X-591-73-69

FINAL REPORT FOR THE MINITRACK TRACKING FUNCTION DESCRIPTION

VOLUME 2

(NASA-TM-X-70482) MINITRACK TRACKING FUNCTION DESCRIPTION, VOLUME 2 Final Report (NASA) ~~315~~ 313 p HC $17.75 CSCL 14B

N73-33156

Unclas

G3/11 19348

MARCH 1973

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

FINAL REPORT FOR THE MINITRACK TRACKING FUNCTION DESCRIPTION

Volume 2

prepared by

Thomas S. Englar
Stephen A. Mango
Charles A. Roettcher
David L. Watters

Control Systems Research, Inc.
Arlington, Virginia 22209

for

Goddard Space Flight Center
Greenbelt, Maryland 20771

March 1973

William M. Rice
Technical Monitor
Contract NAS5-10694

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

The material in this report was prepared in March 1970 under Contract NAS 5-10694 and published as a Goddard Space Flight Center X-Document in March 1973.

[redacted]

CONTENTS

Appendix A

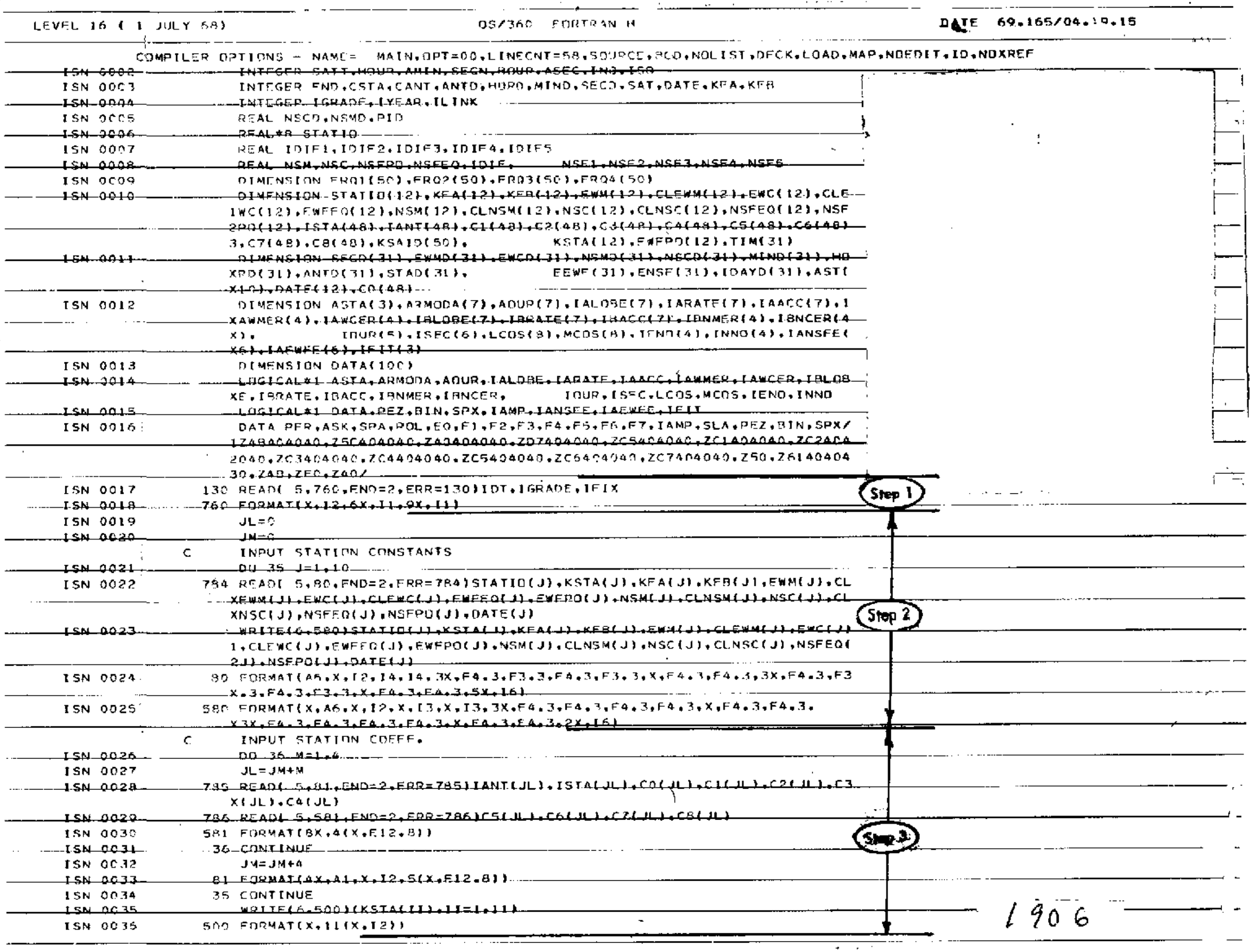
PREPROCESSOR LISTING

This appendix contains a computer listing of the Minitrack cubic preprocessing program, more specifically, listings of the two programs currently used in the preprocessing function, MIN-B and OBS-B, plus listings of the subroutines that are called by one or both of these main programs.

The reader should refer to section 4, of volume 1, on the preprocessor for an explanation of the operational intricacies of these programs. This description progresses through the listings, step by step, as called out by the elliptical step identifiers added to the computer printout.

An explanation of the subroutines is presented in section 5 of volume 1. A technical, analytical discussion of the adequacy of the least-squares fitting technique applied by the preprocessing program and the data-compression routine is given in section 6, volume 1.

MIN-B PROGRAM



```
 LEVEL 16 ( 1 JULY 68)                          OS/360  FORTRAN H                                     DATE  69.165/04.19.15

                COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002             INTEGER SATT,HOUR,AMIN,SEGN,HOUR,ASEC,IN3,ISO
      ISN 0003             INTEGER END,CSTA,CANT,ANTD,HURO,MIND,SECD,SAT,DATE,KFA,KFB
      ISN 0004             INTEGER IGRADE,IYEAR,ILINK
      ISN 0005             REAL NSCD,NSMD,PID
      ISN 0006             REAL*8 STATIO
      ISN 0007             REAL IDIF1,IDIF2,IDIF3,IDIF4,IDIF5
      ISN 0008             REAL NSM,NSC,NSFPO,NSFEQ,IDIF,       NSE1,NSE2,NSE3,NSE4,NSE5
      ISN 0009             DIMENSION FRQ1(50),FRQ2(50),FRQ3(50),FRQ4(50)
      ISN 0010             DIMENSION STATIO(12),KFA(12),KFB(12),EWM(12),CLEWM(12),EWC(12),CLE
                          1WC(12),EWFEQ(12),NSM(12),CLNSM(12),NSC(12),CLNSC(12),NSFEQ(12),NSF
                          2PO(12),ISTA(48),IANT(48),C1(48),C2(48),C3(48),C4(48),C5(48),C6(48)
                          3,C7(48),C8(48),KSAID(50),            KSTA(12),EWFPO(12),TIM(31)
      ISN 0011             DIMENSION SECD(31),EWMD(31),EWCD(31),NSMD(31),NSCD(31),MIND(31),HO
                          XRD(31),ANTD(31),STAD(31),            EEWF(31),ENSF(31),IDAYD(31),AST(
                          X10),DATE(12),C0(48)
      ISN 0012             DIMENSION ASTA(3),ARMODA(7),AOUP(7),IALOBE(7),IARATE(7),IAACC(7),1
                          XAWMER(4),IAWCER(4),IBLOBE(7),IBRATE(7),IBACC(7),IBNMER(4),IBNCER(4
                          X),          IOUR(5),ISEC(6),LCOS(8),MCOS(8),IEND(4),INNO(4),IANSFE(
                          X6),IAEWFE(6),IFIT(3)
      ISN 0013             DIMENSION DATA(100)
      ISN 0014             LOGICAL*1 ASTA,ARMODA,AOUR,IALOBE,IARATE,IAACC,IAWMER,IAWCER,IBLOB
                          XE,IBRATE,IBACC,IBNMER,IBNCER,       IOUR,ISEC,LCOS,MCOS,IEND,INNO
      ISN 0015             LOGICAL*1 DATA,PEZ,BIN,SPX,IAMP,IANSFE,IAEWFE,IFIT
      ISN 0016             DATA PER,ASK,SPA,POL,EQ,F1,F2,F3,F4,F5,F6,F7,IAMP,SLA,PEZ,BIN,SPX/
                          1Z48AC4040,Z5CA04040,Z40404040,ZD7404040,ZC5404040,ZC1A04040,ZC2ACA
                          2040,ZC3404040,ZC4404040,ZC5404040,ZC6404040,ZC7404040,Z50,Z6140404
                          30,ZAB,ZE0,Z40/
      ISN 0017        130 READ( 5,760,END=2,ERR=130)IDT,IGRADE,IFIX                                    Step 1
      ISN 0018        760 FORMAT(X,I2,6X,I1,9X,I1)
      ISN 0019             JL=0
      ISN 0020             JM=0
                    C      INPUT STATION CONSTANTS
      ISN 0021             DO 35 J=1,10
      ISN 0022        784 READ( 5,80,END=2,ERR=784)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CL
                          XEWM(J),EWC(J),CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CL
                          XNSC(J),NSFEQ(J),NSFPO(J),DATE(J)                                            Step 2
      ISN 0023             WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
                          1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
                          2J),NSFPO(J),DATE(J)
      ISN 0024         80 FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
                          X.3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)
      ISN 0025        580 FORMAT(X,A6,X,I2,X,I3,X,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,
                          X3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)
                    C      INPUT STATION COEFF.
      ISN 0026             DO 36 M=1,4
      ISN 0027             JL=JM+M
      ISN 0028        785 READ( 5,81,END=2,ERR=785)IANT(JL),ISTA(JL),C0(JL),C1(JL),C2(JL),C3
                          X(JL),C4(JL)
      ISN 0029        786 READ( 5,581,END=2,ERR=786)C5(JL),C6(JL),C7(JL),C8(JL)
      ISN 0030        581 FORMAT(8X,4(X,E12.8))                                                        Step 3
      ISN 0031         36 CONTINUE
      ISN 0032             JM=JM+4
      ISN 0033         81 FORMAT(4X,A1,X,I2,5(X,E12.8))
      ISN 0034         35 CONTINUE
      ISN 0035             WRITE(6,500)(KSTA(II),II=1,11)
      ISN 0036        500 FORMAT(X,11(X,I2))
```

1906

```
                 C      INPUT SATELLITE CONSTANTS
      ISN 0037          DO 37 J=1,50
      ISN 0038      783 READ( 5,82,END=38,ERR=783)KSAID(J),FRQ1(J),FRQ2(J),FRQ3(J),FRQ4(J)
      ISN 0039       82 FORMAT(I5,19X,F8.3,F8.3,F8.3,F8.3)
      ISN 0040          WRITE( 6,582)KSAID(J),FRQ1(J),FRQ2(J),FRQ3(J),FRQ4(J)
      ISN 0041      582 FORMAT(X,I5,19X,F8.3,X,F8.3,X,F8.3,X,F8.3,X)
      ISN 0042          IF(KSAID(J))37,38,37
      ISN 0043       37 CONTINUE
      ISN 0044       38 KSATCT=J-1
                 C      READ DATA -- CHECK FOR AMPERSINE ************************************
      ISN 0045       39 READ( 9,601,END=1,ERR=39)(DATA(I),I=1,65)
      ISN 0046      601 FORMAT(65A1)
      ISN 0047          IF(DATA(1).NE.IAMP)GO TO 39
      ISN 0049       86 IF(DATA(7).EQ.SPX)GO TO 83
      ISN 0051          IF(DATA(8).EQ.SPX)GO TO 83
      ISN 0053          IF(DATA(4)-BIN.EQ.0)GO TO 85
      ISN 0055          SAT=((DATA(3)-BIN)*10000.)+((DATA(4)-BIN)*1000.)+((DATA(5)-BIN)*10
                       X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      ISN 0056          GO TO 84
      ISN 0057       85 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(5)-BIN)*10
                       X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      ISN 0058          GO TO 84
      ISN 0059       83 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(4)-BIN)*10
                       X0.)+((DATA(5)-BIN)*10.)+(DATA(6)-BIN)
      ISN 0060       84 DO 602 M=1,50
      ISN 0061          IF(SAT.EQ.KSAID(M))GO TO 603
      ISN 0063      602 CONTINUF
      ISN 0064          WRITE( 6,635)
      ISN 0065      635 FORMAT(19H SAID NOT IN TABLE )
      ISN 0066          WRITE(6,183)IAMP,SAT
      ISN 0067      183 FORMAT(X,A1,I5)
      ISN 0068          GO TO 604
      ISN 0069      603 IF(IFIX.EQ.2)GO TO 131
      ISN 0071          IYR=0
      ISN 0072          ILINK=0
      ISN 0073          DO 762 IQ=9,65
      ISN 0074          IF(DATA(IQ).EQ.SPX)GO TO 762
      ISN 0076          IF(ILINK.NE.0)GO TO 763
      ISN 0078          ILINK=DATA(IQ)-BIN
      ISN 0079      762 CONTINUE
      ISN 0080      766 WRITE( 6,765)
      ISN 0081      765 FORMAT(20H DATE NOT IN LINE 1 )
      ISN 0082          WRITE( 6,183)IAMP,SAT
      ISN 0083          GO TO 604
      ISN 0084      763 IYR=((DATA(IQ)-BIN)*10.)+(DATA(IQ+1)-BIN)
      ISN 0085          IYEAR=((DATA(IQ)-BIN)*100000.)+((DATA(IQ+1)-BIN)*10000.)+((DATA(IQ
                       X+2)-BIN)*1000.)+((DATA(IQ+3)-BIN)*100.)+((DATA(IQ+4)-BIN)*10.)+(DA
                       XTA(IQ+5)-BIN)
      ISN 0086          IF(ILINK.NE.1)GO TO 740
      ISN 0088          FREQ=FRQ1(M)
      ISN 0089          GO TO 744
      ISN 0090      740 IF(ILINK.NE.2)GO TO 741
      ISN 0092          FREQ=FRQ2(M)
      ISN 0093          GO TO 744
      ISN 0094      741 IF(ILINK.NE.3)GO TO 743
      ISN 0096          FREQ=FRQ3(M)
```

Step 4

Step 5

continued

continued

```
      ISN 0097            GO TO 744
      ISN 0099        743 IF(ILINK.NE.4)GO TO 745
      ISN 0100            FREQ=FRQ4(M)
      ISN 0101            GO TO 744
      ISN 0102        131 FREQ=FRQ1(M)
      ISN 0103            IYR=IDT
      ISN 0104            IYEAR=IYR*10000
      ISN 0105            ILINK=1                                                          Step 5
      ISN 0106            IF(DATA(2)-BIN.EQ.6)GO TO 302
      ISN 0108            SATT=6*1000000.
      ISN 0109            SATT=SATT+((DATA(2)-BIN)*100000.)+((DATA(3)-BIN)*10000.)+((DATA(4)
                         X-BIN)*1000.)+((DATA(5)-BIN)*100.)+(DATA(6)-BIN)
      ISN 0110            GO TO 764
      ISN 0111        302 SATT=((DATA(2)-BIN)*1000000.)+((DATA(3)-BIN)*100000.)+((DATA(4)-BI
                         XN)*1000.)+((DATA(5)-BIN)*100.)+(DATA(6)-BIN)
      ISN 0112            GO TO 764
      ISN 0113        745 WRITE( 6,746)
      ISN 0114        746 FORMAT(24H FREQ.LINK NOT IN TABLE )
      ISN 0115            WRITE( 6,183)IAMP,SAT
      ISN 0116            GO TO 604
                    C     SEVEN DIGIT SATELLITE FOR DODS ******************************************
      ISN 0117        744 IF(FREQ.LT.136.0)GO TO 745
      ISN 0119            IF(FREQ.GT.138.0)GO TO 745
      ISN 0121            SATT=((DATA(2)-BIN)*1000000.)+((DATA(3)-BIN)*100000.)+((DATA(4)-BI
                         XN)*10000.)+((DATA(5)-BIN)*1000.)+((DATA(6)-BIN)*100.)+((DATA(7)-BI
                         XN)*10.)+(DATA(8)-BIN)
                    C     READ CAL.LINE WITH FORMAT CHECK ********************************
      ISN 0122        764 READ( 9,601,END= 1,ERR=30)(DATA(I),I=1,65)
      ISN 0123            IF(DATA(1).EQ.IAMP)GO TO 85
                    C     CHECK PERIODS IN CAL. LINE
      ISN 0125            IF(DATA(5).NE.PEZ)GO TO 767
      ISN 0127            IF(DATA(13).NE.PEZ)GO TO 767
      ISN 0129            IF(DATA(18).NE.PEZ)GO TO 767
      ISN 0131            IF(DATA(26).NE.PEZ)GO TO 767
      ISN 0133            IF(DATA(31).NE.PEZ)GO TO 767
      ISN 0135            IF(DATA(39).NE.PEZ)GO TO 767
      ISN 0137            IF(DATA(45).NE.PEZ)GO TO 767
      ISN 0139            IF(DATA(53).NE.PEZ)GO TO 767
      ISN 0141            IF(DATA(57).NE.PEZ)GO TO 767
      ISN 0143            IF(DATA(65).NE.PEZ)GO TO 767
      ISN 0145            DO 605 K=1,4                                                     Step 6
      ISN 0146            IF(DATA(K)-BIN.GT.9)GO TO 767
      ISN 0148            IF(DATA(K)-BIN.LT.0)GO TO 767
      ISN 0150            DATA(K)=DATA(K)-BIN
      ISN 0151        605 CONTINUE
      ISN 0152            DO 606 K=6,12
      ISN 0153            IF(DATA(K)-BIN.GT.9)GO TO 767
      ISN 0155            IF(DATA(K)-BIN.LT.0)GO TO 767
      ISN 0157            DATA(K)=DATA(K)-BIN
      ISN 0158        606 CONTINUE
      ISN 0159            DO 607 K=14,17
      ISN 0160            IF(DATA(K)-BIN.GT.9)GO TO 767
      ISN 0162            IF(DATA(K)-BIN.LT.0)GO TO 767
      ISN 0164            DATA(K)=DATA(K)-BIN
      ISN 0165        607 CONTINUE
      ISN 0166            DO 608 K=19,25
```

continued

continued

```
ISN 0167           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0169           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0171           DATA(K)=DATA(K)-BIN
ISN 0172       608 CONTINUE
ISN 0173           DO 609 K=27,30
ISN 0174           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0176           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0178           DATA(K)=DATA(K)-BIN
ISN 0179       609 CONTINUE
ISN 0180           DO 610 K=32,38
ISN 0181           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0183           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0185           DATA(K)=DATA(K)-BIN
ISN 0186       610 CONTINUE
ISN 0187           DO 611 K=40,44
ISN 0188           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0190           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0192           DATA(K)=DATA(K)-BIN
ISN 0193       611 CONTINUE
ISN 0194           DO 612 K=46,52
ISN 0195           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0197           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0199           DATA(K)=DATA(K)-BIN
ISN 0200       612 CONTINUE
ISN 0201           DO 613 K=54,56
ISN 0202           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0204           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0206           DATA(K)=DATA(K)-BIN
ISN 0207       613 CONTINUE
ISN 0208           DO 614 K=58,64
ISN 0209           IF(DATA(K)-BIN.GT.9)GO TO 767
ISN 0211           IF(DATA(K)-BIN.LT.0)GO TO 767
ISN 0213           DATA(K)=DATA(K)-BIN
ISN 0214       614 CONTINUE
ISN 0215           X=DATA(9)+DATA(22)+DATA(35)+DATA(49)+DATA(61)
ISN 0216           IF(X.NE.45.)GO TO 769
ISN 0218           CSTA=(DATA(55)*10.)+(DATA(56))
ISN 0219           DO 616 L=1,12
ISN 0220           IF(KSTA(L).EQ.CSTA)GO TO 617
ISN 0222       616 CONTINUE
ISN 0223           WRITE( 6,618)
ISN 0224       618 FORMAT(27H WRONG STATION IN CAL.LINE )
ISN 0225           GO TO 604
ISN 0226       617 D=DATA(3)*10+DATA(4)
ISN 0227           CEWM=D/100.
ISN 0228           D=DATA(16)*10+DATA(17)
ISN 0229           CEWC=D/100.
ISN 0230           D=DATA(6)*100+DATA(7)*10+DATA(8)
ISN 0231           CEWF1=D/1000.
ISN 0232           D=DATA(19)*100+DATA(20)*10+DATA(21)
ISN 0233           CEWF2=D/1000.
ISN 0234           D=DATA(32)*100+DATA(33)*10+DATA(34)
ISN 0235           CEWF3=D/1000.
ISN 0236           D=DATA(46)*100+DATA(47)*10+DATA(48)
ISN 0237           CEWF4=D/1000.
ISN 0238           D=DATA(58)*100+DATA(59)*10+DATA(60)
```

Step 6

continued

continued

```
      ISN 0239              CEWF5=D/1000.
      ISN 0240              D=DATA(10)*100+DATA(11)*10+DATA(12)
      ISN 0241              CNSF1=D/1000.
      ISN 0242              D=DATA(23)*100+DATA(24)*10+DATA(25)
      ISN 0243              CNSF2=D/1000.
      ISN 0244              D=DATA(36)*100+DATA(37)*10+DATA(38)
      ISN 0245              CNSF3=D/1000.
      ISN 0246              D=DATA(50)*100+DATA(51)*10+DATA(52)
      ISN 0247              CNSF4=D/1000.
      ISN 0248              D=DATA(62)*100+DATA(63)*10+DATA(64)
      ISN 0249              CNSF5=D/1000.
      ISN 0250              D=DATA(29)*10+DATA(30)
      ISN 0251              CNSM=D/100.
      ISN 0252              D=DATA(43)*10+DATA(44)
      ISN 0253              CNSC=D/100.
      ISN 0254              CSTA=DATA(55)*10+DATA(56)
      ISN 0255              WRITE(11,501)IAMP,SAT,CSTA,STATID(L),ILINK,IYEAR
      ISN 0256          501 FORMAT(X,A1,I5,X,I2,X,A6,X,I1,X,I7)
      ISN 0257              CANT=DATA(54)
      ISN 0258              END=DATA(66)
      ISN 0259           43 CEWM=EWM(L)+CEWM
      ISN 0260              CEWC=EWC(L)+CEWC
      ISN 0261              CNSM=NSM(L)+CNSM
      ISN 0262              CNSC=NSC(L)+CNSC
      ISN 0263              KM=5.
      ISN 0264              RATE=0.
      ISN 0265              IDIF1=CEWF2-CEWF1
      ISN 0266              IDIF2=CEWF3-CEWF2
      ISN 0267              IDIF3=CEWF4-CEWF3
      ISN 0268              IDIF4=CEWF5-CEWF4
      ISN 0269              CEWF1=CEWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
      ISN 0270              IF(CANT-2.)162,160,161
                      C     NARROW BAND TRACKING FILTER ****************************************
      ISN 0271          160 IDIF=.0
      ISN 0272              AST(1)=SPA
      ISN 0273              GO TO 164
      ISN 0274          161 IDIF=.0
      ISN 0275              AST(1)=ASK
      ISN 0276              GO TO 164
      ISN 0277          162 IF(CANT.EQ.1.)GO TO 164
      ISN 0279          163 IDIF=.120
                      C     2 CPS TRACKING FILTER ***********************************************
      ISN 0280              AST(1)=PER
      ISN 0281          164 IDIF1=CNSF2-CNSF1
      ISN 0282              IDIF2=CNSF3-CNSF2
      ISN 0283              IDIF3=CNSF4-CNSF3
      ISN 0284              IDIF4=CNSF5-CNSF4
      ISN 0285              CNSF1=CNSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
                      C     CABLE LENGTH INEQUALITIES
      ISN 0286              CEWM=((CLEWM(L)/.846)*(136.5-FREQ))+CEWM
      ISN 0287              CEWC=((CLEWC(L)/.846)*(136.5-FREQ))+CEWC
      ISN 0288              CNSM=((CLNSM(L)/.846)*(136.5-FREQ))+CNSM
      ISN 0289              CNSC=((CLNSC(L)/.846)*(136.5-FREQ))+CNSC
      ISN 0290              WRITE(11,636)
      ISN 0291          636 FORMAT(55H CALIBRATED PHASE READINGS,5 POINT FITTED FINE READINGS)
                      C     CALIBRATED ZENITH ***************************************************
```

Step 6

continued

continued

```
     ISN 0292            WRITE(11,639)
     ISN 0293        639 FORMAT(63H    CEWM          CEWC          CNSM          CNSC          CEWF
                        X   CNSF  )
     ISN 0294            WRITE(11,150)CEWM,CEWC,CNSM,CNSC,CEWF1,CNSF1
     ISN 0295        150 FORMAT(6(X,F10.6))
     ISN 0296            WRITE(11,640)
     ISN 0297        640 FORMAT(74H HRMNSC      EWFINE      EWMEDM      EWCORS      NSFINE
                        XNSMEDM      NSCORS  )
                   C     READ DATA WITH FORMAT CHECK*************************************
     ISN 0298            DO 41 JK=1,60
     ISN 0299            K=JK
     ISN 0300            IF(K.EQ.32.)GO TO 120
     ISN 0302        621 READ( 9,601,END=1,ERR=621)(DATA(I),I=1,65)
     ISN 0303            IF(DATA(10).EQ.SPX)GO TO 122
     ISN 0305            GO TO 124
     ISN 0306        122 IF(DATA(30).EQ.SPX)GO TO 123
     ISN 0308            GO TO 124
     ISN 0309        123 IF(DATA(50).EQ.SPX)GO TO 120
     ISN 0311            IF(DATA(1).EQ.IAMP)GO TO 86
     ISN 0313        124 IF(DATA(6).NE.PEZ)GO TO 621
                   C     CHECK PERIODS IN DATA LINE
     ISN 0315            IF(DATA(13).NE.PEZ)GO TO 621
     ISN 0317            IF(DATA(18).NE.PEZ)GO TO 621
     ISN 0319            IF(DATA(26).NE.PEZ)GO TO 621
     ISN 0321            IF(DATA(31).NE.PEZ)GO TO 621
     ISN 0323            IF(DATA(39).NE.PEZ)GO TO 621
     ISN 0325            IF(DATA(45).NE.PEZ)GO TO 621
     ISN 0327            IF(DATA(53).NE.PEZ)GO TO 621
     ISN 0329            IF(DATA(57).NE.PEZ)GO TO 621
     ISN 0331            IF(DATA(65).NE.PEZ)GO TO 621
     ISN 0333            DO 622 J=1,4
     ISN 0334            IF(DATA(J)-BIN.GT.9)GO TO 621
     ISN 0336            IF(DATA(J)-BIN.LT.0)GO TO 621
     ISN 0338            DATA(J)=DATA(J)-BIN
     ISN 0339        622 CONTINUE
     ISN 0340            DO 623 J=6,12
     ISN 0341            IF(DATA(J)-BIN.GT.9)GO TO 621
     ISN 0343            IF(DATA(J)-BIN.LT.0)GO TO 621
     ISN 0345            DATA(J)=DATA(J)-BIN
     ISN 0346        623 CONTINUE
     ISN 0347            DO 624 J=14,17
     ISN 0348            IF(DATA(J)-BIN.GT.9)GO TO 621
     ISN 0350            IF(DATA(J)-BIN.LT.0)GO TO 621
     ISN 0352            DATA(J)=DATA(J)-BIN
     ISN 0353        624 CONTINUE
     ISN 0354            DO 625 J=19,25
     ISN 0355            IF(DATA(J)-BIN.GT.9)GO TO 621
     ISN 0357            IF(DATA(J)-BIN.LT.0)GO TO 621
     ISN 0359            DATA(J)=DATA(J)-BIN
     ISN 0360        625 CONTINUE
     ISN 0361            DO 626 J=27,30
     ISN 0362            IF(DATA(J)-BIN.GT.9)GO TO 621
     ISN 0364            IF(DATA(J)-BIN.LT.0)GO TO 621
     ISN 0366            DATA(J)=DATA(J)-BIN
     ISN 0367        626 CONTINUE
     ISN 0368            DO 627 J=32,38
```

Step 6

Step 7

continued

continued

```
      ISN 0369          IF(DATA(J)-BIN.GT.9)GO TO 621
      ISN 0371          IF(DATA(J)-BIN.LT.0)GO TO 621
      ISN 0373          DATA(J)=DATA(J)-BIN
      ISN 0374      627 CONTINUE
      ISN 0375          DO 628 J=40,44
      ISN 0376          IF(DATA(J)-BIN.GT.9)GO TO 621
      ISN 0378          IF(DATA(J)-BIN.LT.0)GO TO 621
      ISN 0380          DATA(J)=DATA(J)-BIN
      ISN 0381      628 CONTINUE
      ISN 0382          DO 629 J=46,52
      ISN 0383          IF(DATA(J)-BIN.GT.9)GO TO 621
      ISN 0385          IF(DATA(J)-BIN.LT.0)GO TO 621
      ISN 0387          DATA(J)=DATA(J)-BIN
      ISN 0388      629 CONTINUE
      ISN 0389          DO 630 J=54,56
      ISN 0390          IF(DATA(J)-BIN.GT.9)GO TO 621
      ISN 0392          IF(DATA(J)-BIN.LT.0)GO TO 621
      ISN 0394          DATA(J)=DATA(J)-BIN
      ISN 0395      630 CONTINUE
      ISN 0396          DO 631 J=58,64
      ISN 0397          IF(DATA(J)-BIN.GT.9)GO TO 621
      ISN 0399          IF(DATA(J)-BIN.LT.0)GO TO 621
      ISN 0401          DATA(J)=DATA(J)-BIN
      ISN 0402      631 CONTINUE
      ISN 0403          SECD(K)=DATA(1)*10+DATA(2)
      ISN 0404          MIND(K)=DATA(14)*10+DATA(15)
      ISN 0405          HORD(K)=DATA(27)*10+DATA(28)
      ISN 0406          ANTD(K)=DATA(54)
      ISN 0407          IDAYD(K)=DATA(40)*100+DATA(41)*10+DATA(42)
      ISN 0408          STAD(K)=DATA(55)*10+DATA(56)
      ISN 0409          D=DATA(3)*10+DATA(4)
      ISN 0410          EWMD(K)=D/100.
      ISN 0411          D=DATA(16)*10+DATA(17)
      ISN 0412          EWCD(K)=D/100.
      ISN 0413          D=DATA(29)*10+DATA(30)
      ISN 0414          NSMD(K)=D/100.
      ISN 0415          D=DATA(43)*10+DATA(44)
      ISN 0416          NSCD(K)=D/100.
      ISN 0417          D=DATA(6)*100+DATA(7)*10+DATA(8)
      ISN 0418          EWE1=D/1000.
      ISN 0419          D=DATA(19)*100+DATA(20)*10+DATA(21)
      ISN 0420          EWE2=D/1000.
      ISN 0421          D=DATA(32)*100+DATA(33)*10+DATA(34)
      ISN 0422          EWE3=D/1000.
      ISN 0423          D=DATA(46)*100+DATA(47)*10+DATA(48)
      ISN 0424          EWE4=D/1000.
      ISN 0425          D=DATA(58)*100+DATA(59)*10+DATA(60)
      ISN 0426          EWE5=D/1000.
      ISN 0427          D=DATA(10)*100+DATA(11)*10+DATA(12)
      ISN 0428          NSE1=D/1000.
      ISN 0429          D=DATA(23)*100+DATA(24)*10+DATA(25)
      ISN 0430          NSE2=D/1000.
      ISN 0431          D=DATA(36)*100+DATA(37)*10+DATA(38)
      ISN 0433          D=DATA(50)*100+DATA(51)*10+DATA(52)
      ISN 0434          NSE4=D/1000.
```

