUNT OF Y-COORDINATE FOR VERTICAL LABELS)
Y       I      INPUT - RASTER COUNT OF Y-COORDINATE OF CENTER OF LINE ("X" FOR VERTICAL LABELS)

CALLING SEQUENCE CALL VERLIN(A,N,X,Y)

SYMBOL  TYPE  DESCRIPTION
A       L      INPUT - ALPHANUMERIC INFORMATION TO BE PRINTED
N       I      INPUT - NUMBER OF CHARACTERS TO BE PRINTED
X       I      INPUT - RASTER COUNT OF X-COORDINATE OF CENTER OF LINE
Y       I      INPUT - RASTER COUNT OF Y-COORDINATE OF CENTER OF LINE

CALLING SEQUENCE CALL DIAGLIN(A,N,X,Y,DX,DY)

SYMBOL  TYPE  DESCRIPTION
A       L      INPUT - ALPHANUMERIC INFORMATION TO BE PRINTED
A       I      INPUT - NUMBER OF CHARACTERS TO BE PRINTED
X       I      INPUT - RASTER COUNT OF X-COORDINATE OF CENTER OF LINE
Y       I      INPUT - RASTER COUNT OF Y-COORDINATE OF CENTER OF LINE
DX      I      INPUT - RASTER COUNT BETWEEN CHARACTERS IN X DIRECTION.
SUBROUTINE USED SC4020
COMMON BLOCK CPlots
INPUT FILES NONE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES NONE

SUBROUTINE HORLIN (A,N,X,Y)
  EXPLICIT INTEGER (A-Z)
  LOGICAL*1 A(N)
  COMMON /CPlots/ G1(12),INTENS,G2(4)
C SET INCREMENTS
  DX=8
  DY=0
  GO TO 10
C VERTLin ENTRY
    ENTRY VERTLin (A,N,X,Y)
C SET INCREMENTS
  DX=0
  DY=-16
C DIAGLin ENTRY
    ENTRY DIAGLin (A,N,X,Y,DX,DY)
C SET INITIAL COORDINATES AND OP CODE
  10  IX=X-(t-1)<DX/2
      IY=Y-(t-1)<DY/2
      OP=5
      IF (INTENS.TE.0) OP=6
C PLOT EACH CHARACTER
  DO 20 I=1,N
    CALL SC4020 (OP,IX,IIY,A(I),INTENS)
    IX=IX+DX
    IY=IY+DY
  20  RETURN
END
DESCRIPTION

MINT is a function routine which determines the value of the largest integer which is less than or equal to the value of a floating point number, X, which has been input. Notice that -MINT (-X) can be used to find the smallest integer greater than or equal to X.
NAME MINT

PURPOSE TO TRUNCATE TO THE NEXT ALGEBRAICALLY SMALLER INTEGER

CALLING SEQUENCE MINT(X)

SYMBOL TYPE DESCRIPTION
X M INPUT - VALUE TO BE TRUNCATED
MINT I OUTPUT - LARGEST INTEGER LESS THAN X

(-MINT(-X) TRUNCATES TO THE NEXT ALGEBRAICALLY GREATER INTEGER)

SUBROUTINES USED NONE
COMMON BLOCKS NONE
INPUT FILES NONE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES NONE

FUNCTION MINT(X)
MINT=X
IF (FLOAT(MINT) .GT. X) MINT=MINT-1
RETURN
END

MINT 31
MINT 32
MINT 33
MINT 34
MINT 35
DESCRIPTION

PLOTST is used to initialize the plot package (and produce a leading ID frame) or to terminate the plot package (and produce a trailing ID frame).

PLOTST sets the object space to default values and then calls SC4020 to initialize. If an ID frame is desired, ENTRY IDFRME is used.

IDFRME uses descriptions in data statements to produce an ID frame. DATE is called to put the date on the ID frame.

ENDPLT terminates the plot package by calling SC4020 to terminate (emptying its plot buffers.) Then IDFRME is used to produce the trailing ID frame.
NAME    PLOTST
ENTRY POINT  PURPOSE
PLOTST    TO INITIALIZE THE PLOT PACKAGE AND TO SELECT OUTPUT DEVICES
IDENTIFY TO GENERATE THE IDENTIFICATION FRAME FOR THE PLOT PACKAGE
ENDFLT    TO TERMINATE THE PLOT PACKAGE
CALLING SEQUENCE CALL PLOTST(N,ID)

SYMBOL    TYPE    DESCRIPTION
N          I       INPUT - SUM OF DEVICE NUMBERS DESIRED SUCH THAT
               =1 DESIGNATES THE 33 MM CAMERA
               =2 DESIGNATES THE 5 INCH CAMERA
               =3 DESIGNATES THE PRINTER
IO         L       INPUT - TRUE FOR IO FRAME DESIRED

CALLING SEQUENCE CALL IDENTIFY
CALLING SEQUENCE CALL ENDFLT

SUBROUTINES USED SC4020 MORLIN DATE EMPTY
COMMON BLOCK CPLOTS
INPUT FILES  NONE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES  NONE

SUBROUTINE PLOTST (N,ID)
LOGICAL SWITCH,ID
LOGICAL PRINT,PLOTER,LCGX,LCGY
INTEGER PLUTSF
COMMON /CPLOTS/ PRINT,PLOTER,LCGX,LCGY,XLOLIM,YLOLIM
   XHILIM,YHILIM,SCALEX,SCALEY,SXL3,SYLO,IT,IPRINT,PLUTSF,LINS
   LINEC
   DIMENSION DAT(2)
   DIMENSION A(P),B(2)
   DATA A(32)/"",""/ PLOT PACKAGE FOR IBM 360 /
   DATA B /"RUN ON" /
   DATA I2 /"/ C DEFAULT GRID LIMITS AND SCALE FACTORS

PLOT 43  PLOT 44  PLOT 45  PLOT 46  PLOT 47
PLOT 48  PLOT 49  PLOT 50  PLOT 51  PLOT 52
PLOT 53  PLOT 54  PLOT 55
LUOX = FALSE
LUOY = FALSE
XLULIN = X
YLULIN = Y
XLHILIN = XH
YLHILIN = YH
SCALX = 1
SCALY = 1
XLJX = C
YLJX = D
IT = 0
IPRINT = 6
PLOT 30
LININC = 2
LINECT = 0
C SELECT DEVICES
M=4-N
IF(M.LT.-3)M=-3
CALL SC4->2C(M,D,D,D,D)
00
IF (.*ACT+1C) RETURN
C IDFRE-entry
ENTRY IDFRME
SWITCH=TRUE
GO TO 40
C EMPLT-entry
ENTRY EMPLT
CALL SC4->20 (17,C,D,D,D)
SWITCH=FALSE
C DRAW LARGE SQUARE
40
CALL SC4C2C (10,C,7,1023,D)
CALL SC4C2C (10,1023,C,1023,D)
CALL SC4C2C (9,0,C,1023,D)
CALL SC4C2C (9,C,1023,1023,D)
C DRAW SMALLER SQUARE INSIDE
CALL SC4C2C (10,255,256,767,D)
CALL SC4C2C (10,768,256,767,D)
CALL SC4C2C (9,255,256,768,C)
CALL SC4C2C (9,255,767,768,C)
C DRAW HUMUS
DO 20 J=12,511,64
CALL SC4C2C (11,1,512+1,63,63)
CALL SC4C2C (11,1,511-1,63,63)
CALL SC4C2C (11,1023-1,512+1,-63,63)
CALL SC4C2C (11,1023-1,511-1,-63,-63)
20 CONTINUE
C INSERT TITLE AND DATE
CALL HERLIN (A,31,512,750)
CALL HERLIN (B,1,512,52C)
CALL DATE (DAT)
CALL HERLIN (DAT,B,512,528)
IF (SWITCH) RETURN
C EMPTY BUFFERS AND TERMINATE PLOTER OUTPUT
CALL SC4C2C (17,C,D,D,D)
IF (PLCTER) CALL EMPLT
RETURN
END
DESCRIPTION

QUICKY is a quick plot routine. The user inputs an array of coordinates and QUICKY outputs a plot of his data complete with grid overlay.

QUICKY first calls GRDNUM for the x array to determine x characteristics for the grid overlay. Then GRDNUM is called for the y array. Then GRID is called to output the grid overlay and finally PLOT is called to plot his arrays.
NAME QUICKY

PURPOSE TO PLOT X-Y VALUES ON AN APPROPRIATE GRID

CALLING SEQUENCE CALL QUICKY(X,Y,N,CHAR)

SYMBOL TYPE DESCRIPTION
X R INPUT - ARRAY OF ABSCISSA VALUES TO BE PLOTTED
Y R INPUT - ARRAY OF ORDINATE VALUES TO BE PLOTTED
N I INPUT - NUMBER OF COORDINATES IN THE X-Y ARRAYS
CHAR A INPUT - RIGHT JUSTIFIED CHARACTER TO BE PLOTTED.
            IF CHARACTER IS BLANK, VECTORS WILL BE
            PLOTTED BETWEEN POINTS

SUBROUTINES USED GRDNUM  GRID  PLOT  HORLIN

COMMON BLOCKS  NONE

INPUT FILES  NONE

OUTPUT FILES  NONE

RESTRICTIONS N MUST BE GREATER THAN 1 AND NEITHER THE X NOR THE
               Y ARRAYS MAY HAVE ALL ELEMENTS EQUAL

REFERENCES  NONE

SUBROUTINE QUICKY(X,Y,N,CHAR)
DIMENSION X(N),Y(N)
LOGICAL*1 XFMT(7),YFMT(7)
C GET ESTHETIC GRID LIMITS AND FORMATS
CALL GRDNUM(X,N,XMIN,XMAX,NX,XFMT)
CALL GRDNLM(Y,N,YMIN,YMAX,NY,YFMT)
IF(NX.EQ.0.OR.NY.EQ.0) GO TO 10
C DRAW GRID
CALL GRID(XMIN,XMAX,NX,XFMT,1,YMIN,YMAX,NY,YFMT,1,0)
C PLOT POINTS
CALL PLOT(X,Y,N,CHAR)
RETURN
10 CALL HORLIN('EMPTY ARRAY OR ALL ITEMS EQUAL IN QUICKY')
   40,512,512 RETURN
END
DESCRIPTION

SC4020 is the basic routine which formats the SC4020 instruction. It also simulates the SC4020 by outputing printer plots.

There are two major sections: the printer and the SC4020. The first parameter to the SC4020 call is a operation indicator. If the printer has been selected, then, through a computed GO TO, the operation indicator causes the operation to be done. The same happens for the SC4020 if it has been selected.

The printer section consists of the coding which puts characters into the print buffer. The SC4020 section consists of set up the SC4020 instructions and storing these into a buffer which is output when it is filled.
Special entries in SC4020 are equivalent to calling SC4020 with certain operation indicators. These include FRMADV (frame advance) and EMPTY (empty the buffers). Other entries include NWUNIT (to specify the output units), FRAMES (to return the number of frames produced) and VBAR (to substitute a vertical bar instead of an "I" for vertical plotting on the printer.)
**NAME**

SC4020

**ENTRY POINT**

PURPOSE

SC4020 TO TRANSLATE PLOT COMMANDS INTO SC4020 INSTRUCTIONS AND/OR PRINTER PLOTS

**PURPOSE**

FFMADV TO ADVANCE THE FRAME

**PURPOSE**

NUNIT TO SET THE OUTPUT UNIT NUMBERS

**PURPOSE**

FRAMES TO RETURN A COUNT OF THE NUMBER OF FRAMES PRODUCED

**PURPOSE**

EMPTY TO TERMINATE THE PLOTTER TAPE OUTPUT

**PURPOSE**

YEAR TO USE THE VERTICAL BAR CHARACTER "|" INSTEAD OF "I" FOR VERTICAL LINES OF THE PRINTER PLOTS

**PURPOSE**

CONDNS TO SET A FRAME OF PRINTER AS ONE COMPUTER PAGE INSTEAD OF THE NORMAL TWO

**CALLING SEQUENCES**

BECAUSE EACH OF THE 15 PLOT COMMANDS USES THE ARGUMENT LIST DIFFERENTLY, EACH CALLING SEQUENCE IS LISTED. IN EACH CASE THE ARGUMENT "*D*" IS A DUMMY ARGUMENT, AND THE FIRST ARGUMENT IS THE FUNCTION CODE, "*DP*".

<p>| Call SC4020(-3,D,D,D,D) | -3 | 1 | COMMAND TO SELECT BOTH CAMERAS AND PRINTER |
| Call SC4020(-2,D,D,D,D) | -2 | 1 | COMMAND TO SELECT CAMERA 2 AND PRINTER |
| Call SC4020(-1,D,D,D,D) | -1 | 1 | COMMAND TO SELECT CAMERA 1 AND PRINTER |
| Call SC4020(0,D,D,D,D) | 0 | 1 | COMMAND TO SELECT PRINTER |
| Call SC4020(1,D,D,D,D) | 1 | 1 | COMMAND TO SELECT CAMERA 1 |
| Call SC4020(2,D,D,D,D) | 2 | 1 | COMMAND TO SELECT CAMERA 2 |</p>
<table>
<thead>
<tr>
<th>Call SC4020(D,D,D,D)</th>
<th>3</th>
<th>I</th>
<th>COMMAND TO SELECT BOTH CAMERAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call SC4020(4,D,D,D,D)</td>
<td>4</td>
<td>I</td>
<td>COMMAND TO ADVANCE FILM</td>
</tr>
<tr>
<td>Call SC4020(5,X,Y,CHAR,D)</td>
<td>5</td>
<td>I</td>
<td>COMMAND TO PLOT SINGLE CHARACTER AT COORDINATES X, Y AT CURRENT LIGHT INTENSITY</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>I</td>
<td>RASTER COUNT OF X COORDINATE</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>I</td>
<td>RASTER COUNT OF Y COORDINATE</td>
</tr>
<tr>
<td></td>
<td>CHAR(1)</td>
<td>L*1</td>
<td>CHARACTER TO BE PLOTTED</td>
</tr>
<tr>
<td>Call SC4020(6,X,Y,CHAR,B)</td>
<td>6</td>
<td>I</td>
<td>COMMAND TO PLOT SINGLE CHARACTER AT COORDINATES X,Y WITH B LIGHT INTENSITY</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>I</td>
<td>RASTER COUNT OF X COORDINATE</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>I</td>
<td>RASTER COUNT OF Y COORDINATE</td>
</tr>
<tr>
<td></td>
<td>CHAR(1)</td>
<td>L*1</td>
<td>CHARACTER TO BE PLOTTED</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>I</td>
<td>LIGHT INTENSITY (0-15)</td>
</tr>
<tr>
<td>Call SC4020(7,X,Y,CHAR,D)</td>
<td>7</td>
<td>I</td>
<td>COMMAND TO PLOT SINGLE CHARACTER AT COORDINATES X, Y AND SET LIGHT INTENSITY TO BRIGHT</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>I</td>
<td>RASTER COUNT OF X COORDINATE</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>I</td>
<td>RASTER COUNT OF Y COORDINATE</td>
</tr>
<tr>
<td></td>
<td>CHAR(1)</td>
<td>L*1</td>
<td>CHARACTER TO BE PLOTTED</td>
</tr>
<tr>
<td>Call SC4020(8,X,Y,CHAR,D)</td>
<td>8</td>
<td>I</td>
<td>COMMAND TO PLOT</td>
</tr>
</tbody>
</table>
SINGLE CHARACTER AT COORDINATES X, Y AND SLT LIGHT INTENSITY TO DIM

X I RASTER COUNT OF X COORDINATE

Y I RASTER COUNT OF Y COORDINATE

CHAR(1) L*1 CHARACTER TO BE PLOTTED

CALL SC4020(5,X,Y,STOP,D) 9 I COMMAND TO GENERATE LINE FROM COORDINATES (X,Y) TO (STOP,Y)

X I RASTER COUNT OF X COORDINATE OF STARTING POINT

Y I RASTER COUNT OF Y COORDINATE OF LINE

STOP I RASTER COUNT OF X COORDINATE OF END POINT

CALL SC4020(10,X,Y,STOP,D) 10 I COMMAND TO GENERATE LINE FROM COORDINATES (X,Y) TO (X,STOP)

X I RASTER COUNT OF X COORDINATE OF LINE

Y I RASTER COUNT OF Y COORDINATE OF STARTING POINT

STOP I RASTER COUNT OF Y COORDINATE OF END POINT.

CALL SC4020(11,X,Y,XV,YV) 11 I COMMAND TO GENERATE LINE FROM COORDINATES (X,Y) TO (X+XV, Y+YV)

X I RASTER COUNT OF X COORDINATE
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

<table>
<thead>
<tr>
<th>Y</th>
<th>RASTER COUNT OF Y COORDINATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>XV</td>
<td>SIGNED RASTER COUNT OF X COMPONENT OF THE VECTOR TO BE PLOTTED</td>
</tr>
<tr>
<td>YV</td>
<td>SIGNED RASTER COUNT OF Y COMPONENT OF THE VECTOR TO BE PLOTTED</td>
</tr>
</tbody>
</table>

Call SC4020(12,D,D,D,D) 12 I

Command to force plotter screen square for normal plotting (reduce image)

Call SC4020(13,D,D,D,D) 13 I

Command to force plotter screen rectangular for continuing plot on next frame (expand image)

Call SC4020(14,D,D,D,D) 14 I

Command to project prepared slide onto camera (requires specially prepared slide)

Call SC4020(15,X,Y,CHAR,N) 15 I

Command to begin typewriter mode and type characters starting from (X,Y)

<table>
<thead>
<tr>
<th>X</th>
<th>RASTER COUNT OF X COORDINATE OF STARTING POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>RASTER COUNT OF Y COORDINATE OF STARTING POINT</td>
</tr>
<tr>
<td>CHAR(N)</td>
<td>CHARACTERS TO BE PLOTTED</td>
</tr>
<tr>
<td>N</td>
<td>NUMBER OF CHARACTERS TO</td>
</tr>
</tbody>
</table>

1-170
CALL SC4020(16,D,C,CHAR,N) 16 I

CALL SC4020(17,D,C,D,D) 17 I

CALL SC4020(18,D,C,CHAR,N) 18 I

CALL SC4020(19,D,D,CHAR,N) 19 I

CALL SC4020(20,D,C,D,D) 20 I

BE PLOTTED

COMMAND TO BEGIN TYPEWRITER MODE AND TYPE CHARACTERS STARTING AT THE BEGINNING OF THE LAST VECTOR PLOTTED OR LAST POINT PLOTTED

CHAR(N) L*1 CHARACTERS TO BE PLOTTED

N I NUMBER OF CHARACTERS TO BE PLOTTED

CALL SC4020(17,D,C,D,D) 17 I COMMAND TO ADVANCE FILM, SET LIGHT INTENSITY TO BRIGHT, AND END TYPEWRITER MODE.

CALL SC4020(18,D,C,CHAR,N) 18 I COMMAND FOR CARTRIDGE RETURN AND TYPE CHARACTERS STARTING ON NEXT LINE

CHAR(N) L*1 CHARACTERS TO BE PLOTTED

N I NUMBER OF CHARACTERS TO BE PLOTTED

CALL SC4020(19,D,D,CHAR,N) 19 I COMMAND TO CONTINUE TYPEWRITER MODE ADDING CHARACTERS AFTER LAST CHARACTER TYPED

CHAR(N) L*1 CHARACTERS TO BE PLOTTED

N I NUMBER OF CHARACTERS TO BE PLOTTED

CALL SC4020(20,D,C,D,D) 20 I COMMAND TO STOP TYPEWRITER
CALLING SEQUENCE CALL FRMADV

CALLING SEQUENCE CALL NWUNIT(IPRNT,IPLCTR)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPRNT</td>
<td>I</td>
<td>INPUT - FORTRAN LOGICAL UNIT NUMBER FOR PRINTER PLOTS</td>
</tr>
<tr>
<td>IPLCTR</td>
<td>I</td>
<td>INPUT - FORTRAN LOGICAL UNIT NUMBER FOR PLOTTER DRIVE TAPE</td>
</tr>
</tbody>
</table>

CALLING SEQUENCE CALL FRAMES(FRMONT)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRMONT</td>
<td>I</td>
<td>OUTPUT - NUMBER OF FRAMES PRODUCED</td>
</tr>
</tbody>
</table>

CALLING SEQUENCE CALL EMPTY

CALLING SEQUENCE CALL VBAR

CALLING SEQUENCE CALL CONDENS

SUBROUTINE USED SCHAR

COMMON BLOCK CPlots

INPUT FILES NONE

OUTPUT FILES PRNT - FORTRAN LOGICAL UNIT NUMBER FOR PRINTER PLOTS

PLOTS - FORTRAN LOGICAL UNIT FOR SC4020 PLOTS

RESTRICTIONS NONE

REFERENCES NONE

SUBROUTINE SC4020 (OP, XY, IACHAR, YV)
COMMON /CPlots/PRINT,PLOTER,G1(II),PRNT,PLOTIS;LINING,LINECT
INTEGER SCHAR
INTEGER OP,OP1,XY,XV,YY,PRNT,PLOTIS,FRMCNT,
SHIFT2,SHIFT4,SHIFTS,SHIFT6,SHIFT8,
INTEGER CARETN, STUPTR,RESET,
INTEGER*2 SETPOS,17/P
LOGICAL PRINT, PLOTER, INITIAL,TERM, PERR, TYPING,TYPMOD
LOGICAL*1 CPCODE(25),ERROR(12),OUT(4092),PHIF(128,128)
LOGICAL*1 MINUS,OUT,BLANK,SLASH,PT,BAR
LOGICAL*1 FORMAT(30),"(2H1,6I4,(128A1/2H1,128A1/2X))",/.