Step 7

continued

continued

```
ISN 0435         D=DATA(62)*100+DATA(63)*10+DATA(64)
ISN 0436         NSF5=D/1000.
ISN 0437         END=DATA(65)
ISN 0438     100 IF(ANTD(K)-2)102,101,101
ISN 0439     101 ANT=57.
ISN 0440         GO TO 103
ISN 0441     102 ANT=46.
ISN 0442     103 TIM(K)=((HORD(K)*3600.)+(60.*MIND(K)))+SECD(K)
              C  FIT FIVE FINES EACH LINE
ISN 0443         IDIF1=EWF2-EWF1
ISN 0444         CALL NORMAL(IDIF1)
ISN 0445         IDIF2=EWF3-EWF2
ISN 0446         CALL NORMAL(IDIF2)
ISN 0447         IDIF3=EWF4-EWF3
ISN 0448         CALL NORMAL(IDIF3)
ISN 0449         IDIF4=EWF5-EWF4
ISN 0450         CALL NORMAL(IDIF4)
ISN 0451         EEWF(K)=EWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
ISN 0452         IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
              C  COUNTER DELAY (TIME) **********************************************
ISN 0453         EEWF(K)=(EEWF(K)-(.05*IDIF5*EWF3))
ISN 0454         IDIF1=NSF2-NSF1
ISN 0455         CALL NORMAL(IDIF1)
ISN 0456         IDIF2=NSF3-NSF2
ISN 0457         CALL NORMAL(IDIF2)
ISN 0458         IDIF3=NSF4-NSF3
ISN 0459         CALL NORMAL(IDIF3)
ISN 0460         IDIF4=NSF5=NSF4
ISN 0461         CALL NURMAL(IDIF4)
ISN 0462         ENSF(K)=NSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
ISN 0463         IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
              C  FILTER DELAY (TIME)
ISN 0464         ENSF(K)=ENSF(K)-(.05*IDIF5*NSF3)
ISN 0465         WRITE(11,151)HORD(K),MIND(K),SECD(K),EEWF(K),EWMD(K),EWCD(K),ENSF(
                XK),NSMD(K),NSCD(K)
ISN 0466     151 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)))
ISN 0467      41 CONTINUE
              C  DATA MSG.COMPLETE.START SMOOTHING ********************************
ISN 0468     120 SL=.01
ISN 0469         AST(6)=SPA
ISN 0470         K=K-1
ISN 0471         IF(K.LE.5)GO TO 780
ISN 0473         GO TO 782
ISN 0474     780 WRITE( 6,781)
ISN 0475     781 FORMAT(27H LESS THAN 5 LINES OF DATA )
ISN 0476         KZ=K-1
ISN 0477         GO TO 78
ISN 0478     782 JK=K
ISN 0479         IF(ANT.EQ.57.) GO TO 90
ISN 0481         GO TO 91
ISN 0482      90 CEWF1=CEWF1+EWFPO(L)
ISN 0483         CNSF1=CNSF1+NSFPO(L)
ISN 0484         AST(2)=POL
ISN 0485         GO TO 121
ISN 0486      91 CEWF1=CEWF1+EWFEQ(L)
ISN 0487         CNSF1=CNSF1+NSFEQ(L)
```

Step 7

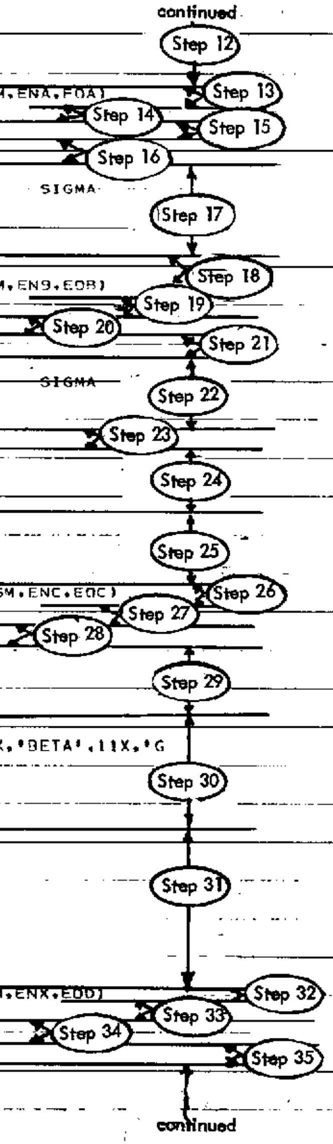
Step 8

continued

```
                                                                              continued
                                                                               Step 8          PAGE 009
      ISN 0488             AST(2)=EQ
      ISN 0489        121  SM=2.0
      ISN 0490             WRITE(11,638)
      ISN 0491        638  FORMAT(32H KC-KS1 USING STATION CONSTANTS )                         Step 9
      ISN 0492             WRITE(11,1152)CEWF1,EWFPO(L),EWFEQ(L),CNSF1,NSFPO(L),NSFEQ(L)
                        C  CHECK TIME SEQUENCE **************************************************
      ISN 0493             IB=0
      ISN 0494             IC=0
      ISN 0495             ID=0
      ISN 0496             IE=0
      ISN 0497             IF=0
      ISN 0498             IG=0
      ISN 0499             IH=0
      ISN 0500             KQ=K-1
      ISN 0501             DO 20 KS=1,KQ
      ISN 0502        806  ITZM=TIM(KS+1)-TIM(KS)
      ISN 0503             IF(ITZM)807,807,808
      ISN 0504        807  TIM(KS+1)=TIM(KS+1)+86400.
      ISN 0505             GO TO 806
      ISN 0506        808  IF(ITZM.NE.1)GO TO 21
      ISN 0508             IB=IB+1
      ISN 0509             GO TO 20
      ISN 0510         21  IF(ITZM.NE.2 )GO TO 22
      ISN 0512             IC=IC+1
      ISN 0513             GO TO 20
      ISN 0514         22  IF(ITZM.NE.10 )GO TO 23
      ISN 0516             ID=ID+1                                                          Step 10
      ISN 0517             GO TO 20
      ISN 0518         23  IF(ITZM.NE.20 )GO TO 24
      ISN 0520             IE=IE+1
      ISN 0521             GO TO 20
      ISN 0522         24  IF(ITZM.NE.60 )GO TO 25
      ISN 0524             IF=IF+1
      ISN 0525             GO TO 20
      ISN 0526         25  IF(ITZM.NE.120 )GO TO 26
      ISN 0528             IG=IG+1
      ISN 0529             GO TO 20
      ISN 0530         26  IF(ITZM.NE.600 )GO TO 20
      ISN 0532             IH=IH+1
      ISN 0533             GO TO 20
      ISN 0534         20  CONTINUE
      ISN 0535             ITZM=MAX0(IB,IC,ID,IE,IF,IG,IH)
      ISN 0536             IF(IB.EQ.ITZM)GO TO 200
      ISN 0538             IF(IC.EQ.ITZM)GO TO 201
      ISN 0540             IF(ID.EQ.ITZM)GO TO 202
      ISN 0542             IF(IE.EQ.ITZM)GO TO 203
      ISN 0544             IF(IF.EQ.ITZM)GO TO 204
      ISN 0546             IF(IG.EQ.ITZM)GO TO 205
      ISN 0548             IF(IH.EQ.ITZM)GO TO 206
      ISN 0550        200  AST(3)=F1
      ISN 0551             TIL1=1.
      ISN 0552             GO TO 207
      ISN 0553        201  AST(3)=F2
      ISN 0554             TIL1=2.
      ISN 0555             GO TO 207
      ISN 0556        202  AST(3)=F3                                                        continued
```

```
ISN 0557           TILI=10.                                              continued
ISN 0558           GO TO 207
ISN 0559       203 AST(3)=F4
ISN 0560           TILI=20.
ISN 0561           GO TO 207
ISN 0562       204 AST(3)=F5                                             Step 10
ISN 0563           TILI=60.
ISN 0564           GO TO 207
ISN 0565       205 AST(3)=F6
ISN 0566           TILI=120.
ISN 0567           GO TO 207
ISN 0568       206 AST(3)=F7
ISN 0569           TILI=600.
ISN 0570           GO TO 207
ISN 0571        72 WRITE( 6,73)
ISN 0572        73 FORMAT(22H TIME OUT OF SEQUANCE )
ISN 0573           KZ=K-1
ISN 0574        78 WRITE( 6,77)SAT,CSTA,STATIO(L),HORD(KZ),MIND(KZ),IDAYD(KZ)
ISN 0575        77 FORMAT(X,I5,X,I2,X,A6,X,I2,I2,X,I3)
ISN 0576           GO TO 65
ISN 0577        74 WRITE( 6,75)                                           Step 11
ISN 0578        75 FORMAT(25H DATA EXCEEDS TIME CHECK )              (see below)
ISN 0579           KZ=K-1
ISN 0580           GO TO 78
ISN 0581        50 WRITE(11,51)
ISN 0582        51 FORMAT(40H EAST MEDIUM CHANNEL EXCEEDS 100 COUNTS )
ISN 0583           AST(6)=F1
ISN 0584           RATE=0.
ISN 0585           GO TO 58                                          Part Step 24
ISN 0586        52 WRITE(11,53)
ISN 0597        53 FORMAT(40H EAST COURSE CHANNEL EXCEEDS 100 COUNTS )
ISN 0588           AST(6)=F1
ISN 0589           RATE=0.
ISN 0590           GO TO 58
ISN 0591        54 WRITE(11,55)
ISN 0592        55 FORMAT(41H NORTH MEDIUM CHANNEL EXCEEDS 100 COUNTS )
ISN 0593           AST(6)=F2
ISN 0594           RATE=0.
ISN 0595           GO TO 59                                          Part Step 43
ISN 0596        56 WRITE(11,57)
ISN 0597        57 FORMAT(41H NORTH COURSE CHANNEL EXCEEDS 100 COUNTS )
ISN 0598           AST(6)=F2
ISN 0599           RATE=0.
ISN 0600           GO TO 59
ISN 0601       804 WRITE(11,805)
ISN 0602       805 FORMAT(27H DATA WILL NOT LOBE ASSIGN )
ISN 0603           KZ=K-1                                      Parts — Steps 25 & 44
ISN 0604           GO TO 78
             C     EW AMBIGUITY LOBE ASSIGN.
ISN 0605       207 SL=.01
ISN 0606           K=JK-1
ISN 0607           DO 70 N=1,K                                            Step 11
ISN 0608           IF(TIM(N+1)-TIM(N))72,72,71
ISN 0609        71 IF(TIM(N+1)-TIM(N)-(6.*TILI))70,70,74             (see above)
ISN 0610        70 CONTINUE
ISN 0611           K=JK
                                                                          Step 12
                                                                         continued
```

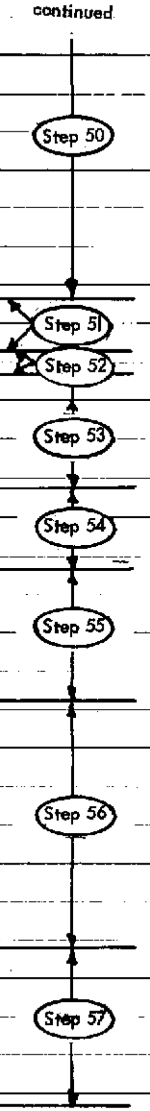
```
ISN 0612          CALL LOBASN(TIM,K,FWMD,RATE,ITD)
ISN 0613          IF(ITD.GE.100)GO TO 804
ISN 0615          CALL LOBASN(TIM,K,EWCO,RATE,ITD)
ISN 0616          IF(ITD.GE.100)GO TO 804
ISN 0618          CALL LSQQUA(TIM,ALPHA,EWMD,K,MID,SIA,SL,FWMB,EWMC,SM,ENA,FOA)
ISN 0619          EWMA=ALPHA+EWMD(MID)
ISN 0620          EWMT=TIM(MID)-.15
ISN 0621          SA=SIA
ISN 0622          WRITE(11,643)
ISN 0623      643 FORMAT(83H          ALPHA,EWM    MID.PT.    RATE          SIGMA
                 X     BETA          GAMMA    )
ISN 0624          WRITE(11,152)ALPHA,MID,RATE,SIA,EWMB,EWMC
ISN 0625      152 FORMAT(X,F14.6,X,I6.X,4(F14.6,X))
ISN 0626          K=JK
ISN 0627          CALL LSQQUA(TIM,ALPHA,FWCO,K,MID,SIA,SL,EWCB,EWCC,SM,EN9,EOB)
ISN 0628          EWCA=ALPHA+FWCO(MID)
ISN 0629          FWCT=TIM(MID)+.05
ISN 0630          SB=SIA
ISN 0631          WRITE(11,644)
ISN 0632      644 FORMAT(83H          ALPHA,EWC    MID.PT.    RATE          SIGMA
                 X     BETA          GAMMA    )
ISN 0633          WRITE(11,152)ALPHA,MID,RATE,SIA,EWCB,EWCC
ISN 0634          RATE=((EWMB*ANT/4.)+(EWCB*ANT/3.5))/2.
              C   EWF LOBE ASSIGN.
ISN 0635          IF(ABS(SB).GT..1)GO TO 52
ISN 0637          IF(ABS(SA).GT..1)GO TO 56
ISN 0639       58 SM=2.5
ISN 0640          K=JK
ISN 0641          CALL LOBASN(TIM,K,EEWF,RATE,ITD)
ISN 0642          IF(ITD.GE.100)GO TO 804
ISN 0644          CALL LSCUB(TIM,ALPHA,EEWF,K,MID,SIA,EWFB,EWFC,EWFE,SM,ENC,EOC)
ISN 0645          EWFT=TIM(MID)
ISN 0646          EWFA=ALPHA+EEWF(MID)
ISN 0647      224 SC=SIA
ISN 0648          IND=SIA*1000.+.5
ISN 0649          CALL ZERO(4,IND,IEND)
ISN 0650          IDAYD(1)=IDAYD(MID)
ISN 0651          WRITE(11,642)
ISN 0652      642 FORMAT(8X,'ALPHA.EWF    MID.PT.   RATE',11X,'SIGMA',9X,'BETA',11X,'G
                 XAMA',10X,'DELTA ')
ISN 0653          WRITE(11,154)ALPHA,MID,RATE,SIA,EWFB,EWFC,EWFE
ISN 0654          WRITE(11,153)EWMA,EWMT,EWCA,EWCT,RATE
ISN 0655      153 FORMAT(6(F14.6,X))
              C   NS AMBIGUITY LOBE ASSIGN.
ISN 0656          RATE=0.
ISN 0657          K=JK
ISN 0658          SM=2.0
ISN 0659          CALL LOBASN(TIM,K,NSMD,RATE,ITD)
ISN 0660          IF(ITD.GE.100)GO TO 804
ISN 0662          CALL LOBASN(TIM,K,NSCD,RATE,ITD)
ISN 0663          IF(ITD.GE.100)GO TO 804
ISN 0665          CALL LSQQUA(TIM,ALPHA,NSMD,K,MID,SIA,SL,SNMB,SNMC,SM,ENX,EDD)
ISN 0666          SNMA=ALPHA+NSMD(MID)
ISN 0667          SNMT=TIM(MID)+.25
ISN 0668          SD=SIA
ISN 0669          WRITE(11,646)
```



```
                                                                                              continued              PAGE 012
ISN 0670        646 FORMAT(83H          ALPHA.NSM    MID.PT.    RATE              SIGMA      Step 36
                   X      BETA              GAMMA    )
ISN 0671            WRITE(11,152)ALPHA,MID,RATE,SIA,SNMA,SNMC
ISN 0672            K=JK                                                                        Step 37
ISN 0673            CALL LSQQUA(TIM,ALPHA,NSCD,K,MID,SIA,SL,SNCB,SNCC,SM,ENE,EDE)
ISN 0674            SNCA=ALPHA+NSCD(MID)                                                    Step 38
ISN 0675            SNCT=TIM(MID)+.45                                                 Step 39
ISN 0676            SE=SIA                                                                    Step 40
ISN 0677            RATE=(((SNMB*ANT/4.)+(SNCB*ANT/3.5))/2.)          *                    Step 42
ISN 0678            WRITE(11,647)
ISN 0679        647 FORMAT(83H          ALPHA.NSC    MID.PT.    RATE              SIGMA
                   X      BETA              GAMMA    )                                      Step 41
ISN 0680            WRITE(11,152)ALPHA,MID,RATE,SIA,SNCB,SNCC
                  C NSF LOBE ASSIGN.
ISN 0681            IF(ABS(SD).GT..1)GO TO 54                                               Step 43
ISN 0683            IF(ABS(SF).GT..1)GO TO 56
ISN 0685         59 SM=2.5
ISN 0686            K=JK                                                                    Step 44
ISN 0687            CALL LOBASN(TIM,K,ENSF,RATE,ITD)
ISN 0688            IF(ITD.GE.100)GO TO 884
ISN 0690            CALL LSCUB(TIM,ALPHA,ENSF,K,MIO,SIA,SNFB,SNFC,SNFE,SM,ENF,EDF)          Step 45
ISN 0691            SNFT=TIM(MID)                                                       Step 46
ISN 0692            SNFA=ALPHA+ENSF(MID)                                          Step 47
ISN 0693        238 SF=SIA
ISN 0694            ISO=SIA*1000.+.5                                                        Step 48
ISN 0695            CALL ZERO(4,ISO,INNO)
ISN 0696        239 WRITE(11,645)
ISN 0697        645 FORMAT(8X,'ALPHA.NSF    MID.PT.    RATE',11X,'SIGMA',9X,'BETA',11X,'G
                   XAMA',10X,'DELTA ')
ISN 0698            WRITE(11,154)ALPHA,MID,RATE,SIA,SNFB,SNFG,SNFE                          Step 49
ISN 0699        154 FORMAT(X,F14.6,X,I6,X,5(F14.6,X))
ISN 0700            WRITE(11,153)SNMA,SNMT,SNCA,SNCT,RATE
ISN 0701            WRITE(11,777)
                  C LOBE ASSIGNED DATA AFTER THE FIT HAS BEEN APPLIED ***************
ISN 0702            WRITE(11,641)
ISN 0703        641 FORMAT(99H FRA.TIME    EWFINE    O-C    EWMEDM    O-C    EWCORS    O-C
                   XNSFINE    O-C    NSMEDM    O-C    NSCORS    O-C )
ISN 0704            K=JK
ISN 0705            DO 790 I=1,K
ISN 0706            T=TIM(I)-EWMT-.15
ISN 0707            AM=EWMA+(EWMB*T)+(EWMC*T**2)
ISN 0708            T=TIM(I)-EWCT+.05
ISN 0709            AN=EWCA+(EWCB*T)+(EWCC*T**2)
ISN 0710            T=TIM(I)-EWFT                                                           Step 50
ISN 0711            AO=EWFA+(EWFB*T)+(EWFC*T**2)+(EWFE*T**3)
ISN 0712            T=TIM(I)-SNMT+.25
ISN 0713            AP=SNMA+(SNMB*T)+(SNMC*T**2)
ISN 0714            T=TIM(I)-SNCT+.45
ISN 0715            AQ=SNCA+(SNCB*T)+(SNCC*T**2)
ISN 0716            T=TIM(I)-SNFT
ISN 0717            AR=SNFA+(SNFB*T)+(SNFC*T**2)+(SNFE*T**3)
ISN 0718            AO=EEWF(I)-AO
ISN 0719            AM=EWMD(I)-AM                                                          continued
ISN 0720            AN=EWCD(I)-AN
ISN 0721            AR=ENSF(I)-AR
ISN 0722            AP=NSMD(I)-AP
```

*Error — in this position will print out fine rate instead of the desired coarse rate

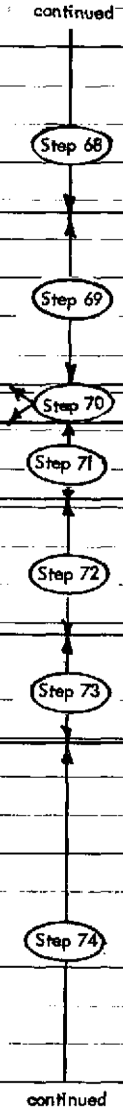
```
ISN 0723          AQ=NSCD(I)-AQ
ISN 0724          WRITE(11,791)TIM(I),EEWF(I),AO,EWMD(I),AM,EWCD(I),AN,ENSF(I),AR,NS
                 XMD(I),AP,NSCD(I),AQ
ISN 0725      791 FORMAT(X,F7.1,6(X,F7.3,X,F6.3))
ISN 0726      790 CONTINUE
ISN 0727          WRITE(11,793)
ISN 0728          WRITE(11,792)ENA,SA,EOA,ENB,SB,EOB,ENC,SC,EOC
ISN 0729          WRITE(11,794)
ISN 0730          WRITE(11,792)ENX,SD,EOD,ENE,SE,EOE,ENF,SF,EOF
ISN 0731      792 FORMAT(3(X,I2,4X,F5.3,8X,I2))
ISN 0732      793 FORMAT( 67H IN     EWM SIGMA     OT IN     EWC SIGMA     OT IN     EWF
                 XSIGMA     OT )
ISN 0733      794 FORMAT( 67H IN     NSM SIGMA     OT IN     NSC SIGMA     OT IN     NSF
                 XSIGMA     OT )
             C    ADJUST TIME TO EWFINE FITTED TIME
ISN 0734          XKFA=(KFA(L)/1000.)+IDIF
ISN 0735          XKFB=(KFB(L)/1000.)+IDIF
ISN 0736          TCOR=EWFT-SNFT-XKFB
ISN 0737          EWFA=EWFA+EWFB*(-XKFA)+EWFC*((-XKFA)**2)+EWFE*((-XKFA)**3)
ISN 0738          EWFB=EWFB+2.*EWFC*(-XKFA)+3.*EWFE*(-XKFA)**2
ISN 0739          EWFC=EWFC+3.*EWFE*(-XKFA)
ISN 0740          SNFA=SNFA+SNFB*TCOR+SNFC*TCOR**2+SNFE*TCOR**3
ISN 0741          SNFB=SNFB+2.*SNFC*TCOR+3.*SNFE*TCOR**2
ISN 0742          SNFC=SNFC+3.*SNFE*TCOR
ISN 0743          EWMA=EWMA+EWMB*(EWFT-EWMT)+EWMC*((EWFT-EWMT)**2)
ISN 0744          EWCA=EWCA+EWCB*(EWFT-EWCT)+EWCC*((EWFT-EWCT)**2)
ISN 0745          SNMA=SNMA+SNMB*(EWFT-SNMT)+SNMC*((EWFT-SNMT)**2)
ISN 0746          SNCA=SNCA+SNCB*(EWFT-SNCT)+SNCC*((EWFT-SNCT)**2)
ISN 0747      118 EWFBB=EWFA-CEWF1
             C    PHASE ANGLE (FITTED) MINUS KS2+KC-KS1
ISN 0748          EWMBB=EWMA-CEWM
ISN 0749          EWCBB=EWCA-CEWC
ISN 0750          SNFBB=SNFA-CNSF1
ISN 0751          SNMBB=SNMA-CNSM
ISN 0752          SNCBB=SNCA-CNSC
             C    REMOVE LOBE INTERGER
ISN 0753          LDIF=EWFBB
ISN 0754          EWFBB=EWFBB-LDIF
ISN 0755          LDIF=EWMBB
ISN 0756          EWMBB=EWMBB-LDIF
ISN 0757          LDIF=EWCBB
ISN 0758          EWCBB=EWCBB-LDIF
ISN 0759          LDIF=SNFBB
ISN 0760          SNFBB=SNFBB-LDIF
ISN 0761          LDIF=SNMBB
ISN 0762          SNMBB=SNMBB-LDIF
ISN 0763          LDIF=SNCBB
ISN 0764          SNCBB=SNCBB-LDIF
ISN 0765          AB=EWMBB-EWCBB
ISN 0766          LDIF=AB
ISN 0767          AB=AB-LDIF
ISN 0768          CALL NORMAL(AB)
ISN 0769          BC=SNMBB-SNCBB
ISN 0770          LDIF=BC
ISN 0771          BC=BC-LDIF
ISN 0772          CALL NORMAL(BC)
```



```
ISN 0773          AB4=AB*8.                                                   Step 58
ISN 0774          BC4=BC*8.
ISN 0775          AB3=AB*7.
ISN 0776          BC3=BC*7.
ISN 0777          EAB4=AB4-EWMBB
ISN 0778          LDIF=EAB4
ISN 0779          EAB4=EAB4-LDIF
ISN 0780          CALL NORMAL(EAB4)
ISN 0781          EBC4=BC4-SNMBB                                              Step 59
ISN 0782          LDIF=EBC4
ISN 0783          EBC4=EBC4-LDIF
ISN 0784          CALL NORMAL(EBC4)
ISN 0785          DE=AB4-EAB4
ISN 0786          FG=BC4-EBC4
ISN 0787          IF(ABS(DE).GT.4.0)DE=DE-SIGN(8.0,DE)                        Step 60
ISN 0789          IF(ABS(FG).GT.4.0)FG=FG-SIGN(8.0,FG)
ISN 0791          EAB3=AB3-EWCBB
ISN 0792          LDIF=EAB3
ISN 0793          EAB3=EAB3-LDIF
ISN 0794          CALL NORMAL(EAB3)
ISN 0795          EBC3=BC3-SNCBB_                                             Step 61
ISN 0796          LDIF=EBC3
ISN 0797          EBC3=EBC3-LDIF
ISN 0798          CALL NORMAL(EBC3)
ISN 0799          DE3=AB3-EAB3                                                Step 62
ISN 0800          FG3=BC3-EBC3
ISN 0801          EWFB=EWFB+2.*EWFC*(-XKFA)+3.*EWFE*(-XKFA)**2 Error already applied once Step 63
ISN 0802          IF(ABS(FG3).GT.3.5)FG3=FG3-SIGN(7.0,FG3)
ISN 0804          IF(ABS(DE3).GT.3.5)DE3=DE3-SIGN(7.0,DE3)                    Step 62
ISN 0806          CORE=DE+DE3
ISN 0807          CORN=FG+FG3                                                 Step 64
ISN 0808          COREF=(ANT/7.5)*CORE
ISN 0809          CORNF=(ANT/7.5)*CORN                                        Step 65
ISN 0810          FE=COREF-EWFBB
ISN 0811          LDIF=FE
ISN 0812          FE=FE-LDIF
ISN 0813          CALL NORMAL(FE)
ISN 0814          FN=CORNF-SNFBB
ISN 0815          LDIF=FN                                                     Step 66
ISN 0816          FN=FN-LDIF
ISN 0817          CALL NORMAL(FN)
ISN 0818          EWLOBE=COREF-FE
ISN 0819          SNLOBE=CORNF-FN
               C  AMBIGUITY ERRORS
ISN 0820          EWMER=4./ANT*EWLOBE-DE
ISN 0821          CALL NORMAL(EWMER)
ISN 0822          EWCER=3.5/ANT*EWLOBE-DE3
ISN 0823          CALL NORMAL(EWCER)                                          Step 67
ISN 0824          SNMER=4./ANT*SNLOBE-FG
ISN 0826          CALL NORMAL(SNMER)
ISN 0826          SNCER=3.5/ANT*SNLOBE-FG3
ISN 0827          CALL NORMAL(SNCER)
ISN 0828          ABB=AB*2
ISN 0829          ADE=DE/4
ISN 0830          ADE3=DE3/3.5
ISN 0831          ACORE=CORE/7.5                                              Step 68
```

continued

```
ISN 0832            AEWLOB=FWLOBE/ANT
ISN 0833            WRITE(11,1152)ABB,ADF3,ADE,ACORE,AEWLOB,EWFT
ISN 0834            ABC=BC*2
ISN 0835            AFG=FG3/3.5
ISN 0836            AFG3=FG3/3.5
ISN 0837            AFG=FG/4
ISN 0838            ACORN=CORN/7.5
ISN 0839            ASNLOB=SNLOBE/ANT
ISN 0840            WRITE(11,1152)ABC,AFG3,AFG,ACORN,ASNLOB,SNFT
ISN 0841       1152 FORMAT(6(X,F14.6))
ISN 0842            IF(ANTD(K)-2)650,651,651
ISN 0843        651 WRITE(11,648)
ISN 0844        648 FORMAT(12H POLAR PASS )
ISN 0845            GO TO 652
ISN 0846        650 WRITE(11,649)
ISN 0847        649 FORMAT(17H EQUATORIAL PASS )
ISN 0848            GO TO 652
ISN 0849        652 IAB=2
ISN 0850            IBO=3
ISN 0851            ACOS=EWLOBE/((FREQ/136.)*ANT)
ISN 0852            BCOS=SNLOBE/((FREQ/136.)*ANT)
               C    SET UP FOR INTERGER OUTPUT
ISN 0853            IACOS=ACOS*1000000+.5
ISN 0854            IBCOS=BCOS*1000000+.5
ISN 0855            CALL ZERO(8,IACOS,LCOS)
ISN 0856            CALL ZERO(8,IBCOS,MCOS)
ISN 0857        226 IEWMER=EWMER*1000
ISN 0858        260 IEWCER=EWCER*1000
ISN 0859        261 ISNMER=SNMER*1000
ISN 0860        262 ISNCER=SNCER*1000
ISN 0861        263 IELOBE=EWLOBE*10000
ISN 0862        264 INLOBE=SNLOBE*10000
ISN 0863            IERATE=EWFB*100000.*TILI
ISN 0864            INRATE=SNFB*100000.*TILI
ISN 0865            IEACC=EWFC*10000000.*(TILI**2)
ISN 0866            INACC=SNFC*10000000.*(TILI**2)
ISN 0867            INSFE=SNFE*100000000.*(TILI**3)
ISN 0868            IEWFE=EWFE*100000000.*(TILI**3)
ISN 0869            IF(EWFT-86400.)115,116,116
ISN 0870        116 EWFT=EWFT-86400.
ISN 0871        115 HOUR=EWFT/3600.
ISN 0872            AMIN=((EWFT-(HOUR*3600.))/60.)
ISN 0873            SECN=(EWFT-((HOUR*3600.)+(AMIN*60.)))
ISN 0874            BOUR=HOUR*100
ISN 0875            BOUR=BOUR+AMIN
ISN 0876            CALL ZERO(5,BOUR,IOUR)
ISN 0877            ASEC=SECN*100C
ISN 0878            CALL ZERO(6,ASEC,ISEC)
ISN 0879            GO TO 117
ISN 0880        117 CALL PYRD(IYR,IDAYD(MID),YRMODA)
ISN 0881            CALL ZERO(3,CSTA,ASTA)
ISN 0882            CALL ZERO(7,YRMODA,ARMODA)
ISN 0883            HOUR=HOUR*10000
ISN 0884            HOUR=AMIN*100+HOUR+SECN
ISN 0885            CALL ZERO(7,HOUR,AOUR)
ISN 0886            CALL ZERO(7,IELOBE,IALOBE)
```



continued

```
ISN 0887          CALL ZERO(4,IEWMER,IAWMER)
ISN 0888          CALL ZERO(4,IEWCER,IAWCER)
ISN 0889          CALL ZERO(7,INLOBE,IBLOBE)
ISN 0890          CALL ZERO(7,IERATE,IARATE)
ISN 0891          CALL ZERO(7,INRATE,IBRATE)
ISN 0892          CALL ZERO(7,IEACC,IAACC)
ISN 0893          CALL ZERO(7,INACC,IBACC)
ISN 0894          CALL ZERO(6,IEWFE,IAEWFE)
ISN 0895          CALL ZERO(6,INSFE,IANSFE)
ISN 0896          CALL ZERO(4,ISNMER,IBNMER)
ISN 0897          CALL ZERO(4,ISNCER,IBNCER)
            C     EAST WEST GRADE
ISN 0898          AST(4)=SPA
ISN 0899          AST(6)=SPA
ISN 0900          IF(ABS(SC).GT..05)AST(6)=F1
ISN 0902          IF(IABS(IEWMER).GT.50)AST(6)=F1
ISN 0904          IF(IABS(IEWCER).GT.50)AST(6)=F1
            C     NORTH SOUTH GRADE
ISN 0906          IF(ABS(SF).GT..05)AST(4)=F2
ISN 0908          IF(IABS(ISNMER).GT.50)AST(4)=F2
ISN 0910          IF(IABS(ISNCER).GT.50)AST(4)=F2
ISN 0912          IF(AST(6).EQ.SPA)GO TO 759
ISN 0914          IF(AST(4).EQ.SPA)GO TO 759
ISN 0916          AST(6)=F3
ISN 0917          AST(4)=F3
ISN 0918      759 IF(IGRADE.EQ.I)GO TO 761
ISN 0920          AST(6)=SPA
ISN 0921          AST(4)=SPA
ISN 0922      761 CALL ZERO(3,K,IFIT)
ISN 0923      215 WRITE(15,751)SATT,ILINK,AST(1),(ASTA(I),I=2,3),AST(2),IAO,AST(6),A
                 XST(3),IFIT,ARMODA,AOUR,IALOBE,IARATE,IAACC,IAEWFE,IENO,IAWMER,IAWC
                 XER,LCOS
ISN 0924          WRITE(11,750)SATT,ILINK,AST(1),(ASTA(I),I=2,3),AST(2),IAO,AST(6),A
                 XST(3),IFIT,ARMODA,AOUR,IALOBE,IARATE,IAACC,IAEWFE,IENO,IAWMER,IAWC
                 XER,LCOS
ISN 0925      216 WRITE(15,751)SATT,ILINK,AST(1),(ASTA(I),I=2,3),AST(2),IBO,AST(4),A
                 XST(3),IFIT,ARMODA,AOUR,IBLOBE,IBRATE,IBACC,IANSFE,INNO,IBNMER,IBNC
                 XER,MCOS
ISN 0926          WRITE(11,750)SATT,ILINK,AST(1),(ASTA(I),I=2,3),AST(2),IBO,AST(4),A
                 XST(3),IFIT,ARMODA,AOUR,IBLOBE,IBRATE,IBACC,IANSFE,INNO,IBNMER,IBNC
                 XER,MCOS
ISN 0927      750 FORMAT(X,I7,X,I1,A1,2A1,A1,I1,A1,A1,3A1,7A1,7A1,7A1,7A1,7A1,6A1,4A
                 X1,4A1,4A1,8A1)
ISN 0928      751 FORMAT(I7,X,I1,A1,2A1,A1,I1,A1,A1,3A1,7A1,7A1,7A1,7A1,7A1,6A1,4A1,
                 X4A1,4A1,8A1)
ISN 0929          WRITE(11,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,AST(2),IAO
ISN 0930          WRITE(11,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,AST(2),IBO
ISN 0931          GO TO 778
ISN 0932      217 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,8A1,3X,A1,I1)
ISN 0933      778 WRITE(11,777)
ISN 0934          GO TO 39
ISN 0935       30 WRITE( 6,31)
ISN 0936       31 FORMAT(17H CAL.LINE PARITY )
ISN 0937          GO TO 39
ISN 0938      604 DO 60 JZ=1,60
ISN 0939          JY=JZ
```