PLUS/++/

1-172
LOGICAL*LX,DUM(1),CHAR(1),XCHAR(256),INCHAR(YV)
C LX IS EQUIVALENCED TO THE 1ST ORDER BYTE IN,
C THIS IS FOR PROCESSING PURPOSES.
EQUIVALENCE (1W,DUM(11),DUM(4),LX)
EQUIVALENCE (XCHAR(1),CHAR(11))
EQUIVALENCE (SETPOS,15)
C INITIALIZE SWITCHES
DATA INITAL,PERR,TERM,TYPING */&/
C SC-4026 DP CCES
DATA EFCODE/ Z21,Z22,Z23,Z26,Z29,Z31,Z32,Z04,  
*  Z18,Z1A,Z30,Z25,Z24,Z15,Z12,Z22,Z2A,Z30,Z0A  
DATA CREAT,STOPTP,RESET /ZC00C02A,ZC00C0CA,ZC00C02E  
C INSERTS SLASHES IN UPPER RIGHT CORNER OF FRAME
DATA ERROR / Z1C,Z1F,Z0F,Z31,Z30,Z31,Z31,Z31,Z31,Z31,Z0A  
C PICT CHARACTERS FOR PRINTER PRINTS
DATA I,MINUS,DOT,BLANK,S/AB/1H,1H-1H,1H*1H*/  
C CONSTANTS FOR BIT MANIPULATION
DATA SHIFT2,SHIFT4,SHIFT5,SHIFT6,*SHIFT8  
*  Z003C00A, Z003C00D, Z0030020, Z0030049, Z00300100  
C MISC. CONSTANTS
DATA 12Z1ZAP  
DATA IBLANK /Z00000C40  
C NORMINAL ENTRY
OPI=1AES(CP)
C TEST FOR DEVICE SELECTION CP CODE
IF (OPI.GT.3) GO TO 1
C SET DEVICE SWITCHES
PRINT=CP.LE.C
PILOT=CP.GT.0
C RETURN IF PRINTER SELECT ONLY
IF (OPI.EQ.0) RETURN
GO TO 1
C FRMADV ENTRY
ENTRY FRMADV
C SET CP CODE FOR RESET
OPI=17
C TEST FOR INITIALIZATION
1 IF (INITAL) GO TO 20
INITAL=*TRUE*
C ZERO FRAME COMMAND AND COMMAND BUFFER INDEX
IFRM=0
ICOUNT=0
C SET UP CHARACTER TRANSLATION MATRIX
IW=0
DO 9 I=1Z,255
9 CHAR(I)=LX
DO 10 I=1Z,63
IW=1
KH=SCF/A(1)
10 CHAR(KH)=LX
C SET PRINT BUFFER TO BLANKS
DO 15 I=1Z,128
DO 15 J=1Z,128
15 PJUJ(J,J)=BLANK
C ZERO CURRENT PRINTER REGISTER VALUES
RX=0.
RY=0.

C TEST FOR INPUT COMMAND IN WRONG MODE
20 IF ((TYING.AND.OP1.LE.17).OR.(.NOT.TYING.AND.OP1.LE.17))
* GO TO 25

C SET PRINTER AND PLOTTER ERROR FLAGS
PERR=.TRUE.
TERR=.TRUE.

C RETURN IF NOT FRAME ADVANCE
IF (OP1.NE.4) RETURN

C SET CP CODE FOR RESET
OP1=17

C SET NEW PROGRAM MODE
25 TYPMOD=TYING

C BEGIN PROCESSING

C TRANSFER ON CP CODE
GO TO (45,45,45,30,45,45,45,45,35,35,45,45,45,45,
* 43,43,39,45,45,30),OP1

C SET FOR PLOTTING MODE
30 TYPMOD=.FALSE.
GO TO 45

C RECOVER X VECTOR COMPONENT OR AXIS END POINT
35 DO 40 I=1,4
40 DV(I)=INCHAR(I)
XV=I
GO TO 45

C SET FOR TYPEWRITER MODE
43 TYPMOD=.TRUE.
45 IC=CP:

C TEST FOR PRINTER PLOTS
IF (.NOT.PRINT) GO TO 150

C TRANSFER ON CP CODE
GO TO (150,150,150,125,65,65,65,55,60,50,150,150,150,
* 90,95,125,85,100,145),OP1

C SET PLOT CHARACTER FOR VECTOR
50 PT=DOT

C CALCULATE MAXIMUM DEFLECTION MAGNITUDE
V=AMAXC(IABS(XV),IABS(YV))

C TEST FOR DEFORMATION CUT OF RANGE
IF (V.LT.64.) GO TO 53
PERR=.TRUE.
GO TO 150

C COMPUTE PRINTER VECTOR COMPONENTS
53 N=V/8.
IF (N.EQ.0) GO TO 70
RXV=FLCAT(XV)/V
RYV=FLCAT(YV)/V
GO TO 70

C SET X AXIS CHARACTER
55 PT=MIALS

C COMPUTE PRINTER VECTOR COMPONENTS
N=(XV-X)/8
RYV=0.
RXV=I.
GO TO 70

C SET Y AXIS CHARACTER
60 PT=I1
C COMPUTE PRINTER VECTOR COMPONENTS
N=(XV-Y)/8
RYV=1*
RXV=0*
GO TO 70
C SET PLOT CHARACTER TO INPUT
65 PT=INCFAR(1)
N=0
C COMPUTE PRINTER DEFLECTIONS FOR ORIGIN
70 RX=FLCAT(X)/8
RY=FLCAT(Y)/8*
C LOOP TO PLOT ALL CHARACTERS IN LINE
DO 80 I=12,N
C COMPUTE PRINTER DEFLECTIONS FOR EACH POINT
R=1
IX=RX+R*RXV
IY=128-INT(RY+R*RYV)
C TEST FOR DEFLECTIONS IN RANGE
IF (IX.GE.0.AND.IX.LE.127.AND.IY.GE.1.AND.IY.LE.128) GO TO 75
PERR=.TRUE.
GO TO 60
C INSERT CHARACTER IN BUFFER
75 PBUF(I+1,IY)=PT
80 CONTINUE
C SET CURRENT POINT INDEX TO BEGINNING OF LINE
85 INDEX=(INDEX+127)/128
GO TO 100
C SET CURRENT POINT INDEX FROM GIVEN POINT
90 INDEX=128*(127-Y/8)+X/8
GO TO 105
C SET CURRENT POINT INDEX FROM CURRENT POINT REGISTERS
95 INDEX=128*(127-INT(RY))+INT(RX)
C TEST FOR CHARACTERS TO PLOT
100 IF (YV.LT.1) GO TO 150
105 INV=0
C LOOP TO PROCESS ALL CHARACTERS
DO 120 I=1,YV
LX=INCFAR(I)
C TEST FOR CARRIAGE RETURN
IF (LX.EQ.CARETN) GO TO 110
C SET CURRENT POINT INDEX
INDEX=(INDEX+127)/128
GO TO 120
C TEST FOR STOP TYPE OR RESET COMMAND
110 IF (LX.EQ.SIGSTOP.AND.INV.EQ.RESET) GO TO 115
C MUST BE LAST CHARACTER IN STRING
IF (LX.EQ.YV) GO TO 113
C SET MODE SWITCH
TYPEMOD=.FALSE.
C TEST FOR STOP TYPE COMMAND
IF (LX.EQ.SIGSTOP) GO TO 145
C SET CP CODE FOR RESET
UP=17
GO TO 125
C SET ERROR INDICATOR
113  PERR=.TRUE.
     GO TO 150
C INCREMENT CURRENT POINT INDEX
115  INDEX=INDEX+1
C INSERT CHARACTER IN BUFFER
120  CONTINUE
     GO TO 150
C TEST FOR ERRORS ON THIS FRAME
125  IF (.NOT.PERR) GO TO 135
C INSERT ERROR SLASHES
130  DO 135 I=121,128
     PBUF(I,1)=SLASH
C RESET ERROR SWITCH
     PERR=.FALSE.
C OUTPUT PRINT BUFFER
135  WRITE(PRNT,FORMAT) PBUF
C INITIALIZE PRINT BUFFER
140  DO 145 I=1,128
     PBUF(I,J)=SLANK
C TEST FOR RESET COMMAND
     IF (OPC#NE.17) GO TO 145
C ZERO CURRENT POINT REGISTER VALUES
     RX=0.
     RY=0.
     GO TO 150
C SET CURRENT POINT REGISTER VALUES
145  RX=INDEX,128
     RY=INDEX/128
C TEST FOR SC4020 PLOTS
150  IF (.NOT.PLOT#) GO TO 300
C INSERT OPCODE IN COMMAND BUFFER
     OPI=OPC
C TEST FOR OPCODE OUT OF RANGE
155  OUT(ICCUNT+1)=OPC#(OPI)
     IW=0
C TRANSFER ON CP CODE
C COMPUTE VECTOR COMPONENTS
160  IXV=IALS(XV)
     IYV=IALS(YV)
C TEST FOR COMPONENTS OUT OF RANGE
     IF (IXV.GT.63.OR.IYV.GT.63) GO TO 350
C INSERT LEADING VECTOR DIGITS IN COMMAND
     LX=OUT(ICCUNT+1)
     IW=IW+1XV/SHIFT2
     OUT(ICCUNT+1)=LX
C SET VECTOR DIGITS CONSTANTS
     IW=IW+YV/SHIFT2
     IF (YV.GT.0) IW=IW+SHIFT4
     IF (XV.GT.0) IW=IW+SHIFT5
     OUT(ICCUNT+4)=LX
     IXV=HEC(IYV,SHIFT2)*SHIFT8
     IYV=HEC(IYV,SHIFT2)*SHIFT8
     GO TO 155
C SET X STOP CCEE
165  IS=-1-XV
SETPOS=1ZAP
GO TO 175
C SET Y STOP CCEE
170  IS=1023-XV
C TEST FOR STOP CODE OUT OF RANGE
175  IF (XV.GT.1023.OR.XV.LT.0) GO TO 350
C SET DEFLECTION BIT CONSTANTS
IWY=MCC(IS,SHIFT2)*SHIFT4
IWX=((IS/SHIFT8)*SHIFT4
IW=MCC(IS/SHIFT2,SHIFT6)
OUT(ICCUNT+4)=LX
GO TO 195
C SET INTENSITY BIT CONSTANTS
180  IW=X=YV/SHIFT2*SHIFT4
IWY=MCC(YV,SHIFT2)*SHIFT4
IW=0
C STORE CHARACTER IN COMMAND BUFFER
190  IX=INCHAR(1)
OUT(ICCUNT+4)=CHAR(IW)
IF (IX.EQ.IOBLANK) OUT(ICCUNT+1)=OPCODE(5)
GO TO 195
C SET BIT CONSTANTS TO ZERODES
195  IWX=0
IWY=0
C STORE CHARACTER IN COMMAND BUFFER
190  IX=0
IX=INCHAR(1)
OUT(ICCUNT+4)=CHAR(IW)
C TEST FOR DEFLECTIONS OUT OF RANGE
195  IF (X.GT.1023.OR.X.LT.0.OR.Y.GT.1023.OR.Y.LT.0) GO TO 350
C INSERT BIT, CONSTANTS AND DEFLECTIONS IN COMMAND BUFFER
IW=IWX(IS/SHIFT6)
OUT(ICCUNT+2)=LX
IW=X
OUT(ICCUNT+3)=LX
IS=1023-Y
IW=IWY+IS/SHIFT6
OUT(ICCUNT+5)=LX
IW=IS
OUT(ICCUNT+6)=LX
C TEST FOR TYPE SPECIFIED PCINT OPERATION
IF (OP1.NE.15) GO TO 285
C INCREMENT COMMAND BUFFER COUNT
ICOUNT=ICOUNT+1
C SET TO BEGIN WITH SECOND CHARACTER
IN=2
GO TO 210
C CORRECT BUFFER COUNT
200  ICOUNT=ICOUNT+1
C START ON FIRST CHARACTER
205  IN=1
C TEST FOR CHARACTERS TO ADD TO BUFFER
210  IF (IN.GT.YV) GO TO 290
IW=0
C LOOP TO PROCESS ALL CHARACTERS
DJ 230 J=IN.YV
IF (ICCUNT.LT.4092) GO TO 215
C OUTPUT BUFFER IF NECESSARY
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT) /
ICCUNT=0
215 LX=INCHAR(J)
C TEST FOR STOP TYPE OR RESET OPERATION
IF (JNE.STOPTP.AND.I=.NE.RESET) GO TO 220
C MUST BE LAST CHARACTER IN STRING
IF (J,NE.YV) GO TO 350
C INSERT COMMAND IN BUFFER AND SET MODE SWITCH
OUT(ICCUNT+1)=LX
TYPMOD=.FALSE.
GO TO 240
220 LX=CHAR(I)
C INSERT CHARACTER IN BUFFER
ICCUNT=ICCUNT+1
230 OUT(ICCUNT)=LX
C TEST TO SEE IF STILL IN TYPEWRITER MODE
240 IF (TYPMOD) GO TO 250
C TEST TO SEE IF LAST CHARACTER WAS STOP TYPE
IF (LX.EQ.STOPTP) GC TO 275
C SET CP CODE FOR RESET
OP1=17
C TEST FOR ERRORS ON THIS FRAME
250 IF (.NOT.TERR) GO TO 275
C TEST TO SEE IF IN TYPEWRITER MODE
IF (.NOT.TYPING) GO TO 260
C INSERT STOP CODE COMMAND IN BUFFER AND ADJUST COUNT
OUT(ICCUNT+1)=OPCODE(20)
ICCUNT=ICCUNT+6/6*6
GO TO 260
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C RESET ERROR SWITCH
260 TERR=.FALSE.
C TEST FOR ROOM IN BUFFER
IF (ICCUNT.LT.4077) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300
C TEST FOR ERRORS ON THIS FRAME
255 IF (.NOT.TERR) GO TO 265
C OUTPUT BUFFER ERROR MARK CODES, AND RESET OR FRAME ADVANCE
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT).ERROR.OPCODE(OP1)
GO TO 295
C INSERT ERROR CODES IN BUFFER
265 DO 270 I=1,12
270 OUT(ICCUNT+1)=ERROR(I)
OUT(ICCUNT+13)=OPCODE(OP1)
ICCUNT=ICCUNT+18
GO TO 290
C ROUND BUFFER COUNT TO AN EVEN COMMAND SIZE
275 ICCUNT=(ICCUNT+6)/6*6
GO TO 290
C INCREMENT BUFFER COUNT BY A FULL COMMAND SIZE
285 ICCUNT=ICCUNT+6
C OUTPUT BUFFER IF FULL
290 IF (ICCUNT.LT.4092) GO TO 300
WRITE (PLOT1$,200C) (OUT(I),I=1,ICCUNT)
GO TO 300

1-178
C ZERO BUFFER COUNT
295 ICOUNT=0
C SAVE MODE OF OPERATION
300 TYPING=TYPMOD
C RETURN IF NOT FRAME ADVANCE OR RESET
IF (IPLINE .NE. 17 .AND. IPLINE .NE. 4) RETURN
C INCREMENT FRAME COUNT AND ZERO LINE COUNT
IFRM=IFRM+1
LINCT=0
C RETURN
RETURN
C SET ERROR INDICATOR AND RETURN
350 TERR=.TRUE.
RETURN
C NUNIT ENTRY
ENTRY NUNIT (IPRT, IPPLTR)
C SET PRINTER AND PLOTTER UNITS
PRNT=IPRT
PLTTR=IPPLTR
RETURN
C FRAMES ENTRY
ENTRY FRAMES (FRMCNT)
C RETURN FRAME COUNT
FRMCNT=IFRM
RETURN
C EMPTY ENTRY
ENTRY EMPTY
C EMPTY PLOT BUFFER AND END FILE OUTPUT UNIT
IF (ICOUNT .GT. 0) WRITE (PLTTR,2000) (OUT(I), I=1,ICOUNT)
ICOUNT=0
END FILE PLOT1S
RETURN
C YEAR ENTRY
ENTRY YEAR
C SET Y AXIS CHARACTER TO VERTICAL BAR
II=BAR
RETURN
C CONDAS ENTRY
ENTRY CONDAS
C SET PRINT FORMAT FOR ONE PAGE INSTEAD OF TWO
FORMAT(18)=PLUS
RETURN
2000 FORMAT (6A1)
END
30
COMMAND WILL SET SC4020 TO PLOTTING MODE

45

35
RECOVER X VECTOR COMPONENT OR AXIS END POINT

45

43
COMMAND WILL SET SC4020 TO TYPERWRITER MODE

45

PRINTER PLOTS

YES

OPCODE 1

YES

150

NO

OPCODE 2

YES

150

NO

OPCODE 3

YES

150

NO

OPCODE 4

YES

125

NO

C
19

OPCODE 19

145

YES

100

NO

53

COMPUTE
PRINTER
VECTOR
COMPONENTS

70

55

SET
X AXIS
CHARACTER

COMPUTE
PRINTER
VECTOR
COMPONENTS

70

50

SET PLOT
CHARACTER
FOR VECTOR

CALCULATE
MAXIMUM
DEFLECTION
MAGNITUDE

DEFLECTION
OUT OF
RANGE

NO

53

YES

SET
ERROR SWITCH

150
F

INSERT CHARACTER IN BUFFER

80

MORE CHARACTERS

YES POINT

NO

150

85

SET CURRENT POINT INDEX TO BEGINNING OF LINE

100

105

SET CURRENT POINT INDEX

110

120

90

SET CURRENT POINT INDEX

105

100

ANY CHARACTERS TO PLOT

NO 150

YES

CARRIAGE RETURN

NO

YES

1-187
RESET ERROR SWITCH

OUTPUT BUFFER, ERROR CODES, AND RESET OR FRAME ADVANCE

RESET ERROR SWITCH

ROUND BUFFER COUNT UP TO AN EVEN COMMAND SIZE

INCREMENT BUFFER COUNT BY FULL COMMAND SIZE

BUFFER FULL

OUTPUT BUFFER

INSERT ERROR CODES IN BUFFER
DESCRIPTION

SCHAR is a function routine which when given an SC4020 character value determines the corresponding EBCDIC character value.

The method involves an array XCHAR which is set up by a data statement so that if I is the input and SCHAR is the output, then basically SCHAR=XCHAR(I).
NAME: SCHAR

PURPOSE: THE FUNCTION VALUE IS THE EBCDIC CHARACTER VALUE CORRESPONDING TO THE INPUT SC4020 CHARACTER VALUE

CALLING SEQUENCE: SCHAR(I)

SYMBOL, TYPE, DESCRIPTION

<table>
<thead>
<tr>
<th>I</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>SC4020 CHARACTER VALUE</td>
</tr>
</tbody>
</table>

| SCHAR | OUTPUT - EBCDIC CHARACTER VALUE |

SUBROUTINES USED: NONE

COMMON BLOCKS: NONE

INPUT FILES: NONE

OUTPUT FILES: NONE

RESTRICTIONS: NONE

REFERENCES: NONE

INTEGER FUNCTION SCHAR(I)

LOGICAL LX,CHAR,XCHAR,DUM(4)

DIMENSION CHAR(1),XCHAR(64)

EQUIVALENCE (IW,DUM(1)),(LX,DUM(4))

EQUIVALENCE (XCHAR(2),CHAR(1))

DATA I, /0/

C TABLE CONTAINS EBCDIC CHARACTER VALUES CORRESPONDING TO SC4020
C CHARACTER SET - NOTE THAT APPROPRIATE SC4020 VALUES RANGE FROM 0 TO 63
C Z4A=CENT,Z5A=EXCLAMATION POINT,Z60=0-2-H PUNCH

DATA XCHAR /

C PERFORM TABLE LOOKUP

LX=CHAR(I)
SCHAR=IW
RETURN
END

SCHA 29
SCHA 30
SCHA 31
SCHA 32
SCHA 33
SCHA 34
SCHA 35
SCHA 36
SCHA 37
SCHA 38
SCHA 39
SCHA 40
SCHA 41
SCHA 42
SCHA 43
SCHA 44
SCHA 45
SCHA 46
SCHA 47
SCHA 48
SCHA 49
SCHA 50
SCHA 51
SCHA 52
DESCRIPTION

TIMING has one entry, NOW. This is used (by DATE, for example) to determine the date and time of day. A system macro is used to get the date in YYDDD integer format and the time of day in hundredths of seconds.
CONTROL SECTION NAME TIMING
ENTRY POINT PURPOSE

NOW OBTAINS THE CURRENT DATE IN IBM PACKED INTEGER
FORMAT (YYDDD) AND THE TIME OF DAY IN INTEGER
HUNDREDTHS OF SECONDS

CALLING SEQUENCE CALL NOW(IYDD, IHM)

SYMBOL TYPE DESCRIPTION
IYDD I OUTPUT - YYDDD FOR CURRENT DATE
IHM I OUTPUT - TIME OF DAY IN HUNDREDTHS OF SECONDS

SUBROUTINES USED NONE
COMMON BLOCKS NONE
INPUT FILES NONE
OUTPUT FILES NONE
RESTRICTIONS NONE
REFERENCES NONE

* TIMING START C
ENTRY NOW

NOW SAVE (14,12),* TIMI 33
BALR 4,0 TIMI 34
USING 4,4 TIMI 35
LH 5,6,0(1) ADDRESSES OF OUTPUT ARGUMENTS TIMI 36
TIME EIN TIMI 37
ST 0,0(6) RETURN TIME OF DAY IN HUNDREDTHS OF SECONDS TIMI 38
ST 1, TEMP+4 MUST CONVERT YYDDD TO BINARY INTEGER TIMI 39
CVB 1, TEMP TIMI 40
ST 1,0(5) BEFORE RETURNING IT TIMI 41
RETURN (14,12) TIMI 42

* TEMP DC C'0'* TIMI 43
END TIMI 44

TIMI 45

TIMI 46

TIMI 47

1-201
DESCRIPTION

TYPLIN is a routine which is used to type line information on the SC4020.

The SC4020 is put into typewriter mode and the carriage control character is inspected. A '1' causes a frame advance for example. The line is typed and the SC4020 is put back into plotting mode.

SETPAG is an entry in TYPLIN so the user can specify where the typing should start. Otherwise, typing starts where it ended last or at the beginning if typing has just started.
NAME  TYPLIN

ENTRY POINT PURPOSE

TYPLIN  TO TYPE A LINE OF INFORMATION

SETPAG  TO SET LINE NUMBER AND COLUMN NUMBER OF TYPLIN

CALLING SEQUENCE CALL TYPLIN(LINE,N)

SYMBOL TYPE DESCRIPTION

LINE  A    INPUT - ARRAY OF CHARACTERS (FIRST CHARACTER IN
           "CARRIAGE CONTROL")

N    I    INPUT - LENGTH OF LINE

CALLING SEQUENCE CALL SETPAG(LINES,ICOL)

SYMBOL TYPE DESCRIPTION

LINES  I    INPUT - LINE NUMBER FOR NEXT CALL TO TYPLIN

ICOL  I    INPUT - COLUMN NUMBER FOR SUBSEQUENT CALLS TO
           TYPLIN

SUBROUTINE USED  SC4020

COMMON BLOCK  CPLOTS

INPUT FILES  NONE

OUTPUT FILES  NONE

RESTRICTIONS  NONE

REFERENCES  NONE

SUBROUTINE TYPLIN(LINE,N)

COMMON /CPLOTS/ GI(15),LININC,LINECT

LOGICAL*1 LINE(1),LX,BLANK,DUM(4),STOPC

EQUIVALENCE (IW,DUM(1)),(LX,DUM(4))

DATA BLANK,STOPCD/1H ,ZDA /

DATA IGE,IZERO /20000CCFI,20000OF0 /

DATA IWO /

DATA IX /0/

: PERFORM INDICATED CARRIAGE CONTROL OPERATION

LX=LINE(1)

IF(IW.EQ.IGE) GO TO 10

IF(IW.EQ.IZERO) LINECT=LINECT+1

IF(LINECT.LT.1024) GO TO 50

10  LINECT=0
CALL SC4020 (17,0,0,0,0)
S0 IY=1023-LINECT*LININC
IF(N.EQ.1) RETURN
C TYPE LINE
  N1=MIN0(N1+1,130)
  LX=LINE(N1)
  LINE(N1)=S10PCD
  CALL SC4020 (15,IX,IY,LINE(21,N)
  LINE(N1)=LX
  LINECT=LINECT+1
  RETURN
C SETPAG ENTRY
ENTRY SETPAG (LINES,ICCL)
C SET LINE NUMBER AND STARTING COLUMN
IX=ICOL*8-8
IF(I'CUL.GT.128.OR.ICOL.LT.1) IX=0
LINECT=LINES
RETURN
END
DESCRIPTION

UCS contain the character description arrays for the standard EBCDIC character font. These are set up in data statements.

If BLKLET is to be used (to produce block letters) than some character description must be input via CSET in BLKLET. In order to aid the programmer, a simple call to UCS will accomplish the above.
NAME
UCS

PURPOSE
TO CALL CSET WITH A STANDS 300 CHARACTER SET

CALLING SEQUENCE
CALL UCS

SUBROUTINE USED
CSET

COMMON BLOCKS
NONE

INPUT FILES
NONE

OUTPUT FILES
NONE

RESTRICTIONS
NONE

REFERENCES
NONE

SUBROUTINE UCS
C

CHARACTER SET
LOGICAL(162)

C

INDEX OF ABOVE CHARACTERS IN IVEC
INTEGER(163)

C

Raster Coordinates of Vectors for Each Character:

INDEX IVEC(100)

INTEGER(164)

F

PACKED AS X,Y,DX, CY
Z6451, Z5620, Z471, ZC961, ZC991, Z20A9, Z19A9, Z6S51, Z6662, Z5571/ UCSI56
*/EQUIVALENCE (IVEC(151), IVEC(1)) UCSI57
*/INTEGER IVEC(1164) UCSI58
*Z3520, Z1551, ZC652, ZC671, Z6451, Z6163, Z5C71, Z502C, Z1551, Z1631, UCSI59
*Z20471, Z1612, Z1CA9, ZC671, Z6167, Z6531, Z5C2C, Z5571, Z6564, Z1451, UCSI60
*Z1160, Z4571, ZF251, Z22AC, Z4271, Z1361, Z1671, Z72U5, Z5551, Z1217, UCSI61
*Z2170, Z2161, Z2270, Z1341, ZC594, Z5014, Z5051, Z4521, Z33C3, Z3672, UCSI62
*Z4871, Z6560, Z366, Z3666, Z3670, Z4571, Z414C, Z3041, Z1151, Z1262, UCSI63
*Z20481, Z2570, Z5541, ZC661, ZC771, Z1861, Z397C, Z5851, Z2C7C, Z2661, UCSI64
*Z2170, Z1061, Z2270, Z2267, Z2970, Z3267, Z3C69, Z3142, Z2160, ZA182, UCSI65
*Z6361, Z64C1, ZC561, ZC632, Z2880, Z6642, Z09C0, Z11A6, Z5126, Z1071, UCSI66
*Z2172, Z3363, Z2652, Z2651, Z257C, Z2551, Z3561, Z2270, Z2261, UCSI67
*Z2370, Z3261, Z2171, ZC5CC, Z6362, Z05CC, Z30C0, 22170, Z2216, Z2270, UCSI68
*Z3161, Z2C71, Z2651, Z2670, Z2671, Z2451, Z295C, Z1761, Z1071, UCSI69
*Z200C9, Z3051, Z2216, Z2271, Z3070, Z4671, Z5161, Z5251, Z3370, Z00CC, UCSI70
*Z16B4, Z6414, Z2C61, Z3262, Z3470, Z4452, Z6561, Z6742, ZC702, Z2960, UCSI71
*Z2570, Z2561, Z2561, Z267C, Z2270, Z2251, Z2370, Z3261, Z1396, Z3386, UCSI72
*Z205C0, Z207C0, Z30A2, Z2042, Z2064, Z2602, Z2075, Z5642, Z4271, UCSI73
*Z5363, Z2270, Z3271, Z4263, Z2251, Z1362, Z1571, Z2670, Z5651, Z3662, UCSI74
*Z204C0, Z206C0, Z2662, Z4663/ CALL CSET(62, ICHAR, IP4OS, IVEC) RETURN
END
1.2 GEODYN DATA HANDLING SUPPORT PROGRAMS

There are five data handling programs used by the GEODYN program: DODS SORT-MERGE, GEOS SORT-MERGE, EPHemeris TAPE GENERATOR, ORB1 CONVERSION and TdF TABLE GENERATOR.

DODS SORT-MERGE sorts and merges DODS formatted data from two tapes onto one tape. The data can be from any number of satellites. GEOS SORT-MERGE performs the same task; however, data from only one satellite should be used. EPHemeris TAPE GENERATOR generates various ephemerides by precessing and nutating the values found on the JPL ephemeris. ORB1 CONVERSION converts an IBM 360 system 9-track tape to the same format on a 7-track tape. TdF TABLE GENERATOR generates tabular information for use with subroutine TdF to compute time differences between systems A.1 and UT1.

Detailed descriptions of the formats of the data tapes are found in Appendix C of Volume III -- GEODYN SYSTEM OPERATIONS DESCRIPTION.
1.2.1 DODS SORT-MERGE

INTRODUCTION

The DODS SORT-MERGE program sorts data from DODS format data tapes by satellite identification numbers into chronological, station and then measurement type order, eliminating duplicate data records.
DESCRIPTION

The main program SRTMRG sorts and merges blocks of 250 sorted records which are obtained from the subroutine RDNSRT. The blocks are sorted onto two scratch disk units, which are then merged and sorted again onto two alternate scratch disk units. The process is repeated until all the records are sorted by satellite identification number and in chronological order. Then the subroutine WRITE is called to write out the data records onto a tape.
MAIN - DODS SRTMRG

PURPOSE
SORTS AND MERGES TWO INPUT DATA TAPES ONTO ONE TAPE

SUBROUTINES USED
NSORT WRITE

COMMON BLOCKS
OSORT UNITS

INPUT FILES
NONE

OUTPUT FILES
NONE

SCRATCH FILES
UNITS - 20, 21, 22, 23

RESTRICTIONS
NONE

REFERENCES
NONE

COMMON/OSORT/AC*G,09(14,250),0821(14,250,2)
COMMON/UNITS/AIN,NOUT,UNIT
DOUBLE PRECISION OB,OB1,LAST(4),EOF
INTEGER NN(2),NT(2),UNIT(2,2)
INTEGER FLIP
EQUIVALENCE (N1,NN(1)),(N2,NN(2))
LOGICAL FRSTIM,LSTPAS,REV,MERGE
DATA FRSTIM,LSTPAS,REV,TRUE*2,N,FAIL$/
* IOU,10H,INH/2,1,2/.
* NSTRING,EOF/1,99999999/.
* FLIP(I)=MOD(I,2)+1

C INITIALIZE INPUT, OUTPUT, AND SCRATCH UNITS
AIN=10
NOUT=11
K=19
DO 5 J=1,2
DO 5 J=1,2
K=K+1
5 UNIT(I,J)=K

C READ AND SRT 250 RECORDS
10 CALL RNSRT
II=1
IF(NO.EQ.0) GG TO 80
IF(FRSTII) GO TO 50

C TEST ORDER OF STRINGS
20 DO 30 I=1,4
30 CONTINUE
IF(II.EQ.KO) GO TO 70
II=II+1
GO TO 20

C FLIP TO ALTERNATE SCRATCH UNIT
40 IOU=FLIP(IOU)
NSTRING=NSTRING+1
C SAVE LAST TIME POINT
50 DO 60 I=1,4
60 LAST(I)=CE(I,NC)
      M=NO-1+1
      IU=UNIT(I,IOH)

C WRITE BLOCK OF RECORDS ON SCRATCH UNIT
WRITE(IU),((CH(I,J),I=1,14),J=11,N))
      FRSTIM=.FALSE.

C TEST IF MORE STRINGS
70 IF(NO .GE. 250).GO TO 10
60 M=0
   DO 90 K=1,2
      IU=UNIT(K,IOH)
   C WRITE EOF AND REWIND
      WRITE(IU),((EOF,I=1,14),J=1,M)
      ENDFILE IU
      REWIND IU

C INITIALIZE SCRATCH UNITS
IU=UNIT(K,INH)
90 REWIND IU
      IOH=INH
      INH=FLIP(INH)
      IU=1
      LSTPAS=NSTRN+LE*2
      PRINT 101,NSTRN
      NSTRN=1
      FRSTIM=.TRUE.
      IOU=1
      IU=UNIT(K,INH)
C READ NEW STRING
READ(IU),((OBI(I,J,K),I=1,14),J=1,M)
100 NN(K)=1
      MERGE=.FALSE.
      IF(NN(K)+NN(2) .NE. 0).GO TO 200
      PRINT 1000
      STOP
200 IF(MERGE).GO TO 230
C SET INDEX FOR FILE TO PROCESS
      IT=1
   DO 210 I=1,4
      IF(OBI(I,1,1)-OBI(I,N2,2))230,210,220
210 CONTINUE
   GO TO 305
220 IT=2
230 N=NN(IT)
      IF(FRSTIM).GO TO 285
C TEST IF RECORDS IN TIME ORDER
   DO 240 I=1,4
      IF(OBI(I,N1,1)-OBI(I,N0))250,240,260
240 CONTINUE
   GO TO 305
250 MERGE=.NOT.MERGE.AND.NN(1).NE.0.AND.NN(2).NE.0
C FLIP INPUT SCRATCH UNIT INDICATOR
      IF=FLIP(IT)
C TEST IF ALTERNATE SCRATCH UNIT IS IN ORDER
   DO 260 I=1,4
IF(MERGE) GO TO 230
REV= .TRUE.
GO TO 270
C TEST IF BUFFER IS FULL
260 IF(NO.LT.250) GO TO 290
C TEST IF LAST STRING IS PROCESSED
270 IF(LSTPAS) GO TO 260
IU=UNIT(IU,ICH)
C WRITE OUTPUT BUFFER
WRITE(IU,NO,((GO(I,J),I=1,14),J=1,N))
NO=0
C TEST IF NEW OUTPUT STRING
IF(.NOT.REV) GO TO 290
REV= .FALSE.
C FLIP OUTPUT INDICATOR
IOU=FLIP(IU)
NSTRING=NSTRING+1
FSTIM=.TRUE.
GO TO 260
C WRITE OUTPUT STRING ON TAPE
280 CALL WRITE(*.FALSE.)
285 NO=0
290 NO=NO+1
FSTIM= .FALSE.
GO 300 I=1,14
C PUT RECORD IN BUFFER
300 IU=UNIT(I1,I1H)
C READ NEW RECCFD FROM SCRATCH UNIT
READ(IOU,((OE1(I,J,IT),I=1,14),J=1,M))
M(IT)=M
M(IT)=1
IF(MERGE.AND.NT(IT).NE.0) GO TO 250
IF(N(1)+N(2).NE.0) GO TO 290
IF(LSTPAS) GO TO 320
IU=UNIT(IU,ICH)
C WRITE BUFFER ON SCRATCH UNIT
WRITE(IU,NO,((OBS(I,J),I=1,14),J=1,N))
GO TO 80
C WRITE OUTPUT TAPE
320 CALL WRITE(*.FALSE.)
CALL WRITE(*.TRUE.)
1000 FORMAT(* IND SORT INPUT*)
1010 FORMAT(* *100 STRINGS*)
STOP
END
B

No more records on unit

YES

Other records out of time order

NO

Both units at end of file

YES

LAST STRING

NO

WRITE OUT BUFFER ON SCRATCH UNIT

WRITE LAST STRING ON TAPE

WRITE END FILE TAPE

STOP

320

250

200

320

80
DESCRIPTION

RDNSRT reads a DODS data tape, sorting each record until a block of 250 records is filled and checking the satellite identification number. If the satellite identification numbers are not the same, a new record is read. When either a block of 250 is reached or the end of the tape is reached, control is returned to SRTMRG.
NAME: DODS RDSRST

PURPOSE: READS AND Sorts 250 RECORDS INTO CORE

CALLING SEQUENCE: CALL RDSRST

SUBROUTINES USED: NONE

COMMON BLOCKS: OSORT UNITS

INPUT FILE: DODS INPIT TAPE

OUTPUT FILES: NONE

RESTRICTIONS: TAPE MUST BE IN DODS FORMAT

REFERENCES: NONE

SUBROUTINE RDSRST
COMMON/OSCRT/HG,OB(14,250)
COMMON/UNITS/NIN,NOUT,SCR(4)
DOUBLE PRECISION OB,SAVE,D
INTEGER*2 ID(4)
EQUIVALENCE (C,ID)
N=0
10 N=N+1
C READ A RECORD
15 READ(NIN,ERR=100,END=80)OB(2,N),OB(3,N),(OB(I,N),I=5,8),ISATID.
* (OB(I,N),I=9,14)
   IF(OB(2,N).LT.0.) GO TO 15
   CB(1,N)=ISATID
   D=OB(13,N)
   GB(4,N)=IC(3)
C IF FIRST REAC, GO READ another
20 IF(N.LT.2) GO TO 10
   DO 30 J=2,N
   JN=N+2-J1
C CHECK TIME ORDER OF RECORDS AND SATELLITE ID NUMBER
30 DO 25 K=1,4
   IF(OB(K,N)-OB(K,J-1))30,25,40
25 CONTINUE
GO TO 15
30 CONTINUE
J=1
40 IF(J.EQ.N) GO TO 70
   J1=N-J
C ARRANGE RECORDS IN ORDER
50 DO 60 K=1,14
   SAVE=OB(K,N)
   DO 50 M=1,J1
   IF=M-H1
   GO TO 50
   GB(K,H+1)=OB(K,N)
50 CONTINUE
60      CO(K,J) = SAVE
70      IF(N.LT.250) GO TO 10
90      IF ARKAY IS FULL, RETURN
          RETURN
80      N = N - 1
          RETURN
100     READ(NIN,ERR=100,END=80)
          GO TO 15
          END
DESCRIPTION

The subroutine WRITE is the output routine of the program. If WRITE is called with a false logical argument, it will write out a block of data records on the output tape. If the satellite identification number changes, it will write out a flagged record with the new identification number. If WRITE is called with a true logical argument, it will write an endfile on the output tape. A flowchart would be superfluous.
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

NAME: DODS WRITE

PURPOSE: WRITES DATA RECORDS AND LDFILES TAPES

CALLING SEQUENCE: CALL WRITE(ENDSW)

SYMBOL TYPE DESCRIPTION
ENDW  L TRUE WILL ENDFILE WHEN TAPE IS COMPLETELY WRITTEN

SUBROUTINES USED: NONE

COMMON BLOCKS: OSORT, UNITS

INPUT FILES: NONE

OUTPUT FILE: MAGNETIC TAPE

RESTRICTIONS: NONE

REFERENCES: NONE

SUBROUTINE WRITE(ENDSW)
COMMON/OSCRT/NT,G,OB(14,250)
DOUBLE PRECISION OB
COMMON/UNIT$/IN,NOUT,SCR(4)
INTEGER ZERO/0/,JSATID/-1/
REAL*8 FLAG/-1.,DO/
LOGICAL ENDSW

C IF REQUESTED, ENDFILE TAPE ONLY
IF(ENDSW) GO TO 20
DO 10 N=1,NT
   JSATID=OE(1,N)+5
   IF(ISATID,NE,JSATID)WRITE(NOUT),FLAG,(ZERO,1=1,10),ISATID,
      (ZERO,1=1,12)
   JSATID=ISATID
C IF SATELLITE ID NUMBERS ARE THE SAME, WRITE OUT THE RECORD
10 WRITE(NOUT),OB(2,N),OB(3,N),(OB(1,N),I=5,8),ISATID,
      (OE(I,N),I=5,14)
   RETURN
20 ENDFILE NCTL
REWIND NCTL
RETURN
END

1-223
1.2.2 GEOS SORT-MERGE

INTRODUCTION

The GEOS SORT-MERGE program sorts data from GEOS format data tapes into chronological, station, and then measurement type order, eliminating duplicate data records.
DESCRIPTION

The main program SRTMRG sorts and merges blocks of 250 sorted records which are obtained from the subroutine RDNSRT. The blocks are sorted onto two scratch disk units which are then merged and sorted again onto two alternate scratch disk units. The process is repeated until all the records are in chronological order. This program does not sort by satellite identification number. Then the subroutine WRITE is called to write out the data records onto a tape.
NAME MAIN = GEOS SRTMRG

PURPOSE SORTS AND MERGES THE INPUT DATA TAPES ONTO ONE TAPE

SUBLRoutines UESD RDISRT WRITE

CCRMCA BLOCKS OSORT UNITS

INPUT FILES NONE

OUTPUT FILES NONE

SCRATCH FILES UNITS = 20,21,22,23

RESTRICTIONS NONE

REFERENCES NONE

COMMON/OSCRT/NO,OB1(19,250),OB2(19,250),2)
COMMON/UNITS/NIN,NDMT,UNIT
INTEGER CE,OB1,NN(2),NT(2),LAST(6),UNIT(2,2),EOF
INTEGER FLIP
EQUIVALENCE (N1,NN(1)),(N2,NN(21))
LOGICAL PRSTIM,LSITAS,REV,MERGE
DATA FRSTIM,LSITAS,REV//TRUE,,,2*FALSE//
* IOU,IOH,NIN/2*1,2/*
* NSTRTD,EOF/1,99999999/
FLIP(I,J)=PCD(I,J)+1
C INITIALIZE INPUT, OUTPUT, AND SCRATCH UNITS
NIN=10
NOUT=11
2 K=19
DO S I=1,2
DO 5 J=1,2
K=K+1
REWIND K
5 UNIT(I,J)=K
REWIND NIN
C READ AND SORT 250 RECORDS
10 CALL RDISRT
11 I=1
IF(NO*EOC) GG TO 80
IF(FRSTIM) GO TO 50
C TEST ORDER OF STRINGS
20 DO 30 I=1,3
IF(LAST(I)~OC(I,11))50,30,40
30 CONTINUE
IF(I,1*EO,M0) GO TO 70
I=I+1
GO TO 20
C FLIP TO ALTERNATE SCRATCH UNIT
40 IOU=FLIP(IOU)
NSTRNG=NSRNG+1
DO 60 I=1,3
C SAVE LAST TIME POINT
LAST(I)=CE(I,N)
K=NO-11+1
IU=UNIT(I(U,1CH))
C WRITE BLOCK OF RECORDS ON SCRATCH UNIT
WRITE(IU),((CB(I,J),I=1,19),J=1,N))
FRSTIM=.FALSE.
C TEST IF MORE STRINGS
IF(N0.GE.150) GO TO 10
80 M=0
CO 90 K=1,2
IU=UNIT(K,1CH)
C WRITE EOF AND REIND
WRITE(IU),((EOF,I=1,19),J=1,M)
ENDFILE IU
REIND IU
C INITIALIZE SCRATCH UNITS
IU=UNIT(K,1NH)
90 REMIND IU
IH=INH
INH=FLIP(INH)
IU=1
LSTRNG=NSRNG.LE.2
PRINT 777,NSTRNG
777 FORMAT( 4*** NUMBER OF STRINGS=",14)
NSTRNG=1
FRSTIM=.TRUE.
DO 100 K=1,2
IU=UNIT(K,1NH)
C READ NEW STRING
READ(IU),((OBI(I,J,K),I=1,19),J=1,M)
NT(K)=M
100 NN(K)=1
MERGE=.FALSE.
IF(NT(1)+NT(2).NE.0) GO TO 200
PRINT 1000
STOP
200 IF(MERGE) GO TO 230
C SET INDEX FOR FILE TO PROCESS
IT=1
DO 210 1=1,3
.CODI (I1,1.I )-OB1(IN1,NO I120,210,220
210 CONTINUE
GO TO 305
220 IT=2
230 N=NN(IT)
IF(FRSTIM) GO TO 285
C TEST IF RECORDS IN TIME ORDER
DO 240 1=1,3
IF(OBI(I,N,I)-OBI(I,N2,2)230,210,220
240 CONTINUE
GO TO 305
250 MERGE=.NOT.,MERGE. AND NT(1).NE.0 AND NT(2).NE.0
C FLIP INPUT SCRATCH UNIT INDICATOR
IT=FLIP(IT)

C TEST IF ALTERNATE SCRATCH UNIT IS IN ORDER
IF(MERGE) GO TO 230
REV=.TRUE.
GO TO 270

C TEST IF BUFFER IS FULL
260 IF(N0.LT.250) GO TO 290

C TEST IF LAST STRING IS PROCESSED
270 IF(LSTPAS) GO TO 290

IU=UNIT(IU,ICH)

C WRITE OUTPUT BUFFER
WRITE(IU,NO.,((OB(I,J),I=1,N0),J=1,N0)
NO=0

C TEST IF NEW OUTPUT STRING
IF(*NOT.REV) GO TO 290
REV=.FALSE.