Step 74

Step 75

Step 76

Step 77

Auxiliary statements for loops and cycling and some error messages

```
                                                                                                PAGE 017

      ISN 0940             IF(JY.EQ.32.)GO TO 39
      ISN 0942          61 READ( 9,601,END=1,ERR=51)(DATA(I),I=1,65)
      ISN 0943             IF(DATA(I).EQ.IAMP)GO TO 86
      ISN 0945             IF(DATA(10).EQ.SPX)GO TO 62
      ISN 0947             GO TO 64
      ISN 0948          62 IF(DATA(30).EQ.SPX)GO TO 63
      ISN 0950             GO TO 64
      ISN 0951          63 IF(DATA(50).EQ.SPX)GO TO 39
      ISN 0953          54 WRITE( 6,660)(DATA(I),I=1,65)
      ISN 0954         660 FORMAT(X,65A1)
      ISN 0955          60 CONTINUE
      ISN 0956             GO TO 39
      ISN 0957          65 K=JK-1
      ISN 0958             DO 66 I=1,K
      ISN 0959             WRITE( 6,67)HORD(I),MIND(I),SFCD(I),EEWF(I),FWMD(I),EWCD(I),ENSF(I
                          X),NSMD(I),NSCD(I),ANTD(I),IDAY9(I)
      ISN 0960          67 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)),I2,X,I3,X)
      ISN 0961          66 CONTINUE
      ISN 0962             GO TO 39
      ISN 0963         769 WRITE( 6,770)
      ISN 0964         770 FORMAT(19H NO 9 IN CAL. LINE )
      ISN 0965             WRITE( 6,183)IAMP,SAT
      ISN 0966             WRITE( 6,660)(DATA(I),I=1,65)
      ISN 0967             GO TO 604
      ISN 0968         767 WRITE( 6,768)
      ISN 0969         768 FORMAT(16H CAL.LINE ERROR )
      ISN 0970             WRITE( 6,183)IAMP,SAT
      ISN 0971             WRITE( 6,660)(DATA(I),I=1,65)
      ISN 0972             GO TO 604
      ISN 0973         777 FORMAT(1H1)
      ISN 0974           2 WRITE( 6,3)
      ISN 0975           3 FORMAT(25H CONSTANTS READ IN ERROR )
      ISN 0976           1 WRITE( 6,990)
      ISN 0977         990 FORMAT(16H JOB IS COMPLETE)
      ISN 0978             END FILE 6
      ISN 0979             END FILE 11
      ISN 0980             STOP
      ISN 0981             END

   COMPILATION DELETED, 3
```

Program Termination

```
LEVEL 16 ( 1 JULY 68)                    OS/360  FORTRAN H                              DATE  69.165/04.20.37

           COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002              SUBROUTINE LSCUB(T,SALPHA,XY,ITO,L,SIGMA,SBETA,SGAMMA,SDELTA,SM,IN
                           XN,IOT)
      ISN 0003              REAL*8 DELSQ,SIG,A,B,ALPHA,BETA,GAMMA,DELTA,TAU,RAPPA,XYNORM,TNORM
      ISN 0004              DIMENSION B(4,5),A(4,5),T(32),XY(32),XYNORM(32),TNORM(32)
      ISN 0005              DO 1 I=1,4
      ISN 0006              DO 1 J=1,5
      ISN 0007            1 A(I,J)=0.0
      ISN 0008              L=(ITO+1)/2
      ISN 0009              A(1,1)=ITO
      ISN 0010              INN=ITO
      ISN 0011              DELSQ=0.
      ISN 0012              ITNO=0
      ISN 0013              DO 10 I=1,ITO
      ISN 0014              RAPPA=XY(I)-XY(L)
      ISN 0015              TAU=T(I)-T(L)
      ISN 0016              XYNORM(I)=RAPPA
      ISN 0017              TNORM(I)=TAU
      ISN 0018              A(1,2)=A(1,2)+TAU
      ISN 0019              A(1,3)=A(1,3)+TAU**2
      ISN 0020              A(1,4)=A(1,4)+TAU**3
      ISN 0021              A(2,4)=A(2,4)+TAU**4
      ISN 0022              A(3,4)=A(3,4)+TAU**5
      ISN 0023              A(4,4)=A(4,4)+TAU**6
      ISN 0024              A(1,5)=A(1,5)+RAPPA
      ISN 0025              A(2,5)=A(2,5)+RAPPA*TAU
      ISN 0026              A(3,5)=A(3,5)+RAPPA*TAU**2
      ISN 0027              A(4,5)=A(4,5)+RAPPA*TAU**3
      ISN 0028           10 DELSQ=RAPPA**2+DELSQ
      ISN 0029           67 A(2,2)=A(1,3)
      ISN 0030              A(2,3)=A(1,4)
      ISN 0031              A(3,3)=A(2,4)
      ISN 0032              A(2,1)=A(2,5)
      ISN 0033              A(3,1)=A(3,5)
      ISN 0034              A(4,1)=A(4,5)
      ISN 0035              N=1
      ISN 0036              DO 20 K=1,4
      ISN 0037              N=N+1
      ISN 0038              DO 15 J=N,5
      ISN 0039           15 B(K,J)=A(K,J)/A(K,K)
      ISN 0040              IF(N-5)12,21,11
      ISN 0041           12 DO 20 I=N,4
      ISN 0042              DO 20 J=I,5
      ISN 0043           20 A(I,J)=A(I,J)-A(K,I)*B(K,J)
      ISN 0044           21 DELTA=B(4,5)
      ISN 0045              GAMMA=B(3,5)-DELTA*B(3,4)
      ISN 0046              BETA=B(2,5)-DELTA*B(2,4)-GAMMA*B(2,3)
      ISN 0047              ALPHA=B(1,5)-BETA*B(1,2)-GAMMA*B(1,3)-DELTA*B(1,4)
      ISN 0048              SALPHA=ALPHA
      ISN 0049              SBETA=BETA
      ISN 0050              SGAMMA=GAMMA
      ISN 0051              SDELTA=DELTA
      ISN 0052              N=A(1,1)
      ISN 0053              SIG=((DELSQ-ALPHA*A(1,5)-BETA*A(2,1)-GAMMA*A(3,1)-DELTA*A(4,1))/A(
                           X1,1))
```

```
ISN 0054          SIGMA=SIG
ISN 0055          IF(SIGMA)70,71,71
ISN 0056       70 SIGMA=0.
ISN 0057       71 SIGMA=SQRT(SIGMA)
ISN 0058          ITNO=ITNO+1
ISN 0059          IF(ITNO-50)35,59,59
ISN 0060       35 DO 61 I=1,4
ISN 0061          DO 61 J=1,5
ISN 0062       61 A(I,J)=0.0
ISN 0063          DELSQ=0.0
ISN 0064          DO 65 I=1,ITO
ISN 0065          RAPPA=XYNORM(I)
ISN 0066          TAU=TNORM(I)
ISN 0067          R=RAPPA-ALPHA-BETA*TAU-GAMMA*TAU**2-DELTA*TAU**3
ISN 0068          IF(ABS(R).GT..200)GO TO 65
ISN 0070          IF(ABS(R).GT.SM*SIGMA)GO TO 65
ISN 0072       32 A(1,1)=A(1,1)+1.
ISN 0073          A(1,2)=A(1,2)+TAU
ISN 0074          A(1,3)=A(1,3)+TAU**2
ISN 0075          A(1,4)=A(1,4)+TAU**3
ISN 0076          A(2,4)=A(2,4)+TAU**4
ISN 0077          A(3,4)=A(3,4)+TAU**5
ISN 0078          A(4,4)=A(4,4)+TAU**6
ISN 0079          A(1,5)=A(1,5)+RAPPA
ISN 0080          A(2,5)=A(2,5)+RAPPA*TAU
ISN 0081          A(3,5)=A(3,5)+RAPPA*TAU**2
ISN 0082          A(4,5)=A(4,5)+RAPPA*TAU**3
ISN 0083          DELSQ=RAPPA**2+DELSQ
ISN 0084       65 CONTINUE
ISN 0085          IF(A(1,1)-5)59,66,66
ISN 0086       66 IF(A(1,1)-N)67,59,67
ISN 0087       59 IST=N
ISN 0088          RETURN
ISN 0089       11 WRITE(6,7)
ISN 0090        7 FORMAT(39H THIS MSG.HAS A ERROR IN THE CUBIC FIT )
ISN 0091          RETURN
ISN 0092          END
```

```
LEVEL 16 ( 1 JULY 68)                              OS/360  FORTRAN H                               DATE  69.165/04.20.57

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
     ISN 0002              SUBROUTINF LSQQUA(T,ALPHA,XY,ITO,L,SIGMA,F,BETA,GAMMA,SM,INN,IOT)
     ISN 0003              DIMENSION B(3,4),A(3,4),T(32),XY(32)
     ISN 0004              DO 1 I=1,3
     ISN 0005              DO 1 J=1,4
     ISN 0006            1 A(I,J)=0.0
     ISN 0007              L=(ITO+1)/2
     ISN 0008              A(1,1)=ITO
     ISN 0009              INN=ITO
     ISN 0010              DELSQ=0.
     ISN 0011              DO 10 I=1,ITO
     ISN 0012              RAPPA=XY(I)-XY(L)
     ISN 0013              TAU=T(I)-T(L)
     ISN 0014              A(1,2)=A(1,2)+TAU
     ISN 0015              A(1,3)=A(1,3)+TAU**2
     ISN 0016              A(1,4)=A(1,4)+RAPPA
     ISN 0017              A(2,3) =A(2,3)+TAU**3
     ISN 0018              A(2,4)=A(2,4)+RAPPA*TAU
     ISN 0019              A(3,3) =A(3,3) +TAU**4
     ISN 0020              A(3,4)=A(3,4)+RAPPA*TAU**2
     ISN 0021           10 DELSQ=RAPPA**2+DELSQ
     ISN 0022           67 A(2,2)=A(1,3)
     ISN 0023              A(2,1)=A(2,4)
     ISN 0024              A(3,1)=A(3,4)
     ISN 0025              N=1
     ISN 0026              DO22 K=1,3
     ISN 0027              N=N+1
     ISN 0028              DO15 J=N,4
     ISN 0029           15 B(K,J)=A(K,J)/A(K,K)
     ISN 0030              IF(N-4)12,21,11
     ISN 0031           12 DO20I=N,3
     ISN 0032              DO20J=I,4
     ISN 0033           20 A(I,J)=A(I,J)-A(K,I)*B(K,J)
     ISN 0034           21 GAMMA=B(3,4)
     ISN 0035              BETA=B(2,4)-GAMMA*B(2,3)
     ISN 0036              ALPHA    =B(1,4)-BETA*B(1,2)-GAMMA*B(1,3)
     ISN 0037              N=A(1,1)
     ISN 0038              SIGMA=((DELSQ-ALPHA*A(1,4)-BETA*A(2,1)-GAMMA*A(3,1))/A(1,1))
     ISN 0039              IF(SIGMA)70,71,71
     ISN 0040           70 SIGMA=0.
     ISN 0041           71 SIGMA=SQRT(SIGMA)
     ISN 0042              IF(SIGMA-F    )59,59,35
     ISN 0043           35 DO61I=1,3
     ISN 0044              DO61J=1,4
     ISN 0045           61 A(I,J)=0.0
     ISN 0046              DELSQ=0.0
     ISN 0047              DO65I=1,ITO
     ISN 0048              RAPPA=XY(I)-XY(L)
     ISN 0049              TAU= T(I)-T(L)
     ISN 0050              R=RAPPA-ALPHA-BETA*TAU-GAMMA*TAU**2
     ISN 0051              IF(R)2,3,3
     ISN 0052            2 R=-R
     ISN 0053            3 IF(R-SM*SIGMA)32,32,65
     ISN 0054           32 A(1,1)=A(1,1)+1.
     ISN 0055              A(1,2)=A(1,2)+TAU
ISN 0056             A(1,3)=A(1,3)+TAU**2
ISN 0057             A(1,4)=A(1,4)+RAPPA
ISN 0058             A(2,3) =A(2,3)+TAU**3
ISN 0059             A(2,4)=A(2,4)+RAPPA*TAU
ISN 0060             A(3,3) =A(3,3) +TAU**4
ISN 0061             A(3,4)=A(3,4)+RAPPA*TAU**2
ISN 0062             DELSQ=RAPPA**2+DELSQ
ISN 0063          65 CONTINUE
ISN 0064             IF(A(1,1)-S)59,66,66
ISN 0065          66 IF(A(1,1)-N    )67,59,67
ISN 0066          59 ITO=A(1,1)
ISN 0067             IOT=ITO
ISN 0068             RETURN
ISN 0069          11 WRITE( 6,7)
ISN 0070           7 FORMAT(30H THIS MSG. HAS EXCESSIVE NOISE)
ISN 0071             RETURN
ISN 0072             END
```

```
LEVEL 16 ( 1 JULY 68)                                  OS/360  FORTRAN H                                DATE  69.165/04.21.13

               COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002              SUBROUTINE LOBASN(TEM,I,A,RATE,L)
      ISN 0003              DIMENSION A(31),TEM(31)
      ISN 0004              K=I-1
      ISN 0005              IF(RATE.GT.0.)A(1)=A(1)+1.
      ISN 0007              IF(RATE.LT.0.)A(1)=A(1)-1.
      ISN 0009              DO 10 J=1,K
      ISN 0010              L=0
      ISN 0011              JJ=J+1
      ISN 0012              DELTA=RATE*(TEM(JJ)-TEM(J))
      ISN 0013           11 X=A(JJ)-A(J)
      ISN 0014              L=L+1
      ISN 0015              IF(L.GE.100)GO TO 12
      ISN 0017              IF(ABS(DELTA-X)-.500)10,10,6
      ISN 0018            6 IF(DELTA)4,5,3
      ISN 0019            4 A(JJ)=A(JJ)-1
      ISN 0020              GO TO 11
      ISN 0021            3 A(JJ)=A(JJ)+1
      ISN 0022              GO TO 11
      ISN 0023            5 IF(X)3,10,4
      ISN 0024           10 CONTINUE
      ISN 0025           12 RETURN
      ISN 0026              END
```

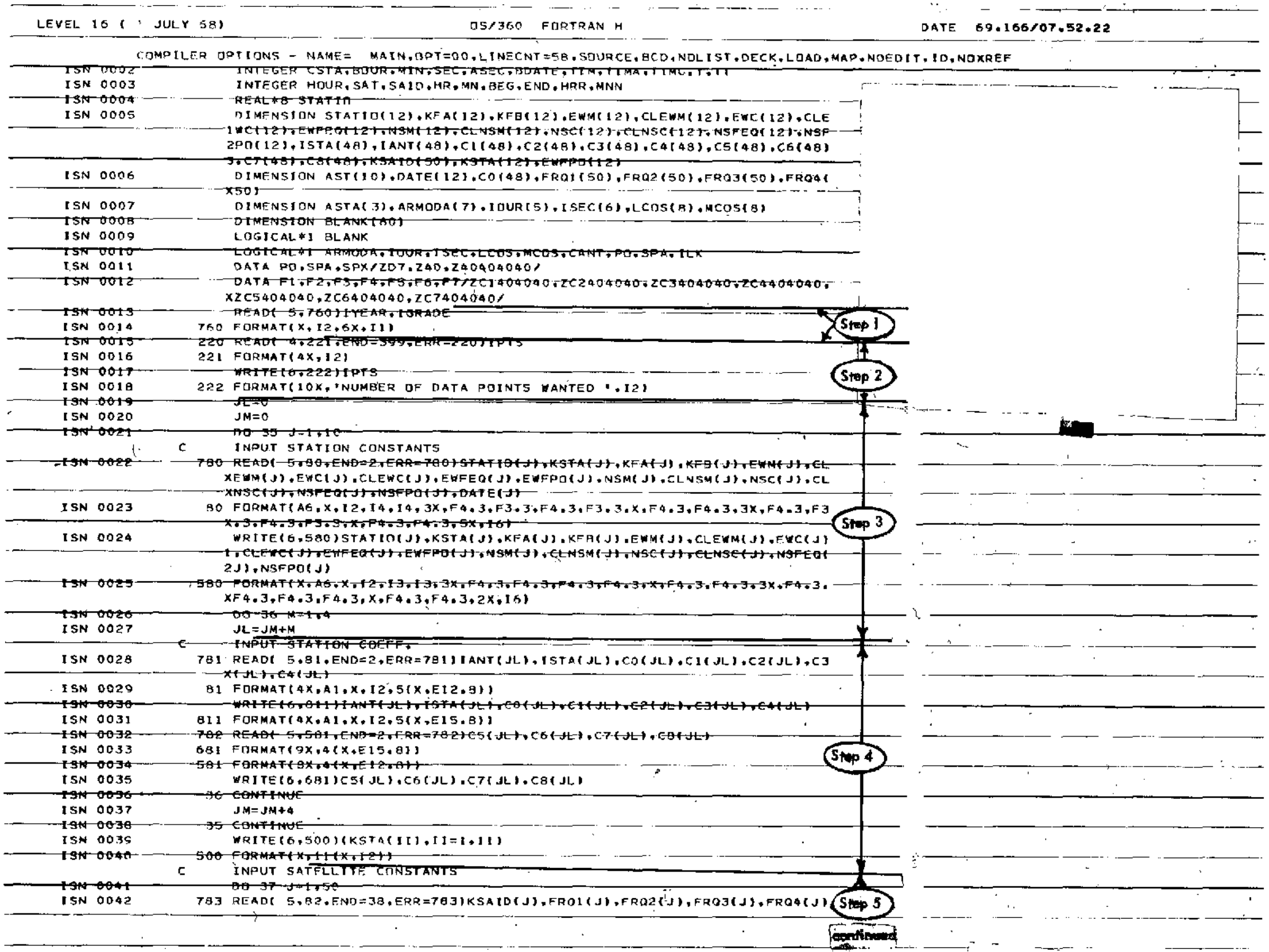
```
LEVEL 16 ( 1 JULY 68)                     OS/360  FORTRAN H                           DATE  69.166/07.52.32

      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002           SUBROUTINE DCOUNT(IDATE)
  ISN 0003           DIMENSION IA(12)
              C
  ISN 0004           DATA IA(1),IA(2),IA(3),IA(4),IA(5),IA(6),IA(7),IA(8),IA(9),
                    1IA(10),IA(11),IA(12)
                    1/0,31,59,90,120,151,181,212,243,273,304,334/
  ISN 0005           JSUM=0
  ISN 0006           IYR=IDATE/10000
  ISN 0007           IMO=(IDATE-(IYR*10000))/100
  ISN 0008           IDA=IDATE-(IYR*10000)-(IMO*100)
  ISN 0009           JSUM=JSUM&IA(IMO)&IDA
  ISN 0010           ISUM=0
  ISN 0011           ICO=0
  ISN 0012           J=1
  ISN 0013           IRDA=365
  ISN 0014         1 IF(ICO.EQ.0)GO TO 4
  ISN 0016           ISUM=ISUM&IRDA
  ISN 0017           J=J&1
  ISN 0018           IF(J.EQ.4)GO TO 2
  ISN 0020           IF(J.LE.4)GO TO 3
  ISN 0022           J=1
  ISN 0023           IRDA=364
  ISN 0024         2 IRDA=IRDA&1
  ISN 0025         3 ICO=ICO-1
  ISN 0026           GO TO 1
  ISN 0027         4 LYR=0
  ISN 0028           FYR=IYR
  ISN 0029           SYR=IYR/4
  ISN 0030           IF(FYR/4.0.NE.SYR)GO TO 5
  ISN 0032           IF(IMO.LT.3) GO TO 5
  ISN 0034           LYR=LYR&1
  ISN 0035         5 IDATE=ISUM&JSUM&LYR
  ISN 0036           RETURN
  ISN 0037         6 STOP
  ISN 0038           END
```

```
LEVEL 16 ( 1 JULY 68)                        OS/360  FORTRAN H                            DATE  69.165/04.21.28

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002           SUBROUTINE ZERO(N,IIN,AREA)
      ISN 0003           DIMENSION DIV(7)
      ISN 0004           INTEGER DIV
      ISN 0005           LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)
      ISN 0006           DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8-
                        X),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
                        X/
      ISN 0007           DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
                        X00,10000,1000,100,10,1/
      ISN 0008           AREA(1)=PLUS
      ISN 0009           IF(IIN)1,2,2
     .ISN 0010         1 AREA(1)=MINUS
      ISN 0011           IIN=IABS(IIN)
      ISN 0012         2 IJ=9-N
      ISN 0013           K=2
      ISN 0014           J=N-1
      ISN 0015           DO 4 I=1,J
      ISN 0016           ITEMP=IIN/DIV(IJ)
      ISN 0017           IF(ITEMP.LE.9.)GO TO 6
      ISN 0019           AREA(K)=ASK
      ISN 0020           GO TO 5
      ISN 0021         6 AREA(K)=IC(ITEMP+1)
      ISN 0022         5 IIN=IIN-(ITEMP*DIV(IJ))
      ISN 0023           IJ=IJ+1
      ISN 0024           K=K+1
      ISN 0025         4 CONTINUE
      ISN 0026           RETURN
      ISN 0027           END
```

```
LEVEL 16 ( 1 JULY 68)                          OS/360  FORTRAN H                              DATE  69.165/04.21.19

           COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002               SUBROUTINE NORMAL(X)
      ISN 0003               IF(X)113,114,115
      ISN 0004           113 IF(ABS(X)-.5)114,114,116
      ISN 0005           115 X=X+1.0
      ISN 0006               GO TO 114
      ISN 0007           115 IF(ABS(X)-.5)114,114,117
      ISN 0008           117 X=X-1.0
      ISN 0009           114 RETURN
      ISN 0010               END
```

```
LEVEL 16 ( 1 JULY 68)                    OS/360  FORTRAN H                    DATE  69.165/04.21.23

            COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
      ISN 0002           SUBROUTINF PYRD(NYR,JDAY,NYMODA)
      ISN 0003           DIMENSION N(24)
      ISN 0004           DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10),N(11),N(12
                        1),N(13),N(14),N(15),N(16),N(17),N(18),N(19),N(20),N(21),N(22),N(23
                        2),N(24)/0,31,59,90,120,151,181,212,243,273,304,334,0,31,60,91,121,
                        3152,182,213,244,274,305,335/
      ISN 0005           J=0
      ISN 0006           A=NYR/4.
      ISN 0007           L=A
      ISN 0008           A=A-L
      ISN 0009           IF(A)2,1,2
      ISN 0010         1 J=12
      ISN 0011         2 DO 5 K=1,12
      ISN 0012           M=J+K
      ISN 0013           IF(JDAY.LE.N(M))GO TO 4
      ISN 0015         5 CONTINUE
      ISN 0016         4 M=J+K-1
      ISN 0017           NDAY=JDAY-N(M)
      ISN 0018           K=K-1
      ISN 0019           NYMODA=(NYR*10000)+(K*100)+NDAY
      ISN 0020           RETURN
      ISN 0021           END
```

OBS-B PROGRAM



```
LEVEL 16 ( ' JULY 58)                       OS/360   FORTRAN H                       DATE  69.166/07.52.22

           COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
    ISN 0002              INTEGER CSTA,BDUR,MIN,SEC,ASEC,BDATE,TIM,TIMA,TIMC,I,II
    ISN 0003              INTEGER HOUR,SAT,SAID,HR,MN,BEG,END,HRR,MNN
    ISN 0004              REAL*8 STATIN
    ISN 0005              DIMENSION STATIO(12),KFA(12),KFB(12),EWM(12),CLEWM(12),EWC(12),CLE
                         1WC(12),EWFEQ(12),NSM(12),CLNSM(12),NSC(12),CLNSC(12),NSFEQ(12),NSF
                         2PO(12),ISTA(48),IANT(48),C1(48),C2(48),C3(48),C4(48),C5(48),C6(48)
                         3,C7(48),C8(48),KSAID(50),KSTA(12),EWFPO(12)
    ISN 0006              DIMENSION AST(10),DATE(12),C0(48),FRQ1(50),FRQ2(50),FRQ3(50),FRQ4(
                         X50)
    ISN 0007              DIMENSION ASTA(3),ARMODA(7),IDUR(5),ISEC(6),LCOS(8),MCOS(8)
    ISN 0008              DIMENSION BLANK(80)
    ISN 0009              LOGICAL*1 BLANK
    ISN 0010              LOGICAL*1 ARMODA,IDUR,ISEC,LCOS,MCOS,CANT,PO,SPA,ILK
    ISN 0011              DATA PO,SPA,SPX/ZD7,Z40,Z40404040/
    ISN 0012              DATA F1,F2,F3,F4,F5,F6,F7/ZC1404040,ZC2404040,ZC3404040,ZC4404040,
                         XZC5404040,ZC6404040,ZC7404040/
    ISN 0013              READ( 5,760)IYEAR,IGRADE
    ISN 0014          760 FORMAT(X,I2,6X,I1)                                                      Step 1
    ISN 0015          220 READ( 4,221,END=399,ERR=220)IPTS
    ISN 0016          221 FORMAT(4X,I2)
    ISN 0017              WRITE(6,222)IPTS                                                        Step 2
    ISN 0018          222 FORMAT(10X,'NUMBER OF DATA POINTS WANTED ',I2)
    ISN 0019              JL=0
    ISN 0020              JM=0
    ISN 0021              DO 35 J=1,10
                    C     INPUT STATION CONSTANTS
    ISN 0022          780 READ( 5,80,END=2,ERR=780)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CL
                         XEWM(J),EWC(J),CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CL
                         XNSC(J),NSFEQ(J),NSFPO(J),DATE(J)
    ISN 0023           80 FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
                         X,3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)                                     Step 3
    ISN 0024              WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
                         1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
                         2J),NSFPO(J)
    ISN 0025          580 FORMAT(X,A6,X,I2,I3,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,3X,F4.3,
                         XF4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)
    ISN 0026              DO 36 M=1,4
    ISN 0027              JL=JM+M
                    C     INPUT STATION COEFF.
    ISN 0028          781 READ( 5,81,END=2,ERR=781)IANT(JL),ISTA(JL),C0(JL),C1(JL),C2(JL),C3
                         X(JL),C4(JL)
    ISN 0029           81 FORMAT(4X,A1,X,I2,5(X,E12.8))
    ISN 0030              WRITE(6,811)IANT(JL),ISTA(JL),C0(JL),C1(JL),C2(JL),C3(JL),C4(JL)
    ISN 0031          811 FORMAT(4X,A1,X,I2,5(X,E15.8))
    ISN 0032          782 READ( 5,581,END=2,ERR=782)C5(JL),C6(JL),C7(JL),C8(JL)
    ISN 0033          681 FORMAT(9X,4(X,E15.8))
    ISN 0034          581 FORMAT(9X,4(X,E12.8))                                                  Step 4
    ISN 0035              WRITE(6,681)C5(JL),C6(JL),C7(JL),C8(JL)
    ISN 0036           36 CONTINUE
    ISN 0037              JM=JM+4
    ISN 0038           35 CONTINUE
    ISN 0039              WRITE(6,500)(KSTA(II),II=1,11)
    ISN 0040          500 FORMAT(X,11(X,I2))
                    C     INPUT SATELLITE CONSTANTS
    ISN 0041              DO 37 J=1,50
    ISN 0042          783 READ( 5,82,END=38,ERR=783)KSAID(J),FRQ1(J),FRQ2(J),FRQ3(J),FRQ4(J) Step 5
```