C FLIP OUTPUT INDICATOR
IU=FLIP(IU)
ASTRING=NSTRING+1
FRSTIN=.TRUE.
GO TO 200

C WRITE OUTPUT STRING ON TAPE
280 CALL WRITE(.FALSE.)
285 NO=0
290 NO=NO+1
FRSTIN=.FALSE.

C PUT RECORD IN BUFFER
DO 300 I=1,N0
300 OB(I,N0)=OB(I,IT)
305 IF(NN(IT).EQ.NT(IT)) GO TO 310
NN(IT)=NN(IT)+1
GO TO 200

310 IU=UNIT(IU,I1,INH)

C READ NEW RECORD FROM SCRATCH UNIT
READ(IU,INO.,((OB(I,J,IT),I=1,N0),J=1,M)
NT(IT)=M
NN(IT)=1
IF(MERGE.AND.NT(IT).EQ.0) GO TO 250
IF(NN(IT).EQ.NT(2).NE.0) GO TO 200
IF(LSTPAS) GO TO 320
IU=UNIT(IU,ICH)

C WRITE BUFFER ON SCRATCH UNIT
WRITE(IU,NO.,((OB(I,J,IT),I=1,N0),J=1,N0)
GO TO 80

C WRITE OUTPUT TAPE
320 CALL WRITE(.FALSE.)
CALL WRITE(.TRUE.)

1000 FORMAT(*IND SORT INPUT*)
STOP
END
80

WRITE END OF FILE ANDREWIND

INITIALIZE SCRATCH UNITS

SET INDEX FOR FILE TO PROCESS

RECORDS IN TIME ORDER

250

FLIP INPUT SCRATCH UNIT INDICATOR

ALTERNATE SCRATCH UNIT IN ORDER

270

RECORDS IN TIME ORDER

260

MORE ROOM IN BUFFER

290

260

270

230
DESCRIPTION

RDNSRT reads a GEOS data tape, sorting each record into a block of 250 records. When either a block is full or the end of the tape is reached, control is returned to SRTMRG.
NAME GEOS RDNSRT

PURPOSE TO READ AND SORT 250 RECORDS INTO CORE

CALLING SEQUENCE CALL RDNSRT

SUBROUTINES USED NONE

COMMON BLOCKS OSORT UNITS

INPUT FILE GEOS INPUT TAPE

OUTPUT FILES NONE

RESTRICTIONS TAPE MUST BE IN GEOS FORMAT AND CONTAIN DATA FROM
THE SAME SATELLITE

REFERENCES NONE

SUBROUTINE RDNSRT
COMMON/OSORT/N,OB(19,250)
COMMON/UNIT5/NIN,NOUT,SCR(4)
INTEGER CE,SAVE

READ A RECORD
READ(NIN,10000,END=80,ERR=15)GE(4,N),OB(5,N),M TYPE,ID,OB(6,N),
+ OB(7,N),ISANG,OB(1,N),OB(2,N),OB(1,N),I=0,15

OB(3,N)=ISTAN+100+M TYPE*10+9-ID

IF FIRST REAC, GO READ ANOTHER
IF(N.LT.2) GO TO 10
DO 30 J=2,N
J=N+2-J1

IF ARRAY IS FULL, RETURN
RETURN

C CHECK TIME ORDER OF RECORDS
DO 25 K=1,3
IF(OB(K,N)-OB(K,J-1))30,25,40

CONTINUE
GO TO 15

CONTINUE
J=1

IF(N.EQ.1) GO TO 70
J=N-J

C ARRANGE RECORDS IN ORDER
DO 60 K=1,19
SAVE=OB(K,N)
DO 50 M=1,J1
K=N-M1

50 OB(K+1)=SAVE
60 OB(K,J)=SAVE
70 IF(N.LT.250) GO TO 10

C IF ARRAY IS FULL, RETURN
RETURN

1-234
80  A=N-1
RETURN
10000  FORMAT (A4,A2,2I1,A4,A1,2I5,11X,A4,A2)
END
DESCRIPTION

The subroutine WRITE is the output routine of the program. If WRITE is called with a false logical argument it will write out a block of data records on the output tape. If it is called with a true logical argument, it will write an endfile on the output tape. A flowchart would be superfluous.
NAME GEOS WRITE
PURPOSE WRITES DATA RECORDS AND ENDFILES TAPES
CALLING SEQUENCE CALL WRITE(ENDSW)
SYMBOL TYPE DESCRIPTION
ENDSW L TRUE WILL ENDFILE WHEN TAPE IS COMPLETELY WRITTEN
SUBROUTINES USED NONE
COMMON BLOCKS OSORT UNITS
INPUT FILES NONE
OUTPUT FILE MAGNETIC TAPE
RESTRICTIONS NONE
REFERENCES NONE

SUBROUTINE WRITE(ENDSW)
COMMON/OSORT/N, OR(19, 250)
INTEGER CE
COMMON/UNITS/NIN, NOUT, SCR(4)
LOGICAL ENDSW
C IF REQUESTED, ENDFILE TAPE ONLY
IF(ENDSW) GO TO 20
D0 12 N=1, NT
ISTAND=OE(3, N)/100
ITEMP=OE(2, N)-ISTAND*100
MTYPE=ITEMP/10
ID=9-(ITEMP-MTYPE*10)
C WRITE OUT THE RECORD
10 WRITE(NOUT, 1CC0) OB(4, N), OB(5, N), MTYPE, ID, OB(6, N), OB(7, N),
* ISTAND, OB(1, N), OB(2, N), OB(3(1, N), 1=9, 19)
12 CONTINUE
RETURN
20 ENDFILE ACUT
RETURN
10000 FORMAT(A4, A2, 211, A4, A1, 15, 216, 11A4, A2)
END
1.2.3 EPHEMERIS TAPE GENERATOR

INTRODUCTION

The ephemeris tape contains the following ephemerides:

- geocentric lunar positions at half day intervals
- heliocentric positions of the earth-moon barycenter at 4-day intervals
- heliocentric positions of the planets, Venus, Mars, Jupiter and Saturn at 4-day intervals
- nutation in obliquity at half day intervals

The ephemerides are obtained by precessing and nutating to true of date coordinates the values found on the JPL planetary ephemeris tape.
The positions of the Earth-moon barycenter and the planets are heliocentric on the JPL tape; however, the moon is geocentric. Subroutine READE uses Everett's 5th order interpolation formula which is written as follows:

\[
y(t_j + sh) = P(s) = y_j F_0(1-s) + d_j^2 F_2(1-s) + d_j^4 F_4(1-s) + y_{j+1} F_0(s) + d_{j+1}^2 F_2(s) + d_{j+1}^4 F_4(s)
\]

where

\[
F_0(s) = s
\]

\[
F_2(s) = [(s-1) (s) (s+1)]/6
\]

\[
F_4(s) = [(s-2) (s-1) (s) (s+1) (s+2)]/120
\]

\[
d_j^2, d_j^4\] are the second and fourth modified central differences contained on the JPL tape.

\[
y_j, j=1, 2, \ldots, n\] denotes successive tabular values of one of the quantities contained in the ephemeris.

\[
t_j\] denotes the corresponding time points.

\[
h = t_{j+1} - t_j
\]

\[
s = \frac{t - t_j}{h}, t\] is time at which information is requested.
All the coordinates are converted to geocentric positions and placed in common. The MAIN program, by calling subroutine EQUATR, precesses and nutates the coordinate system from Mean of 1950 to True of Date. The planets are reconstructed by MAIN to heliocentric positions, while the moon remains geocentric. The sun is converted to Earth-moon barycentered positions. The second and fourth modified central differences are then recomputed. These differences and the positions are written on the tape.

The modified second and fourth differences for Everett interpolation are computed as follows:

\[
\begin{align*}
\delta_j^2 &= \delta_j^2 + a_{26} \delta_j^6 + a_{28} \delta_j^8 \\
\delta_j^4 &= \delta_j^4 + a_{46} \delta_j^6 + a_{48} \delta_j^8
\end{align*}
\]

where

\[
\begin{align*}
a_{26} &= -0.013120 & a_{28} &= 0.004299 \\
a_{46} &= -0.278269 & a_{48} &= 0.068489
\end{align*}
\]

and the ordinary central differences are defined:

\[
\begin{align*}
\delta_j^0 &= y_j \\
\delta_j^1 &= \delta_j^0 - \delta_{j-1}^0 \\
\delta_j^2 &= \delta_j^1 - \delta_{j-0.5}^1 \\
\delta_j^3 &= \delta_j^2 - \delta_{j-0.5}^2 \\
\delta_j^4 &= \delta_j^3 - \delta_{j-0.5}^3 \\
\end{align*}
\]

etc.
## SUBROUTINE CROSS REFERENCE CHART

### Calling Routines

<table>
<thead>
<tr>
<th></th>
<th>MAIN</th>
<th>DATES</th>
<th>DIFF</th>
<th>DJUL</th>
<th>EQUATR</th>
<th>MATRIX</th>
<th>NUTATE</th>
<th>PRECES</th>
<th>READE</th>
<th>YMDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDYMD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQUATR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETTAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULMAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUTATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRECES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>READE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROTMAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RYMDI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YMDAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COMMON BLOCK CROSS REFERENCE

<table>
<thead>
<tr>
<th>COMMON BLOCKS</th>
<th>ROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>CETBL1</td>
<td></td>
</tr>
<tr>
<td>CETBL2</td>
<td></td>
</tr>
<tr>
<td>CETBL3</td>
<td></td>
</tr>
<tr>
<td>CETBL4</td>
<td></td>
</tr>
<tr>
<td>CETBL5</td>
<td></td>
</tr>
<tr>
<td>CETBL9</td>
<td></td>
</tr>
<tr>
<td>REC1</td>
<td></td>
</tr>
<tr>
<td>REC2</td>
<td></td>
</tr>
<tr>
<td>TAPE</td>
<td></td>
</tr>
</tbody>
</table>

1-243
DESCRIPTION

The MAIN program, EPHEM, reads the JPL ephemeris tape for positions of the Earth-moon barycenter, the planets and the moon. All the coordinates are converted to geocentric positions and placed in common. MAIN, by calling subroutine EQUATR, precesses and nutates the coordinate systems from Mean of 1950 to True of Date. The planets are reconverted by MAIN to heliocentric positions while the moon remains geocentric. The sun is converted to Earth-moon bary-centered positions. The second and fourth modified central differences are then recomputed. These differences and the positions are written on the tape.
**NAME**  | MAIN - EPHEM
---|---
**PURPOSE** | GENERATES HELIOCENTRIC POSITIONS OF VENUS, MARS, JUPITER, SATURN AND THE EARTH-MOON NODYCENC, GEOCENTRIC LUNAR POSITIONS AND NUTATIONS IN DILIGUITY
**SUBROUTINES USED** | CLEAR, DATES, EQUAT, READ, TDIF, YMDAY
**COMMON BLOCKS** | CETBL1, CETBL2, CETBL3, CETBL4, TAPE
**INPUT FILES** | 5 - READER
**OUTPUT FILES** | 6 - PRINTER
**REFERENCES** | JPL DEVELOPMENT EPIERERIS INO. 10 TECHNICAL REPORT 32-1181 - C.J. DEVINE JPL, CALIF. INST. OF TECH., PASADENA, CALIF. NOV. 15, 1967

```
REAL*3 JD1, DJ, SEC1, DELJD, DELSEC, AU, REM, TPD, EMAT, TABOUT
REAL*4 SUN, NUT, NUTATE, J2O, TSEC, SOLARS(3,51,10), BASE, DJBASE,
2BUF1(3,3,17), BUF3(3,3,5), GOUTE, TAB3, BUF3(133), BUF3(135).
3DAYEND, DAY, YMDAY, A(4), 06, 09
EQUIVALENCE (BUF3(1,1,11), (BUF3(1,1,1)), (BUF3(1,1,11), BUF2(1))
REAL*8 FACTOR
DATA FACTOR/1.215037316452D-02/
DATA ITIME/1/
DATA NUTO/6/
DATA V20(10)/11, 10, 2, 4, 5, 6, 7, 8, 9, 1/
REAL NUTAT
LOGICAL TIN

DIMENSION NUTATE(A1), BUF2(3,17)
INTEGER RECORD/0/
INTEGER OUT/10/, 1Y/50/
DATA NSODY,NSTS/10,16/
DATA A/-1.2242990-3, -2.782685-1, 6.94992-2/
DATA JD1, SEC1, DELJD, DELSEC/35, 19, 6, 500, 0, 000, 0, 500, 0, 000/
DATA DJBASE, DAYEND/2433281, 500, 2440791, 500/
DATA TIN/SHA FALSE/
E(X)=(1.4990197-21E-0+0214A10-15)X+6.21979599-9)*X+4.09310760
READ 1003, TIMES
JD1=TIMES(1TIME)
DAYEND=TIMES(1TIME+1)
IN=12
```

**Page 2 of 5**
30 September 1972
ICW=1
NCENTR=3
AU=1.455973011
RF=6176.149500
TD=4.6404
TSFC=0
EMPAT=61.30200
CALL CLEAR(1,REQ+13,1)
DO 600 I=1,NEO
II=1EO(I)
605 IREQ(I)=1
IREQ(I13)=1
CALL READE(JD1,TSEC,IERR)
JD1=TA93(1)
JED=JD1
C DAYS = 5 1990.0
JD1=JD1-24515020.000
DJ=JED+14.000
C 1950.0
BASE=YMDAY(500100,0,00)
DOUT=BASE+JED-DJ
C SET UP ARRAY FIRST TIME
DO 606 I=1,81
CALL READE(JED,TSEC,IERR)
C PRECESS AND NUTATE
CALL EQUAT(^SUN,BASE,TIN,SOLARS(1,1,1),DOUT,.TRUE.,1EQ,NEO)
C SUBTRACT VECTOR TO SUN FROM PLANETS
DO 900 J=1,NEO
DO 900 L=1,3
SOLARS(L,1,J)=SOLARS(L,1,J)-SOLARS(L,1,2)
C SUBTRACT EARTH-MOON BARYCENTER FROM VECTOR TO SUN
DO 910 J=1,3
SOLARS(J,1,2)=SOLARS(J,1,2)-FACTOR*SOLARS(J,1,1)
610 CONTINUE
NUTATE(I)=5COS(E(JD1)+NUT(2))*NUT(I)
DOUT=DOUT+DELJD
JD1=JD1+DELJD
JED=JED+DELJD
606 CONTINUE
GO TO 610
C READ ONE RECORD
620 CONTINUE
DO 621 I=66.81
CALL READE(JFD,TSEC,IERR)
C PRECESS AND NUTATE
CALL EQUAT(\SUN,BASE,TIN,SOLARS(1,1,1),DOUT,.TRUE.,1EQ,NEO)
C SUBTRACT VECTOR TO SUN FROM PLANETS
DO 900 J=1,NEO
DO 900 L=1,3
SOLARS(L,1,J)=SOLARS(L,1,J)-SOLARS(L,1,2)
C SUBTRACT EARTH-MOON BARYCENTER FROM VECTOR TO SUN
DO 910 J=1,3
SOLARS(J,1,2)=SOLARS(J,1,2)-FACTOR*SOLARS(J,1,1)
910 CONTINUE
NUTATE(I)=5COS(E(JD1)+NUT(2))*NUT(I)
DOUT=DOUT+DELJD
C COMPUTE DIFFERENCES

610 DO 608 J=1,17
II=II+32
BUF(2,1,1)=NUTATE(II)
BUF(2,2,1)=NUTATE(II-1)+NUTATE(II+1)-2.00*NUTATE(II)
BUF(3,1,1)=NUTATE(II+2)+NUTATE(II-2)-4.00*NUTATE(II+1)
\(-4.00\times NUTATE(II+1) + 6.00 \times NUTATE(II)\)
D6=NUTATE(II-3)+NUTATE(II+3)-6.00*(NUTATE(II-2)+NUTATE(II+2))
\(-15.00 \times NUTATE(II+1) + 20.00 \times NUTATE(II)\)
D8=NUTATE(II-4)+NUTATE(II+4)-6.00*(NUTATE(II-3)+NUTATE(II+3))
\(+26.00\times NUTATE(II+1) + 20.00 \times NUTATE(II)\)
D2=NUTATE(II+1)+70.00*NUTATE(II)
BUF(2,2,1)=BUF(2,2,1)+A(1)*D6+A(2)*DB
BUF(2,3,1)=BUF(2,3,1)+A(3)*D6+A(4)*DB
DO 608 J=1,3
BUF(1,1,1)=SOLARS(J,II,1)
BUF(2,1,1)=SOLARS(J,II-1,1)+SOLARS(J,II+1,1)-2.00*SOLARS(J,II,1)
BUF(3,1,1)=SOLARS(J,II+2,1)+SOLARS(J,II-2,1)-4.00*SOLARS(J,II+1,1)
\(-4.00 \times SOLARS(J,II+1,1) + 6.00 \times SOLARS(J,II,1)\)
D6=SOLARS(J,II-3,1)+SOLARS(J,II+3,1)-6.00*(SOLARS(J,II-2,1)+SOLARS(J,II+2,1))
\(+5.00 \times SOLARS(J,II+1,1) + 6.00 \times SOLARS(J,II,1)\)
D8=SOLARS(J,II-4,1)+SOLARS(J,II+4,1)-6.00*(SOLARS(J,II-3,1)+SOLARS(J,II+3,1))
\(+30.00 \times SOLARS(J,II+1,1) + 70.00 \times SOLARS(J,II,1)\)
BUF(2,2,1)=BUF(2,2,1)+A(1)*D6+A(2)*DB
BUF(2,3,1)=BUF(2,3,1)+A(3)*D6+A(4)*DB
DO CONTINUE

608 CONTINUE
DO 611 K=1,3
II=(II-1)*8+33
DO 611 K=2,60
DO 611 K=1,3
BUF3(1,1,1)=SOLARS(J,II,K)
BUF3(2,1,1)=SOLARS(J,II-1,1)+SOLARS(J,II+1,1)-2.00*SOLARS(J,II,K)
\(+4.00 \times SOLARS(J,II+1,K) \times SOLARS(J,II,K)\)
BUF3(3,1,1)=SOLARS(J,II+2,1)+SOLARS(J,II-2,1)-4.00*SOLARS(J,II+1,1)
\(+14.00 \times SOLARS(J,II+1,1) + 15.00 \times SOLARS(J,II,K)\)
D6=SOLARS(J,II-2,1)+SOLARS(J,II+2,1)-6.00*(SOLARS(J,II-1,1)+SOLARS(J,II+1,1))
\(+23.00 \times SOLARS(J,II+1,1) + 20.00 \times SOLARS(J,II,K)\)
D8=SOLARS(J,II-3,1)+SOLARS(J,II+3,1)-6.00*(SOLARS(J,II-2,1)+SOLARS(J,II+2,1))
\(+30.00 \times SOLARS(J,II+1,1) + 70.00 \times SOLARS(J,II,K)\)
BUF3(2,2,1)=BUF3(2,2,1)+A(1)*D6+A(2)*DB
BUF3(3,2,1)=BUF3(3,2,1)+A(3)*D6+A(4)*DB

611 CONTINUE
C WRITE OUTPUT
DAY=DOJD+DOJBASE+BASE
DAY=DAY+64E4-32.1576+64E4
CALL DATES(DAY,YY,IIY,IIYM,IIYH,SEC)
WRITE(OUT) IIYH,IIYM,IIY,SEC,BUF(1,1)=1,1,27)
WRITE(OUT) BUF(1,1)=1,51)
WRITE(OUT) BUF(1,1)=52,102)
C SHUFFLE BACK VALUES

DO 630 I=1,16
   NUTATE(I)=NUTATE(11)
DO 630 J=1,7
DO 630 K=1,NEO
   SOLARS(J,K)=SOLARS(J,11+K)

CONTINUE
   IF(ICW.EQ.0) GO TO 100
   IF(OJ.GE.DAYEND) GO TO 100
   GO TO 620

C TEST TO SEE IF LAST TAPE HAS BEEN READ

100 IF(IN.EQ.14) GO TO 200
   ITIME=ITIME+12.
   IF(TIMES(ITIME).LE.0.) GO TO 200
   DAYEND=TIMES(ITIME+1)
   REWIND IN
   IN=IN+1
   ICW=1
   GO TO 620

200 WRITE(6,1000) DJ
   IYMD=0
   WRITE(OUT) IYMD,IHM,SEC,BUF2.(BUFF(I),I=1,27)
   WRITE(OUT) (BUFF(I),I=1,51)
   WRITE(OUT) (BUFF(I),I=52,102)
   WRITE(OUT) (BUFF(I),I=103,153)
   WRITE(OUT) (BUFF(I),I=28,81)
   WRITE(OUT) (BUFF(I),I=82,135)
   END FILE OUT
   REWIND OUT
   REWIND IN
   STOP

1000 FORMAT('EPHEMERIS TAPE GENERATION COMPLETE! LAST DATE',G25.16)
1001 FORMAT('B1/B1/G25.16')
1002 FORMAT('B10/5G25.16')
1003 FORMAT(6D12.6)
END
DESCRIPTION

(See GEODYN)
DESCRIPTION

(See GEODYN)
DATES

DESCRIPTION

DATES converts a number of days elapsed from Jan 0.0 of the arc reference year into a three-word date of the form: YYMMDD, HHMM, SEC.
NAME: DATES

PURPOSE: CONVERTS DAYS ELAPSED FROM JAN 0,0 OF THE REFERENCE YEAR INTO A 3-YEAR DATE OF THE FORM: YYYY, HHMM, SEC

CALLING SEQUENCE: CALL DATES(DAYN, IY, IYMD, IHM, SEC)

SYMBOL DESCRIPTION
DAYN OP INPUT - DAYS ELAPSED FROM JAN 0,0 OF THE REFERENCE YEAR
IY I INPUT - 1950 REPRESENTED BY THE LAST TWO DIGITS IN THE FORM YY
IYMD I OUTPUT - YEAR, MONTH, DAY IN THE FORM YYYYDD
IH M I OUTPUT - HOUR, MINUTES IN THE FORM HHMM
SEC R OUTPUT - SECONDS

SUBROUTINES USED: ADDYNDF

COMMON BLOCKS: NONE

INPUT FILES: NONE

OUTPUT FILES: NONE

SUBROUTINE: DATES(DAYN, IY, IYMD, IHM, SEC)

DOUBLE PRECISION DAYN, IY, IYMD, IHM, SEC

C NUMBER OF DAYS FROM JAN 1 OF REFERENCE YEAR
DAY = DAYN + 630 + 4/6/6404
I DAY = I DAY - 1.

C NUMBER OF DAYS FROM JAN 1 OF THE REFERENCE YEAR
IYMD = IYMD + 100/3601

C CALCULATE YEAR, MONTH, DAY OF INTEREST
CALL ADOYK(IYMD, I DAY)

C CALCULATE THE NUMBER OF SECONDS REMAINING
S = 6.6254*(DAY-FLOAT(I DAY+1))
SEC = S

C CONVERT TO HOUR, MINUTE FORMAT
IH M = 60*(SEC/3600) + SEC/60

C REMAINING SECONDS
SEC = SEC - FLOAT(60*(SEC/3600)) + SEC/60

RETURN
END

DATE 34
DATE 35
DATE 36
DATE 37
DATE 28
DATE 39
DATE 40
DATE 41
DATE 42
DATE 43
DATE 44
DATE 45
DATE 46
DATE 47
DATE 48
DATE 49
DATE 50
DATE 51

1-252
DESCRIPTION

DIFF calculates the difference in days and seconds between any two time points in the 20th century.
**NAME**

DIFF

**PURPOSE**

CALCULATES THE DIFFERENCE BETWEEN ANY TWO TIME POINTS IN THE 20TH CENTURY

**CALLING SEQUENCE**

CALL DIFF(IYMD1, IHMS1, IYMD2, IHMS2, IDAY, ISEC)

**SYMBOL** | **TYPE** | **DESCRIPTION**
---|---|---
IYMD1 | I | INPUT - FIRST DATE IN THE FORM YYMMDD
IHMS1 | I | INPUT - TIME ON IYMD1 IN THE FORM HMMSS
IYMD2 | I | INPUT - SECOND DATE IN THE FORM YYMMDD
IHMS2 | I | INPUT - TIME ON IYMD2 IN THE FORM HMMSS
IDAY | I | OUTPUT - ELAPSED FULL DAYS DIFFERENCE
     |     | IDAY IS NEGATIVE IF IYMD2, IHMS2 IS THE EARLIER TIME
ISEC | I | OUTPUT - REMAINDER OF DIFFERENCE IN SECONDS
     |     | ISEC HAS THE SAME SIGN CONVENTION AS IDAY

**SUBROUTINES USED**

PYMDI

**COMMON BLOCKS**

MONTHS

**INPUT FILES**

NONE

**OUTPUT FILES**

NONE

**SUBROUTINE**

DIFF(IYMD1, IHMS1, IYMD2, IHMS2, IDAY, ISEC)

DIMENSION MONTH(13, 2)
DATA MONTH/O, 360, 9) / 21, 152.1, 2, 213.24, 3, 274, 305, 335, 366, 5, 366.

ISUB1(IY) = MINO(MOD(IY, 4), 1)
ISEC = 0

IF(IYMD1.EQ.IYMD2) GOTO 4000
CALL PYMDI(IYMD1, IY1, IMI1, ID1)
CALL PYMDI(IYMD2, IY2, IM2, ID2)
L1 = ISUB1(IY1)
IYEAR1 = 36525*(IY1-1)/100+MONTH(IM1, L1)+ID1
L2 = ISUB1(IY2)
IYEAR2 = 36525*(IY2-1)/100+MONTH(IM2, L2)+ID2
ISEC = (IYEAR2-IYEAR1)*86400

4000 ISEC1 = IHMS1/40*(IHMS1/100) - 2400 - (IHMS1/10000)
     ISEC2 = IHMS2/40*(IHMS2/100) - 2400 - (IHMS2/10000)
     ISEC = ISEC1 + ISEC2 - ISEC1
     IDAY = ISEC/86400
     ISEC = ISEC - IDAY*86400
     RETURN

END

1-254
DESCRIPTION

DJUL computes the Julian date for a time input in days from Jan. 0.0 of the reference year.
NAME     DJUL

PURPOSE  TO COMPUTE JULIAN DATE FOR AN INPUT TIME IN DAYS FROM JAN C.C. OF THE REFERENCE YEAR FOR THE ARC

CALLING SEQUENCE  X=DJUL(DAY)

SYMBOL TYPE DESCRIPTION
DAY     DF  INPUT — TIME IN DAYS FROM JAN C.C. OF THE REFERENCE YEAR
DJUL   DF  OUTPUT — JULIAN DATE

SUBROUTINE  NEED YMDAY

COMMON BLOCKS  NONE

INPUT FILES  NONE

OUTPUT FILES  NONE

RESTRICTIONS  NONE

REFERENCES  NONE

DOUBLE PRECISION FUNCTION DJUL(DAY)
REAL*8, DJ, DAY, YMDAY
LOGICAL NOTIST/.FALSE./
IF(NOTIST) GO TO 10
NOTIST=.TRUE.
DJ=2433201.5D0-YMDAY(56111010, C.)
10 DJUL=DJ+DAY
RETURN
END
DESCRIPTION

(See GEODYN)
DESCRIPTION

EQUATR rotates a set of vectors from mean or true equator and equinox of one epoch to mean or true equator and equinox of another epoch.
NAME EQUATR

PURPOSE TO ROTATE A SET OF VECTORS FROM MEAN OR TRUE EQUATOR AND EQUINOX OF ONE EPOCH TO MEAN OR TRUE EQUATOR AND EQUINOX OF ANOTHER EPOCH

CALLING SEQUENCE CALL EQUATR(X, DIN, TIN, Y, DOUT, TOUT, IEO, NEO)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>DP</td>
<td>INPUT - SET OF VECTORS TO BE ROTATED</td>
</tr>
<tr>
<td>DIN</td>
<td>DP</td>
<td>INPUT - DAY NUMBER OF THE COORDINATES SINCE JAN 0.0 OF THE REFERENCE YEAR</td>
</tr>
</tbody>
</table>
| TIN    | L    | INPUT - TYPE OF INPUT  
  *TRUE* = TRUE COORDINATE SYSTEM  
  *FALSE* = MEAN COORDINATE SYSTEM |
| Y      | DP   | OUTPUT - ROTATED SET OF VECTORS |
| DOUT   | DP   | OUTPUT - DAY NUMBER OF OUTPUT VECTOR SET SINCE JAN 0.0 OF THE REFERENCE YEAR |
| TOUT   | L    | INPUT - TYPE OF OUTPUT  
  *TRUE* = TRUE COORDINATE SYSTEM  
  *FALSE* = MEAN COORDINATE SYSTEM |
| IEO    | I    | INPUT - INDICATES WHICH MEMBERS OF THE SET ARE TO BE ROTATED |
| NEO    | I    | INPUT - NUMBER OF MEMBERS OF THE SET TO BE ROTATED |

SUBROUTINES USED NUTATE PRECES

COMMON BLOCKS NONE

INPUT FILES NONE

OUTPUT FILES NONE

SUBROUTINE EQUATR(S, DIN, TIN, Y, DOUT, TOUT, IEO, NEO)  
REAL*8 X(6,1), Y(3,1), NP(3*3,4), T(3), DIN, DOUT, TEMP  
DIMENSION IEO(1)  
LOGICAL TIN, TOUT  
M=2  
IF(.NOT.TIN) GO TO 10  
M=1  
C OBTAIN MATRIX TO NUTATE FROM TRUE TO MEAN OF INPUT EPOCH  
CALL NUTATL(DIN, NP(1,1,1))  
C OBTAIN MATRIX TO PRECES FROM INPUT EPOCH TO 1950  
10 CALL PRECES(DIN, NP(1,1,2))
CALL PRECESS(DOUT,NP(1,1,3))
N=3
IF(.NOT.TJUT) GO TO 20
N=6

CALL MUTATE(DOUT,NP(1,1,4))

CALL NUTATE(DOUT,NP(1,1,3))

DO 30 J=1,3
DO 30 K=3,N
NP(I,J,K)=NP(J,K)
30 NP(I,J,K)=NP(I,J,K)

DO 70 II=1,NEO
JJ=I=0(I1)
LL=(II-1)*81+1
DO 40 I=1,3
40 Y(I,LL)=X(I,JJ)

DO 60 K=M,N
60 Y(I,LL)=0.00

DO 60 J=1,3
DO 60 I=1,3
60 T(I)=Y(I,LL)+NP(I,J,K)*T(J)

Y(I,LL)=Y(I,LL)+NP(I,J,K)*T(J)

RETURN
END
DESCRIPTION

GETTAP obtains a Julian date through common block CETBL9 and then reads the JPL ephemeris tape one record at a time. The record containing the information desired is loaded into common block CETBL3.
TAPE  GETTAP

PURPOSE  READS JPL TAPE ONE RECORD AT A TIME GIVEN A JULIAN DATE

CALLING SEQUENCE  CALL GETTAP

SUBROUTINES USED  NONE

COMMON BLOCKS  CETBL2 CETBL3 CETBL9 REC1 REC2

INPUT FILES  IN - JPL EPHEMERIS TAPE

OUTPUT FILES  NONE

RESTRUCTIONS  NONE

REFERENCE  JPL DEVELOPMENT EPHEMERIS NUMBER 19
        TECHNICAL REPORT 32-1131 - C.J. DEVINE
        JPL, CALIF. INST. OF TECH., PASADENA, CALIF.
        NOV. 15, 1967

EXTERNAL GETTAP
COMMON/CETBL2/ICW, ICENT, IREQ(12)  GETT 27
COMMON/CETBL3/TAB3(329), NUTAT, CKSUM  GETT 28
COMMON/CETBL9/JD1, TCAY, JDIF, IERF1  GETT 29
COMMON/REC1/REC2(24)  GETT 30
COMMON/REC2/TBODY, TYPE, AJC, BJD, STEP, DUM20(20)  GETT 31
COMMON/TAPE/IN  GETT 32
REAL REC2(25)  GETT 33
DOUBLE PRECISION TAB3, CJ, JD1, TCAY, JDIF  GETT 34
EQUIVALENCE (REC2(1), TBODY)  GETT 35
IF (ICW .NE. 1) CJ = 1, CD20  GETT 36
IERF1 = 0  GETT 37
JDIF = JD1 - CJ  GETT 38
IF (JD1 .LE. CJ + 8, CDG) GO TO 100  GETT 39
IF (JD1 .LE. CJ) RETURN  GETT 40
REWIND IN  GETT 41
READ(IN) REC1  GETT 42
READ(IN) REC2  GETT 43
READ(IN) TAB3, NUTAT, CKSUM  GETT 44
ICW = 2  GETT 45
CJ = TAB3(1)  GETT 46
JDIF = JD1 - CJ  GETT 47
IF (JD1 .LE. CJ + 8, CDG) GO TO 100  GETT 48
IF (JD1 .LE. CJ) RETURN  GETT 49
PRINT 26C, JD1, DJ, IN  GETT 50
PRINT 3GC, TAB3, NUTAT, CKSUM  GETT 51
STOP 516  GETT 52
READ(IN, END=150) TAB3, NUTAT, CKSUM  GETT 53
CJ = TAB3(1)  GETT 54

1-262
JDIF=JD1-CJ
IF(JULGE;DJ+9.5DC) GO TO 100
RETURN
150 PRINT 250;JD1;DJ+IN
PRINT 350;TAJ;NUTAT;CKSUM
STOP 51316
200 FORMAT('11***** DATA REQUESTED AT JULIAN DATE ',G16.9,' SMALLER ')
   ' THAN FIRST DATE ',G16.9,' ON INPUT UNIT ',I3)
250 FORMAT('11***** DATA REQUESTED AT JULIAN DATE ',G16.9,' GREATER ')
   ' THAN LAST DATE ',G16.9,' ON INPUT UNIT ',I3)
300 FORMAT(I4C/(5G23.16))
END
DESCRIPTION

MATRIX calls subroutine NUTATE to find the nutation matrix for a time specified in the calling sequence. It then multiplies the nutation matrix by another 3x3 matrix passed in through the calling sequence.
NAME: MATRIX

PURPOSE: MULTIPLIES TWO 3X3 MATRICES

CALLING SEQUENCE: CALL MATRIX(DAY,A,B)

SYMBOL    TYPE   DESCRIPTION
DAY       DP      INPUT - TIME OF NUTATION MATRIX
A          DP      OUTPUT - PRODUCT OF THE TWO MATRICES
B          DP      INPUT - MATRIX TO BE MULTIPLIED BY THE NUTATION MATRIX

SUBROUTINES USED: NUTATE

COMMON BLOCKS: NONE

INPUT FILES: NONE

OUTPUT FILES: NONE

RESTRICTIONS: NONE

REFERENCES: NONE

SUBROUTINE MATRIX(DAY,A,B)
DOBLE PRECISION A(3,3),B(3,3),CT(3,3),DAY
CALL NUTATE(DAY,CT)
DO 10 I=1,3
DO 10 J=1,3
A(I,J)=0.000
DO 10 K=1,3
10 A(I,J)=A(I,J)+B(I,K)*CT(J,K)
RETURN
END
DESCRIPTION

(See GEODYN)
DESCRIPTION

(See GEODYN)
DESCRIPTION

(See GEODYN)
DESCRIPTION

READE interpolates the JPL ephemeris quantities to find values on the date and time specified in the calling sequence. It then performs coordinate transformations as specified in common block CETBL2 and unit transformations as specified in common block CETBL1.
SUBROUTINE READ(JED,TSEC,IERR)
JED CALLING MESA SCIENTIFIC CORP. 1965 SEPT 15
CRED LASSON: JPL, 1965 MAR 17.
READ JPL EPHemeris AT THE JULIAN EPHemeris DATE
GIVEN BY (JED+TSEC/66400.D0)

** ITEMS COMMUNICATED THROUGH THE CALLING SEQUENCE **

JED REFERENCE JULIAN EPHemeris DATE.
TSEC SECONDS OF EPHemeris TIME PAST JED.
ANY COMBINATION OF VALUES OF JED AND TSEC
IS ACCEPTABLE AS LONG AS (JED+TSEC/36400.D0)
IS WITHIN THE RANGE OF THE EPHemeris TAPE
BEING USED, HOWEVER TO OBTAIN THE
FINEST POSSIBLE RESOLUTION IN INTERPOLATION
THE NUMBER JED MUST BE AN EXACT MACHINE
NUMBER, FOR EXAMPLE JED COULD BE A DATE ENDING
WITH .0 OR .5.
IERR ERROR FLAG
1=NEG ERROR
2=(JED+TSEC/36400.D0) LESS THAN FIRST DATE
ON TAPE
3=(JED+TSEC/36400.D0) GREATER THAN LAST DATE
ON TAPE
4=ICENm NOT IN THE RANGE 1 THRU 11
5=ICW IS NOT 1, 2, OR 3

** THE FOLLOWING ITEMS ARE INPUT THROUGH COMMON **

* COMMON BLOCK CETBL1 *

AU A*U EXPRESSED IN DESIRED OUTPUT UNITS
RE EQUATORIAL RADIUS OF EARTH IN DESIRED OUTPUT UNITS
TPD DESIRED NUMBER OF TIME UNITS PER DAY
EMRAT EARTH MOON MASS RATIO: SUGGESTED VALUE=31.3

SUGGESTED VALUES FOR AU AND RE DEPEND UPON
DESIZED OUTPUT UNITS AS FOLLOWS:
FOR OUTPLT IN EARTH RADIUS AU=2.245,73400122511700, RE =1.D9
FOR OUTPLT IN KILOMETERSAU=1.49593546.D0, RE =6378.16900
FOR OUTPLT IN A.U. AU=1.0, FE =2.63522711150350%.5

SET TPD=86400.D0 FOR VELOCITY IN LINEAR UNITS PER SECOND.
SET TPD= 1.D0 FOR VELOCITY IN LINEAR UNITS PER DAY.

* COMMON BLOCK CETBL2 *

ICW FLAG INDICATING STATUS OF COMMON BLOCKS REC2 AND CETBL3
1 MEANS NEITHER BLOCKS CONTAIN VALID DATA
2 MEANS BOTH BLOCKS CONTAIN VALID DATA
3 MEANS REC2 IS VALID, CETBL3 IS NOT.
USER MUST SET ICW=1 BEFORE INITIAL CALL

ICENTR SPECIFIES CENTRAL BODY FOR COORDINATE
TRANSLATION AS FOLLOWS:
1:MEC 5 JUP 9 PLUTO
2:VENUS 6 SAT 10 SUN
3 EARTH  7 URANUS 11 MOON
4 MARS   8 NEP

IEOE(J)
IEOE(J) SPECIFIES OUTPUT DESIRED FOR
BODY NO. J.
IEOE(J)=7 NO OUTPUT
1 POSITION
2 POSITION AND VELOCITY
J RUNS FROM 1 TO 11 AS FOLLOWS:
1 MERC  5 JUP  9 PLUTO
2 VENUS 6 SAT 10 SUN
3 EARTH 7 URANUS 11 MOON
4 MARS  8 NEP 12 EARTH-MOON-BARYCENTER
13 NUTATION

* COMMON BLOCK CETBL3 *
TABS 329 DOUBLE PREC. WORD BUFFER TO ACCOMODATE J.D. AND EPHEMERIS.
NUTAT 254 SINGLE PREC. WORD BUFFER TO ACCOMODATE NUTATION DATA.
CKSUM 1 SINGLE PREC. WORD FOR CHECKSUM.
** THE FOLLOWING ITEMS ARE OUTPUT THROUGH COMMON **
* COMMON BLOCK CETPLA *
TABOUT(I,J) PLANETARY AND LUNAR OUTPUT, SCALED AND
TRANSLATED WITH RESPECT TO CENTER.
TABOUT(I,J) CONTAINS OUTPUT FOR
BODY NO. J. (1..IE. J-1-12)
The index I IDENTIFIES COMPONENTS AS FOLLOWS:
1=X 2=Y 3=Z
4=XDOT 5=YDOT 6=ZDOT
NUT1(I) NUTATION OUTPUT
NUT1(1)=DELTA LONGITUDE
NUT1(2)=DELTA DELICUITY
NUT1(3)=TIME DERIVATIVE OF NUT1(1)
NUT1(4)=TIME DERIVATIVE OF NUT1(2)

* COMMON BLOCK CETBL9 *
BIVECT(I,J) WORKING ARRAY. CONTENTS ARE INTERPOLATED
AND SCALED BUT NOT TRANSLATED. 1ST INDEX RUNS
OVER X,Y,Z,XDOT,YDOT,ZDOT AS IN TABOUT
BUT 2ND INDEX IS DIFFERENT AS FOLLOWS:
EDDIES 1 THRU 9 ARE HELIOCENTRIC.
1 MERC  5 JUP  9 PLUTO
2 VENUS 6 SAT 10 MOON REL TO EARTH
3 EARTH 7 URANUS 11 EARTH REL TO EARTH
4 MARS  8 NEP 12 EARTH-MOON REL TO MOON
13 SEE 4092+

THE COMMON BLOCK 'CETBL9' IS FOR COMMUNICATION
BETWEEN REEPR2 AND GETR2.