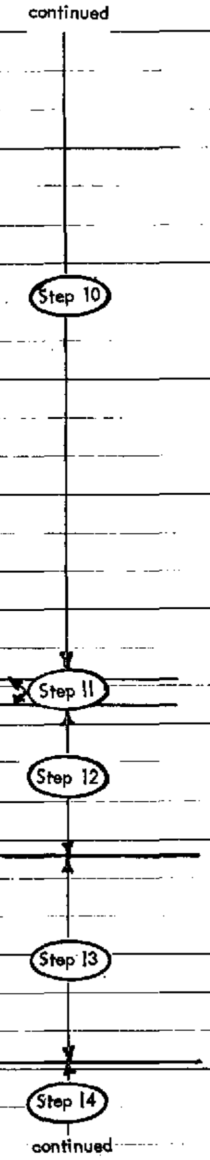
continued

continued

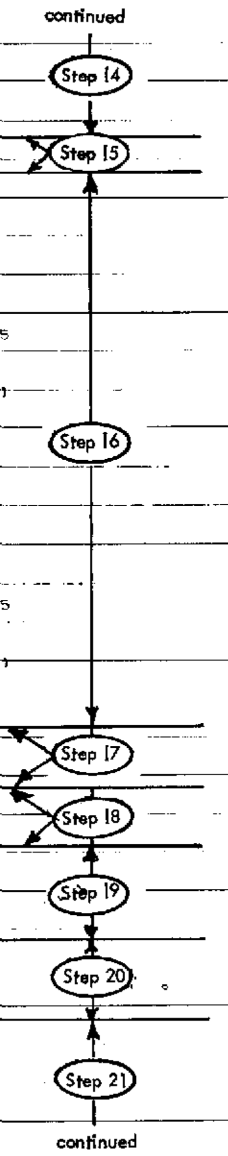
```
ISN 0043          82 FORMAT(I5,19X,F8.3,F8.3,F8.3,F8.3)
ISN 0044             WRITE( 6,582)KSAID(J),FRQ1(J),FRQ2(J),FRQ3(J),FRQ4(J)
ISN 0045         582 FORMAT(X,I5,19X,F8.3,X,F8.3,X,F8.3,X,F8.3)
ISN 0046             IF(KSAID(J))37,38,37                                            Step 5
ISN 0047          37 CONTINUF
ISN 0048          38 KSATCT=J-1
ISN 0049         398 IAO=0
ISN 0050             IBO=0
ISN 0051         399 READ( 2,499,END=219,ERR=399)IA,IB,IC,ID,IE,IF,ILINK,AST(1),CSTA,CA
                    XNT,IAO,AST(7),AST(4),IFIT,BDATE,HOUR,MIN,SEC,EWLOBE,EWRATE,EWACC,E
                    XWO,IENO,EWMFR,EWCER,ACOS
ISN 0052         499 FORMAT(I1,I1,I1,I2,I1,I1,X,I1,A1,I2,A1,I1,A1,A1,I3,X,I6,X,I2,I2,I2
                    X,F7.4,F7.5,F7.7,F6.9,I4,I4,I4,I8)
ISN 0053             WRITE( 6,394)IA,IB,IC,ID,IE,IF,ILINK,AST(1),CSTA,CANT,IAO,AST(7),A   Step 6
                    XST(4),IFIT,BDATE,HOUR,MIN,SEC,EWLOBE,EWRATE,EWACC,EWD,IEND,EWMER,E
                    XWCER,ACOS
ISN 0054             IF(IGRADE.EO.2)GO TO 761
ISN 0056             IF(AST(7).NE.SPX)GO TO 398
ISN 0058         761 IF(IAO.EO.2)GO TO 397
ISN 0060             GO TO 398
ISN 0061         397 READ( 2,499,END=219,ERR=399)IA,IB,IC,ID,IE,IF,ILINK,AST(1),JSTA,JA
                    XNT,IBO,AST(6),AST(3),JFIT,JDATE,JHOR,JMN,JSC,SNLOBE,SNRATE,SNACC,S
                    XND,INNO,SNMEP,SNCFR,BCOS
ISN 0062             WRITE( 6,394)IA,IB,IC,ID,IE,IF,ILINK,AST(1),JSTA,JANT,IBO,AST(6),A
                    XST(3),JFIT,JDATE,JHOR,JMN,JSC,SNLOBE,SNRATE,SNACC,SND,INNO,SNMER,S
                    XNCER,BCOS                                                         Step 7
ISN 0063         394 FORMAT(2X,I1,I1,I1,I2,I1,I1,X,I1,A1,I2,A1,I1,A1,A1,I3,X,I6,X,I2,I2
                    X,I2,F9.4,F9.5,F10.7,F10.8,2X,I4,X,I4,X,I4,X,I8)
ISN 0064             IF(IGRADE.FO.2)GO TO 762
ISN 0066             IF(AST(6).NE.SPX)GO TO 398
ISN 0068         762 IF(IBO.EQ.3)GO TO 763
ISN 0070             GO TO 398
ISN 0071         763 IF(IAO+IBO.NE.5)GO TO 398
ISN 0073             IF(BDATE.NE.JDATE)GO TO 398
ISN 0075             IF(HOUR.NE.JHOR)GO TO 398                                       Step 8
ISN 0077             IF(MIN.NE.JMN)GO TO 398
ISN 0079             IF(SEC.NE.JSC)GO TO 398
ISN 0081             IF(IC.NE.0)GO TO 764
ISN 0083             SAT=(IA*10000)+(IB*1000)+(ID*10)+IF
ISN 0084             GO TO 765                                                       Step 9
ISN 0085         764 SAT=(IB*10000)+(IC*1000)+(ID*10)+IF
ISN 0086         765 DO 490 L=1,12
ISN 0087             IF(KSTA(L).FO.CSTA)GO TO 302
ISN 0089         490 CONTINUE
ISN 0090         302 IF(CANT.FQ.PO)GO TO 489
ISN 0092             ANT=46.
ISN 0093             GO TO 488
ISN 0094         489 ANT=57.
ISN 0095         488 DO 487 M=1,50
ISN 0096             IF(SAT.EQ.KSAID(M))GO TO 307                                    Step 10
ISN 0098         487 CONTINUF
ISN 0099             GO TO 398
ISN 0100         307 IF(AST(4).EQ.F1)GO TO 401
ISN 0102             IF(AST(4).FQ.F2)GO TO 402
ISN 0104             IF(AST(4).FQ.F3)GO TO 403
ISN 0106             IF(AST(4).FQ.F4)GO TO 404
```

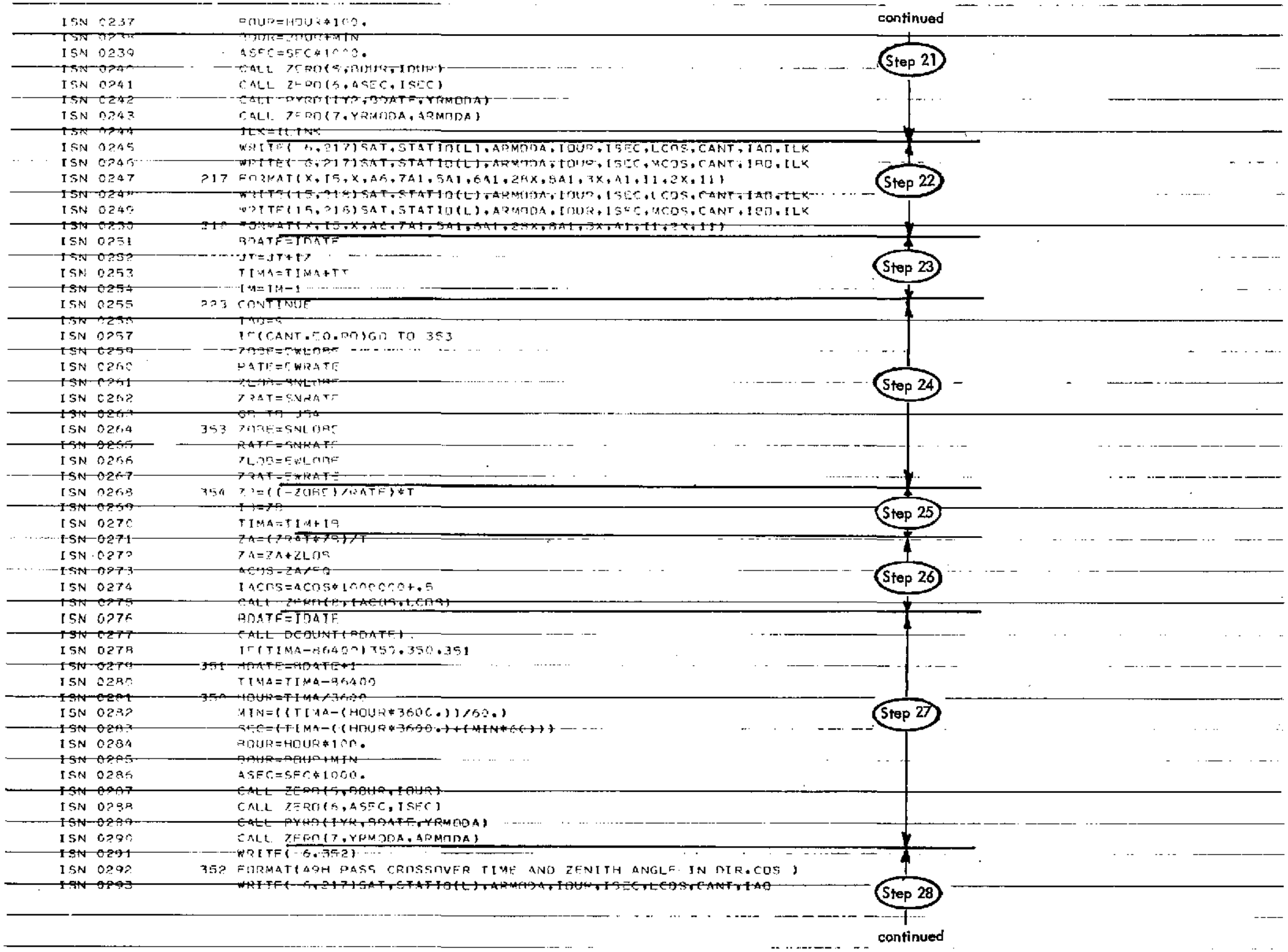
continued

```
ISN 0109           IF(AST(4).EQ.F5)GO TO 405
ISN 0111           IF(AST(4).EQ.F6)GO TO 406
ISN 0112           IF(AST(4).EQ.F7)GO TO 407
ISN 0114           GO TO 390
ISN 0115       401 T=1.
ISN 0116           GO TO 400
ISN 0117       402 T=2.
ISN 0118           GO TO 400
ISN 0119       403 T=10.
ISN 0120           GO TO 400
ISN 0121       404 T=30.
ISN 0122           GO TO 400
ISN 0123       405 T=60.
ISN 0124           GO TO 400
ISN 0125       406 T=120.
ISN 0126           GO TO 400
ISN 0127       407 T=600.
ISN 0128       400 A=EWLDPF
ISN 0129           B=SNLDPF
ISN 0130           IF(ILINK.NE.1)GO TO 740
ISN 0132           FREQ=FR01(M)
ISN 0133           GO TO 744
ISN 0134       740 IF(ILINK.NE.2)GO TO 741
ISN 0136           FREQ=FR02(M)
ISN 0137           GO TO 744
ISN 0138       741 IF(ILINK.NE.3)GO TO 743
ISN 0140           FREQ=FR03(M)
ISN 0141           GO TO 744
ISN 0142       743 IF(ILINK.NE.4)GO TO 745
ISN 0144           FREQ=FR04(M)
ISN 0145           GO TO 744
ISN 0146       745 WRITE( 6,746)ILINK,FREQ
ISN 0147       746 FORMAT('FREQ.LINK NOT IN TABLE.LINK= ',I1,' FREQ= ',F8.3,X)
ISN 0148           GO TO 390
ISN 0149       744 IF(FREQ.LT.136.0)GO TO 745
ISN 0151           IF(FREQ.GT.138.0)GO TO 745
ISN 0153           TIM=((HOUR*3600)+(60*MIN))+SEC
ISN 0154           DO 21 IM=1,42
ISN 0155           IF(CSTA.EQ.ISTA(IM))GO TO 24
ISN 0157        21 CONTINUE
ISN 0158        24 IF(CANT.NE.P0)GO TO 25
ISN 0160           IM=IM+2
ISN 0161        25 PI=6.2831853
ISN 0162           F0=((FREQ/136.)*ANT)
ISN 0163           IY=(IPTS-1)/2
ISN 0164           IF(IY.NE.0)GO TO 70
ISN 0166           IZ=1
ISN 0167           JT=0
ISN 0168           TT=0
ISN 0169           GO TO 71
ISN 0170        70 IZ=(IFIT/2)/IY
ISN 0171           JT=-IZ*IY
ISN 0172           TT=T*JT
ISN 0173        71 TIMA=TIM+TT
ISN 0174           TT=T*IZ
ISN 0175           DO 223 II=1,IPTS
```



```
ISN 0176          AA=A+(EWRATE*JT)+(EWACC*JT**2)+(EWD*JT**3)
ISN 0177          BB=B+(SNRATE*JT)+(SNACC*JT**2)+(SND*JT**3)
ISN 0178       53 AO=AA
ISN 0179          BO=BB
ISN 0180          CC=AO/FQ
ISN 0181          DD=BO/FQ
ISN 0182          IF(CANT.EQ.PO)GO TO 42
ISN 0184          IF(ABS(CC).GT..08716)GO TO 54
ISN 0186          IF(ABS(DD).GT..64279)GO TO 54
ISN 0188          EE=BO**3
ISN 0189          GO TO 43
ISN 0190       42 EE=AO**3
ISN 0191          IF(ABS(CC).GT..64279)GO TO 54
ISN 0193          IF(ABS(DD).GT..08716)GO TO 54
ISN 0195       43 C=(C0(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
                 X(IM)*BO**2)+(C6(IM)*EE)+(C7(IM)*SIN(AO*PI))+(C8(IM)*COS(AO*PI))
ISN 0196          GO TO 59
ISN 0197       54 C=(C0(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(AO*PI)
                 X)+(C8(IM)*COS(AO*PI))
ISN 0198       59 IM=IM+1
ISN 0199          IF(CANT.EQ.PO)GO TO 48
ISN 0201          IF(ABS(CC).GT..08716)GO TO 60
ISN 0203          IF(ABS(DD).GT..64279)GO TO 60
ISN 0205          FF=BO**3
ISN 0206          GO TO 49
ISN 0207       48 FF=AO**3
ISN 0208          IF(ABS(CC).GT..64279)GO TO 60
ISN 0210          IF(ABS(DD).GT..08716)GO TO 60
ISN 0212       49 E=(C0(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
                 X(IM)*BO**2)+(C6(IM)*FF)+(C7(IM)*SIN(BO*PI))+(C8(IM)*COS(BO*PI))
ISN 0213          GO TO 63
ISN 0214       60 E=(C0(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(BO*PI)
                 X)+(C8(IM)*COS(BO*PI))
ISN 0215       63 AA=C
ISN 0216          BB=E
ISN 0217       27 IYR=BDATE/10000
ISN 0218          IDATE=BDATE
ISN 0219          CALL DCOUNT(BDATE)
ISN 0220          IF(TIMA)28,29,29
ISN 0221       28 TIMA=86400+TIMA
ISN 0222          BDATE=BDATE-1
ISN 0223       29 IAA=2
ISN 0224          IBB=3
               C  MAXIUM OBSERVATIONS
ISN 0225          ACOS=AA/FQ
ISN 0226          BCOS=BB/FQ
ISN 0227          IACOS=ACOS*1000000+.5
ISN 0228          IBCOS=BCOS*1000000+.5
ISN 0229          CALL ZERO(8,IACOS,LCOS)
ISN 0230          CALL ZERO(8,IBCOS,MCOS)
ISN 0231          IF(TIMA-86400)303,303,304
ISN 0232      304 BDATE=BDATE+1
ISN 0233          TIMA=TIMA-86400
ISN 0234      303 HOUR=TIMA/3600.
ISN 0235          MIN=((TIMA-(HOUR*3600.))/60.)
ISN 0236          SEC=(TIMA-((HOUR*3600.)+(MIN*60.)))
```





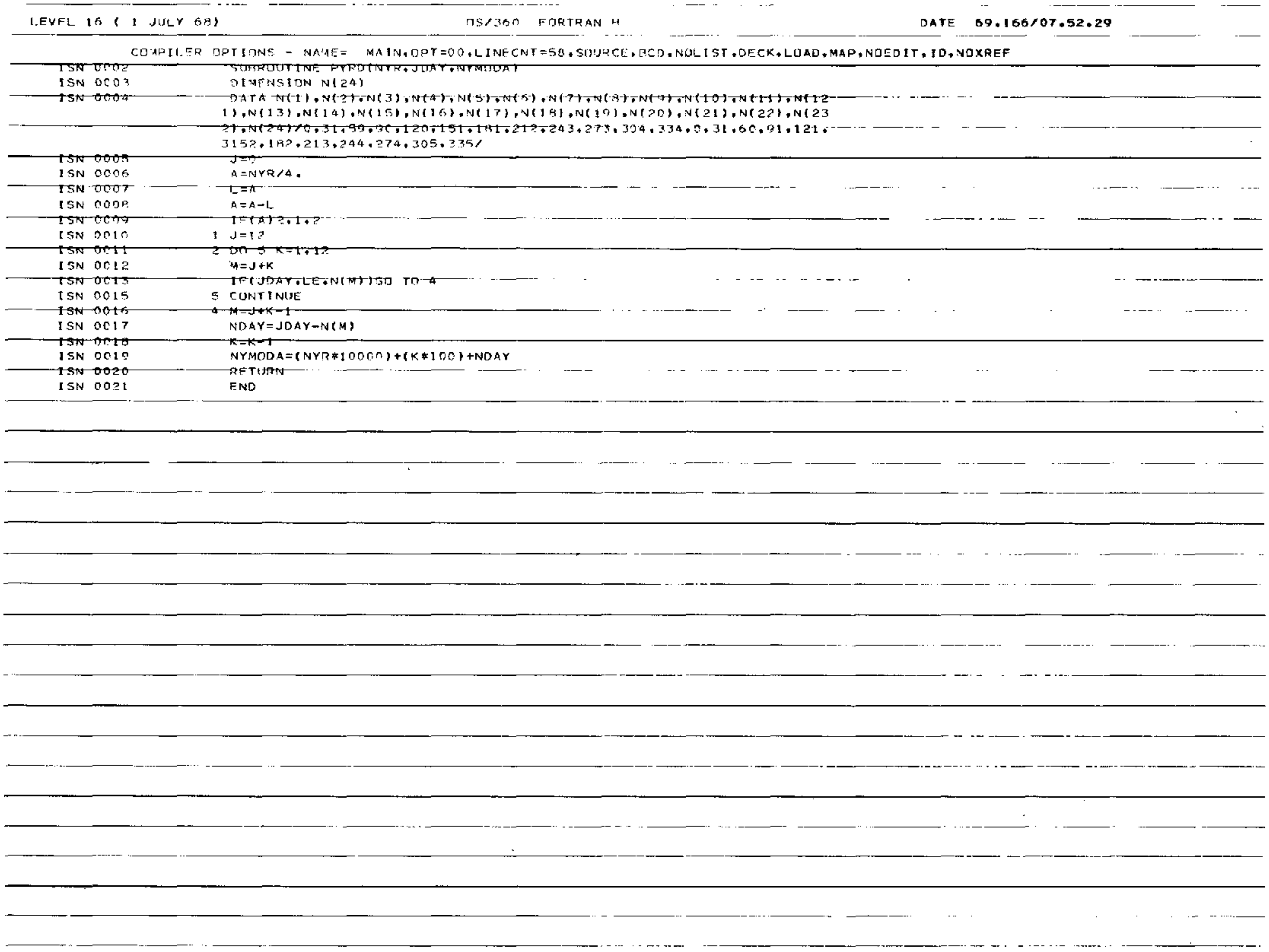
```
ISN 0237          BOUR=HOUR*100.
ISN 0238          BOUR=BOUR+MIN
ISN 0239          ASEC=SEC*1000.
ISN 0240          CALL ZERO(5,BOUR,IOUR)
ISN 0241          CALL ZERO(6,ASEC,ISEC)
ISN 0242          CALL PYRD(IYR,BDATE,YRMODA)
ISN 0243          CALL ZERO(7,YRMODA,ARMODA)
ISN 0244          IEX=ILINK
ISN 0245          WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO,ILK
ISN 0246          WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO,ILK
ISN 0247      217 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,5A1,3X,A1,I1,2X,I1)
ISN 0248          WRITE(15,216)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO,ILK
ISN 0249          WRITE(15,216)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO,ILK
ISN 0250      216 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,5A1,3X,A1,I1,2X,I1)
ISN 0251          BDATE=IDATE
ISN 0252          JT=JT+IZ
ISN 0253          TIMA=TIMA+TT
ISN 0254          IM=IM-1
ISN 0255      223 CONTINUE
ISN 0256          IAO=5
ISN 0257          IF(CANT.EQ.RO)GO TO 353
ISN 0258          ZOBE=EWLOBE
ISN 0260          RATE=EWRATE
ISN 0261          ZLOB=SNLOBE
ISN 0262          ZRAT=SNRATE
ISN 0263          GO TO 354
ISN 0264      353 ZOBE=SNLOBE
ISN 0265          RATE=SNRATE
ISN 0266          ZLOB=EWLOBE
ISN 0267          ZRAT=EWRATE
ISN 0268      354 ZB=((-ZOBE)/RATE)*T
ISN 0269          IB=ZB
ISN 0270          TIMA=TIM+IB
ISN 0271          ZA=(ZRAT*ZB)/T
ISN 0272          ZA=ZA+ZLOB
ISN 0273          ACOS=ZA/EQ
ISN 0274          IACOS=ACOS*1000000+.5
ISN 0275          CALL ZERO(8,IACOS,LCOS)
ISN 0276          BDATE=IDATE
ISN 0277          CALL DCOUNT(BDATE)
ISN 0278          IF(TIMA-86400)350,350,351
ISN 0279      351 BDATE=BDATE+1
ISN 0280          TIMA=TIMA-86400
ISN 0281      350 HOUR=TIMA/3600
ISN 0282          MIN=((TIMA-(HOUR*3600.))/60.)
ISN 0283          SEC=(TIMA-((HOUR*3600.)+(MIN*60)))
ISN 0284          BOUR=HOUR*100.
ISN 0285          BOUR=BOUR+MIN
ISN 0286          ASEC=SEC*1000.
ISN 0287          CALL ZERO(5,BOUR,IOUR)
ISN 0288          CALL ZERO(6,ASEC,ISEC)
ISN 0289          CALL PYRD(IYR,BDATE,YRMODA)
ISN 0290          CALL ZERO(7,YRMODA,ARMODA)
ISN 0291          WRITE( 6,352)
ISN 0292      352 FORMAT(49H PASS CROSSOVER TIME AND ZENITH ANGLE IN DIR.COS )
ISN 0293          WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
```

continued

Step 28

```
    ISN 0294             GO TO 398
    ISN 0295           2 WRITE( 6,3)
    ISN 0296           3 FORMAT(25H CONSTANTS READ IN ERROR )
    ISN 0297         219 END FILE 15
    ISN 0298             REWIND 15
    ISN 0299           1 STOP
    ISN 0300             END
  ADCONS FOR EXTERNAL REFERENCES
```

Error Message
and
Program Temination



```
LEVEL 16 ( 1 JULY 68)                              OS/360  FORTRAN H                          DATE  69.166/07.52.29

         COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
   ISN 0002            SUBROUTINE PYRD(NYR,JDAY,NYMODA)
   ISN 0003            DIMENSION N(24)
   ISN 0004            DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10),N(11),N(12
                      1),N(13),N(14),N(15),N(16),N(17),N(18),N(19),N(20),N(21),N(22),N(23
                      2),N(24)/0,31,59,90,120,151,181,212,243,273,304,334,0,31,60,91,121,
                      3152,182,213,244,274,305,335/
   ISN 0005            J=0
   ISN 0006            A=NYR/4.
   ISN 0007            L=A
   ISN 0008            A=A-L
   ISN 0009            IF(A)2,1,2
   ISN 0010          1 J=12
   ISN 0011          2 DO 5 K=1,12
   ISN 0012            M=J+K
   ISN 0013            IF(JDAY.LE.N(M))GO TO 4
   ISN 0015          5 CONTINUE
   ISN 0016          4 M=J+K-1
   ISN 0017            NDAY=JDAY-N(M)
   ISN 0018            K=K-1
   ISN 0019            NYMODA=(NYR*10000)+(K*100)+NDAY
   ISN 0020            RETURN
   ISN 0021            END
```

```
LEVEL 16 ( 1 JULY 68)                    OS/360  FORTRAN H                      DATE  69.166/07.52.32

       COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
  ISN 0002            SUBROUTINE DCOUNT(IDATE)
  ISN 0003            DIMENSION IA(12)
                C
  ISN 0004            DATA IA(1),IA(2),IA(3),IA(4),IA(5),IA(6),IA(7),IA(8),IA(9),
                     1IA(10),IA(11),IA(12)
                     1/0,31,59,90,120,151,181,212,243,273,304,334/
  ISN 0005            JSUM=0
  ISN 0006            IYR=IDATE/10000
  ISN 0007            IMO=(IDATE-(IYR*10000))/100
  ISN 0008            IDA=IDATE-(IYR*10000)-(IMO*100)
  ISN 0009            JSUM=JSUM&IA(IMO)&IDA
  ISN 0010            ISUM=0
  ISN 0011            ICO=0
  ISN 0012            J=1
  ISN 0013            IRDA=365
  ISN 0014          1 IF(ICO.EQ.0)GO TO 4
  ISN 0016            ISUM=ISUM&IRDA
  ISN 0017            J=J&1
  ISN 0018            IF(J.EQ.4)GO TO 2
  ISN 0020            IF(J.LE.4)GO TO 3
  ISN 0022            J=1
  ISN 0023            IRDA=364
  ISN 0024          2 IRDA=IRDA&1
  ISN 0025          3 ICO=ICO-1
  ISN 0026            GO TO 1
  ISN 0027          4 LYR=0
  ISN 0028            FYR=IYR
  ISN 0029            SYR=IYR/4
  ISN 0030            IF(FYR/4.0.NE.SYR)GO TO 5
  ISN 0032            IF(IMO.LT.3) GO TO 5
  ISN 0034            LYR=LYR&1
  ISN 0035          5 IDATE=ISUM&JSUM&LYR
  ISN 0036            RETURN
  ISN 0037          6 STOP
  ISN 0038            END
  LEVEL 16 ( 1 JULY 68)                          OS/360  FORTRAN H                                      DATE  69.166/07.52.35

                COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=58,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,ID,NOXREF
         ISN 0002              SUBROUTINE ZERO(N,IIN,AREA)
         ISN 0003              DIMENSION DIV(7)
         ISN 0004              INTEGER DIV
         ISN 0005              LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)
         ISN 0006              DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8
                              X),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
                              X/
         ISN 0007              DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
                              X00,10000,1000,100,10,1/
         ISN 0008              AREA(1)=PLUS
         ISN 0009              IF(IIN)1,2,2
         ISN 0010            1 AREA(1)=MINUS
         ISN 0011              IIN=IABS(IIN)
         ISN 0012            2 IJ=9-N
         ISN 0013              K=2
         ISN 0014              J=N-1
         ISN 0015              DO 4 I=1,J
         ISN 0016              ITEMP=IIN/DIV(IJ)
         ISN 0017              IF(ITEMP.LE.9.)GO TO 6
         ISN 0019              AREA(K)=ASK
         ISN 0020              GO TO 5
         ISN 0021            6 AREA(K)=IC(ITEMP+1)
         ISN 0022            5 IIN=IIN-(ITEMP*DIV(IJ))
         ISN 0023              IJ=IJ+1
         ISN 0024              K=K+1
         ISN 0025            4 CONTINUE
         ISN 0026              RETURN
         ISN 0027              END
```

Appendix B

MINITRACK HARDWARE

THEORY OF OPERATION

The Minitrack radio interferometer performs angular position measurements by phase comparisons between multiple pairs of antennas of known separation distance. Antenna pairs are aligned along east-west and north-south baselines.

The accuracy of angle measurement increases with the length of the baseline between the antennas; two pairs of antennas have baselines many wavelengths long to obtain good angular resolution. These are called fine antennas. Ambiguity results because the phase meters repeat their reading cycle every wavelength of path difference. This is resolved by employing several progressively shorter baselines to produce fewer integral numbers of wavelength changes while the satellite is within the antenna pattern. These are termed "medium" and "coarse antennas." Each antenna pair feeds a channel in the receiver, yielding six separate phase measurements. Ambiguity antenna information identifies the integral number of path differences on the fine antenna (fig. B-1); data from the fine antenna define the direction cosines of a satellite and are the basis for the orbital calculations.

ANTENNAS

The antennas are a slot type with ground screens. They are pedestal mounted, parallel to the ground plane, precisely positioned geographically, and accurately leveled (fig. B-2). The antenna beam is fan shaped, the plane of the fan being perpendicular to the long dimension. The fine antenna array produces a fan beam 76° (north-south) and 11° (east-west) at the 3-dB points. This alignment will accommodate low- and medium-inclination equatorial orbits. The stations have a second set of fine antennas producing a fan beam with the wide dimension of the fan in the east-west direction. This configuration is adapted to high-inclination orbits. The equatorial or polar tracking modes are selected by switching to the desired set of antennas. The ambiguity antennas have a beamwidth of 78° (east-west) and 108° (north-south) at the 3-dB points. Antennas feed the receivers through coaxial transmission lines.

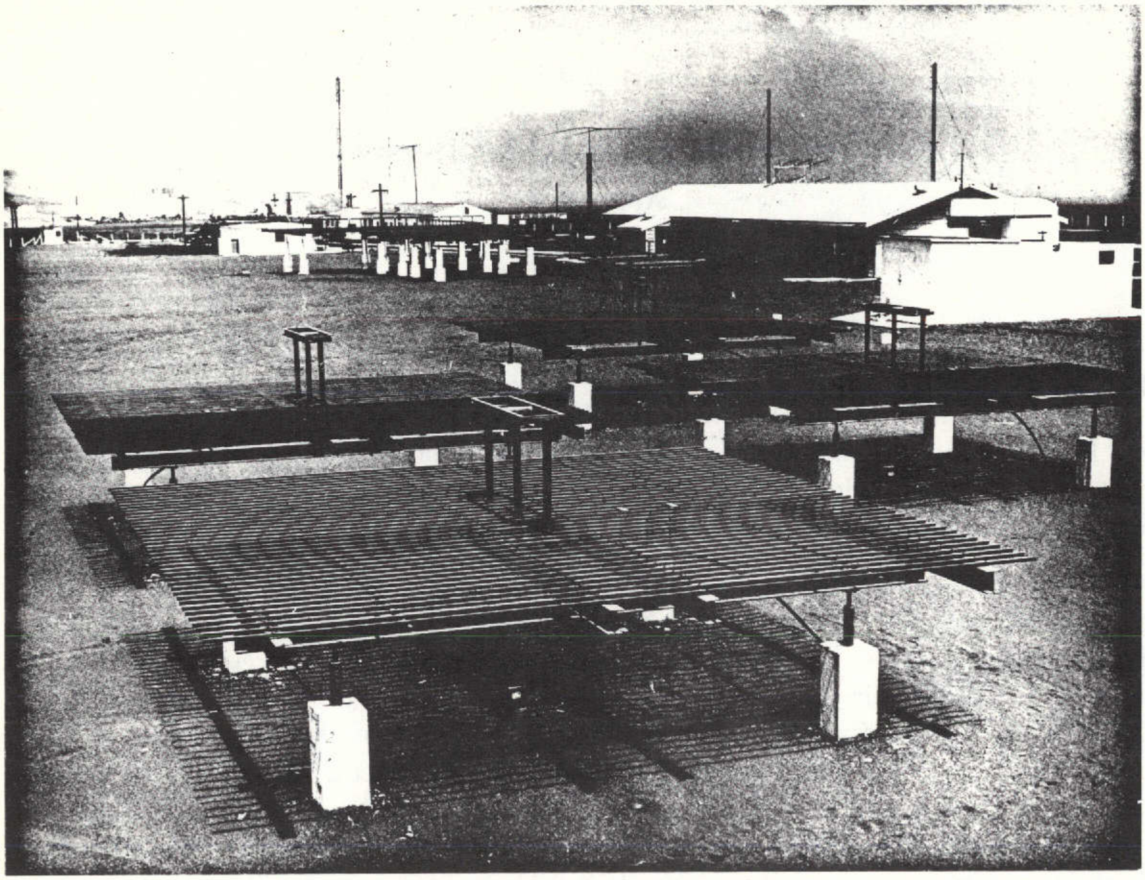


Figure B-1.—MHz ambiguity-resolution antenna.

RECEIVER

The Minitrack receiver is a seven-channel, triple-conversion superheterodyne instrument tunable over 136 to 138 MHz in 1-kHz steps (fig. B-3). Six channels carry satellite information; one channel is reserved for system calibration. A low-noise (3 dB or less) preamplifier-converter is connected to each antenna. Each preamplifier-converter has two outputs enabling certain ambiguity antennas to be used in more than one combination.

A radiofrequency calibrating source is used to check sensitivity, noise, and phase-shift characteristics of the receiver. An attenuator provides calibrated signal levels covering the dynamic range of the system.

Separate AGC generators furnish control voltage proportional to input signal levels. One unit controls the fine channels and another unit controls the ambiguity channels. Different AGC response speeds may be selected, and manual gain control is included for receiver alignment.

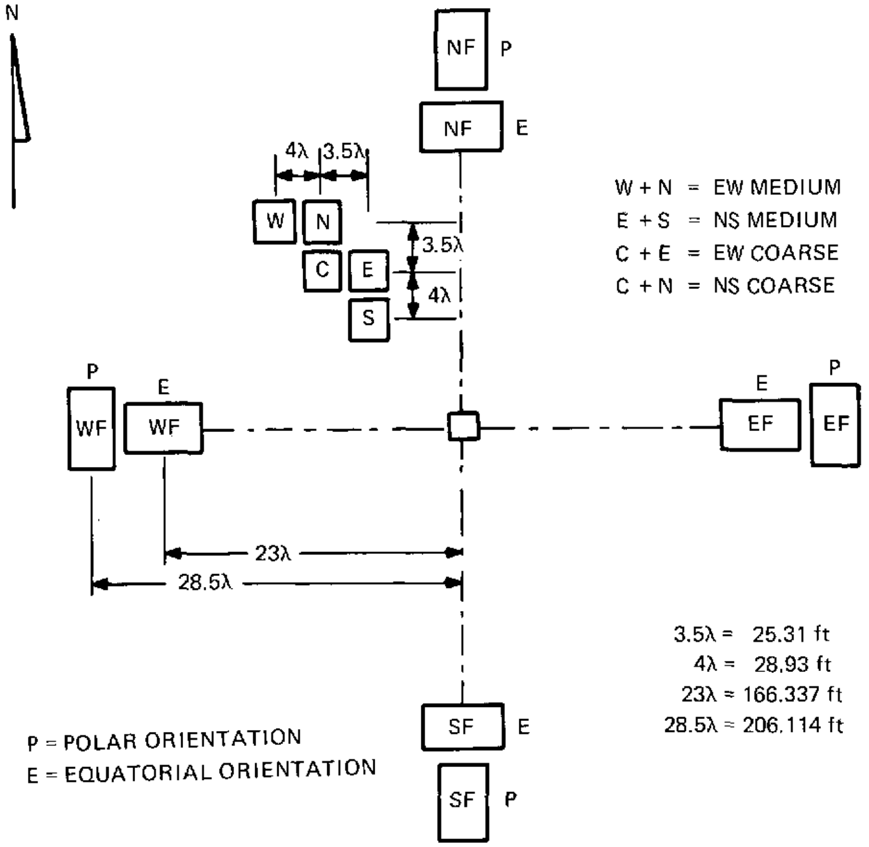


Figure B-2.—136-MHz Minitrack antenna field.

MEASUREMENT AND DATA SYSTEM

Phase meters provide phase comparisons to the accuracy needed and furnish a permanent record of this phase-angle per time information. The phase meter inputs are the 100-Hz detected signals plus noise and the 100-Hz reference signal developed by the local oscillator. The phase relationship between the output signals and the reference signal is the difference between their respective antenna pairs. These are processed by narrowband filters, amplified, and applied to analog phase meters and a reference pulse generator.

The signals are reshaped into triggering pulses that control gates whose widths are proportional to the relative phase between the signal and reference. The outputs of the phase meters are dc voltages proportional to gate widths and thus to phase differences. Analog outputs are available for all channels, fine and ambiguity, and are also displayed on panel meters.

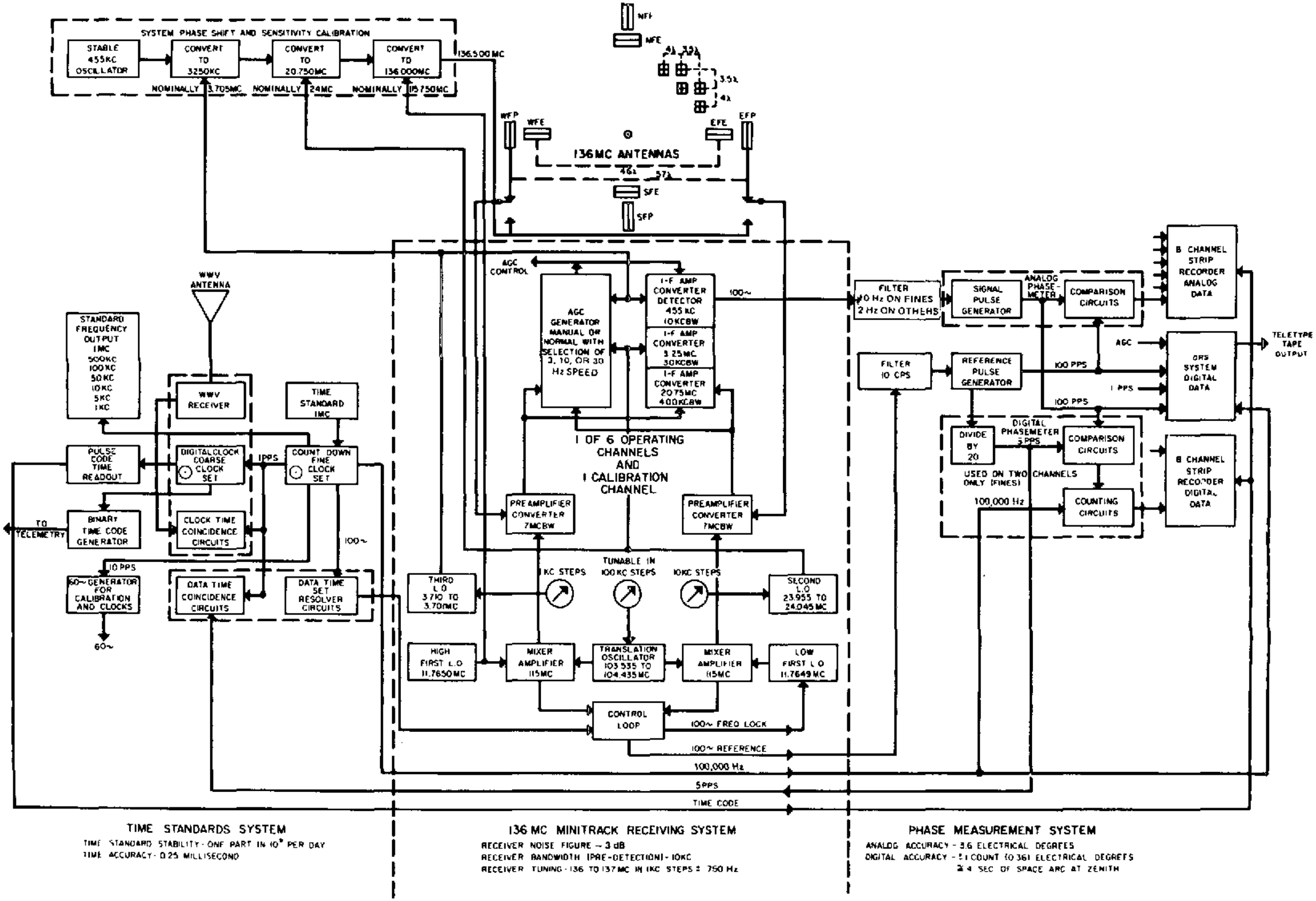

SYSTEM PHASE SHIFT AND SENSITIVITY CALIBRATION
STABLE 455KC OSCILLATOR
CONVERT TO 3250KC
CONVERT TO 20.750MC
CONVERT TO 136.000MC
NOMINALLY 3.705MC
NOMINALLY 24MC
NOMINALLY 115.750MC
136.500 MC
NFP
NFE
4λ
3.5λ
3.5λ
4λ
WFP
WFE
EFE
EFP
136MC ANTENNAS
46λ
57λ
SFE
SFP
WWV ANTENNA
STANDARD FREQUENCY OUTPUT 1MC 500KC 100KC 50KC 10KC 5KC 1KC
WWV RECEIVER
TIME STANDARD 1MC
PULSE CODE TIME READOUT
DIGITAL CLOCK COARSE CLOCK SET
1PPS
COUNT DOWN FINE CLOCK SET
TO TELEMETRY
BINARY TIME CODE GENERATOR
CLOCK TIME COINCIDENCE CIRCUITS
100~
10 PPS
60~ GENERATOR FOR CALIBRATION AND CLOCKS
60~
DATA TIME COINCIDENCE CIRCUITS
DATA TIME SET RESOLVER CIRCUITS
AGC CONTROL
AGC GENERATOR MANUAL OR NORMAL WITH SELECTION OF 3, 10, OR 30 Hz SPEED
I-F AMP CONVERTER DETECTOR 455 KC 10KCBW
100~
I-F AMP CONVERTER 3.25MC 30KCBW
I-F AMP CONVERTER 20.75MC 400KCBW
1 OF 6 OPERATING CHANNELS AND 1 CALIBRATION CHANNEL
PREAMPLIFIER CONVERTER 7MCBW
PREAMPLIFIER CONVERTER 7MCBW
TUNABLE IN 100KC STEPS
1KC STEPS
10KC STEPS
THIRD L.O 3.710 TO 3.70MC
SECOND L.O 23.955 TO 24.045 MC
HIGH FIRST L.O 11.7650 MC
MIXER AMPLIFIER 115MC
TRANSLATION OSCILLATOR 103.535 TO 104.435 MC
MIXER AMPLIFIER 115MC
LOW FIRST L.O 11.7649 MC
CONTROL LOOP
100~ FREQ LOCK
100~ REFERENCE
100,000 Hz
5PPS
TIME CODE
FILTER 10 Hz ON FINES 2 Hz ON OTHERS
SIGNAL PULSE GENERATOR
ANALOG PHASE-METER
COMPARISON CIRCUITS
8 CHANNEL STRIP RECORDER ANALOG DATA
AGC
ORS SYSTEM DIGITAL DATA
TELETYPE TAPE OUTPUT
FILTER 10 CPS
REFERENCE PULSE GENERATOR
100 PPS
1 PPS
100 PPS
DIGITAL PHASEMETER
DIVIDE BY 20
5 PPS
COMPARISON CIRCUITS
USED ON TWO CHANNELS ONLY (FINES)
100,000 Hz
COUNTING CIRCUITS
8 CHANNEL STRIP RECORDER DIGITAL DATA
TIME STANDARDS SYSTEM
TIME STANDARD STABILITY - ONE PART IN 10⁹ PER DAY
TIME ACCURACY - 0.25 MILLISECOND
136 MC MINITRACK RECEIVING SYSTEM
RECEIVER NOISE FIGURE – 3 dB
RECEIVER BANDWIDTH (PRE-DETECTION) - 10KC
RECEIVER TUNING - 136 TO 137MC IN 1KC STEPS ± 750 Hz
PHASE MEASUREMENT SYSTEM
ANALOG ACCURACY - 3.6 ELECTRICAL DEGREES
DIGITAL ACCURACY - ±1 COUNT (0.36) ELECTRICAL DEGREES
≅ 4 SEC OF SPACE ARC AT ZENITH

Figure B-3.—Block diagram of the 136-MHz Minitrack interferometer system.

Analog data and AGC voltages from the fine and ambiguity channels of the receiver are recorded on a Sanborn recorder. Precise resolution of the fine data is furnished by a digital phase meter, which gates a 100,000-pps pulse train to a three-decade decimal counter during the time between a reference pulse and a signal pulse. East-west and north-south fine data are read digitally five times per second. Data-readout time is synchronized with clock time to an accuracy of 100 μs. A 100-Hz precision signal, continuously variable in phase, is used for calibration.

The time standard drives a digital clock that gives time in days of the year, hours, minutes, and seconds for the digital recording unit. AGC levels from the receiver are presented, ambiguity information from both medium and coarse channels is sampled once per second, and the fine channels are sampled five times per second. All data are punched on a five-level perforated Teletype tape.

MINITRACK OPTICAL TRACKING SYSTEM (MOTS)

The equatorially mounted astrographic camera used for periodic aircraft calibration of the interferometer system has been adapted for optical tracking of brighter satellites up to the fifth magnitude. The camera has an ultralinear *f* 5.0, 40-in. focal length lens and uses 8- by 10-in. spectroscopic plates affording an ultimate star-resolution accuracy of better than 1 arcsec over an 11° by 14° field of view. The camera tracks the star field, thus permitting stars as faint as 11th magnitude to be photographed.

The station serial time code is used to actuate a solenoid that moves a plunger to displace the film plate within its holder. The satellite photographs as a trail of light against a star background interrupted by breaks corresponding to time-code pulses. The photographic plates are compared to star charts, and preliminary reductions are made at the tracking stations. Whenever possible, photographs are taken while the satellite is in the main antenna beam of the Minitrack interferometer system, and the corresponding radio records are mailed to GSFC along with the photographic plates.

SPECIAL PROBLEMS

It is beyond the scope of this report to describe in detail each electronic chassis of the system. Much of that information is available from other sources.[1] Any functional descriptions, transfer functions, or nonlinear analysis beyond that already in print would require extensive laboratory and bench testing–tasks much removed from the intent of this effort.

Nonetheless, certain observations came to our attention in the process of our analyses and on-site inquiry that are worth mentioning here.

[1] Bendix Corp.: *Instruction Manual for 136 Mc Minitrack Interferometer System.* Vols. I and II, Bendix Radio Division, 1965.

Transmission Lines

Each tracking station employs semiflexible, 50Ω coaxial transmission lines between the antennas deployed on the field of the facility and the preamplifier converters in the electronic racks inside the building. These transmission lines are commercial grade, 7/8-in., Neoprene rubber covered RG-254/U, with air (gas) dielectric. Our investigation indicates that the lines used in the installation of the Minitrack stations around the world were procured from different manufacturers. Fortunately, all 7/8-in. coaxial cable at any site is from the same manufacturer.

The difficulty, however, lies in the fact that the velocity factors of the coaxial cable from different manufacturers range from 0.83 to 0.96. At first glance this would appear not to be a problem. One could argue that the most important criterion is that the velocity factor for the cable at any one site be constant under a wide range of environmental conditions and that the line exhibit a good characteristic of phase stability. Because the transmission lines are of equal length, the absolute value of the velocity factor is not important, provided, of course, that both lines for the interferometer pair come from the same manufacturer's process batch and have the same propagation delay time.

Unfortunately, such arguments are flawed. The reader will remember that all fine-antenna transmission-line pairs were cut to the same length; this, therefore, poses no problem. At many sites, however, the pairs of transmission lines between respective sets of ambiguity antennas are significantly different in length. The effect of this differential is supposedly corrected in the preprocessing program MIN B. (See pp. 42 and 50 in vol. 1 and p. 2 in vol. 2.) The reader will observe that a constant said to include the coaxial-cable velocity factor is part of the equations of the inequality calculation. This constant is written as ".846." If the velocity factors of all transmission lines at all stations were the same, namely 0.846, there would be no problem. Unfortunately this is not the case. Thus, the cables (or the preprocessor) introduce error in the tracking equations.

At this point, we cannot say just what magnitude of error is being introduced by this effect. Such a determination would require considerable field and laboratory effort. Correction of this differential velocity factor effect would require field measurement and modification to the preprocessing program. It will be noted that the ultimate angular tracking resolution of a Minitrack interferometer facility is primarily dependent on the fine-antenna system, specifically, the spacing of the fine interferometer antenna pairs. The theoretical angular resolution of the ambiguity antenna pairs is much less than that of the fine-antenna system. This, of course, is to be expected from the nature of interferometry. The ambiguity system has the sole function of pointing out which fine-antenna phase segment the spacecraft is operating through. A little leeway in the ambiguity system is not important insofar as the system continues to indicate the correct fine-antenna phase segment; that is, the correct zero-phase cone angle. We suspect, however, that this is not the case at those stations having unequal ambiguity transmission lines whose velocity factors vary from the standard 0.846. The error introduced would be a step function; that is, when the differential velocity factor effect is operating, it causes an error which is a multiple of 1000 fine counts. This, we believe, is a serious problem.

Another set of problems related to the transmission lines include drying and pressurizing the lines. We found, for instance, that an unadvisable purging procedure is being used at some of the tracking sites. It has been maintained that as long as dry-nitrogen pressure is kept on the lines and the lines will hold their pressures, the lines are dry.

Dewpoint measurements made in the field showed that in each test case nitrogen highly saturated with moisture was purged from the lines while dry nitrogen was being introduced at the other (building) end. Further, it was found that in some cases the purge plug at the antenna ends of certain transmission lines were sealed with black tape and silicone rubber compound. Other transmission lines were found to be stopped up with gas-barrier-type bullet insulators instead of having the perforated type. This latter condition prevented the dry gas from passing into the power-splitting networks and baluns.[2]

Antenna Switches

At certain tracking sites, the solid-state coaxial switches were inadequately grounded.[2] These solid-state switches consist of passive, lumped-constant components and back-biased diodes. They are well designed and exhibit characteristics of high phase stability. 'They replace the old Ledex-type hard-contact coaxial switches used in Minitrack earlier. In some of the tracking sites, the new switches were merely set in place over the holes in the old Ledex switch cabinet with no attempt to securely ground the case of the switch to the cabinet.

Such a condition may be introducing severe errors in the accuracy of the phase-measurement process, how much cannot be ascertained without field and laboratory studies. We found that it was virtually impossible to attain repeatability of the antenna voltage standing wave ratio measurements when taken through the inadequately grounded solid-state coaxial switches.

Phase Centers

Investigation of the hardware-development work done on the Minitrack antennas revealed that little, if any, effort was directed to identifying the region of the antenna phase centers. There is reason to believe that the significant displacement of the phase centers from the geometric centers of the fine antenna arrays is introducing error into the Minitrack system.[3]

INVENTORY

The following pages are reproductions of a typical Minitrack station equipment inventory. Slight variations from facility to facility will occur, but, by and large the hardware

[2] D. Watters: "St. Johns Minitrack Facility Inspection." Memorandum, Contract file for NAS5-10694, GSFC, Sept. 3, 1969.

[3] Control Systems Research, Inc., has a paper entitled "Minitrack Beam Switching" that attempts to identify the phase-center displacement problem and recommends a means for correction. This paper is available to serious investigators upon request.

complements are the same. The forms that follow were prepared to assist in the process of generating an accurate Minitrack inventory. They are designed to enlist the services of field technicians in the gathering of data.

Sheet No. 1

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Tracking Station Name: ____________

Inventory Completed By:

______________________(Signature)

______________________(Date)

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
	ANTENNA SYSTEM							
1.1	Antenna - Fine	8			TACO			
1.2	Antenna - Ambiguity	5			TACO			
1.3	Calibration Antenna - Dipole	2						
1.4	Coaxial Cable - (7/8" pressurized transmission lines)	1						
	Enter manufacturer, stock no. & RG no. (Important)							
1.5	Transmission Line Pressure System	1						
	Enter type (nitrogen tank, dry air pump, etc.)							

Tracking Station Name: ____________

Inventory Completed By:

______________________(Signature)

______________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 2

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
1.6	Pressure Gages							
	(Enter qty., type, manufacturer & stock or model no.)							
1.7	Antenna Select Switch	11			American Electronic, Inc.	SNB-8718		
1.8	Power Supply - VAC VDC	1			Trygon Electronics	HR-40		
1.9	Transmission Line Term Box	1			Custon Built			
1.10	Calibrator Power Divider	1			Custom Built			
	RECEIVER RACK							
2.1	Preamp Converter	11			Bendix			
2.2	Translation Oscillator	1			Bendix			
2.3	High First Local Oscillator	1			Bendix			
2.4	Low First Local Oscillator	1			Bendix			
2.5	Second Local Oscillator	1			Bendix			
2.6	Third Local Oscillator	1			Bendix			
2.7	IF Amplifier	7			Bendix			
2.8	AGC Generator	3			Bendix			

Tracking Station Name: ____________

Inventory Completed By:

____________________(Signature)

____________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 3

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
2.9	Control Loop	1			Bendix			
2.10	Calibrator	1			Bendix			
	POWER SUPPLY - RACK A							
3.1	AC Control Panel	1			Custom Built			
3.2	Power Supply	1			Power Designs	305M		
	VAC VDC							
3.3	Power Supply	4			Power Designs	204M-9		
	VAC VDC							
	POWER SUPPLY - RACK B							
3.4	AC Voltage Regulator	1			Sorensen	2000-S		
3.5	Power Supply	3			Power Designs	204M-9		
	VAC VDC							
3.6	Power Supply	1			Power Designs	204M		
	VAC VDC							

Tracking Station Name: ____________

Inventory Completed By:

____________________(Signature)

____________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 4

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
	PHASE MEASUREMENT RACK							
4.1	Filter	2			Bendix	BP-1545		
4.2	Buffer	2			Bendix	BP-1514		
4.3	Reference Pulse Generator	1			Bendix	BP-1516		
4.4	Analog Phase Meter	6			Bendix	BP-1512		
4.5	Digital Phase Meter	1			Bendix			
4.6	Data Readout Swith & Fine Channel							
4.7	Bandwidth Selector	1			Custom Built			
4.8	Narrow Band Track Filter	1			Electrac, Inc.	219		
	TIME STANDARD RACK							
	TIME STANDARD RACK MOD							
5.1	Ultra Stable Oscillator	1			Hermes Electronics	101CS		
5.2	Frequency Divider	1						
5.3	Frequency Divider Mod.	1						
5.4	Plug-in Multiplier	1						
5.5	GOHz Generator Amplifier	1						
5.6	Signal Distribution System	1						

Tracking Station Name: ____________

Inventory Completed By:

____________________(Signature)

____________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 5

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
5.7	Digital Clock	1						
5.8	Binary Time Code Generator	1						
5.9	WWV Receiver	1						
5.10	Time Comparison Equipment	1						
5.11	Power Supply	1			Tektronix	160A		
	VAC VDC							
5.12	Waveform Generator	1			Tektronix (Mod.)	162		
5.13	Indicator	1			Tektronix (Mod.)	360		
5.14	60 Hz Power Amplifier	1						
5.15	Converter Regulator	1						
5.16	12V Battery Charger	1						
5.17	Power Supply	1			Power Designs	323M		
	VAC VDC							
5.18	Power Supply	1			Power Designs	305M		
	VAC VDC							
5.19	Power Supply	1			Harrison Labs	400D		
	VAC VDC							

Tracking Station Name: ____________

Inventory Completed By:

____________________(Signature)

____________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 6

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
6.1	Timing System Subassembly	1			Astrodata	6600		
6.2	Distribution Amplifier	1			Monotronics	203-11		
6.3	Battery Pack - 24V	1			Astrodata	6600-1800		
6.4	Power Supply Battery Charger	1			Astrodata	8100		
6.5	Power Supply Regulator	1			Astrodata			
	CONTROL CONSOLE							
7.1	Phase Shifter & Data Time	1			Custom Built			
7.2	Power Supply	3			Power Designs	305-M		
	VAC VDC							
7.3	Power Supply	2			Power Designs	323M		
	VAC VDC							
7.4	Power Supply	2			Tektronix	160A		
	VAC VDC							
7.5	Waveform Generator	1			Tektronix (Mod.)	162		
7.6	Indicator (CR tube type)	8			Tektronix (Mod.)	360		

Sheet No. 7

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Tracking Station Name: ____________

Inventory Completed By:

______________________(Signature)

______________________(Date)

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
8.1	Automatic Digital Recording System	1			Consolidated Systems, Inc.			
8.2	Spare High Speed Reperferator	1			Consolidated Systems, Inc.			
9.1	Recorder Junction Box	1			Bendix - Custom built			
10.1	Storage Battery - 12V	1						
11.1	Storage Battery - 6V	2						
12.1	Coaxial Cable Set (W201	1						
	through W292) - Enter manu-							
	facturer, manufacturer stock							
	or type no., RG no.							

Tracking Station Name: ____________

Inventory Completed By:

____________________(Signature)

____________________(Date)

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Sheet No. 8

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
13.1	Coaxial Cable Set (W201	1						
	through W278) - Enter manu-							
	facturer, manufacturer stock							
	or type no., RG no.							
14.1	Power Cable Set (W101	1						
	through W120)							
15.1	Power Cable Set (W101	1						
	through W116, W121 through							
	W124)							
16.1	Cable Stringer							
17.1	Tactical Intercommunications	1			Cooke Electronics			
	System - Operator Station							

Sheet No. 9

MINITRACK EQUIPMENT INVENTORY (136 MHz)

Tracking Station Name: ____________

Inventory Completed By:

____________________ (Signature)

____________________ (Date)

ITEM No.	EQUIPMENT NAME (If non-existent, write none)	Standard Quantity	Quantity installed	Spares at your site	STANDARD MANUFACTURER (Check and correct if non-standard)	MFG. STOCK OR MODEL NO.	SERIAL NUMBER	COMMENTS
18.1	Primary Power Generator							

PRECEDING PAGE BLANK NOT FILMED

Appendix C

SYSTEM CALIBRATION

Each Minitrack station is calibrated once or twice a year to obtain constants and coefficients that are used in compensating the system equipment. Such compensation is done, in part, at the respective worldwide tracking facilities. The remainder of the compensatory adjustments are achieved through the Minitrack preprocessing computer program.

The techniques employed in the calibration of the Minitrack system use an astrographic camera at each tracking site. This camera is mounted precisely at the intersection of the fine baselines. An aircraft with a high-intensity flashing light is employed in the procedure. The aircraft is equipped with a radiofrequency beacon transmitter operating in the Minitrack frequency spectrum. At night, while the high-flying aircraft is being tracked by the Minitrack station, the astrographic camera is made to photograph the flashing light against a fixed known star field background.

The constants and coefficients of calibration are derived by comparing the known position of the aircraft beacon transmitter (and flashing light) with the instantaneous positions of the beacon as defined by the Minitrack station. The process of obtaining the calibration constants from the information imaged on the astrographic camera plates and the reduction of the data to meaningful terms is accomplished by detailed use of precision measuring machines and the general-purpose digital computer (IBM 360). The purpose of this section is to introduce the reader to the mechanics of the computer reduction of the astrographic-plate data that ultimately define the respective calibration coefficients. A previously unpublished description of the operations and functions of three computer programs used in the calibration data reduction process is published here. Additionally, a listing of the programs is provided for review and reference.

Readers unfamiliar with the Minitrack calibration procedure should refer to several documents already in print. Among these is the excellent description of the Minitrack calibration system[1] by Berbert, Oosterhout, Engels, and Habib and a paper on the reduction of the Minitrack astrographic plates[2] by Good, Berbert, and Oosterhout. These documents are recommended for an understanding of the program descriptions that follow. The basic

[1] J. H. Berbert, J. D. Oosterhout, P. D. Engels, and E. J. Habib: "Minitrack Calibration System." *Photograph. Sci. Eng.* 7(2): 78-83, Mar.-Apr. 1963.

[2] Emily W. Good, J. H. Berbert, and J. D. Oosterhout: "Reduction of the Minitrack Astrographic Plates." *Photograph. Sci. Eng.* 6(6): 324-327, Nov.-Dec. 1962.

information in these articles will not be repeated because such effort would be redundant; they are important to understanding, nonetheless. Our emphasis is directed toward the more detailed, hitherto unpublished descriptive material.

The reduction of the calibration data must be keyed to the fundamental star positions used for the reference grid on the calibration plates. Optical images of the strobe light mounted on the calibration aircraft are shifted by tropospheric refraction from the atmosphere between the aircraft and the ground station, whereas the optical images of the standard stars are shifted by refraction from the entire atmosphere. The fundamental star positions quoted in the star catalogs are corrected for this refraction from the entire atmosphere; equivalently, they are the true positions or "zenith" positions. Therefore, the optical images of the aircraft must be corrected for tropospheric refraction to deduce its true or zenith position.[3] In this manner, the difference in plate coordinates of the images of the aircraft and stars yields the difference in their true coordinates. By comparing the 136.5-MHz radio position of the aircraft determined from the standard reduction of the interferometer data and the optical position of the aircraft for the same instant of time, the calibration procedure is established. For commensurability, the radio position must also be corrected for the tropospheric refraction from the atmosphere between the aircraft and the ground station.

The calibration computations are performed by three computer programs at the Physical Science Laboratory of New Mexico State University, Las Cruces, N. Mex. These programs in the order of use are DR01J, astrographic plate reduction; DR01K, PROOFREAD; and DR01E, antenna-field error models. Each of these programs as supplied by the Physical Science Laboratory is described in the next section. A listing of the calibration computer programs is given at the end of this appendix.

DR01J

Problem Definition

This program is designed for reduction of Minitrack astrographic plates or PACT data. The objective is to compute east-west and north-south Minitrack parameters versus time for all associated antenna systems.

Plate Reduction

A flashing light aboard an aircraft, offset a known amount from a radiofrequency source, is photographed against a star background. Updated star catalog positions and plate measurements can be used to find plate constants that can be used with the plate measurement of any image to yield the direction of the line between the camera and the object.

[3] This correction for the optical radiation between the aircraft and the ground station is from F. O. Vonbun: *Correction for Atmospheric Refraction at the NASA Minitrack Stations.* NASA TN D-1448, Aug. 1962.

When the object is the flashing light, the offset coordinates can be used to compute the direction to the radiofrequency source at the corresponding time. This direction then allows computation of the desired Minitrack parameters.

The star positions updated to the time of plate exposure and the Cartesian coordinates of the images with respect to the approximate principal point must be known. There can be up to 75 stars.

The plate constants we wish to solve for are the coefficients $a, b, \ldots, f'$ in the following equations:

$$\xi = a + bx + cy + dxy + ex^2 + fx(x^2 + y^2) \tag{C-1a}$$

$$\eta = a' + b'x + c'y + d'xy + e'y^2 + f'y(x^2 + y^2) \tag{C-1b}$$

Here x and y must be referred to the approximate principal point whose plate measurement is x_c, y_c. The variables ξ and η are standard coordinates that can be found from the star position as follows. Denote the right ascension and declination of the star by α^* and δ^*, respectively, and of the approximate principal point by α^c and δ^c, respectively. Then,

$$\xi = \frac{\cos \delta^* \sin (\alpha^* - \alpha^c)}{D} \tag{C-2}$$

$$\eta = \frac{\sin (\delta^* - \delta^c) - \sin \delta^c \cos \delta^* [\cos (\alpha^* - \alpha^c) - 1]}{D} \tag{C-3}$$

where

$$D = \cos (\delta^* - \delta^c) + \cos \delta^c \cos \delta^* [\cos (\alpha^* - \alpha^c) - 1] \tag{C-4}$$

The plate coordinates of each star image are measured from one to five times. If more than three readings are taken, the set showing the highest deviation in either x or y should be discarded. The values of x and y to be used in all computations are then the average values of the remaining measurement sets less x_c and y_c, the coordinates of the plate center.

The plate constants are found by solving equation (C-1a) by the method of least squares:

$$
\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} N & \sum x & \sum y & \sum xy & \sum x^2 & \sum xr \\ \sum x & \sum x^2 & \sum xy & \sum x^2 y & \sum x^3 & \sum x^2 r \\ \sum y & \sum xy & \sum y^2 & \sum xy^2 & \sum x^2 y & \sum xyr \\ \sum xy & \sum x^2 y & \sum xy^2 & \sum x^2 y^2 & \sum x^3 y & \sum x^2 yr \\ \sum x^2 & \sum x^3 & \sum x^2 y & \sum x^3 y & \sum x^4 & \sum x^2 r \\ \sum xr & \sum x^2 r & \sum xyr & \sum x^2 yr & \sum x^3 r & \sum x^2 r^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum \xi \\ \sum x\xi \\ \sum y\xi \\ \sum xy\xi \\ \sum x^2 \xi \\ \sum xr\xi \end{bmatrix} \qquad \text{(C-5)}
$$

where the summations are over N stars and $r = x^2 + y^2$. The variables $a', b', \ldots, f'$ are found from the similar set of linear equations associated with the equation for η in equation (C-1b).

Once the plate constants are known, equation (C-1) will yield standard coordinates ξ and η for any x and y. The corresponding right ascension α and declination δ are found from the following relations:

$$
\alpha = \alpha^c + \Delta\alpha \qquad \text{(C-6a)}
$$

$$
\delta = \tan^{-1} \left(\frac{\sin \delta^c + \eta \cos \delta^c}{\cos \delta^c - \eta \sin \delta^c} \right) \cos \Delta\alpha \qquad \text{(C-6b)}
$$

where

$$
\Delta\alpha = \tan^{-1} \frac{\xi}{\cos \delta^c - \eta \sin \delta^c}
$$

$$
= \sin^{-1} \frac{\xi \sin \delta^*}{\cos \delta^c \cos \delta^* \; (\eta + \sin \delta^c / \cos \delta^c)} \qquad \text{(C-7)}
$$

There will be some error associated with each star because of distortion, refraction, catalog errors, measurement errors, misidentification, or other blunders. To eliminate stars with errors greater than some acceptable r_{max}, the following procedure is used.

Initially use all input stars to solve for plate constants. Then compute residuals for each star:

$$r_\alpha = (\alpha - \alpha^*) \cos \delta^* \tag{C-8a}$$

$$r_\delta = \delta - \delta^* \tag{C-8b}$$

where α and δ are found from the plate measurements, plate constants, and equations (C-6) and (C-7). If any residual exceeds $r_{\max}$, remove the star showing the largest residual (in either right ascension or declination) and compute a new set of plate constants. Again examine the residuals and remove another star if necessary. Keep repeating this process until all residuals are less than or equal to $r_{\max}$ or until more than eight stars would have to be removed from the solution. The root mean square (rms) values for r_α and r_δ should be computed also.

For each flashing light image on the plate, compute ξ and η by equation (C-1), and then α and δ by equations (C-6) and (C-7). Direction cosines are given by

$$\lambda = \sin \delta \cos \phi - \cos \delta \sin \phi \cos H \tag{C-9a}$$

$$\mu = - \cos \delta \sin H \tag{C-9b}$$

$$\nu = \sin \delta \sin \phi + \cos \delta \cos \phi \cos H \tag{C-9c}$$

where ϕ is the latitude and H is the local hour angle given by

$$H = \mathrm{STG} + \mathrm{C} - \mathrm{L} - \alpha + \mathrm{UT} \tag{C-10}$$

where STG is the sidereal time, C is 9.8565 s of time per hour of UT, L is the longitude, and UT is the universal time of the flash.

PACT Data

When PACT data are to be reduced, the direction cosines are given as input, and therefore the computation of plate constants and direction cosines may be omitted.