SUBROUTINE READE(JEC,TSEC,IERR)
COMMON /CETBL1/ AU.RE, TPD, EMAT
COMMON /CETBL2/ ICW, ICENT, IEOE(13)
COMMON /CETBL3/ TAH(3825), NUTAT(254), CKSUM
COMMON /CETBL4/ TABOUT(6,12), NUT1(4)
COMMON /CETBL5/ BIVECT(6,13)
COMMON /CETBL9/ JD1, TDAY, JDIF, IERR
CATA FAC/1.157474.747407405/
CATA STP/2.8*4.2*5/
CATA MC1/15.15.6*15.10.5/
CATA M1/11.11.11.D12.0.12.0.3.3.3.6.0.6.13.0.0/}
CATA
* IPOS /02.092.146.209.254.353.362.316.470.524.1/
* IVEL /47.119.173.272.281.338.389.443.457.677.103/
JDI=JED
TDAY=TSECFAC
CALL GETTAP
IF(IEK1 NE.0) GO TO 5000
IF(IEKT NE.12)GO TO 10
ER=4
GO TO 5000
10 CONTINUE
SET JREG(1) TO CONTROL INTERPOLATION
DO 20 1=1,10
IF(IREQ(1) .GE. 0 .AND. IREQ(1) .LE. 2) GO TO 20
IRE=1
GO TO 5000
20 JREQ(I)=IRE(1)
BARYCENTER FLAG
JREQ(3)=IREQ(12)
MXPX=JREQ(1)
DO 24 1=2,10
24 MXPX=MAX(MAXP,MREQ(1))
MXPX=MAX(MAXP,MREQ(3),MREQ(11))
MXPX=MAX(MAXP,MAXK)
MXPX=MAX(MAXP,MAXK)
IF(ICEKT .EQ. 3 .OR. ICENT .EQ. 11) GO TO 28
CENTER IS NOT EARTH OR MOON
10=M0ON,3=EARTH
32 JREQ(10)=MAXE
JREQ(3)=MAXI(JREQ(3),MAXE)
JREQ(11)=MAXALL
GO TO 32
CENTER IS EARTH OR MOON
10=M0ON,3=EARTH
32 JREQ(10)=MAXALL
JREQ(3)=MAXP
32 JREQ(11)=IREQ(13)
LUNAR=JREQ(10)
IBARY=JREG(3)*3
JREQ( ) IS NOW SET
SAVE=0.
DO 240 IECDY=1,11
IF(JREQ(I IODY)) 240.240,40
40 IF(ISTP(I IODY)) .EQ. SAVE) GO TO 165
SAVE=STEP(I IODY)
READ 112
READ 113
READ 114
READ 115
READ 116
READ 117
READ 118
READ 119
READ 120
READ 121
READ 122
READ 123
READ 124
READ 125
READ 126
READ 127
READ 128
READ 129
READ 130
READ 131
READ 132
READ 133
READ 134
READ 135
READ 136
READ 137
READ 138
READ 139
READ 140
READ 141
READ 142
READ 143
READ 144
READ 145
READ 146
READ 147
READ 148
READ 149
READ 150
READ 151
READ 152
READ 153
READ 154
READ 155
READ 156
READ 157
READ 158
READ 159
READ 160
READ 161
READ 162
READ 163
READ 164
READ 165
READ 166
READ 167
160 TEMP=JULF/SAVE
  KK=TCMP
  U(1,1)=TEMP-FLUAT(KK)
  IF(U(1,1))IC1,IC5,161
161  CONTINUE:
  U(2,1)=1+U(1,1)
  DO 163 IC=1,2
  U(1,1)=U(1,1)+U(1,1)
  U(1,2)=((U(1,3)-1.DO)**2/8.DO)
  U(1,3)=((U(1,3)-4.DO)**2/8.DO)
163  IF(U(1,3)<-200) GO TO 170
  IC=IC+1
  CONTINUE:
164  C=AU
165  GO TO 172
166  C=AU
INTERPOLATE IBOODY=1:2,....,10
172  IGrt1=1POS(IBOODY)+KK*9
    IC1=1:
200  CONTINUE:
    IF(U(1,1))221,203,222
203  IGrt2=IGrt1+6
    GO TO 201
201  IC2=IC1+2
    GO 203
    IC1=IC1+1
204  GO TO 205
205  CONTINUE:
    JR=9(IBOODY)=JRED(IBOODY)+1
    IF(JRED(IBOODY)) 240,240,240
240  CONTINUE:
    IC1=IC1+4
    C=C/TPD
    GO TO 206
207  CONTINUE:
    IGrt1=IVEL(IBOODY)+KK*9
    IC1=4:
    C=C/TPD
    GO TO 206
INTERPOLATE IBOODY=11 NUTATION
220  C=1.DO
    IGrt1=IPOS(IBOODY)+KK*6
    IC1=1
221  IGrt2=IGrt1+3
222  IF(U(1,1))223,226,228
225  DC 23U IGrt1,IGrt2,3
    NUT(IC1)=
      * C=(U(2,1)+(NUTAT(IGrt1)))
      * U(2,2)+(NUTAT(IGrt1+1))
      * U(2,3)+(NUTAT(IGrt1+2))
      * U(1,1)+(NUTAT(IGrt1+3))
      * U(1,2)+(NUTAT(IGrt1+4))
      * U(1,3)+(NUTAT(IGrt1+5))
230  IC1=IC1+1
GO TO 222
222 DO 227 ICET=ICET1,1GET2,3
READ 224
227 IC1=IC1+1
READ 225
232 CONTINUE
READ 226
JEC0(JECGY)=JEC0(1BODY)-1
READ 227
IF(JEC0(1BODY)) 247,249,236
READ 228
IC1=ICET
READ 229
GO TO 222
READ 230

CONTINUE
INTERPOLATION IS FINISHED
READ 231
RESULTS ARE IN BIVECT ( ) AND NUT ( )
READ 232
IF(LUNAR ) 420,422,4013
READ 233
TEST MOON REQUEST
READ 234
NOTE..EMRAT=EARTH MASS/MOON MASS
READ 235
SET BIVECT ( ,11)=ERTH MN CENTERED AT EARTH
READ 236
SET BIVECT ( ,12)=ERTHMN CENTERED AT MOON
READ 237
4010 RAT=1.0/(EMRAT+1.0)
READ 238
IMAX=LUNAR + 3
READ 239
DO 4016 I=1,IMAX
READ 240
BIVECT(I,11)=RAT*BIVECT(I,10)
READ 241
4016 BIVECT(I,12)=-EMRAT*BIVECT(I,11)
READ 242
IF(IREQ(1BODY)) 410,410A,410C
READ 243
KASE=KCENT+KREQ(1BODY)
READ 244
KI=M1(KASE)
READ 245
GO TO (4122,4106,4062,4063,4064,4028,4029,4030)
READ 246
KASE=15
READ 247
BEGIN TRANSLATION LOOP
READ 248
4328 K1=IBODY
READ 249
KASE=1.2.6.11
READ 250
DO 4336 I=1,IMAX
READ 251
4336 TAHOUT(I,1BODY)=BIVECT(I,K1)
READ 252
GO TO 4106
READ 253
KASE=2.8.14
READ 254
DO 4044 I=1,IMAX
READ 255
4044 TAHOUT(I,1BODY)=0.0
READ 256
GO TO 4106
READ 257
KASE=4.5
READ 258
K2=3
READ 259
GO TO 4160
READ 260
KASE=5
READ 261
L2=11
READ 262
GO TO 4060
READ 263
KASE=10
READ 264
L2=12
READ 265
K1=1BODY
READ 266
K2=13
READ 267
1-274
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

C C G T U 4052
C KASE=19
C 4064 K1=ICENT
C C KASE=7
C 4068 LD 4072 I=1,I*MAX
C 4072 TAOOUT(I,IBODY)=BIVECT(I,K1)
C GO TO 4100
C KASE=12,13
C 4076 K2=KASE-1
C GO TO 4100
C KASE=22
C 4084 K1=IBODY
C KASE=16
C 4088 K2=ICENT
C GO TO 4100
C KASE=17,18
C 4088 L2=ICENT
C K2=KASE-6
C 4092 IF(FLAG) GO TO 4100
C FLAG=.TRUE.
C BIVECT(1,13) IS AN AUXILIARY VECTOR
C NEEDED WHEN KASE=5,16,17,18.
C FOR KASE=5 BIVECT(1,13)=EARTH CENTERED AT SUN
C FOR KASE=10 BIVECT(1,13)=MOON CENTERED AT SUN
C FOR KASE=17,18 BIVECT(1,13)=ERTHNN CENTERED AT ICENT
C G T U 4096 I=1,IBARY
C 4096 BIVECT(I,13)=BIVECT(I,3)-BIVECT(I,L2)
C 4100 DO 4104 I=1,I*MAX
C 4104 TAOOUT(I,IBODY)=BIVECT(I,K1)-BIVECT(I,K2)
C 4106 CONTINUE
C 5000 ICEN=1ERR1
C RETURN
C END
DESCRIPTION

(See GEODYN)
DESCRIPTION

RYMDI separates a six digit number representing a date in the form YYMMDD into three two digit numbers representing the year, month, and day.
NAME                   RYMDI
PURPOSE               TO SEPARATE PACKED SIX-DIGIT DECIMAL DATES INTO
                      TWO-DIGIT YEAR, MONTH, AND DAY
CALLING SEQUENCE     CALL RYMDI(YMD,Y,M,D)
SYMBOl TYPE           DESCRIPTION
YMD                I   INPUT - DATE TO BE SEPARATED
Y                 I   OUTPUT - TWO-DIGIT YEAR
M                 I   OUTPUT - TWO-DIGIT MONTH
D                 I   OUTPUT - TWO-DIGIT DAY
SUBROUTINES USED      NONE
COMMON BLOCKS        NONE
INPUT FILES           NONE
OUTPUT FILES         NONE
RESTRICTIONS         NONE
REFERENCES           NONE

SUBROUTINE RYMDI (YMD,Y,M,D)  
INTEGRATE YMD,Y,M,D
Y = YMD/10000
I = YMD/100
M = I - Y*100
D = YMD - I*100
RETURN
END

RYMD 34
RYMD 35
RYMD 36
RYMD 37
RYMD 38
RYMD 39
RYMD 40
RYMD 41
DESCRIPTION

YMDAY is a real valued DOUBLE PRECISION function used to compute from a given date and time the number of days from January 0.0 of a given reference year.
NAME: YMDAY

PURPOSE: Given a date, computes the number of days from January 0, 0 of the reference year for the arc.

CALLING SEQUENCE: X = YMDAY(YYMD, IHM, SEC)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>DP</td>
<td>Output - number of days from January 0, 0 of the reference year.</td>
</tr>
<tr>
<td>YYMD</td>
<td>I</td>
<td>Input - date in the form YYMD.</td>
</tr>
<tr>
<td>IHM</td>
<td>I</td>
<td>Input - hours and minutes in the form HHMM.</td>
</tr>
<tr>
<td>SEC</td>
<td>R</td>
<td>Input - seconds.</td>
</tr>
</tbody>
</table>

SUBROUTINES USED: ADDYMD

COMMON BLOCKS: NONE

INPUT FILES: NONE

OUTPUT FILES: NONE

DOUBLE PRECISION FUNCTION YMCAY(YYMD, IHM, SEC)

LOGICAL NOTIST

DATA NOTIST = .FALSE.

IF (NOTIST) GO TO 10

NOTIST = .TRUE.

IY = (YYMD / 10000) + 10000 + 101

10

IHM = IHM * 100

CALL JIFF(IY, 0, YYMD, IHM, IH, IS)

YYDAY = 86400 * (IY + 1) + IS

YYDAY = (YYDAY + SEC) / 8.6444

RETURN

END
1.2.4 ORB1 CONVERSION

INTRODUCTION

The ORB1 CONVERSION program is used to convert a 9-track 360 double-precision ORB1 tape to a 7-track 7094 single-precision ORB1 tape.

The main routine reads in IBM 360 double-precision words and writes on a 7-track tape the equivalent IBM 7094 single-precision words.

The subroutine WORD94 does the conversion from the IBM 360 64-bit floating point format to the IBM 7094 36-bit floating point format.
DESCRIPTION

The main program for ORB1 reads a block, converts each double precision word in the block to the IBM 7094 single precision format using subroutine WORD94, and then outputs the converted block. This procedure continues until all blocks on the input tape have been processed.
NAME          MAIN - OR81 CONVERSION
PURPOSE       TO CONVERT A 5-TRACK IBM 367 FORMAT OR81 TAPE TO
               7-TRACK IBM 7094 FORMAT
SUBROUTINE USED WORD94
COMPILED BLOCKS NONE
INPUT FILE    IN - FORTRAN LOGICAL UNIT NUMBER FOR INPUT TAPE
OUTPUT FILE   OUT - FORTRAN LOGICAL UNIT NUMBER FOR OUTPUT TAPE
RESTRICTIONS NONE
REFERENCES    GSFC ORBIT TAPE - FORMAT 1

LOGICAL*1 BUF(6,350)                              CR81 21
REAL*8 CELF(350)                                  OR81 22
INTEGER IN/OUT/END/OUT/11/                       OR81 23
C READ EACH RECORD
10 READ(IN,END=30) DBUF                            OR81 24
C CALL WORD94 IC CONVERT EACH INPUT WORD TO OUTPUT FORMAT
DO 20 I=1,350                                       OR81 25
20 CALL WCRC64(DEUF(I),BUF(1,I))                  OR81 26
C OUTPUT RECORD
   WRITE(CUT,100) BUF                               OR81 27
   GO TO 10                                        OR81 28
C END FILE OUTPUT TAPE AND TERMINATE
30 ENDFILE CUT                                    OR81 29
STOP                                                OR81 30
1000 FORMAT(21)11                                  OR81 31
END                                                 OR81 32

1-283
DESCRIPTION

Subroutine WORD94 converts a word in 64 bit IBM 360 floating point format to 36 bit IBM 7094 floating point format.

The order of computation is as follows:

- Bits 8-38 (the fraction) of the 360 word are extracted and placed in bits 2-31 of an integral word (NUM).

- The sign (bit 0) and exponent (bits 1-7) are extracted and stored as integer.

- $40_{16}$ is subtracted from the exponent and the result multiplied by 4 to change to base 2.

- Bits 30-28 are sequentially tested for non zero to obtain a normalization count, N.
N is added to the exponent and the fraction (NUM) is shifted right 4-N bits.

The fraction is then stored six bits at a time from the right (bits 76-31, 20-25 into the output characters (6,5...1).

In 2d character WORD94 stores the low order 3 bits of the exponent and bits 28-30 of the fraction.

In the 1st character WORD94 stores the high order bits of the exponent and the sign.
NAME  WORD94

PURPOSE  TO CONVERT FROM 360, 36 BIT FORMAT TO 7094, 36 BIT FORMAT (FLOATING POINT)

CALLING SEQUENCE  CALL WORD94(W360,W94)

SYMBOL TYPE DESCRIPTION
W360  L*1  INPUT - 360 DOUBLE PRECISION WORD

W94  L*1  OUTPUT - 7094 SINGLE PRECISION WORD

SUBROUTINES USED  NONE

COMMON BLOCKS  NONE

INPUT FILES  NONE

OUTPUT FILES  NONE

RESTRICTIONS  NONE

REFERENCES  IBM 360 AND 7094 PRINCIPLES OF OPERATION MANUALS

SUBROUTINE WORD94(W360,W94)
LOGICAL*1 W360(8),W94(6),L(4),L

EQUIVALENCE (L1,IN),(L1(4),L)

C BIT CONTAINS ALL INTEGER POWERS OF 2 WHICH FIT IN INT 4 WORD

INTEGER EIT(32)/Z00000001,Z00000002,Z00000004,Z00000008,
   Z00000010,Z00000020,Z00000040,Z00000080,
   Z00000100,Z00000200,Z00000400,Z00000800,
   Z00001000,Z00002000,Z00004000,Z00008000,
   Z00010000,Z00020000,Z00040000,Z00080000,
   Z00100000,Z00200000,Z00400000,Z00800000,
   Z01000000,Z02000000,Z04000000,Z08000000,
   Z10000000,Z20000000,Z40000000,Z80000000/)

C EXTRACT MANTISSA (FRACTIONAL PART)

IN=0
L=W360(5)
NUM=IN/4
L=W360(4)
NUM=NUM+IN*BIT(7)
L=W360(3)
NUM=NUM+IN*BIT(15)
L=W360(2)
NUM=NUM+IN*BIT(23)

C EXTRACT SIGN BIT
L=W360(1)
ISGN=IN*BIT(8)

C EXTRACT EXPONENT

EXP=(IN-ISGN*BIT(0)-BIT(6))*BIT(3)
A=0
TEST FOR ZERO WORD

1-286
IF (IEXP.GT.0) GO TO 10
IEXP=0
NUM=0

C ADJUST EXPONENT FOR BINARY NORMALIZATION INSTEAD OF HEX NORMALIZATION
10 GO 20 L=1:3
IF(NUM.GE.BIT(31-1).OR.IEXP.EQ.0) GO TO 30
IEXP=IEXP-1

20 N=N+1

C SHIFT MANTISSA TO ACCOUNT FOR NORMALIZATION
30 NUM=NUM/BIT(4-N)
C OUTPUT LOW MANTISSA
DO 40 L=1,5
IN=NUM
#94(7-L)=L
40 NUM=NUM/EIT(7)

C PUT LOW ORDER THREE BITS OF EXPONENT IN WITH FIRST 3 BITS OF MANTISSA
IEXP=IEXP+SIGN*BIT(9)
IN=IN+IEXP*BIT(4)
#94(2)=L
C INSERT HIGH ORDER BITS OF EXPONENT AND SIGN
IN=IEXP/EIT(4)
#94(1)=L
C ALL DONE
RETURN
END
1.2.5 TDIF TABLE GENERATOR

INTRODUCTION

The TDIF TABLE GENERATOR generates tabular differences between time systems A.1 and UT1. It reads as input the differences between systems UT1 and UTC which are obtained from B.I.H. Using the differences between A.1 and UTC computed by subroutine TDIF in conjunction with the difference between UT1 and UTC, the TDIF TABLE GENERATOR computes the differences between A.1 and UT1.