Phase Center Corrections

The azimuth A and elevation ϵ are found from the direction cosines by

$$A = \tan^{-1} \frac{\mu}{\lambda} \tag{C-11a}$$

$$\epsilon = \sin^{-1} \nu \tag{C-11b}$$

The quadrant of A is determined by the fact that μ and λ have the same signs as $\sin A$ and $\cos A$, respectively.

Given phase center offsets F and P, the problem is to find direction cosines to the radio-frequency source before performing any Minitrack computations.

The course of the aircraft must be known. The angle C is defined as the angle between the east direction and the line of flight of the aircraft, measured clockwise from east. Because two successive observations are required to determine the direction of aircraft flight, it will be necessary to consider the course as constant for the first two data points. Also, if two observations are more than 2 s apart, the first should be treated as a final point and the second as an initial point in these computations. Denote the aircraft height by H_A and station height by H_S. Then for point i,

$$E_i = \frac{\mu_i}{\nu_i}(H_A - H_S) \tag{C-12a}$$

$$N_i = \frac{\lambda_i}{\nu_i}(H_A - H_S) \tag{C-12b}$$

Denote any two successive observations by the subscripts i and $i-1$. Then

$$C_i = \tan^{-1} \frac{N_i - N_{i-1}}{E_i - E_{i-1}} \tag{C-13}$$

for $|N_i - N_{i-1}| \leqslant |E_i - E_{i-1}|$. Add π to C_i if $E_i < E_{i-1}$. If $E_i \geqslant E_{i-1}$ and $C_i < 0$, add 2π to C_i.

$$C_i = \frac{\pi}{2} - \tan^{-1} \frac{E_i - E_{i-1}}{N_i - N_{i-1}} \tag{C-14}$$

for $|E_i - E_{i-1}| < |N_i - N_{i-1}|$. Add π to C_i if $N_i < N_{i-1}$.

Once C_i has been properly determined, we can find the desired direction cosines:

$$E_i' = F \cos C_i - P \sin C_i + E_i \tag{C-15a}$$

$$N_i' = F \sin C_i + P \cos C_i + N_i \tag{C-15b}$$

from which

$$\lambda_i' = N_i'/R_i' \tag{C-16a}$$

$$\mu_i' = E_i'/R_i' \tag{C-16b}$$

$$\nu_i' = \nu_i \tag{C-16c}$$

where

$$R_i' = [E_i'^2 + N_i'^2 + (H_A - H_S)^2]^{½} \tag{C-17}$$

The direction cosines defined in equations (C-16) are the ones to be used in subsequent Minitrack computations.

Baseline in Wavelengths

It will be necessary to know the number of wavelengths in the baseline of all antenna systems. The number of wavelengths in the baseline for a vacuum is given by

$$n = \frac{\text{baseline length} \times \text{frequency}}{\text{vacuum velocity of light}} \tag{C-18}$$

using 9.83569229 X 10^8 ft/s for the vacuum velocity of light.[4] The actual number of wavelengths in the baseline is

$$n' = (1 - R)\bar{\mu}n \tag{C-19}$$

where R is a quantity based on the optical index of refraction between the station and the aircraft, which is given by

$$R = \frac{77.6P}{T} \, \frac{1-e^{-\gamma(H_A - H_S)}}{\gamma(H_A - H_S)} \times 10^{-6} \tag{C-20}$$

The factor γ is a constant dependent on atmospheric conditions, for which we use the value 3.048 X 10^{-5}. The units of R in equation (C-20) are arcseconds and must be converted to radians for use in equation (C-19). For PACT data computation, R is set equal to zero. In equation (C-19), $\bar{\mu}$ is the index of refraction for the radiofrequency between the station and the aircraft and is computed from

$$\bar{\mu} = 1 + N_S \frac{1-e^{-k(H_A - H_S)}}{k(H_A - H_S)} \tag{C-21}$$

where

$$N_S = \frac{77.6}{T}\left(P + \frac{4810\, P\nu R_H}{T}\right) \times 10^{-6} \tag{C-22}$$

[4] This value is derived from the value given in NASA SP-7012 (1969), which is 2.9979250 X 10^{10} cm/s. The conversion factor is 1 m = 3.2808333333 U.S. survey ft.

in which

P = pressure in millibars, which is 0.02953144 in. Hg
Pv = saturated vapor pressure (given in tabular form as a function of temperature in degrees Celsius in the *Handbook of Chemistry and Physics*)
R_H = relative humidity
T = temperature in kelvins, which is 273.13 + (5/9) (°F - 32)

In equation (C-21)

$$k = \ln \frac{N_S}{N_S + \Delta N} \tag{C-23}$$

where

$$\Delta N = -7.32 e^{5577 N_S} \times 10^{-6} \tag{C-24}$$

The variable k as given by equation (C-23) is in units of reciprocal kilometers.

Fine Antennas

The Minitrack parameters A_c and B_c are given by

$$\left.\begin{aligned} A_c &= a_0 - F_{ew} - V_{ew} \\ B_c &= b_0 - F_{ns} - V_{ns} \end{aligned}\right\} \tag{C-25}$$

where

$$a_0 = \frac{n'\mu'}{\left[1 + (1/4r^2)(n^2 - n^2\mu'^2)\right]^{1/2}} \tag{C-25a}$$

$$b_0 = \frac{n'\lambda'}{\left[1 + (1/4r^2)(n^2 - n^2\lambda'^2)\right]^{1/2}} \tag{C-25b}$$

$$r = \frac{H_A - H_S}{\nu'} \times \frac{\text{frequency}}{\text{velocity of light}} \tag{C-25c}$$

$$F_{ew} = \frac{\Delta a_0}{\Delta \text{UT}} \times \text{(east-west filter constant)} \tag{C-25d}$$

$$F_{ns} = \frac{\Delta b_0}{\Delta \text{UT}} \times \text{(north-south filter constant)} \tag{C-25e}$$

$$V_{\mathrm{ew}} = \frac{-\Delta a_0}{\Delta \mathrm{UT}} V_f A_m \qquad \text{(C-25f)}$$

$$V_{\mathrm{ns}} = \frac{-\Delta b_0}{\Delta \mathrm{UT}} V_f B_n \qquad \text{(C-25g)}$$

The east-west and north-south filter constants and the velocity constant V_f are given. The variables A_m and B_m are the decimal portions of the Minitrack values. $\Delta a_0/\Delta \mathrm{UT}$ and $\Delta b_0/\Delta \mathrm{UT}$ are the rates of change in a_0 and b_0 computed from two successive points. It is necessary to consider the rates as constant for the first two points associated with a plate.

The variables n and n' are defined in equations (C-18) and (C-19) where the east-west fine baseline is used for computing a_0 and the north-south fine baseline is used for computing b_0.

Ambiguity Antennas

Minitrack parameters for the ambiguity antennas are computed only when input time is an integral number of seconds.

A_c and B_c for medium and coarse antennas are computed by equations (C-25), setting V_{ew} and V_{ns} equal to zero and using the appropriate baseline lengths. For A_c medium, use baseline east-west medium; for A_c coarse, use baseline east-west coarse. For B_c medium, use baseline north-south medium; and for B_c coarse, use baseline north-south coarse.

In addition, the direction cosines to be used are derived from the ones that are used to compute the fine data. The coordinates of the radiofrequency source with respect to the camera (located at the center of the fine-antenna system) are

$$X_A = \frac{(H_A - H_S)\lambda'}{\nu'} \qquad \text{(C-26a)}$$

$$Y_A = \frac{(H_A - H_S)\mu'}{\nu'} \qquad \text{(C-26b)}$$

$$Z_A = H_A - H_S \qquad \text{C-26c)}$$

The direction cosines of the radiofrequency source with respect to any other point P are

$$\lambda_P = \frac{X_A - X_i}{r} \qquad \text{(C-27a)}$$

$$\mu_P = \frac{Y_A - Y_i}{r} \qquad \text{(C-27b)}$$

$$\nu_P = \frac{Z_A - Z_i}{r} \qquad \text{(C-27c)}$$

where

$$r = [(X_A - X_i)^2 + (Y_A - Y_i)^2 + (Z_A - Z_i)^2]^{1/2}$$

and X_i, Y_i, and Z_i are the coordinates of point P with respect to the camera. λ_P, μ_P, ν_P are the direction cosines to be used for λ, μ, and ν in equations (C-25). We will assume $Z_i = 0$ in all cases. The variables X_i and Y_i are given as translation coordinates. For A_c medium, use $X_i = X_{\text{ewm}}$ and $Y_i = Y_{\text{ewn}}$; for A_c coarse, use $X_i \fallingdotseq X_{\text{ewc}}$ and $Y_i = Y_{\text{ewc}}$. For B_c medium, use $X_i = X_{\text{nsm}}$ and $Y_i = Y_{\text{nsm}}$; for B_c coarse, use $X_i = X_{\text{nsc}}$ and $Y_i = Y_{\text{nsc}}$.

Program Description

The program consists of one main procedure that controls three external procedures. The flowchart (fig. C-1) gives more detail about the sequence of operations in each procedure. The equation numbers 1 through 25 in the flowchart refer to equations (C-1) through (C-25).

Main Procedure

This procedure reads the Start card (fig. C-2). If a new data set is to be processed, INITIAL should be punched on the Start card. All preexisting data are discarded from this area of the disk. If plates are being added to or replacing data already on the disk (signaled by ADDITION on the Start card), the OUTPUT procedure is invoked to save the data already on the disk in a special file.

A Plate card (fig. C-3) is read in. If PACT data are indicated, the MTRACK procedure is invoked; if plate constants are indicated, as input, the cards are read. If they are to be computed, the STRSOL procedure is invoked. After plate constants have been computed or read in, the MTRACK procedure is invoked.

After all plates have been processed in the above manner, the OUTPUT procedure is invoked.

STRSOL Procedure

This procedure computes plate constants using updated star catalog positions and the coordinates of the star images on photographic plates as described earlier. All of the plate coordinates for the stars are read in, after which the updated star catalog positions are read.

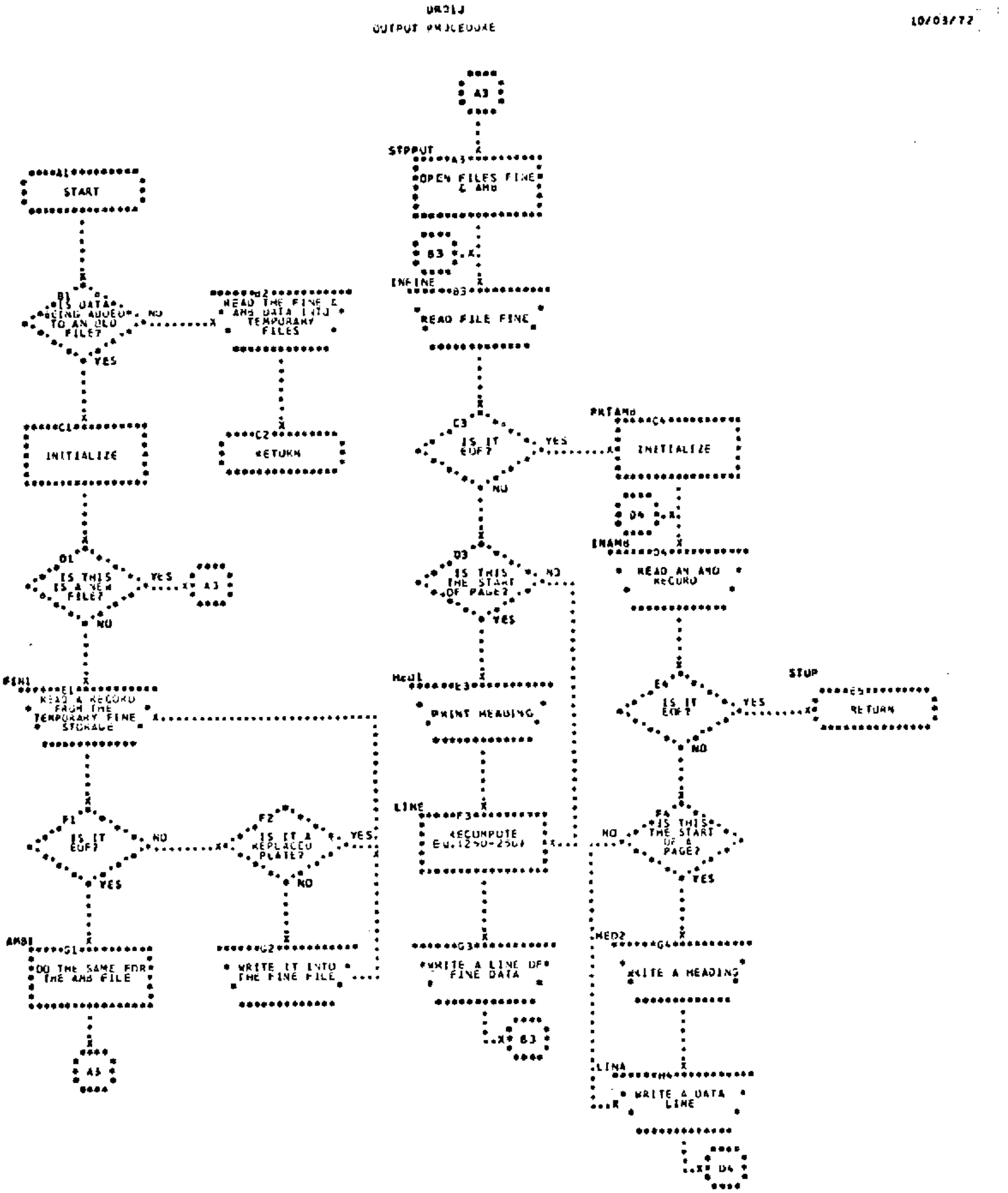


Figure C-1.—Flowchart for DR01J.

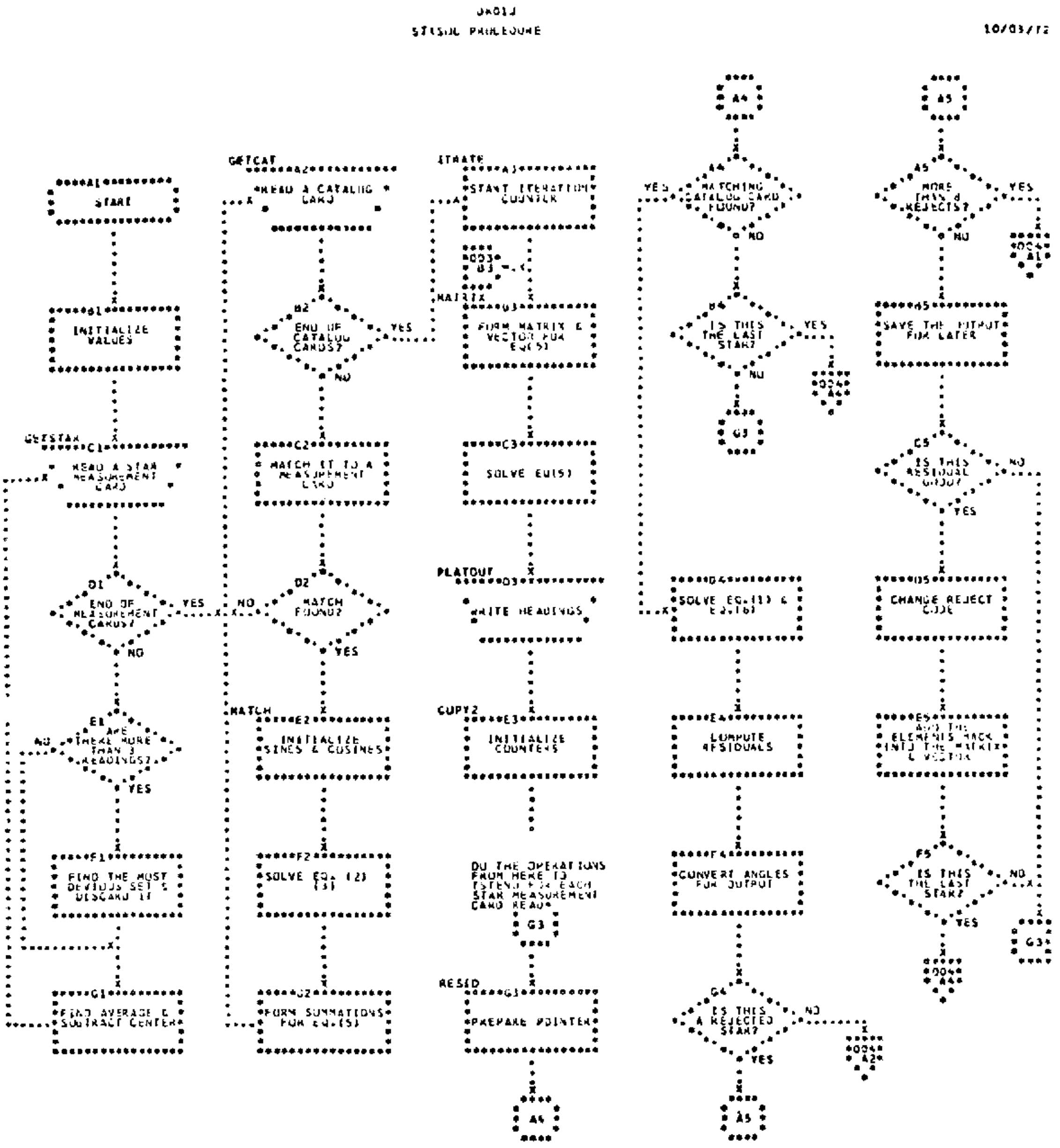


Figure C-1 (continued).–Flowchart for DR01J.

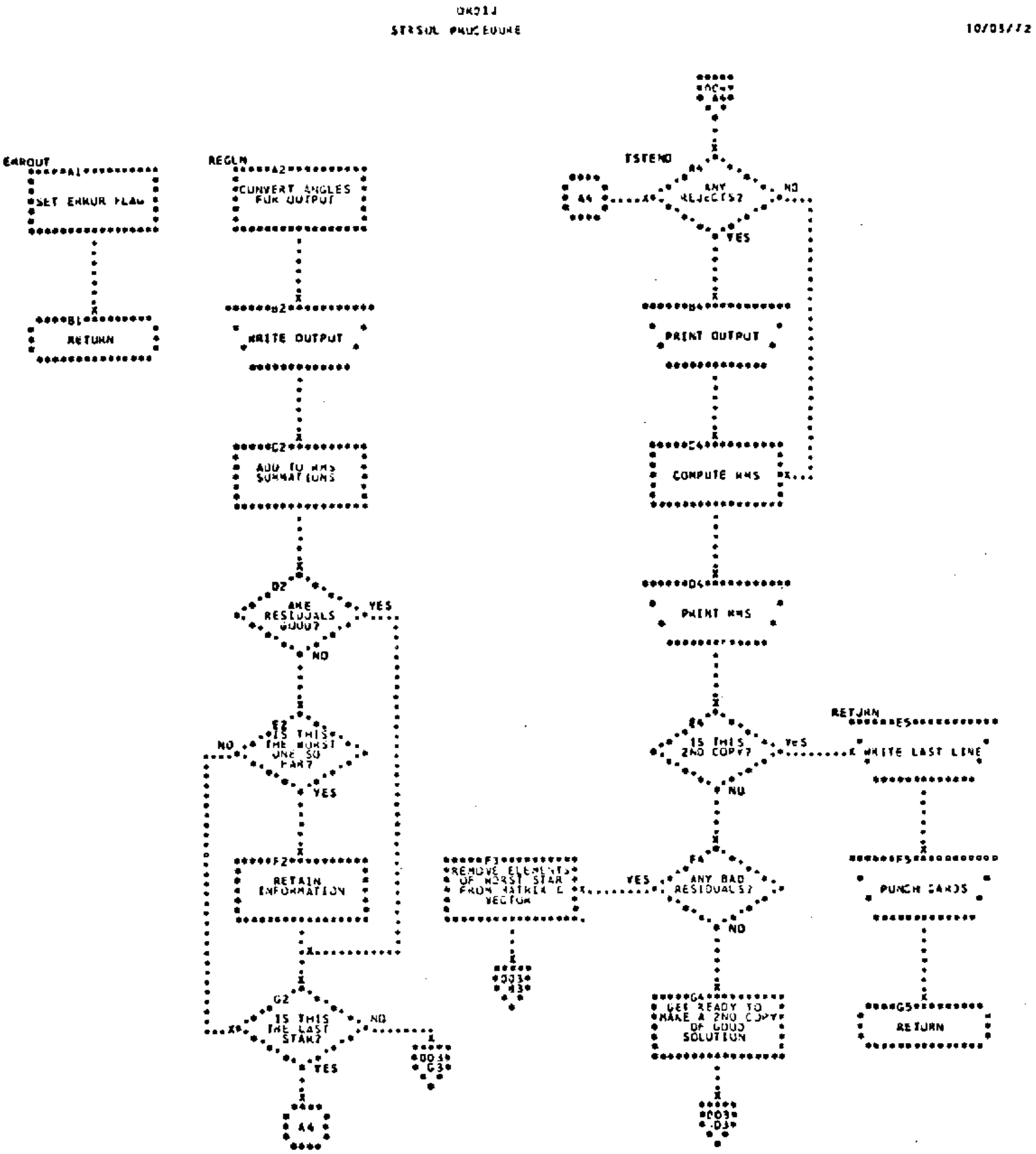


Figure C-1 (continued).—Flowchart for DR01J.

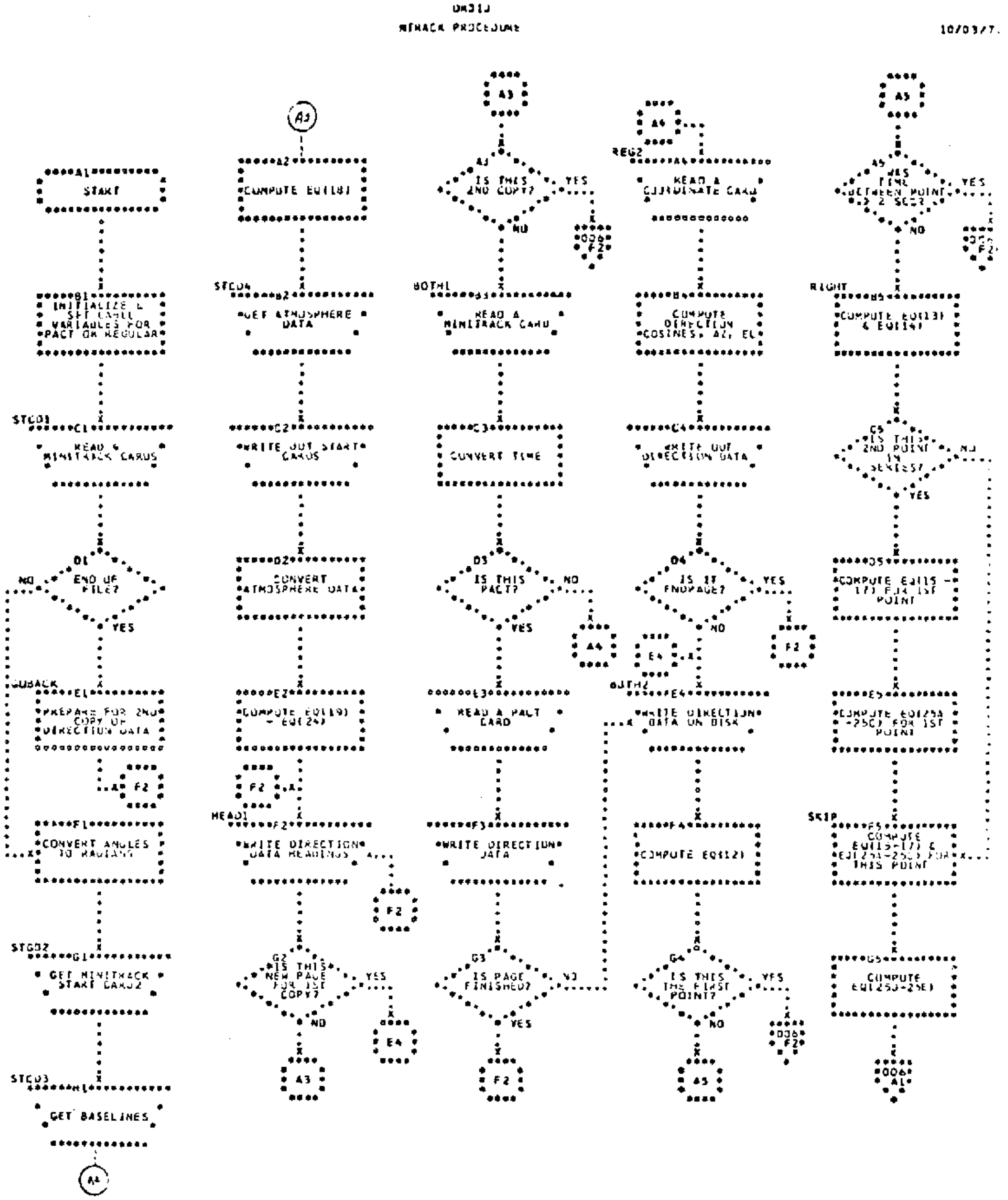


Figure C-1 (continued).—Flowchart for DR01J.

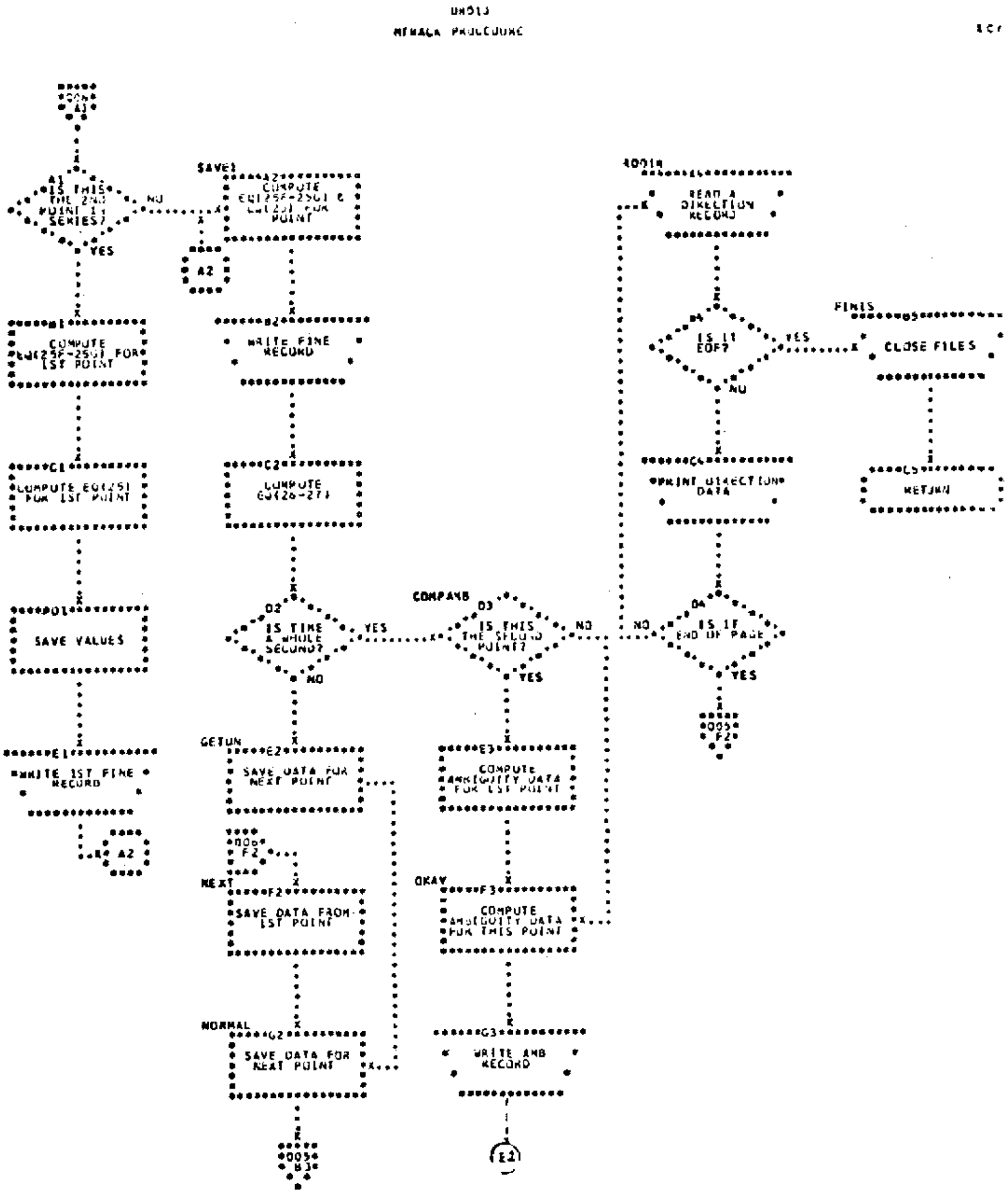

DR01J
NFWACK PROCEDURE
1C7
005 A1
A1 IS THIS THE 2ND POINT IN SERIES?
NO
YES
SAVE1
A2 COMPUTE EQ(25F-25G) & EQ(23) FOR POINT
A2
B1 COMPUTE EQ(25F-25G) FOR 1ST POINT
B2 WRITE FINE RECORD
C1 COMPUTE EQ(25) FOR 1ST POINT
C2 COMPUTE EQ(26-27)
D1 SAVE VALUES
D2 IS TIME A WHOLE SECOND?
YES
NO
COMPAMB
D3 IS THIS THE SECOND POINT?
NO
YES
E1 WRITE 1ST FINE RECORD
A2
GETUN
E2 SAVE DATA FOR NEXT POINT
E3 COMPUTE AMBIGUITY DATA FOR 1ST POINT
006 F2
NEXT
F2 SAVE DATA FROM 1ST POINT
OKAY
F3 COMPUTE AMBIGUITY DATA FOR THIS POINT
NORMAL
G2 SAVE DATA FOR NEXT POINT
G3 WRITE AMB RECORD
005 B3
E2
RDDIR
A4 READ A DIRECTION RECORD
B4 IS IT EOF?
YES
NO
FINIS
B5 CLOSE FILES
C4 PRINT DIRECTION DATA
C5 RETURN
D4 IS IT END OF PAGE
NO
YES
005 F2

Figure C-1 (continued).—Flowchart for DR01J.

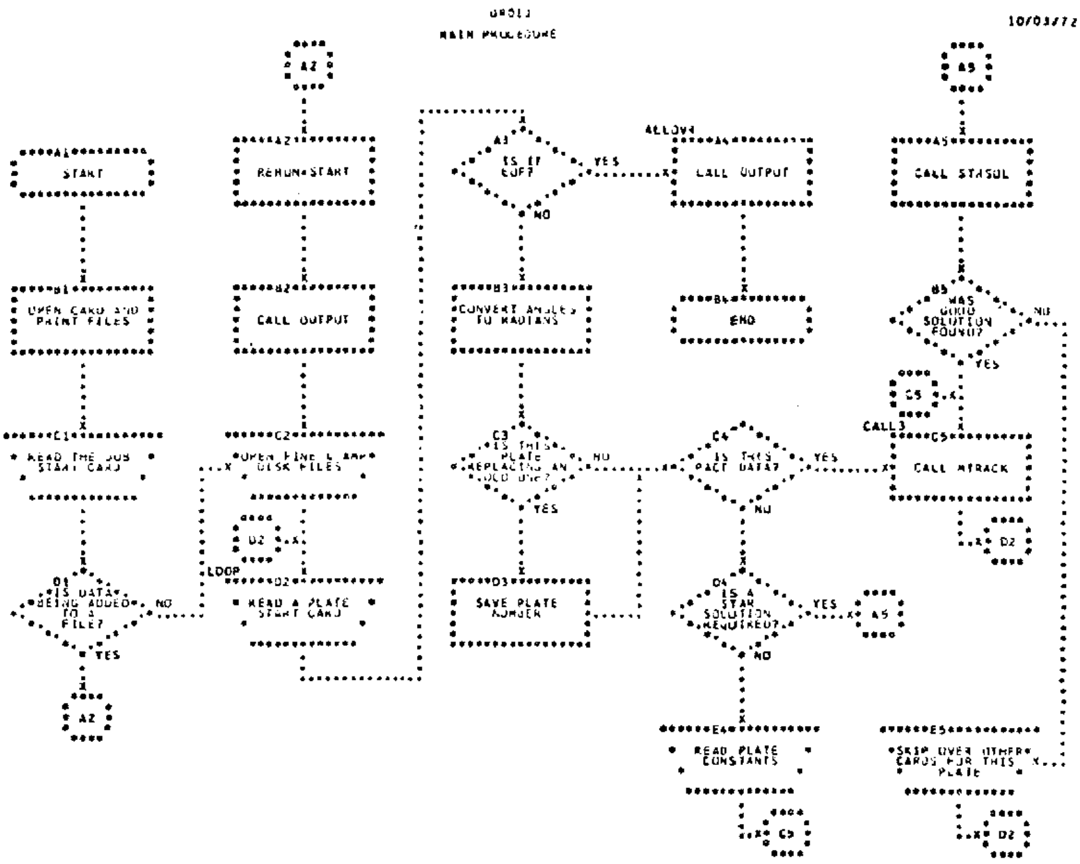


Figure C-1 (concluded).—Flowchart for DR01J.