Continual maintenance is required to keep these tables up-to-date.
**SUBROUTINE CROSS REFERENCE CHART**

<table>
<thead>
<tr>
<th>CALLED ROUTINES</th>
<th>MAIN</th>
<th>DJUL</th>
<th>TDIFF</th>
<th>YMDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDIFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YMDAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*DJUL is an entry point in DPFCT.*

**COMMON BLOCK CROSS REFERENCE CHART**

<table>
<thead>
<tr>
<th>COMMON BLOCKS</th>
<th>ROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAIN</td>
</tr>
<tr>
<td>CONSTS</td>
<td></td>
</tr>
<tr>
<td>CSTHET</td>
<td></td>
</tr>
<tr>
<td>CTIME</td>
<td></td>
</tr>
<tr>
<td>INITBK</td>
<td></td>
</tr>
<tr>
<td>MONTHS</td>
<td></td>
</tr>
</tbody>
</table>
DESCRIPTION

The MAIN routine reads the UT1-UTC time differences and uses subroutine TDIF to obtain the A.1-UTC differences. Then MAIN subtracts these differences [(A.1-UTC) - (UT1-UTC)] to obtain the time differences between systems A.1 and UT1 (A.1-UT1).
NAME           MAIN - TDFGEN

PURPOSE        COMPUTES TIME DIFFERENCES BETWEEN A1 AND UT1

SUBROUTINES USED DJUL TIF YMDAY

COMMON BLOCKS  CTIME INITBK

INPUT FILE     5 - CARD INPUT

OUTPUT FILES   6 - PRINTER
                7 - PUNCHES CARDS

RESTRICTIONS  NONE

REFERENCES     NONE

REAL*8 YMDAY, DAY, DJUL, DJ

DIMENSION A1UTC(1000)  TDFG 21
COMMON/CTIME/CUM(22), IYREF  TDFG 22
COMMON/INITBK/NOTIST(57)  TDFG 23
DO 5 I=1,57
 5 NOTIST(I)=0  TDFG 24
  IYREF=66  TDFG 25
  NUM=0  TDFG 26
  HM=0  TDFG 27
  SEC=0  TDFG 28
10 READ(5,1000,END=100) IYMD,UTIUTC
   DAY=YMDAY(IYMD, HM, SEC)  TDFG 29
   A1UTC=TDF(4,3,DAY)  TDFG 30
   TAIUTC=TDF(4,1,DAY)  TDFG 31
   NUM=NUM+1  TDFG 32
   CJ=DJUL(IY)
   A1UTC(NUM)=A1UTC-UTIUTC  TDFG 33
   IF(MOD(NUM,50).EQ.1) PRINT 230C  TDFG 34
   IF(MOD(NUM,5).EQ.0) PRINT 205  TDFG 35
   PRINT 201C, IYMD, HM, SEC, DJ, A1UTC, UTIUTC, A1UTC(NUM), TAIUTC  TDFG 36
   GO TO 10  TDFG 37
100 PUNCH 300C,(A1UTC(I),I=1,NUM)  TDFG 38
   PRINT 3CCC,(A1UTC(I),I=1,NUM)  TDFG 39
   STOP 41  TDFG 40
1000 FORMAT(16,F10.5)  TDFG 41
2000 FORMAT('1YMMDD HHHMM SS.SSSS JULIAN A1-UTC UTI-UTC ', TDFG 42
   ' A1-UTC',4X,'A1-UTC TDF1')  TDFG 43
2005 FORMAT(1X)  TDFG 44
2010 FORMAT(1X,15,15,F8.4,2X,F11.1,2X,F7.4,F9.4,F3.4,3F11.4)  TDFG 45
3000 FORMAT(5X,1H1,1X,F7.4,1H1,,F7.4,1H1,,F7.4,1H1,,F7.4,1H1,)  TDFG 46
   * F7.4,1H1,,F7.4,1H1,,F7.4,1H1,,F7.4,1H1,)  TDFG 47
END  TDFG 48
DESCRIPTION

(See GEODYN)
DESCRIPTION

The block data routine initializes values for $\pi$, $2\pi$, and the conversion factors for converting degrees to radians and arc seconds to radians. It also gives the day number of the first day of each month in a regular year and in a leap year starting from Jan. 0.0 of that year.
NAME BLOCK DATA

PURPOSE DATA INITIALIZING OF PI, 2*PI, CONVERSION FACTOR OF DEGREES TO RADIANS, CONVERSION FACTOR OF ARC SECONDS TO RADIANS, AND THE DAY NUMBER OF THE FIRST DAY OF EACH MONTH IN A YEAR

COMMON BLOCK MONTH

BLOCK DATA
IMPLICIT REAL*8 (A-H,O-Z) BLOCK 13
COMMON/CONSTS/PI,2PI,DRAD,DRSEC BLOCK 14
COMMON/MONTHS/MONTH(26) BLOCK 15
DATA PI/3.14159265358979323800/, BLOCK 16
• DRAD/1.745329251994329500/, BLOCK 17
• DRSEC/484813681159536000/, BLOCK 18
DATA MONTH/0,31,60,91,121,152,182,213,244,274,305,335,366, BLOCK 19
• 0,31,60,91,121,152,182,213,244,273,304,335,365/ BLOCK 20
END
DESCRIPTION

(See GEODYN)
DESCRIPTION

(See GEODYN)
DESCRIPTION

(See GEODYN)
SECTION 2.0
OPERATIONS DESCRIPTION OF GEODYN SUPPORT PROGRAMS.

2.1 GEODYN ANALYSES AND GRAPHICS SUPPORT PROGRAMS

The GEODYN Analyses and Graphics Support Programs constitute an integral part of the GEODYN System. Included within this set of programs are the following:

- DELTA - DELTA computes, prints and plots satellite trajectory differences.
- GEORGE - GEORGE performs a linear regression analysis on GEODYN residuals.
- GROUNDTRACK - GROUNDTRACK plots subsatellite groundtracks used for analysis of tracking station-satellite pass geometric relationships.
- WRDC SC4020 PLOT PACKAGE - The Plot Package is a group of subroutines that may be used to generate plots.

The operation of the programs DELTA, GEORGE, and GROUNDTRACK will be described in the following pages of this section.
2.1.1 DELTA

DELTA is a GEODYN support program which reads satellite trajectory tapes written by GEODYN and computes, prints and plots orbital differences. DELTA reads inertial Cartesian coordinates and computes trajectory differences in the more physically meaningful radial, cross track, and along track directions. Optional output from DELTA is a plot of these trajectory differences. By calling WRDC SC4020 PLOT PACKAGE subroutines, DELTA will plot these differences on the printer and/or will write an SC4020 Plotter Driver Tape which may be used to obtain microfilm and/or hard copy plots of the DELTA trajectory differences from the SC4020 plotter.

The following pages will describe in detail the setup and operation of the DELTA support program of the GEODYN System.
2.1.1.1 DELTA Input Cards

The entire card input to DELTA consists of four cards per case with no limit on the number of cases. The four cards input to DELTA consist of the DELTA Option Card and three title cards. These cards are described below.

1. The DELTA Option Card

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>I2</td>
<td>Change of unit for first satellite trajectory tape (default is 21 or that value used by the previous case).</td>
</tr>
<tr>
<td>3-4</td>
<td>I2</td>
<td>Change of unit for second satellite trajectory tape (default is 22 or that value used by the previous case).</td>
</tr>
<tr>
<td>5</td>
<td>I1</td>
<td>T- Plot requested. F or blank--no plot requested.</td>
</tr>
<tr>
<td>6</td>
<td>I1</td>
<td>Request for type of output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = microfilm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = hardcopy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 = printer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Any combination of the above may be used by simple summation. (Default is 7).</td>
</tr>
<tr>
<td>7</td>
<td>I1</td>
<td>=1 Specifies that the input tapes are ORB1 tapes. (Default is RV tapes)</td>
</tr>
<tr>
<td>8-9</td>
<td>I2</td>
<td>(\leq 0) Plots every point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(=1) Plots every point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(=n) Plots every (n^{th}) point.</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>FORMAT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>10</td>
<td>I1</td>
<td>≠0 Specifies that another case will follow. =0 This is the last case.</td>
</tr>
<tr>
<td>11-22</td>
<td>F12.6</td>
<td>Y-scale upper limit for plots. A suitable default value will be used if no value is input here.</td>
</tr>
<tr>
<td>23-34</td>
<td>F12.6</td>
<td>Y-scale lower limit for plots. A suitable default value will be used if no value is input here.</td>
</tr>
<tr>
<td>35-46</td>
<td>F12.6</td>
<td>Y-scale divisions interval. A suitable default value will be used if no value is input here. If a value is input here DELTA will assume that values also have been input in columns 11-22 and 23-34.</td>
</tr>
</tbody>
</table>

2. The DELTA Title Cards

Any information may be punched on these title cards in columns 1-56. Information punched on these cards will appear on the first frame of all plots for this case. These cards should be present only when plotting is requested.
2.1.1.2 DELTA Job Control Language and Hardware and Software Restrictions

2.1.1.2.1 Job Control Language

The DELTA program may be executed by use of the LINKGO procedure as follows:

// EXEC LINKGO, REGION.GO=250K
// LINK.SYSLIN DD *
   INCLUDE LOADLIB(ZCTVMDEL)
   INCLUDE LOADLIB(ZCRGWTP)
   ENTRY MAIN
/*
// GO.FT20F001 DD LABEL=(,BLP), UNIT=2400-7,
// DCB=(RECFM=FB, LRECL=6, BLKSIZE=4092, DEN=1),
// VOL=SER=PL0T2
// GO.FT21F001 DD UNIT=2400-9, VOL=SER=RVTAP1,
// DCB=(RECFM=VBS, LRECL=72, BLKSIZE=7204),
// LABEL=(,BLP)
// GO.FT22F001 DD UNIT=2400-9, VOL=SER=RVTAP2,
// DCB=(RECFM=VBS, LRECL=72, BLKSIZE=7204),
// LABEL=(,BLP)
// GO.DATA5 DD *

<<<The DELTA Input Cards go here.<<<

/*

Unit 20 is used for output of the SC4020 Plotter Driver tape.

Units 21 and 22 and any other units which the user wishes to specify are used for input of RV tapes and/or ORBI tapes.
2.1.1.2.2 Hardware and Software Restrictions

DELTA requires an IBM 360 computer with a minimum of 250K bytes of user accessible core, two 9 track tape drives, one 7 track tape drive, one card reader and one high speed printer.

The current DELTA program is executable under versions 14, 16, and 18 of the IBM 360 operating system.

The compilation of DELTA requires an IBM FORTRAN IV Level G compiler.

There are no DELTA software or hardware restrictions other than that the above mentioned hardware and software be available and working properly.
2.1.1.3 DELTA Example Job

The example job for DELTA is included with Example Three for GEODYN in section 4.3 of Volume 3. Shown in this example is the normal mode of operation for the DELTA program. However, as shown in section 2.1.1.2.1, tape input may also be used for DELTA rather than concatenating the DELTA execution with a GEODYN execution.
2.1.1.4 DELTA Error Messages

There are no DELTA error messages other than those which may be printed by the WRDC SC4020 PLOT PACKAGE. These messages are described in section 2.1.4 of this document.
2.1.2 GEORGE

GEORGE is a GEODYN support program which reads a Binary Residual tape written by GEODYN and using the information obtained from this tape performs statistical linear regression computations to determine tracking instrument zero-set biases and timing biases in the GEODYN residuals. Optional output from GEORGE is a residual plot which may be obtained on the printer and/or a WRDC SC4020 Plotter Driver tape from which may be obtained microfilm and/or hard copy plots of the GEODYN residuals. To perform such plotting functions, GEORGE calls subroutines in the WRDC SC4020 PLOT PACKAGE.

The following pages will describe in detail the setup and operation of the GEORGE support program of the GEODYN System.
2.1.2.1 GEORGE Input Cards

Specific functions of the GEORGE program may be requested by input cards. The GEORGE Input Cards are separated into two categories:

- GEORGE Mandatory Cards - these are cards that must be present for each case.

- GEORGE Option Cards - these are cards that as the name implies are optional.

A set of these cards will define a case. More than one case may be present.

This section of the manual describes the format and usage of the GEORGE Input Cards.
### MEASUREMENT CARD

#### COLUMNS

<table>
<thead>
<tr>
<th>1-6</th>
<th>11-16</th>
<th>21-26</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANGE</td>
<td>C BAND</td>
<td>NWALIB</td>
</tr>
</tbody>
</table>

#### Format

<table>
<thead>
<tr>
<th>1-6</th>
<th>A6</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>Alphanumeric measurement type, left adjusted in field. Measurement types are: RT ASC, R RATE, X ANGL, RANGE, ALPHA, AZMUTH.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11-16</th>
<th>A6</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-16</td>
<td>Alphanumeric network name, left adjusted in field. Network names are: STADAN, DOPLER, USAF, C BAND, SECOR, USC+GS, SPEOPT, INTERL, SAO.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>21-26</th>
<th>A6</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21-26</td>
<td>Alphanumeric station name.</td>
<td></td>
</tr>
</tbody>
</table>

#### Notes

* One Measurement Card is mandatory for each case. The measurement type must be specified. The network and station name are optional. If left blank, all networks and stations will be analyzed.
### COLUMNS FORMAT DESCRIPTION

<table>
<thead>
<tr>
<th>1-6</th>
<th>A6</th>
<th>The word &quot;EL CUT&quot; requests that data elevation cutoff be made.</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-20</td>
<td>F10.5</td>
<td>Desired data elevation cutoff angle.</td>
</tr>
</tbody>
</table>
The word "HISTGM" requests that histogram plots be made.

- 0. Histograms of residuals.
- 1. Histograms of residual ratios to sigma.
- 2. Histograms of residuals plus a final histogram of all residuals.
- 3. Histograms of residual ratios to sigma plus a final histogram of all residual ratios to sigma.
## OPTION CARD

**PLOT**

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>A4</td>
<td>The word &quot;PLOT&quot; requests that plots of analysis be made.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.5</td>
<td>=0. Printer plots only. =1. Printer plots and SC4020 Plotter Driver tape.</td>
</tr>
</tbody>
</table>
OPTION CARD
REJECT

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>The word &quot;REJECT&quot; requests data editing.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.5</td>
<td>Value of the rejection criterion.</td>
</tr>
</tbody>
</table>

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.
OPTION TERMINATION CARD*

COLUMNs | FORMAT | DESCRIPTION
---------|--------|-------------------
1-4      | A4     | The word "DATA" signifies the end of the option cards.

Note: Termination Card - This card must always be present, even if no option cards are used.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.
CASE TERMINATION CARD

The word "LAST" appearing here indicates that this is the last case. If left blank this card indicates that another case will follow.

Note: *CASE TERMINATION CARD - This card must always be present to terminate each case.
2.1.2.2 GEORGE Job Control Language and Hardware and Software Restrictions

2.1.2.2.1 Job Control Language

The GEORGE program may be executed by use of the LINKGO procedure as follows:

// EXEC LINKGO, REGION.GO=525K
//LINK.SYSLIN DD *

INCLUDE LOADLIB(ZCMILDGRG)
INCLUDE LOADLIB(ZCRGWYP)
ENTRY MAIN

GO.FT15F001 DD UNIT=2400-9, VOL=SER=BRESID,
DCB=(RECFM=VBS, LRECL=80, BLKSIZE=3204),
LABEL=(, BLP)
GO.FT20F001 DD LABEL=(, BLP), UNIT=2400-7,
DCB=(DEN=1, RECFM=FB, LRECL=6, BLKSIZE=4092),
VOL=SER=PLOT1
GO.DATA5 DD *

The GEORGE Input Card deck goes here.

Unit 15 is used to input the Binary Residual Tape.
Unit 20 is used for output of the SC4020 Plotter Driver tape.

2.1.2.2.2 Hardware and Software Restrictions

GEORGE requires a large scale IBM 360 computer with a minimum of 525K bytes of user accessible core, one 9 track tape drive, one 7 track tape drive, one card reader and one high speed printer.
The current GEORGE program is executable under version 18 of the IBM 360 operating system.

The compilation of GEORGE requires an IBM FORTRAN IV Level G compiler.

There are no GEORGE software or hardware restrictions other than that the above mentioned hardware and software be available and working properly.
2.1.2.3 GEORGE Example Job

The example job for GEORGE is included with Example Two for GEODYN in Volume 3, Section 4.2. Shown in this example is the normal mode of operation for the GEORGE program. However, as shown in Section 2.2.2.1, tape input may also be used for GEORGE rather than concatenating the GEORGE execution with a GEODYN execution.
2.1.2.4 GEORGE Error Messages

In addition to those error messages that may be printed by the WRDC SC4020 PLOT PACKAGE the following error messages may be printed during the execution of the GEORGE program.

a) ILLEGAL MEASUREMENT TYPE--SKIPPING TO NEXT CASE

b) ILLEGAL NETWORK NAME--SKIPPING TO NEXT CASE

c) ILLEGAL OPTION CARD--REMAINING OPTIONS IGNORED--SKIPPING TO DATA

d) NO DATA OF THE TYPE SPECIFIED FOUND--SKIPPING TO NEXT CASE

e) TOO MANY OBSERVATIONS--REMAINDER IGNORED

With the exception of error e all of these messages are self-explanatory.

e) The GEORGE program will process a maximum of 4000 observations per case. Observations in excess of 4000 will be ignored.
2.1.3 GROUNDTRACK

GROUNDTRACK is a GEODYN support program which reads a subsatellite groundtrack tape written by GEODYN and plots, using the WRDC SC4020 PLOT PACKAGE, the geometry of satellite passes across the stations tracking the satellite. Only those passes on which tracking data is taken are written on the groundtrack tape. Plots from GROUNDTRACK may be obtained on the printer and/or an SC4020 Plotter Driver tape.

The following pages will describe in detail the setup and operation of the GROUNDTRACK support program of the GEODYN System.
2.1.3.1 GROUNDTRACK Input Cards

Specific functions of the GROUNDTRACK program may be requested by cards. The GROUNDTRACK Input Cards are separated into two categories:

- **GROUNDTRACK Mandatory Cards** — these are cards that must be present for each case.

- **GROUNDTRACK Option Cards** — these are cards that, as the name implies, are optional.

A set of these cards will define a case. More than one case may be present.

This section of the manual describes the format and usage of the GROUNDTRACK Input Cards.
## STATION POSITION CARDS

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>Station name. ( \alpha )</td>
</tr>
<tr>
<td>7-10</td>
<td>I4</td>
<td>Station number.</td>
</tr>
<tr>
<td>11</td>
<td>A1</td>
<td>Sign of Latitude</td>
</tr>
<tr>
<td>12-13</td>
<td>I2</td>
<td>Degrees ( \phi ) Station</td>
</tr>
<tr>
<td>14-15</td>
<td>I2</td>
<td>Minutes ( \lambda ) East</td>
</tr>
<tr>
<td>16-25</td>
<td>F10.5</td>
<td>Seconds ( \phi ) Latitude</td>
</tr>
<tr>
<td>26-28</td>
<td>I3</td>
<td>Degrees ( \lambda ) East</td>
</tr>
<tr>
<td>29-30</td>
<td>I2</td>
<td>Minutes ( \lambda ) East</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.5</td>
<td>Seconds ( \lambda ) Longitude</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.5</td>
<td>Station height in meters. ( h )</td>
</tr>
</tbody>
</table>

**Note:** Station position cards are mandatory for all stations for which plotting is requested. A maximum of 10 stations is permitted per case.
The word "END" specified here indicates the end of the Station Position cards for the case.

This card must be present.
**OPTION CARD**

**GRDSET**

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>A6</td>
<td>The word &quot;GRDSET&quot; specified here indicates that this card will specify the grid parameters. If this card is not present, GROUNDTRACK will compute appropriate grid limits.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.5</td>
<td>Maximum longitude west of station.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.5</td>
<td>Minimum longitude east of station.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.5</td>
<td>Number of longitudinal grid intervals.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.5</td>
<td>Minimum latitude.</td>
</tr>
<tr>
<td>51-60</td>
<td>F10.5</td>
<td>Maximum latitude.</td>
</tr>
<tr>
<td>61-70</td>
<td>F10.5</td>
<td>Number of latitudinal grid intervals.</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>FORMAT</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>1-6</td>
<td>A6</td>
<td>The word &quot;LNDPLT&quot; specified here indicates the plot is to be superimposed over the land contour plot for the region of the earth specified.</td>
</tr>
</tbody>
</table>
The word "PLOTS" specified here indicates that groundtrack plots are requested with the option specified on this card.

=0. Printer plots only.

=1. Printer plots and an SC4020 Plotter Driver tape.
## OPTION CARD

### TIME

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>A4</td>
<td>The word &quot;TIME&quot; specified here indicates that the groundtrack plot times will be specified on this card.</td>
</tr>
<tr>
<td>11-20</td>
<td>F10.5</td>
<td>Start date in YYMMD.</td>
</tr>
<tr>
<td>21-30</td>
<td>F10.5</td>
<td>Start time in HHMM.</td>
</tr>
<tr>
<td>31-40</td>
<td>F10.5</td>
<td>Stop date in YYMMD.</td>
</tr>
<tr>
<td>41-50</td>
<td>F10.5</td>
<td>Stop time in HHMM.</td>
</tr>
</tbody>
</table>
**OPTION TERMINATION CARD**

DATA

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>FORMAT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>A4</td>
<td>The word &quot;DATA&quot; specified here indicates the end of the optional GROUNDTRACK Input Cards.</td>
</tr>
</tbody>
</table>

Note: *OPTION TERMINATION CARD*—This card must always be present for each arc.
CASE TERMINATION CARD

COLUMN  FORMAT   DESCRIPTION
1-4     A4          The word "LAST" appearing here indicates that this is the last case. If left blank this card indicates that another case will follow.