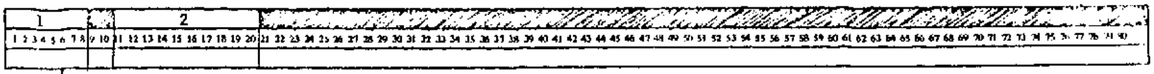
As each catalog card is read, the coordinate cards are searched for a match. If both cards are found for a star, then the components for that star are added into the matrix of equation (C-5).

Processing continues as described earlier until a satisfactory plate constant solution is obtained or until more than eight stars have been rejected. Control is then returned to MAIN. A sample output is shown in figure C-4.

MTRACK Procedure

This procedure computes the Minitrack parameters versus time for each antenna system. The Minitrack Start cards are read. Values that are constant for a given plate are computed, and direction data headings are printed.

The input cards for each point of time are assumed to be interfiled: a Minitrack card followed by its corresponding PACT card or Measurement card. For each point, direction cosines are either computed from the plate constants as described earlier or given on the PACT card. Then computations for phase center corrections, baselines, and fine and ambiguity antennas are performed. The results of these computations are saved in the FINE and AMB disk files.



FIELD(S)

1 INITIAL - If a new disk file is being created (left-justify)

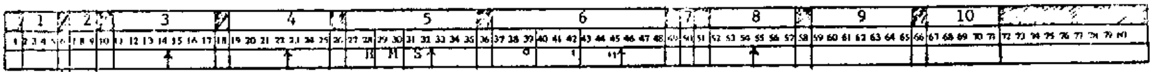
ADDITION - If plates are being added to files already on the disk

2 POLAR - If this is polar data

EQUATORIAL - If it is equatorial data

This parameter is used only for output listings. For best appearance POLAR should be centered in the field.

Figure C-2.—Start card for DR01J.



FIELD(S)

1 Station Number }
2 Plate Number } Character data

3 X_c }
4 Y_c } The coordinates (in mm) of the approximate principal point (floating point)

5 α_c Right ascension (hrs,min,sec) of the approximate principal point. Right-justify the hours and minutes (fixed integers) and allow 3 decimal places for seconds.

6 δ_c Declination (in degrees,min,and sec) of the approximate principal point. Right-justify degrees and minutes (fixed integers) and allow 3 decimal places for seconds. If the angle is negative, put a minus sign in front of each of the components. If any component is zero, it may be left blank.

7 The number of measurements (1-5) taken on the coordinates of each star and each flashing light. This may be zero or blank for PACT data.

8 The maximum residual (r_α or r_δ in eq. (C-8)) allowed for a star to stay in the solution. This may be zero or blank for PACT data or if plate constants are given.

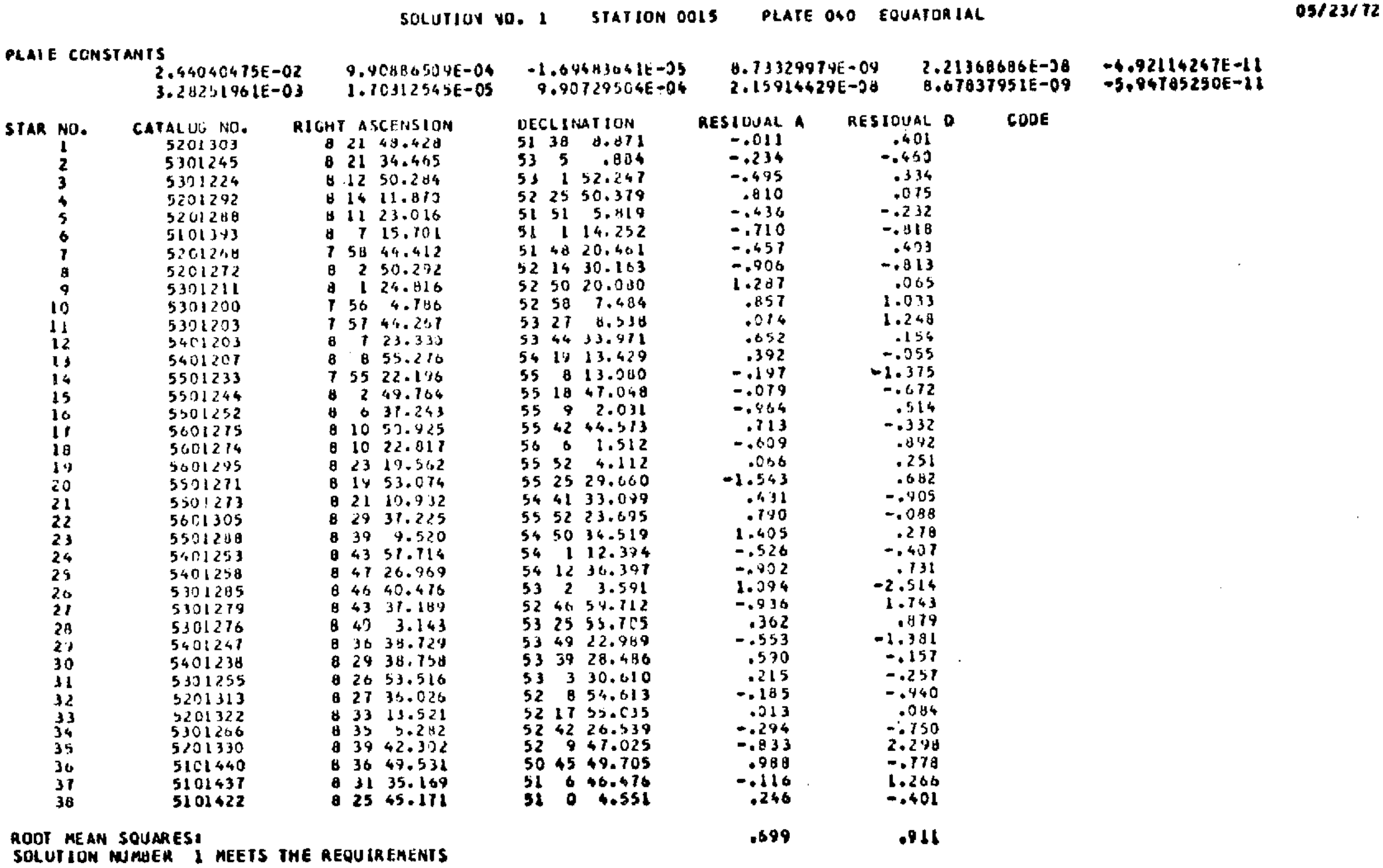
9 REPLACE If this plate is replacing the data for a plate with the same number already in the disk file. Leave the field blank if this is the first time data for this plate has been run.

10 PLATE If the plate constants for this plate have already been computed and appear on plate constant cards which follow.

or

PACT (Left-justified) If there are PACT data for this plate. Leave the field blank if plate constants are to be computed.

Figure C-3.—Plate card for DR01J.



```
                              SOLUTION NO. 1    STATION 0015    PLATE 040  EQUATORIAL                          05/23/72

PLATE CONSTANTS
                2.44040475E-02   9.90886509E-04  -1.69483641E-05   8.73329979E-09   2.21368686E-38  -4.92114247E-11
                3.28251961E-03   1.70312545E-05   9.90729504E-04   2.15914429E-08   8.67837951E-09  -5.94785250E-11

STAR NO.    CATALOG NO.    RIGHT ASCENSION      DECLINATION     RESIDUAL A   RESIDUAL D     CODE
   1          5201303        8 21 48.428        51 38  8.871       -.011        .401
   2          5301245        8 21 34.465        53  5   .804       -.234       -.460
   3          5301224        8 12 50.284        53  1 52.247       -.495        .334
   4          5201292        8 14 11.870        52 25 50.379        .810        .075
   5          5201288        8 11 23.016        51 51  5.819       -.436       -.232
   6          5101393        8  7 15.701        51  1 14.252       -.710       -.818
   7          5201268        7 58 44.412        51 48 20.461       -.457        .493
   8          5201272        8  2 50.292        52 14 30.163       -.906       -.813
   9          5301211        8  1 24.816        52 50 20.080       1.287        .065
  10          5301200        7 56  4.786        52 58  7.484        .857       1.033
  11          5301203        7 57 44.267        53 27  8.538        .074       1.248
  12          5401203        8  7 23.330        53 44 33.971        .652        .154
  13          5401207        8  8 55.276        54 19 13.429        .392       -.055
  14          5501233        7 55 22.196        55  8 13.080       -.197      -1.375
  15          5501244        8  2 49.764        55 18 47.048       -.079       -.672
  16          5501252        8  6 37.243        55  9  2.031       -.964        .514
  17          5601275        8 10 50.925        55 42 44.573        .713       -.332
  18          5601274        8 10 22.817        56  6  1.512       -.609        .892
  19          5601295        8 23 19.562        55 52  4.112        .066        .251
  20          5501271        8 19 53.074        55 25 29.660      -1.543        .682
  21          5501273        8 21 10.932        54 41 33.099        .431       -.905
  22          5601305        8 29 37.225        55 52 23.695        .790       -.088
  23          5501288        8 39  9.520        54 50 34.519       1.405        .278
  24          5401253        8 43 57.714        54  1 12.394       -.526       -.407
  25          5401258        8 47 26.969        54 12 36.397       -.902        .731
  26          5301285        8 46 40.476        53  2  3.591       1.094      -2.514
  27          5301279        8 43 37.189        52 46 59.712       -.936       1.743
  28          5301276        8 40  3.143        53 25 55.705        .362        .879
  29          5401247        8 36 38.729        53 49 22.989       -.553      -1.381
  30          5401238        8 29 38.758        53 39 28.486        .590       -.157
  31          5301255        8 26 53.516        53  3 30.610        .215       -.257
  32          5201313        8 27 36.026        52  8 54.613       -.185       -.940
  33          5201322        8 33 13.521        52 17 55.035        .013        .084
  34          5301266        8 35  5.282        52 42 26.539       -.294       -.750
  35          5201330        8 39 42.302        52  9 47.025       -.833       2.298
  36          5101440        8 36 49.531        50 45 49.705        .988       -.778
  37          5101437        8 31 35.169        51  6 46.476       -.116       1.266
  38          5101422        8 25 45.171        51  0  4.551        .246       -.401

ROOT MEAN SQUARES:                                                  .699         .911
 SOLUTION NUMBER  1 MEETS THE REQUIREMENTS
```

Figure C-4.–Sample output listing of star solution.

The direction data are printed as they are computed for each point. Another copy is printed after all computations are complete and control is returned to MAIN.

OUTPUT Procedure

This procedure can be invoked by MAIN in two instances. If plates are being added to a file already on the disk, the procedure is invoked. At that time the data already in the file are read and saved in a separate file, and control is returned to MAIN. After the last plate in a run has been processed by MTRACK, this procedure is invoked to edit the disk files and produce the fine-antenna and ambiguity-antenna printed output. If data are being added to a file or replacing plates already on the file, the data saved by the first call of this procedure are read in. If the plate number is one of those designated to be replaced by the Plate card, the data are skipped; otherwise, the data are added to the file containing the more recently computed data. That process is repeated for both the fine and ambiguity data.

All of the data in the fine and ambiguity files are then printed with a separate listing for each plate.

The values from equations (C-25d) to (C-25g) are computed in this procedure because they are not retained in the FINE disk file.

After all listings are complete, control is returned to MAIN.

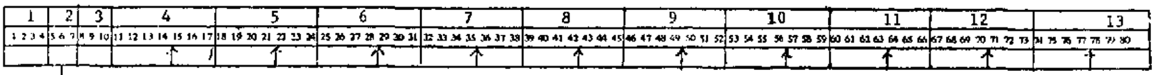
Input

Punched Cards

The cards that constitute a data deck are

(1) Start card
(2) Plate card
(3) Star Measurement cards followed by a card with a star number <75 (fig. C-5)
(4) Catalog cards followed by a card with a star number <75 (fig. C-6)
(5) Minitrack Start card 1 (fig. C-7)
(6) Minitrack Start card 2 (fig. C-8)
(7) Minitrack Start card 3 (fig. C-9)
(8) Minitrack Start card 4 (fig. C-10)
(9) Minitrack cards (fig. C-11) }
(10) Flashing light Measurement cards (fig. C-12) or PACT data cards } Interfiled
(fig. C-13)
(11) /* end of file
(12) /* end of file

Cards 2 through 11 may be repeated for as many plates as the user wishes. The only limitation is the size of the area allotted to the FINE and AMB files. Cards 3 and 4 may be replaced by a Plate Constant card (fig. C-14), or they may be omitted if PACT data are given. Figure C-15 is a listing of typical Minitrack Start cards 1 through 4.



FIELD(S)		
1	Station number	Character data not checked by the program
2	Plate Number	
3	Star number - Fixed integer, right-justified	
4	X_1 coordinate measurement	All measurements in mm with 3 decimal places allowed. The number of sets of measurements used by the program is in field 7 of the Plate Card.
5	Y_1 coordinate measurement	
6	X_2	
7	Y_2	
8	X_3	
9	Y_3	
10	X_4	
11	Y_4	
12	X_5	
13	Y_5	

Figure C-5.–Star Measurement card for DR01J.

Disk

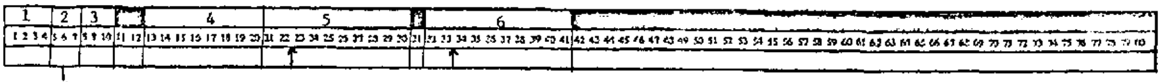
If ADDITION is specified on the Start card, FINE and AMB files are expected to be in the output disk area. The data in those areas will be saved and added to the file after the plates of the run are processed. Therefore any plate added to a file will be first in order in that file.

Output

Printed Output

As the plates are processed a listing is made of direction data (fig. C-16). For PACT data the time, right ascension, declination, azimuth, elevation, and direction cosines are printed for each point; for other data the time, azimuth, and elevation are printed. As each line is printed a copy is saved in disk file DIROUT. After all the points for one plate have been read and processed, a second copy is made of the direction data from the data saved on DIROUT.

After all the plates for a run have been processed and data saved from a previous run have been added back into the FINE and AMB files, all of the fine data are listed (time, A_c, B_c, A_m, B_m, Δa_0, Δb_0, N_A, N_B, course, range, F_{ew}, F_{ns}, V_{ew}, and V_{ns}) for each point with a separate listing started for each plate in the file (fig. C-17). Then all of the ambiguity data are listed (flash time, A_c medium, A_c coarse, B_c medium, and B_c coarse) for each point, again with a separate listing for each plate (fig. C-18).



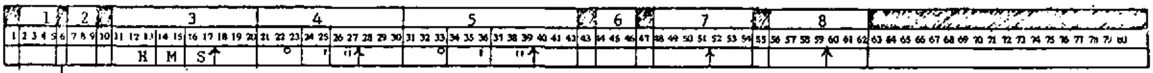
1 Station number
2 Plate number } Character data not checked by the program

3 Star number - Fixed integer number must match a star number on the star measurement cards

4 Star catalogue number - character data

5 α right ascension
6 δ declination } location of the star from the catalog

Figure C-6.—Catalog card for DR01J.



FIELD(S)

1 Station number
2 Plate number } Character data not checked by the program

3 STG - The sidereal time at Greenwich at 0 hours UT on the day of observation in hours, minutes(Fixed integers), and seconds(three decimal places allowed).

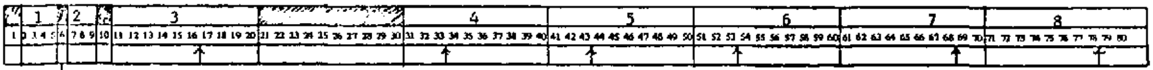
4 West longitude of the camera site in degrees, minutes(fixed integers), and seconds (three decimal places allowed).

5 Latitude of the camera site in degrees, minutes(fixed integers), and seconds (three decimal places allowed). If the angle is negative, each component must have a sign. If any component is zero it may be left blank.

6 Aircraft code - Character data not used by the program.

7 Forward offset
8 Port offset } Offset coordinates of the RF source from the flashing light in feet.

Figure C-7.—Minitrack Start card 1 for DR01J.



FIELD(S)

1 Station number
2Φ Plate number } Character data not used by the program

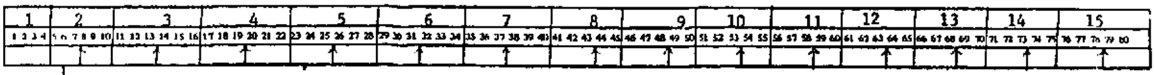
3 Frequency in MHz of the transmission

4 Velocity constant, used in equations (C-25f) and (C-25g)

5 EW
6 NS } Filter constants used in equations (C-25d) and (C-25e)

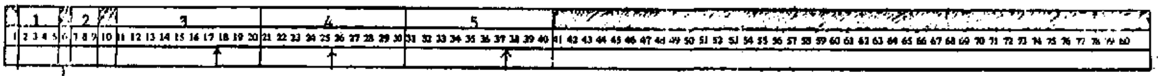
7 Station Height
8 Aircraft Height } In feet, above mean sea level.

Figure C-8.—Minitrack Start card 2 for DR01J.



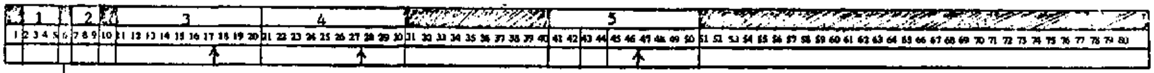
FIELD(S)		
1	Station number - Character data not used by the program	
2	EWF	East-West and North-South baselines for the fine,coarse, and medium antennas in feet
3	NSF	
4	EWC	
5	NSC	
6	EWM	
7	NSM	
.8	X - EWC	Translation coordinates in feet.
9	X - NSC	
10	X - EWM	
11	X - NSM	
12	Y - EWC	
13	Y - NSC	
14	Y - EWM	
15	Y - NSM	

Figure C-9.—Minitrack Start card 3 for DR01J.



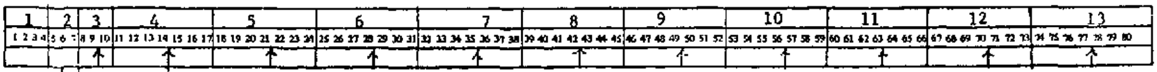
FIELD(S)		
1	Station Number	Character data not used by the program
2	Plate Number	
3	Temperature in degrees Farenheit	
4	Pressure in inches mercury (Hg)	
5	Relative humidity in percent (3 decimal places allowed).	

Figure C-10.—Minitrack Start card 4 for DR01J.



FIELD(S)		
1	Station Number	Character data not used by the program
2	Plate Number	
3	AM	Fine antenna Minitrack measurements.
4	BM	
5	Universal time of the flash in hours, minutes, and seconds.	

Figure C-11.—Minitrack card for DR01J.



FIELD(S)

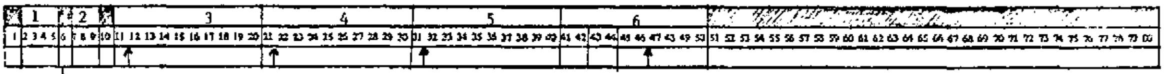
1 Station Number
2 Plate Number
3 UT seconds for time of flash

} Data not used by the program

4 X_1
5 Y_1
6 X_2
7 Y_2
8 X_3
9 Y_3
10 X_4
11 Y_4
12 X_5
13 Y_5

} Coordinate measurements of flash. All measurements are in mm with 3 decimal places allowed. The number of sets of measurements used by the program is in field 7 of the Plate Card.

Figure C-12.—Flashing light Measurement card for DR01J.



1 Station Number
2 Plate Number

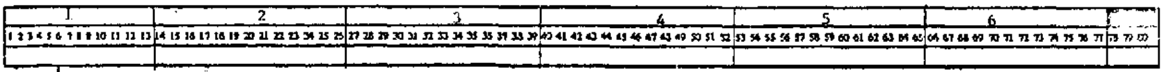
} Data not used by the program

3 λ
4 μ
5 ν

} Direction cosines

6 UT time of flash - Data not used by the program

Figure C-13.—PACT data card for DR01J.

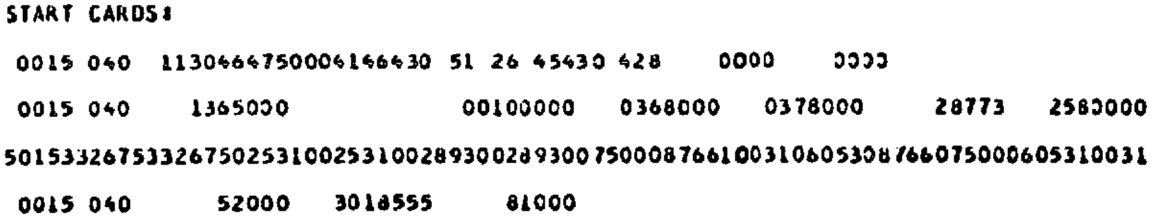


1 a
2 b
3 c
4 d
5 e
6 f

} Plate constants in floating point exponential format with 8 significant digits and an exponent to place the decimal point.

PLATE CONSTANT CARD 2 has exactly the same format with plate constants a', b', c', d', e', and f'.

Figure C-14.—Plate Constant card 1 for DR01J.



```
START CARDS:

 0015 040   1130464750004146630  51  26  45430 428      0000      0000

 0015 040     1365000                  00100000   0368000   0378000     28773    2580000

50153326753326750253100253100289300289300750008766100310605308766075000605310031

 0015 040       52000    3018555      81000
```

Figure C-15.—Sample output listing of Minitrack Start cards for DR01J.

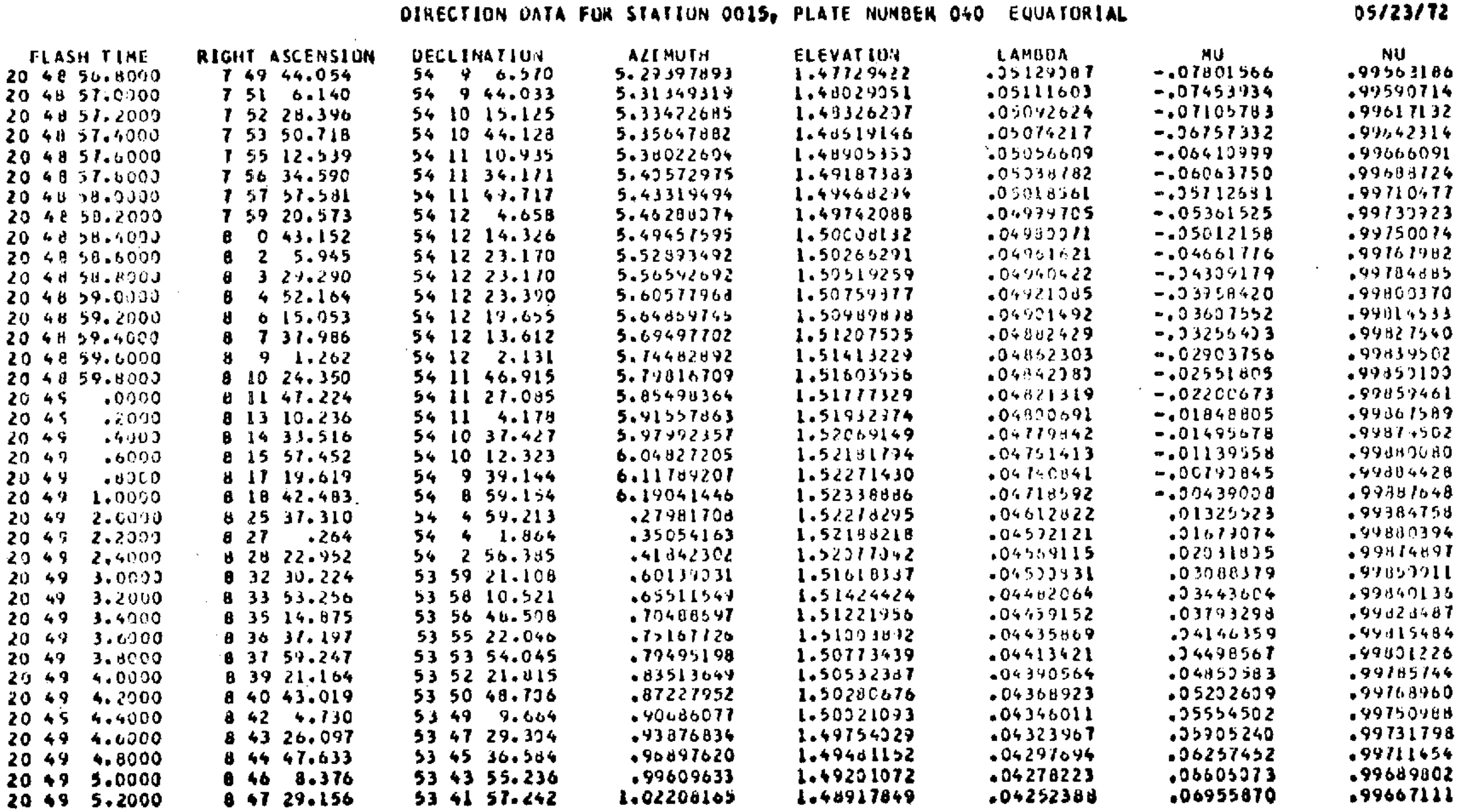
Disk Output

Permanent disk output consists of the fine and ambiguity data files. The fine file records are in blocks of 3400 bytes with 136 bytes per logical record. Each record has the following format:

Bytes	*Data*		
1 to 4	Hours	Time of flash in single-precision floating point	
5 to 8	Minutes		
9 to 12	Seconds		
13 to 16	Null area character data reserved for later use		
17 to 24	A_c	From equation (C-25)	Double-precision floating point
25 to 32	B_c		
33 to 40	A_m	Minitrack card input values	
41 to 48	B_m		
49 to 56	C	From equation (C-13) or (C-14)	
57 to 64	R	Equation (C-17)	
65 to 72	Δa_0	$\Delta a_0/\Delta$UT and $\Delta b_0/\Delta$UT from equations (C-25d) to (C-25g)	
73 to 80	Δb_0		
81 to 88	N_A	As defined in equation (C-19) with fine baselines	
89 to 96	N_B		
97 to 100	Station number	Character	
101 to 103	Plate number		
104	Code for later use		
105 to 136	Four double-precision floating-point numbers for later use		

The ambiguity file has blocks of 1680 bytes with 56 bytes per logical record. Each record has the following data:

Bytes	*Data*	
1 to 4	Hours	Time of flash in single-precision floating point
5 to 8	Minutes	
9 to 12	Seconds	

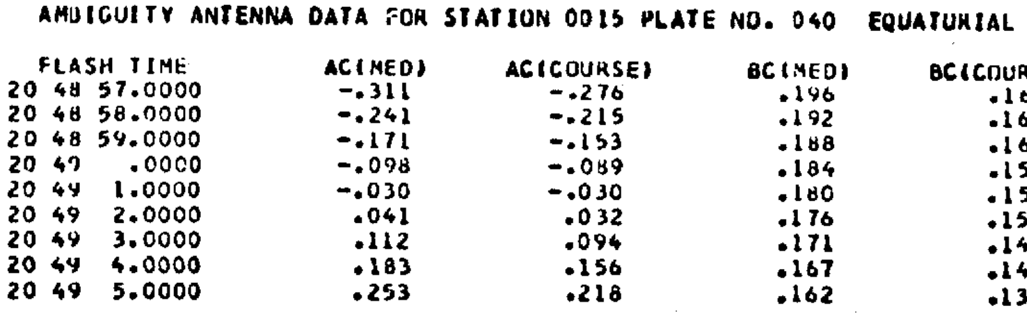


```
                                     DIRECTION DATA FOR STATION 0015, PLATE NUMBER 040  EQUATORIAL                05/23/72

  FLASH TIME      RIGHT ASCENSION    DECLINATION       AZIMUTH       ELEVATION       LAMBDA          MU            NU
20 48 56.8000       7 49 44.054     54  9  6.570     5.29397893     1.47729422     .05129087     -.07801566     .99563186
20 48 57.0000       7 51  6.140     54  9 44.033     5.31349319     1.48029051     .05111603     -.07453934     .99590714
20 48 57.2000       7 52 28.396     54 10 15.125     5.33472685     1.48326207     .05092624     -.07105783     .99617132
20 48 57.4000       7 53 50.718     54 10 44.128     5.35647882     1.48619146     .05074217     -.06757332     .99642314
20 48 57.6000       7 55 12.539     54 11 10.935     5.38022694     1.48905353     .05056609     -.06410999     .99666091
20 48 57.8000       7 56 34.590     54 11 34.171     5.40572975     1.49187383     .05038782     -.06063750     .99688724
20 48 58.0000       7 57 57.581     54 11 49.717     5.43319494     1.49468294     .05018561     -.05712681     .99710477
20 48 58.2000       7 59 20.573     54 12  4.658     5.46280074     1.49742088     .04999705     -.05361525     .99730923
20 48 58.4000       8  0 43.152     54 12 14.326     5.49457595     1.50008132     .04980071     -.05012158     .99750074
20 48 58.6000       8  2  5.945     54 12 23.170     5.52893492     1.50266291     .04961621     -.04661776     .99767982
20 48 58.8000       8  3 29.290     54 12 23.170     5.56592692     1.50519259     .04940422     -.04309179     .99784885
20 48 59.0000       8  4 52.164     54 12 23.390     5.60577968     1.50759377     .04921085     -.03958420     .99800370
20 48 59.2000       8  6 15.053     54 12 19.655     5.64859745     1.50989838     .04901492     -.03607552     .99814533
20 48 59.4000       8  7 37.986     54 12 13.612     5.69497702     1.51207505     .04882429     -.03256403     .99827540
20 48 59.6000       8  9  1.262     54 12  2.131     5.74482892     1.51413229     .04862303     -.02903756     .99839502
20 48 59.8000       8 10 24.350     54 11 46.915     5.79816709     1.51603556     .04842080     -.02551805     .99850100
20 49     .0000     8 11 47.224     54 11 27.085     5.85498364     1.51777329     .04821319     -.02200673     .99859461
20 49     .2000     8 13 10.236     54 11  4.178     5.91557863     1.51932374     .04800691     -.01848805     .99867589
20 49     .4000     8 14 33.516     54 10 37.427     5.97992357     1.52069149     .04779842     -.01495678     .99874502
20 49     .6000     8 15 57.452     54 10 12.323     6.04827205     1.52181794     .04751413     -.01139558     .99880080
20 49     .8000     8 17 19.619     54  9 39.144     6.11789207     1.52271430     .04740841     -.00790845     .99884428
20 49    1.0000     8 18 42.483     54  8 59.154     6.19041446     1.52338886     .04718592     -.00439008     .99887648
20 49    2.0000     8 25 37.310     54  4 59.213      .27981708     1.52278295     .04612822      .01325523     .99884758
20 49    2.2000     8 27   .264     54  4  1.864      .35054163     1.52188218     .04572121      .01679074     .99880394
20 49    2.4000     8 28 22.952     54  2 56.385      .41842302     1.52077042     .04559115      .02031835     .99874897
20 49    3.0000     8 32 30.224     53 59 21.108      .60139031     1.51618337     .04500931      .03088379     .99859911
20 49    3.2000     8 33 53.256     53 58 10.521      .65511549     1.51424424     .04482064      .03443604     .99840136
20 49    3.4000     8 35 14.875     53 56 46.508      .70488697     1.51221956     .04459152      .03793298     .99828487
20 49    3.6000     8 36 37.197     53 55 22.046      .75167726     1.51003892     .04435869      .04146359     .99815484
20 49    3.8000     8 37 59.247     53 53 54.045      .79495198     1.50773439     .04413421      .04498567     .99801226
20 49    4.0000     8 39 21.164     53 52 21.815      .83513649     1.50532387     .04390564      .04850583     .99785744
20 49    4.2000     8 40 43.019     53 50 48.706      .87227952     1.50280676     .04368923      .05202609     .99768960
20 49    4.4000     8 42  4.730     53 49  9.664      .90486077     1.50021093     .04346011      .05554502     .99750988
20 49    4.6000     8 43 26.097     53 47 29.304      .93876834     1.49754029     .04323967      .05905240     .99731798
20 49    4.8000     8 44 47.633     53 45 36.584      .96897620     1.49481152     .04297694      .06257452     .99711654
20 49    5.0000     8 46  8.376     53 43 55.236      .99609633     1.49201072     .04278223      .06605073     .99689802
20 49    5.2000     8 47 29.156     53 41 57.242     1.02208165     1.48917849     .04252388      .06955870     .99667111
```