Note:  *CASE TERMINATION CARD--This card must always be present to terminate each case.
2.1.3.2 GROUNDTRACK Job Control Language and Hardware and Software Restrictions

2.1.3.2.1 Job Control Language

The GROUNDTRACK program may be executed by use of the LINKGO procedure as follows:

```plaintext
// EXEC LINKGO,REGION.GO=500K
//LINK.SYSLIN DD *
INCLUDE LOADLIB(ZCMLDGRK)
INCLUDE LOADLIB(ZCRJGWRL)
INCLUDE LOADLIB(ZCRGWTYP)
ENTRY MAIN
/*
//GO.FT11F001 DD UNIT=2400-9,VOL=SER=GTRACK,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
// LABEL=(,BLP)
//GO.FT20F001 DD UNIT=2400-7,LABEL=(,BLP),
// DCB=(DEN=1,RECFM=FB,LRECL=6,BLKSIZE=4092),
// VOL=SER=PLOT3
//GO.DATA5 DD *

The GROUNDTRACK Input Card deck goes here.

/*

Unit 11 is used for input of the groundtrack tape.
Unit 20 is used for output of the SC4020 Plotter Driver tape.

2.1.3.2.2 Hardware and Software Restrictions

GROUNDTRACK requires a large scale IBM 360 computer with a minimum of 500K bytes of user accessible core, one 9
track tape drive, one 7 track tape drive, one card reader and one high speed printer.

The current GROUNDTRACK program is executable under version 18 of the IBM 360 operating system.

The compilation of GROUNDTRACK requires an IBM FORTRAN IV Level G compiler.

There are no GROUNDTRACK software or hardware restrictions other than that the above mentioned hardware and software be available and working properly.
2.1.3.3 GROUNDTRACK Example Job

The example job for GROUNDTRACK is included with Example One for GEODYN in Section 4.1 of Volume 3. Shown in this example is the normal mode of operation for the GROUNDTRACK program. However, as shown in Section 2.1.3.2.1, tape input may also be used for GROUNDTRACK rather than concatenating the GROUNDTRACK execution with a GEODYN execution.
2.1.3.4 GROUNDTRACK Error Messages

In addition to those error messages that may be printed by the WRDC SC4020 PLOT PACKAGE, the following error message may be printed during the execution of the GROUNDTRACK program.

ILLEGAL OPTION CARD _______ IGNORED REMAINING OPTIONS,
EXECUTION CONTINUING
2.1.4 WRDC SC4020 PLOT PACKAGE

The WRDC SC4020 PLOT PACKAGE is a group of subroutines which may be called from FORTRAN programs and which may be used to plot information. The WRDC SC4020 PLOT PACKAGE has no main program and therefore no setup and operation procedures. However, since DELTA, GEORGE, and GROUNDTRACK all use the WRDC SC4020 PLOT PACKAGE, it is appropriate to herein describe all error messages which may be printed by the WRDC SC4020 PLOT PACKAGE during the execution of the above mentioned GEODYN support programs.
The WRDC SC4020 PLOT PACKAGE prints three different error messages, all of which result from inappropriate PLOT PACKAGE input. Consequently, any errors which have occurred will be caused by errors in the calling programs (i.e. DELTA, GEORGE, and GROUNDTRACK). The proper response to all of these messages is to examine the input to these calling programs for misplaced, out-of-order, or mispunched input cards or incorrectly specified input tape parameters.

The WRDC SC4020 PLOT PACKAGE error messages are:

1) SETGRD ARGUMENTS OUT OF RANGE -- LIMITS NOT RESET

2) EMPTY ARRAY OR ALL ITEMS EQUAL IN QUICKY

3) /////\\\\\

Slashes in upper right corner of a plot indicate an attempt was made to plot outside of the device limits.

These errors always result in the following program action.

1) Plot frame advance.

2) No program corrective action.

3) No program corrective action.
2.2 GEODYN DATA HANDLING SUPPORT PROGRAMS

The GEODYN Data Handling Support Programs are used for data management. The five data handling support programs are:

- DODS SORT-MERGE
- GEOS SORT-MERGE
- EPHEMERIS TAPE GENERATOR
- ORB1 CONVERSION (9-7 track)
- TDIF TABLE GENERATOR

The operation of these programs is described in the following pages.
2.2.1 DODS SORT-MERGE

DODS SORT-MERGE reads an unspecified number of data tapes in DODS Data Tape Format, assuming these tapes to be one continuous file not in time order. Scratch files are written containing strings of time ordered data which are iteratively merged with other strings decreasing the number of strings by half until one time ordered string of data in DODS Data Tape Format exists.

There is no card input to the DODS SORT-MERGE program. On the following pages will be described

- Job Control Language and Job Submittal and

- DODS SORT-MERGE Printer Output.
2.2.1.1 Job Control Language and Job Submittal

To submit a DODS SORT-MERGE job requires only the preparation of the job control language (JCL) and submittal of the job to the computer with the proper job identification slip.

DODS SORT-MERGE may be invoked by usage of the FORTRAN and LINKGO procedures.

```
// EXEC FORTRAN
//SOURCE.SYSIN DD *

<<<The DODS SORT-MERGE FORTRAN deck goes here.<<<
```

```
/*
 // EXEC LINKGO,REGION.GO=400K,TIME=1440
 //GO.FT10F001 DD UNIT=2400-9,LABEL=(,BLP),
 // DCB=(RECFM=VBS,LRECL=104,BLKSIZE=1044),
 // VOL=SER=(INPUT1,INPUT2,INPUT3,......)
 //GO.FT11F001 DD UNIT=2400-4,VOL=SER=OUTPUT,
 // DCB=(RECFM=VBS,LRECL=104,BLKSIZE=8324),
 // LABEL=(,BLP)
 //GO.FT20F001 DD UNIT=DISK,SPACE=(28008,100),
 // DCB=(RECFM=VST,BLKSIZ=28008)
 //GO.FT21F001 DD UNIT=DISK,SPACE=(28008,100),
 // DCB=(RECFM=VST,BLKSIZ=28008)
 //GO.FT22F001 DD UNIT=DISK,SPACE=(28008,100),
 // DCB=(RECFM=VST,BLKSIZ=28008)
 //GO.FT23F001 DD UNIT=DISK,SPACE=(28008,100),
 // DCB=(RECFM=VST,BLKSIZ=28008)
```

2-40
Data tape input is on unit 10.
Data tape output is on unit 11.
Units 20, 21, 22, 23 are used for temporary scratch data storage and will each hold 25,000 observations.

The combined disk space requested by units 20-23 comes to a total of 1600 tracks. To allocate 1600 tracks is extremely difficult; therefore, if more than 25,000 observations are to be processed, units 20-23 should be specified as 9-track, high density tapes with the following DCB parameters.

```
DCB=(RECFM=VBS,LRECL=28008,BLKSIZE=28012,DEN=3)
```
DOES SORT-REVERSE EXAMPLE SHIP DECK

// ... JOR ...
// EXEC FORTRAN
// SOURCE, SYST, ON 

THE DOES SORT-REVERSE FORTRAN DECK GOES HERE

/*
 // EXEC LINKER, REGION, GO=LOOK
 // G0, 0710001 ON UNIT=2400-0, DCR=(RECFM=VST, LRECF=104, BLKSIZE=2048),
 // DFM=3), LABEL=(*.RIP), VOL=599=1'UNIT
 // G0, 0711001 ON UNIT=2400-0, DCR=(RECFM=VST, LRECF=104, BLKSIZE=2248,
 // DFM=3), LABEL=(*.RIP), VOL=599=SCRATCH
 // G0, 0720001 ON UNIT=DISK, DCR=(RECFM=VST, BLKSIZE=2004),
 // SPACE=(20004, 10)
 // G0, 0715000 ON UNIT=DISK, DCR=(RECFM=VST, BLKSIZE=2004),
 // SPACE=(20004, 10)
 // G0, 0722001 ON UNIT=DISK, DCR=(RECFM=VST, BLKSIZE=2004),
 // SPACE=(20004, 10)
// G0, 0723001 ON UNIT=DISK, DCR=(RECFM=VST, BLKSIZE=2004),
// SPACE=(20004, 10)

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

2-42
2.2.1.2 DODS Sort-Merge Printer Output

During normal operation DODS SORT-MERGE prints the number of strings of data before each merge process.

Only one error message may be printed and that message is

NO SORT INPUT

The following example job sorted and merged a data tape with 2130 measurements in random time order.

The core and time required were

CORE = 396k  CPU = 0.91m  I/O = 0.19m
2.2.2 GEOS SORT-MERGE

GEOS SORT-MERGE reads an unspecified number of data tapes in GEOS Data Tape Format assuming these tapes to be one continuous file not in time order. Scratch files are written containing strings of time ordered data which are iteratively merged with other strings decreasing the number of strings by half until one time ordered string of data in GEOS Data Tape Format exists.

There is no card input to the GEOS SORT-MERGE program. On the following pages will be described

- Job Control Language and Job Submittal.
- GEOS SORT-MERGE Printer Output.
2.2.2.1 Job Control Language and Job Submittal

To submit a GEOS SORT-MERGE job requires only the preparation of the job control language (JCL) and submittal of the job to the computer with the proper job identification slip.

GEOS SORT-MERGE may be invoked by usage of the FORTRAN and LINKGO procedures.

// EXEC FORTRAN
//SOURCE.SYSIN DD *

The GEOS SORT-MERGE FORTRAN deck goes here.

/*
// EXEC LINKGO,REGION.GO=250K,TIME=1440
//GO.FT10F001 DD UNIT=(2400-9,2),LABEL=(,BLP),
//DCB=(RECFM=FB,LRECL=80,BKSIZE=3200),
//VOL=SER=(INPUT1,INPUT2,INPUT3,......)
//GO.FT11F001 DD UNIT=2400-4, LABEL=(,BLP),
//DCB=(RECFM=FB,S,LRECL=80,BKSIZE=8000),
//VOL=SER=OUTPUT
//GO.FT20F001 DD UNIT=DISK,SPACE=(19008,100),
//DCB=(RECFM=VST,BKSIZE=19008)
//GO.FT21F001 DD UNIT=DISK,SPACE=(19008,100),
//DCB=(RECFM=VST,BKSIZE=19008)
//GO.FT22F001 DD UNIT=DISK,SPACE=(19008,100),
//DCB=(RECFM=VST,BKSIZE=19008)
//GO.FT23F001 DD UNIT=DISK,SPACE=(19008,100),
//DCB=(RECFM=VST,BKSIZE=19008)
Data tape input is on unit 10.
Data tape output is on unit 11.
Units 20, 21, 22, 23 are used for temporary scratch data storage and will each hold 25,000 observations.

The combined disk space requested by units 20-23 comes to a total of 1100 tracks. To allocate 1100 tracks is extremely difficult; therefore, if more than 25,000 observations are to be processed, units 20-23 should be specified as 9-track, high density tapes with the following DCB parameters.

DCB=(RECFM=VBS,LRECL=19008,BLKSIZExE=19012,DEN=3)
GENS SORT-MERGE EXAMPLE SETUP DECK

//... JOB ...
// EXEC FORTRAN
// SOURCE, SYSIN ON *

THE GEN'S SORT-MERGE FORTRAN DECK GOES HERE

/*
EXEC LINCOM, REGION, GD=27SK
EXEC F110001 ON UNIT=2400, DCR=(RFILE=FR, LREFL=80, ALK1ZE=2000),
       LABEL=1, SQL, MIL=SDF, SRCIN=0.00
EXEC F120001 ON UNIT=DISK, DCR=(RFILE=VST, ALK1ZE=1900A),
       SPACE=(1900A,10)
EXEC F210001 ON UNIT=DISK, DCR=(RFILE=VST, ALK1ZE=1900A),
       SPACE=(1900A,10)
EXEC F220001 ON UNIT=DISK, DCR=(RFILE=VST, ALK1ZE=1900A),
       SPACE=(1900A,10)
EXEC F230001 ON UNIT=DISK, DCR=(RFILE=VST, ALK1ZE=1900A),
       SPACE=(1900A,10)
EXEC F100001 ON *

THE GEN'S DATA CARDS TO BE MERGED GO HERE

*/

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.
2.2.2.2 GEOS Sort-Merge Printer Output

During normal operation GEOS SORT-MERGE prints the number of strings data before each merge process.

Only one error message may be printed and that message is

NO SORT INPUT

The following example job sorted and merged 720 GEOS data cards in random time order.

The core and time required were

CORE = 256k      CPU = 0.12m      I/O = 0.08m
<table>
<thead>
<tr>
<th>Time</th>
<th>Action</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00:00</td>
<td>START</td>
<td>System initialized</td>
<td></td>
</tr>
<tr>
<td>0:00:12</td>
<td>RUN</td>
<td>Program execution started</td>
<td></td>
</tr>
<tr>
<td>0:00:22</td>
<td>STOP</td>
<td>Program execution stopped</td>
<td></td>
</tr>
<tr>
<td>0:00:23</td>
<td>STOP</td>
<td>System shut down</td>
<td></td>
</tr>
<tr>
<td>0:00:24</td>
<td>STOP</td>
<td>System shut down</td>
<td></td>
</tr>
</tbody>
</table>

**System Status:**
- CPU: 0.10 MINS
- Disk: 0.00 MINS
- Tape: 0.00 MINS
- Cell: 0.00 MINS
- Other: 0.00 MINS

**System Resources:**
- **ALOCC.**
  - ZCTVMNT, CC
  - FT121, CC
  - FT216, CC
  - FT117, CC
  - FT218, CC

**Allocations:**
- ZCTVMNT, CC
- FT121, CC
- FT216, CC
- FT117, CC
- FT218, CC

**System Configuration:**
- 1564K LEC, OK
2.2.3 EPHEMERIS TAPE GENERATOR

The EPHEMERIS TAPE GENERATOR generates geocentric lunar positions at half day intervals, heliocentric positions of the Earth-moon barycenter, and the planets, Venus, Mars, Jupiter and Saturn at four day intervals and the nutation in obliquity at half day intervals.
2.2.3.1 Input Card

The EPHEMERIS TAPE GENERATOR can read a maximum of three tapes since the JPL ephemeris is broken into three pieces. The input card of the EPHEMERIS TAPE GENERATOR consists of specification of start and stop times for taking information from each input tape. This is done in the following manner:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12</td>
<td>F12.5</td>
<td>Julian start time of ephemeris.</td>
</tr>
<tr>
<td>13-24</td>
<td>F12.5</td>
<td>Julian stop time for taking information from first input tape.</td>
</tr>
<tr>
<td>25-36</td>
<td>F12.5</td>
<td>Julian start time for taking information from second input tape. If zero, second input tape will not be read. If second tape is to be read, start time must be the same as the stop time of first tape.</td>
</tr>
<tr>
<td>37-48</td>
<td>F12.5</td>
<td>Julian stop time for taking information from second input tape.</td>
</tr>
<tr>
<td>49-60</td>
<td>F12.5</td>
<td>Julian start time for taking information from third input tape. If zero, third tape will not be read. If third tape is to be read, start time must be the same as the stop time of second tape.</td>
</tr>
<tr>
<td>61-72</td>
<td>F12.5</td>
<td>Julian stop time for taking information from third input tape.</td>
</tr>
</tbody>
</table>

First, second and third tapes must use units 12, 13, and 14, respectively.
2.2.3.2 Job Control Language and Job Submittal

To submit an EPHEMERIS TAPE GENERATOR job requires only the preparation of the job control language (JCL) and input card.

The EPHEMERIS TAPE GENERATOR may be invoked using the following procedure:

```
//-----JOB-----
// EXEC FORTRAN
//SYSIN DD *

<<<The EPHEMERIS TAPE GENERATOR Source Deck goes here.<<<

/*
 // EXEC LINKGO,REGION=250K
 //GO.FT10F001 DD UNIT=2400-9,DCB=(RECFM=VBS,
 // LRECL=436,BLKSIZE=7294,DEN=3),LABEL=(1,BLP),
 // VOL=SER=OUTPUT
 //GO.FT12F001 DD UNIT=2400-9,DCB=(RECFM=VBS,
 // LRECL=7456,BLKSIZE=29828,DEN=3),LABEL=(,BLP),
 // VOL=SER=INPUT1
 //GO.SYSUDUMP DD SYSOUT=C,SPACE=(CYL,(10,2))
 //GO.DATA5 DD *

<<<The EPHEMERIS TAPE GENERATOR Input Card goes here.<<<

*/

2-53
2.2.4 ORB1 CONVERSION (9-7) Tracks

The ORB1 CONVERSION program reads a double precision, 9-track, IBM 360 ORB1 tape written by GEODYN and writes a single precision, 7-track, IBM 7094 ORB1 tape in the same format.

There is no card input and no printer output for the ORB1 CONVERSION program and therefore, complete program operation is described by Job Control Language (JCL). The JCL necessary is described below and requires only to be submitted to the computer with the proper job identification slip.

ORB1 CONVERSION may be invoked by usage of the FORTRAN and LINKGO procedures.

```
// EXEC FORTRAN
//SOURCE:SYSIN DD *
```

The ORB1 CONVERSION FORTRAN deck goes here.

```
/*
// EXEC LINKGO
//GO.FT10F001 DD UNIT=2400-9,LABEL=(,BLP).
// DCB=(RECFM=VBS,LRECL=2804,BLKSIZE-2808),
// VOL=SER=INPUT9
//GO.FT11F001 DD UNIT=2400-7,LABEL=(,BLP),
// DCB=(RECFM=FB,BLKSIZE=2100,LRECL=21)
// VOL=SER=OUTPUT
```

Unit 10 is the IBM 360, 9-track, ORB1 tape input.
Unit 11 is the IBM 7094, 7-track, ORB1 tape output.
The example job for ORB1 CONVERSION is included with Example Three for GEODYN in Volume 3, Section 4.3.
2.2.5 TDIF TABLE GENERATOR

The TDIF TABLE GENERATOR generates tabular differences between the time systems A.1 and UT1. It reads tables showing the differences between systems UT1 and UTC (UT1-UTC) which are obtained from B.I.H. The tables require continual up-dating as this information is received directly from B.I.H.
2.2.5.1 Data Deck

The values of UT1-UTC are received from B.I.H. on Circular D.

The values of UT1-UTC are input to the program at 10-day intervals in the following manner:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>I6</td>
<td>Date in YYMMD</td>
</tr>
<tr>
<td>7-16</td>
<td>F10.5</td>
<td>Value in seconds of UT1-UTC as given on B.I.H. circular D.</td>
</tr>
</tbody>
</table>

An example of this circular is given on the following page.

The program then punches the values of A.1-UT1 on cards in tabular form.
1 - UNIVERSAL TIME AND COORDINATES OF THE POLE

Date | J.D. | smoothed values | raw values
--- | --- | --- | ---
(Oh UT) | 2400000.5 | x | y | UT2-UTC | UT1-UTC | UT2-UTC | UT1-UTC
1972 | + | 0.001 | 0.000 | 0.0001s | 0.0001s | - | -
June 1 | 41 469 | -145 | +356 | -5252 | -5553 | -151 | +341 | -5541 | -16.555
6 | 474 | -134 | +366 | -5416 | -5710 | -129 | +371 | -5743 | 571
11 | 479 | -120 | +376 | -5579 | -5861 | -114 | +359 | -5872 | 586
16 | 484 | -105 | +385 | -5740 | -6245 | -92 | +374 | -6019 | 666
21 | 489 | -89 | +394 | -5899 | -6145 | -92 | +421 | -6165 | 614
26 | 494 | -72 | +402 | -6057 | -6279 | -51 | +382 | -6263 | 627
July 1 | 499 | -54 | +409 | +3786 | +3591 | -58 | +407 | +3616 | 646

IAT-UTC is exactly 10s in June 1972
IAT-UTC is exactly 11s since 1972 July 1st, Oh UTC.

2 - EMISSION TIME OF TIME SIGNALS, for June 1972

| Signal | E | Signal | E | Signal | E |
--- | --- | --- | --- | --- | --- |
CHU | 0 | FTH42, FTK77, FTN87 | 0 | NSS (o.c.) | + 9 |
DAM, DAN, DAO | 0 | HEG | 0 | OBRS (2) | |
DCF77 | 0 | IA (2) | 0 | CRA (2) | |
DGI | 0 | IBF | + 3 | PPE | - 5 |
DTZ | 0 | JJJ | 0 | RPM (1) | 0 |
FFH | 0 | LOL | - 5 | VNG | 0 |
FTA91 | 0 | NSF | + 1 | WWV, WWVB, WWVH | 0 |
GBC | 0 | GBZ (3) | - 3 | ZUO (2) | 0 |

(1) and other signals from USSR
(2) no data available
(3) corrected values : April 1972, E = - 3 ; May 1972, E = - 2

3 - COORDINATED UNIVERSAL TIME (approximation UTC(i) of UTC, kept by the laboratory i.
Ref. CCIR Recommendation 458, 1970)

- From LORAN-C and Television pulses receptions

| Date | J.D. 2400000.5 | June 11 | June 21 | July 1 |
--- | --- | --- | --- | --- |
J. D. 2400000.5 | + | 41 479 | 41 489 | 41 499 |
Laboratory i | UTC-UTC(i) | (unit : 1 µs) | UTC-UTC(i) | (unit : 1 µs) | UTC-UTC(i) | (unit : 1 µs) |
--- | --- | --- | --- | --- | --- | --- |
PTB (Braunschweig) | + 2.9 | + 3.0 | + 2.9 | |
USNO (Washington) | - 6.6 | - 6.5 | - 6.2 | - 6.6 |
OP (Paris) | + 1.6 | + 1.6 | + 1.6 | + 1.6 |
NBS (Boulder) | - 2.4 | - 2.5 | - 2.7 | - 2.7 |
RCO (Herstmonceux) | + 3.2 | + 3.8 | + 4.4 | + 4.4 |
NRC (Ottawa) | + 0.8 | + 0.9 | + 1.2 | + 1.2 |
FOA (Stockholm) | + 23.9 | + 26.3 | + 28.6 | + 28.6 |
DHI (Hamburg) | - 16.9 | - 15.3 | - 13.9 | - 13.9 |
ON (Neuchâtel) | + 20.6 | + 20.7 | + 20.6 | + 20.6 |

P. T. C
2.2.5.2 Job Control Language and Job Submittal

To submit a TDIF TABLE GENERATOR job requires only the preparation of the job control language (JCL) and the data deck.

The TDIF TABLE GENERATOR may be invoked using the following procedure:

```bash
//-----JOB-----
// EXEC FORTRANH,PARM='ID,OPT=2'
//SOURCE.SYSIN DD *

<<<THE TDIF TABLE GENERATOR Fortran deck goes here.<<<

// EXEC LOADER
//GO.DATAS DD *

<<<The TDIF TABLE GENERATOR data deck goes here<<<

/*

2-59