Figure C-16.—Sample output listing of direction data for DR01J.

```
                                 FINE ANTENNA DATA FOR STATION C015 PLATE NO.041  EQUATORIAL                                05/23/72

    FLASH TIME        AC        BC     AM     BM     OA/DT   OB/DT       NA         NB        COURSE       RANGE       FEW     FNS     VEW     VNS
 20 53 42.0000       3.582     .484   .827   .594   -.814    .027   46.168825 46.168825   -1.53818     25589.82    -.030    .001    .007   -.000
 20 53 42.2000       3.420     .489   .666   .600   -.814    .027   46.168825 46.168825   -1.53818     25583.01    -.030    .001    .005   -.000
 20 53 42.4000       3.258     .495   .508   .605   -.816    .027   46.168825 46.168825   -1.53827     25576.52    -.030    .001    .004   -.000
 20 53 43.0000       2.772     .511   .023   .620   -.817    .028   46.168825 46.168825   -1.53767     25558.98    -.030    .001    .000   -.000
 20 53 43.2000       2.602     .516   .850   .625   -.816    .022   46.168825 46.168825   -1.54492     25553.78    -.030    .001    .007   -.000
 20 53 44.0000       1.954     .535   .209   .644   -.817    .024   46.168825 46.168825   -1.54192     25536.21    -.030    .001    .002   -.000
 20 53 44.2000       1.791     .541   .043   .649   -.822    .027   46.168825 46.168825   -1.53790     25532.61    -.030    .001    .000   -.000
 20 53 44.4000       1.621     .546   .869   .653   -.817    .025   46.168825 46.168825   -1.54091     25529.35    -.030    .001    .007   -.000
 20 53 44.6000       1.458     .550   .704   .658   -.821    .021   46.168825 46.168825   -1.54610     25526.39    -.030    .001    .006   -.000
 20 53 44.8000       1.295     .555   .544   .662   -.822    .024   46.168825 46.168825   -1.54151     25523.76    -.030    .001    .004   -.000
 20 53 45.0000       1.132     .559   .386   .667   -.821    .022   46.168825 46.168825   -1.54405     25521.45    -.030    .001    .003   -.000
 20 53 45.2000        .969     .564   .225   .671   -.819    .023   46.168825 46.168825   -1.54276     25519.48    -.030    .001    .002   -.000
 20 53 45.4000        .806     .568   .060   .577   -.821    .019   46.168825 46.168825   -1.54750     25517.82    -.030    .001    .000   -.000
 20 53 45.6000        .636     .572   .886   .680   -.819    .025   46.168825 46.168825   -1.54098     25516.49    -.030    .001    .007   -.000
 20 53 45.8000        .473     .577   .721   .685   -.820    .021   46.168825 46.168825   -1.54545     25515.47    -.030    .001    .006   -.000
 20 53 46.0000        .310     .581   .559   .689   -.821    .022   46.168825 46.168825   -1.54419     25514.78    -.030    .001    .005   -.000
 20 53 46.2000        .148     .586   .401   .694   -.819    .023   46.168825 46.168825   -1.54290     25514.42    -.030    .001    .003   -.000
 20 53 46.4000       -.015     .589   .241   .698   -.819    .018   46.168825 46.168825   -1.54847     25514.36    -.030    .001    .002   -.000
 20 53 46.6000       -.177     .595   .077   .703   -.821    .026   46.168825 46.168825   -1.53851     25514.65    -.030    .001    .001   -.000
 20 53 46.8000       -.348     .599   .902   .707   -.820    .021   46.168825 46.168825   -1.54497     25515.25    -.030    .001    .007   -.000
 20 53 47.0000       -.511     .603   .737   .712   -.821    .022   46.168825 46.168825   -1.54434     25516.17    -.030    .001    .006   -.000
 20 53 47.2000       -.673     .608   .575   .716   -.817    .022   46.168825 46.168825   -1.54351     25517.41    -.030    .001    .005   -.000
 20 53 47.4000       -.836     .612   .416   .721   -.821    .023   46.168825 46.168825   -1.54292     25518.97    -.030    .001    .003   -.000
 20 53 47.6000       -.999     .617   .256   .725   -.822    .023   46.168825 46.168825   -1.54286     25520.87    -.030    .001    .002   -.000
 20 53 47.8000      -1.161     .622   .093   .730   -.817    .025   46.168825 46.168825   -1.54041     25523.08    -.030    .001    .001   -.000
 20 53 48.0000      -1.332     .626   .919   .734   -.819    .022   46.168825 46.168825   -1.54309     25525.61    -.030    .001    .008   -.000
 20 53 48.2000      -1.494     .631   .755   .739   -.820    .023   46.168825 46.168825   -1.54233     25528.47    -.030    .001    .006   -.000
 20 53 48.4000      -1.656     .635   .592   .743   -.817    .023   46.168825 46.168825   -1.54185     25531.64    -.030    .001    .005   -.000
 20 53 48.6000      -1.819     .640   .433   .748   -.819    .023   46.168825 46.168825   -1.54167     25535.14    -.030    .001    .004   -.000
 20 53 48.8000      -1.981     .645   .273   .752   -.819    .025   46.168825 46.168825   -1.54034     25538.96    -.030    .001    .002   -.000
 20 53 49.0000      -2.143     .650   .110   .757   -.813    .023   46.168825 46.168825   -1.54214     25543.08    -.030    .001    .001   -.000
 20 53 49.2000      -2.313     .654   .939   .761   -.821    .022   46.168825 46.168825   -1.54321     25547.56    -.030    .001    .008   -.000
 20 53 49.4000      -2.475     .659   .775   .766   -.814    .025   46.168825 46.168825   -1.53913     25552.33    -.030    .001    .006   -.000
 20 53 49.6000      -2.637     .663   .612   .770   -.816    .021   46.168825 46.168825   -1.54433     25557.43    -.030    .001    .005   -.000
 20 53 49.8000      -2.799     .668   .453   .774   -.816    .022   46.168825 46.168825   -1.54258     25562.86    -.030    .001    .004   -.000
 20 53 50.0000      -2.960     .672   .293   .778   -.814    .024   46.168825 46.168825   -1.54032     25568.60    -.030    .001    .002   -.000
 20 53 50.2000      -3.121     .676   .131   .783   -.811    .018   46.168825 46.168825   -1.54713     25574.63    -.030    .001    .001   -.000
```

Figure C-17.–Sample output listing of fine-antenna data for DR01J.



```
      AMBIGUITY ANTENNA DATA FOR STATION 0015 PLATE NO. 040  EQUATURIAL

   FLASH TIME             AC(MED)       AC(COURSE)        BC(MED)        BC(COURSE)
20 48 57.0000              -.311           -.276            .196             .168
20 48 58.0000              -.241           -.215            .192             .164
20 48 59.0000              -.171           -.153            .188             .161
20 49    .0000             -.098           -.089            .184             .157
20 49  1.0000              -.030           -.030            .180             .154
20 49  2.0000               .041            .032            .176             .150
20 49  3.0000               .112            .094            .171             .146
20 49  4.0000               .183            .156            .167             .142
20 49  5.0000               .253            .218            .162             .138
```

Figure C-18.—Sample output listing of ambiguity-antenna data for DR01J.

Bytes	*Data*	
13 to 16	Character data to pad for alignment	
17 to 24	A_c medium	Equation (C-25) as modified for ambiguity data—double-precision floating point
25 to 32	A_c coarse	
33 to 40	B_c medium	
41 to 48	B_c coarse	
49 to 52	Station number	Character
53 to 55	Plate number	
56	Code for later use	

Disk files are also used for temporary storage of fine and ambiguity data already in the file and direction data.

Problem Statement

The PROOFREAD program has three main purposes. First, PROOFREAD assigns the proper integral portions to the input Minitrack readings, which contain only the decimal portions. Second, the decimal portions of the Minitrack readings are corrected by the calibration constant correction value KS1. Finally, PROOFREAD determines the average values of the calibration constants for all antenna systems and the simple residuals associated with these constants. It is to be run in conjunction with and after program DR01J.

Mathematical Procedure

The symbol A identifies the east-west direction; B identifies the north-south direction. The camera data are denoted by AC and BC, and the decimal part of the readings from the Minitrack system are denoted by AMQ and BMQ. On the DR01J output listings, AMQ and BMQ are labeled AM and BM. The calibration constants are KCA and KCB. The calibration constant correction factors are denoted by KS1A and KS1B. Then for any point *i*,

$$KCA_i = \text{Decimal portion of } (100 + AMQ_i - AC_i - KS1A) + IA_i$$

$$KCB_i = \text{Decimal portion of } (100 + BMQ_i - BC_i - KS1B) + IB_i$$

where IA_i and IB_i are integral numbers and are always equal either to 49 to 50. They are determined in the following manner for any particular antenna system. For the first data point in a set,

$$IA1 = \begin{cases} 49 \text{ if decimal portion of } (100 + AMQ_1 - AC_1 - KS1A) > 0.850 \\ 50 \text{ if decimal portion of } (100 + AMQ_1 - AC_1 - KS1A) \leqslant 0.850 \end{cases}$$

and similarly for IB_1. For any data point i in a set after the first point,

$$IA_i = \begin{cases} 50 \text{ if } KCA_1 \geqslant 50.100 \\ 50 \text{ if } KCA_1 < 50.100 \text{ and decimal portion of } (100 + AMQ_i - AC_i - KS1A) < 0.500 \\ 49 \text{ if } KCA_1 < 50.100 \text{ and decimal portion of } (100 + AMQ_i - AC_i - KS1A) \geqslant 0.500 \end{cases}$$

and similarly for IB_i.

Final Minitrack values AM_i and BM_i are given by

$$AM_i = IA_i + AC_i + \text{decimal portion of } (100 + AMQ_i - AC_i - KS1A)$$

$$BM_i = IB_i + BC_i + \text{decimal portion of } (100 + BMQ_i - BC_i - KS1B)$$

Final calibration constants KCA and KCB are simply the averages of the individually computed KCA_i and KCB_i values; that is,

$$KCA = \frac{\sum KCA_i}{n}$$

$$KCB = \frac{\sum KCB_i}{n}$$

where n is the total number of data points.

Simple residuals RA_i and RB_i are computed for every point:

$$RA_i = AC_i + KCA - AM_i$$

$$RB_i = BC_i + KCB - BM_i$$

All of the above equations apply to fine-antenna data. The same equations are used for ambiguity-antenna data except that medium antennas and the coarse antennas have their own unique constant correction factors to be used in place of KS1A and KS1B. Also, ambiguity data are processed only at integral seconds, and because of the way the ambiguity Minitrack data are recorded, the data must be interpolated.

AM medium is recorded for the integral second, AM coarse at that time + 0.2 s, BM medium for that time + 0.4 s, and BM coarse for that time + 0.6 s. Linear interpolation is used.

Program Description

Fine-Antenna Data

This program can be preceded by a sort routine for ambiguity data to merge, arrange, and interfile two files (polar and equatorial) in ascending order by plate number. The merged files of ambiguity data are left on a work disk, and the data on the Minitrack disk are not changed.

The program DR01K reads first a Lead card, to determine whether fine data or ambiguity data are to be processed. The Lead card is followed by a Start card and up to 50 Correction cards and *n* fine KS1 cards, where *n* is the number of plates in the calibration. The quantities on the fine KS1 cards are read into an array, from which they are matched with the proper plate on the disk. The camera and fractional Minitrack values for fine data are both found on the Minitrack disk, written there as output by program DR01J. The revised Minitrack values and calibration constants are computed from the data on the Minitrack disk and KS1 values from the fine KS1 cards and are stored in a temporary file on the work disk. After all the data points have been read, the calibration constants are averaged. The residuals are computed, and the data are written back onto the Minitrack disk in the same place from which they came. The revised Minitrack values and the residuals assign values to variables that were empty on input. None of the input data are changed. However, there will be an additional record at the beginning of the file that contains Start card information.

Ambiguity-Antenna Data

The ambiguity data are computed separately from the fine data. The input data consist of camera values on the Minitrack disk and Minitrack values on cards. The Minitrack cards for each plate are preceded by a KS1 card. This is read and followed by a series of ambiguity Minitrack cards and disk records. All data points without both card and disk data are ignored. The Minitrack data must be interpolated to the camera data; otherwise, the process for computing revised Minitrack values, calibration constants, average calibration constants, and residuals is the same as for fine data. Medium and coarse intermediate results are stored in separate temporary storage areas. Two listings of each are printed, and there is no disk output.

All listing, fine, medium, and coarse, is done using the PRINT listing procedure.

The program must interpolate the Minitrack ambiguity data because it is not all recorded for times at integral seconds. For example, the east-west coarse Minitrack value is recorded for 0.2 s plus the integral second. The value that the Minitrack would have been had it been found at the integral second instead of 0.2 s later must be computed. The present Minitrack value and the previous Minitrack value are linearly interpolated to find the value in between. Thus the first ambiguity data point of every plate will be lost because there is no previous value to use.

When there is a jump of more than 1 s between data points, the program assumes that the change in Minitrack value is evenly distributed over the time gap.

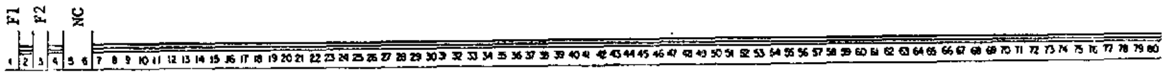
Corrections

A correction option is provided for the fine-antenna data in this program. It is to be used in case a few data points on the disk have incorrect Minitrack values that can be corrected or there are points with incorrect camera data that can be deleted. The Minitrack errors must be small enough in magnitude that the errors were not propagated into the camera data, and the number of points to be deleted must be small enough that they can be omitted without lowering the quality of the output.

In making a correction, either AM or BM, or both, can be corrected. A point may be deleted regardless of whether the error is in the camera data or the Minitrack. Up to 50 points can be corrected or deleted.

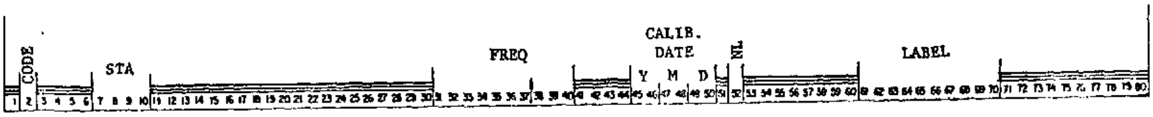
Input Description

This program uses as input the Minitrack disk created by the program employing the BLCAM procedure (DR01J). Its card input is seen in figures C-19 to C-24. Carets are used to denote implied decimal points.



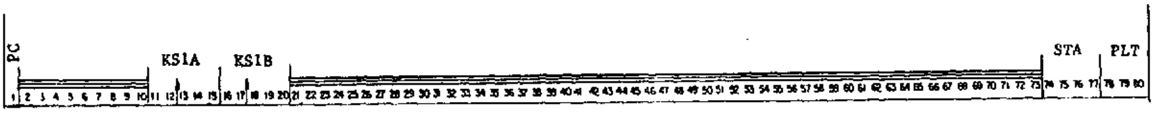
LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
F1	FINE-FLAG	1	Flag = 1 if fine antenna data are to be processed, = 0 if fine data are not to be processed.
F2	AMB-FLAG	3	Flag = 1 if ambiguity data are to be processed, = 0 if ambiguity data are not to be processed.
NC	CORECT	5-6	Total number of data points to be corrected or rejected by means of Correction Cards. This number must be equal to the number of Correction Cards which are included in the input deck.

Figure C-19.—Lead card for DR01K.



LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
CODE	CODE	2	Code = 1 if calibration is polar = 0 if calibration is equatorial
STA	SSTA	7-10	Station number
FREQ	FREQ	31-40	Frequency in megahertz
Y	YR	45-46	Year of calibration date
M	MON	47-48	Month of calibration date
D	DY	49-50	Day of calibration date
NL	NO	52	Number of listings desired
LABEL	EOP	61-70	A label which is printed as part of the title of the output listing. This can be any 10 character title.

Figure C-20.—Start card for DR01K.



LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
PC	PC	1	Plot Code = 1 for East-west run = 2 for North-south run
KS1A	KSA	11-15	Calibration correction constant for east-west fine array, in counts
KS1B	KSB	16-20	Calibration correction constant for north-south fine array, in counts
STA	ST	74-77	Station number
PLT	PLT	78-80	Plate number

Note: If PC = 1, all data points whose AC (Camera east-west) exceeds 5.000 in absolute value will be omitted from the output. To be used only when data will be plotted.

If PC = 2, all data points whose BC (Camera north-south) exceeds 5.000 in absolute value will be omitted from the output. To be used only when data will be plotted.

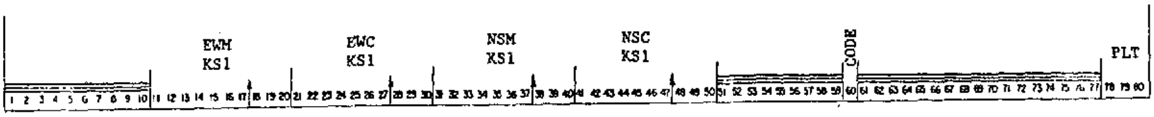
If PC = blank, all camera data (AC or BC) which exceeds |5.000| will be included in output. Used when data is not to be plotted.

Figure C-21.—Fine KS1 card for DR01K.

Disk Input

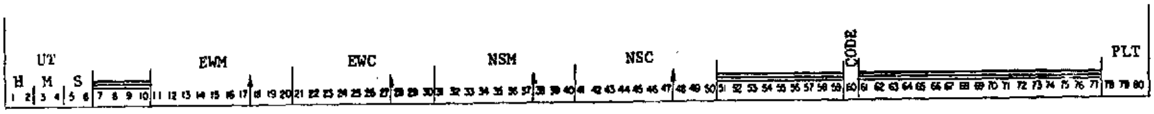
Camera data and decimal Minitrack values for the fine antennas are read from a Minitrack disk (table C-1). Polar and equatorial data are in separate files and must be processed separately. Camera data for all ambiguity antennas are read from a different area on the same disk (table C-2).

The polar and equatorial files may be sorted together and processed together.



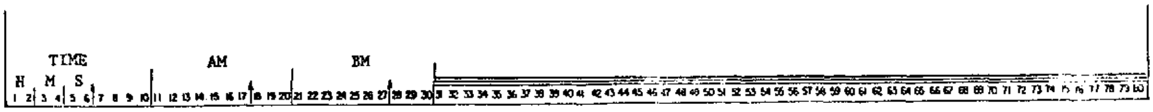
LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
EWM KSI	KSAM	11-20	Calibration correction constant for east-west medium array, in counts.
EWC KSI	KSAC	21-30	Calibration correction constant for east-west coarse array, in counts.
NSM KSI	KSBM	31-40	Calibration correction constant for north-south medium array, in counts.
NSC KSI	KSBC	41-50	Calibration correction constant for north-south coarse array, in counts.
CDE	CDE	60	This must be set equal to 1 in order to get the KS1's subtracted out.
PLT	CPLATE	78-80	Plate identification number.

Figure C-22.–Ambiguity KS1 card for DR01K.



LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
UT	AMB-UT	1-6	Time of data point in hrs, min, sec.
EWM	M-QAM	11-20	Minitrack value as measured by EWM array.
EWC	QAM-C	21-30	Minitrack value as measured by EWC array.
NSM	QBM-M	31-40	Minitrack value as measured by NSM array.
NSC	QBM-C	41-50	Minitrack value as measured by NSC array.
CODE		60	Must be blank.
PLT	CPLATE	78-80	Plate number identification.

Figure C-23.–Ambiguity Minitrack data card for DR01K.



LAYOUT NAME	PROGRAM NAME	COLUMNS	COMMENTS
H	HCD	1-2	Hours } Time of data point to
M	MCD	3-4	Minutes } be corrected
S	SCD	5-10	Seconds } or deleted
AM	CORAM	11-20	If the decimal Minitrack east-west value is to be corrected, the correct value is put here. Otherwise field is blank.
BM	CORBM	21-30	If the decimal Minitrack north-south value is to be corrected, the correct value is put here. Otherwise field is blank.

Note: If the data point is to be deleted, both AM and BM fields are left blank and only the time of the point is put on the card.

Figure C-24.—Correction card for DR01K.

Table C-1.—Fine Disk Input Record (136 Bytes)

Program name	Bytes	Comments
HR	1 to 4	Hours } Data point time in UT
MN	5 to 8	Minutes }
SEC	9 to 12	Seconds }
AC	17 to 24	Camera east-west data
BC	25 to 32	Camera north-south data
AMQ	33 to 40	Minitrack east-west decimal portion
BMQ	41 to 48	Minitrack north-south decimal portion
STA	97 to 100	Station number
F_PLATE	101 to 103	Plate number
CODE[a]	104	PC from FINE KSI card
RA[a]	105 to 112	Fine residual, east-west
RB[a]	113 to 120	Fine residual, north-south
AM[a]	121 to 128	Minitrack value, east-west
BM[a]	129 to 136	Minitrack value, north-south

[a] On the input record these fields are filled with dummy data. The quantities described refer to the disk output record.

Card Input Deck

The input decks for the various options, including system and program control cards and cards needed for sorting, are given in figures C-25 to C-27.

Output Description

Tabular and disk output are generated.

Table C-2.—Ambiguity Disk Input Record

Program name	Bytes	Comments
AHR	1 to 4	Hours
AMN	5 to 8	Minutes } Data point time in UT
ASC	9 to 12	Seconds
AC_M	17 to 24	Camera east-west medium data
AC_C	25 to 32	Camera east-west coarse data
BC_M	33 to 40	Camera north-south medium data
BC_C	41 to 48	Camera north-south coarse data
ASTA	49 to 52	Station number
PLATE	53 to 55	Plate number
ACODE	56	Print code

```
    //  PAUSE MOUNT DRZ001 ON 133
    //  ASSGN SYS018,X'133'
    //  ASSGN SYS019,X'132'
**  //  DLBL FINE,'FINE DATA   GUS TEST                          136.500',68/001
    //  EXTENT SYS018,DRZ001,,,2700,25
**  //  DLBL FINEIN,'FINE DATA   GUS TEST                         136.500'
    //  EXTENT SYS018,DRZ001
    //  DLBL TFINE,'FINE ANTENNA TEMPORARY AREA',68/001
    //  EXTENT SYS019,111111,,,20,140
    //  DLBL TFINEI,'FINE ANTENNA TEMPORARY AREA'
    //  EXTENT SYS019,111111
       ASSGN SYSCLB,X'135'
    //  EXEC DR01K


            Lead Card

            Start Card

            0-50 Correction Cards

            Fine KS1 Cards - one for each plate in the calibration

    /*
    /*
    /&
```

** Both cards must contain exactly the same punches between the quotation marks and they must exactly match what was used on the corresponding cards in Program DR01J.

Note: The correction cards must be in the order in which the points addressed are written on the disk.

Figure C-25.—Fine-antenna data control cards and input deck for DR01K.

```
    //  ASSGN SYS001,X'132'       OUTPUT FILE - COMBINED AMBIGUITY
    //  ASSGN SYS002,X'133'       POLAR AMB
    //  ASSGN SYS003,X'133'       EQUATORIAL AMB
    //  ASSGN SYS004,X'131'       WORK FILE
**  //  DLBL SORTIN1,'AMBIGUITY   WINK EQ                     137.500'
 $  //  EXTENT SYS002, DRZ001,,,2975,5
**  //  DLBL SORTIN2,'AMBIGUITY   WINK   POL                  137.000'
 $  //  EXTENT SYS003,DRZ001,,,2925,5
    //  DLBL SORTOUT,'EQU AND POL AMBIGUITY DATA',69/001
    //  EXTENT SYS001,111111,,,500,40
    //  DLBL SORWK1,,69/001
    //  EXTENT SYS004,111111,,,20,40
    //  EXEC SORT

$$  70001     SORT FIELDS=(53,3,CH,A,1,4,FL,A,5,4,FL,A,9,4,FL,A),FILES=2,WORK=1
    70001  RECORD TYPE=F,LENGTH-(56,,56)
    70001  INPFIL BLKSIZE=(1680)
    70001  OUTFIL BLKSIZE=(1680)
    70001   OPTION PRINT=ALL
    70001  END
    //  ASSGN SYS018,X'132'
    //  ASSGN SYS019,X'132'
    //  DLBL AMBIN,'EQU AND POL AMBIGUITY DATA'
    //  EXTENT SYS018,111111
    //  DLBL M_OUT,'MEDIUM TEMPORARY AREA',68/001
    //  EXTENT SYS019,111111,,,540,40
    //  DLBL M_IN,'MEDIUM TEMPORARY AREA'
    //  EXTENT SYS019,111111
    //  DLBL C_OUT,'COARSE TEMPORARY AREA',68/001
    //  EXTENT SYS019,111111,,,580,40
    //  DLBL C_IN,'COARSE TEMPORARY AREA'
    //  EXTENT SYS019,111111
       ASSGN SYSCLB,X'135'
    //  EXEC DR01K

        Lead Card

        Start Card

       {Ambiguity KS1 Card for Plate i  }  repeated for
       {Minitrack data Cards for Plate i}  i=1 to i=n


        /*

        /*

        /&
```

** Both cards must contain exactly the same punches between the quotation marks and they must exactly match what was used on the corresponding cards in Program DR01J.

$ The extent must match what was used on the corresponding card in Program DR01J.

$$ The number of files in this example is two. If only one is to be processed put FILES=1 on this card.

Figure C-26.—Ambiguity-antenna data control cards and input deck for DR01K, sorted.

```
//  ASSGN SYS018,X'132'
//  ASSGN SYS019,X'132'
    The 4 title cards from Program DR01J
//  DLBL M_OUT,'MEDIUM TEMPORARY AREA',68/001
//  EXTENT SYS019,111111,,,540,40
//  DLBL M_IN,'MEDIUM TEMPORARY AREA'
//  EXTENT SYS019,111111
//  DLBL C_OUT,'COARSE TEMPORARY AREA',68/001
//  EXTENT SYS019,111111,,,580,40
//  DLBL C_IN,'COARSE TEMPORARY AREA'
//  EXTENT SYS019,111111
   ASSGN SYSCLB,X'135'
//  EXEC DR01K

     Lead Card

     Start Card

    {Ambiguity KS1 Card for Plate i  }  repeated for
    {Minitrack data Cards for Plate i}  i=1 to i=n

     /*

     /*

     /&
```

Figure C-27.–Ambiguity-antenna data control cards and input deck for DR01K, unsorted.

Tabular Output

The number of listings of fine data to be generated is specified on the Start card (column 52). Two listings of medium and coarse data are always generated for ambiguity data. A sample output is shown in figures C-28 to C-30.

Disk Output

There is disk output for fine data only. It is written on the same area of the Minitrack disk from which the input data were read.

The first record of fine disk output will contain the Start card information, which will be assigned to the input record variables in the following manner:

AC = FINE_KCA
BC = FINE_KCB
AM = FREQ (a "get string" must be performed)
HR = YR
MN = MON
DEC = DY

CODE will be set equal to 9 to indicate a Start card record.

The output data record is the same as the input record, except that the input dummy data in bytes 104 to 136 are replaced by output from this program. The disk output is used by program DR01E.

Accuracy and Limitations

Camera and Minitrack values, residuals, and average calibration constants are rounded to the third decimal place by adding 0.005 times the sign (-1 or +1) of the variable. The residuals are multiplied by 1000 before being listed.

```
      STATION        KCA         KCB          FREQ        CALIB DATE

       0011        50.097      49.864        136515            720216                TEST P

        UT            AC          BC            AM            BM              RA            RB  P STA PLATE

   19 27  6.200     -2.076       4.628        48.018        54.495            3             -3  1 0011 001
   19 27  6.400     -2.065       4.426        48.029        54.290            3              0
   19 27  6.600     -2.054       4.234        48.042        54.097            1              1
   19 27  6.800     -2.042       4.042        48.054        53.907            1             -1
   19 27  7.000     -2.031       3.849        48.065        53.715            1             -2
   19 27  7.200     -2.020       3.657        48.075        53.521            2              0
   19 27  7.400     -2.010       3.454        48.083        53.315            4              3
   19 27  7.600     -2.000       3.261        48.093        53.122            4              3
   19 27  7.800     -1.990       3.068        48.104        52.932            3              0
   19 27  8.000     -1.980       2.875        48.116        52.741            1             -2
   19 27  8.200     -1.970       2.681        48.127        52.547           -0             -1
   19 27  8.400     -1.961       2.478        48.135        52.340            1              3
   19 27  8.600     -1.952       2.284        48.141        52.148            4              1
   19 27  8.800     -1.943       2.091        48.150        51.957            4             -2
   19 27  9.000     -1.935       1.897        48.160        51.766            2             -5
   19 27  9.200     -1.926       1.703        48.171        51.572            0             -5
   19 27  9.400     -1.917       1.500        48.180        51.365            0             -1
   19 27  9.600     -1.909       1.306        48.188        51.172            0             -2
   19 27  9.800     -1.901       1.112        48.194        50.980            2             -4
   19 27 10.000     -1.893        .918        48.201        50.787            3             -5
   19 27 10.200     -1.886        .724        48.209        50.592            2             -4
   19 27 11.000     -1.854       -.062        48.244        49.803           -1             -0
   19 27 11.200     -1.845       -.256        48.250        49.609            2             -1
   19 27 11.400     -1.838       -.450        48.256        49.412            3              2
   19 27 11.600     -1.829       -.654        48.265        49.207            3              3
   19 27 11.800     -1.821       -.847        48.275        49.014            1              3
   19 27 12.000     -1.813      -1.041        48.285        48.824           -1             -1
   19 27 12.200     -1.805      -1.235        48.294        48.632           -2             -3
   19 27 13.000     -1.773      -2.018        48.322        47.850            2             -4
   19 27 13.200     -1.764      -2.211        48.332        47.657            1             -4
   19 27 13.400     -1.756      -2.404        48.342        47.462           -1             -2
   19 27 13.600     -1.747      -2.607        48.352        47.255           -2              3
```

Figure C-28.—Sample tabular output of fine-antenna data for DR01K.

```
     STATION        KCA         KCB         FREQ        CALIB DATE          MEDIUM

      0011        50.423      50.580       136514          710302

        UT             AC          BC          AM          BM            RA           RB   P STA   PLATE

 20  0 32.000        -.079       .437      50.340      51.030            4          -14     0011 002
 20  0 33.000        -.078       .385      50.340      50.980            5          -15
 20  0 34.000        -.076       .334      50.340      50.924            6          -11
 20  0 35.000        -.074       .282      50.350      50.870           -2           -8
 20  0 36.000        -.072       .230      50.350      50.820            1          -10
 20  0 37.000        -.069       .178      50.350      50.764            4           -6
 20  0 38.000        -.065       .126      50.360      50.710           -2           -4
 20  0 39.000        -.061       .074      50.360      50.660            2           -7
 20  0 40.000        -.057       .022      50.370      50.610           -4           -9
 20  0 41.000        -.053      -.030      50.370      50.560           -0          -11
 20  0 42.000        -.049      -.082      50.370      50.510            4          -13
 20  0 43.000        -.044      -.134      50.380      50.460           -2          -15
 20  0 44.000        -.040      -.186      50.380      50.410            3          -17

 20  6   1.000        .292       .091      50.720      50.656           -5           14     0011 003
 20  6   2.000        .279       .153      50.700      50.722            1           11
 20  6   3.000        .265       .216      50.690      50.786           -2            9
 20  6   4.000        .251       .278      50.680      50.846           -6           12
 20  6   5.000        .237       .340      50.660      50.912           -0            8
 20  6   6.000        .223       .402      50.650      50.976           -5            6
 20  6   7.000        .208       .464      50.630      51.036            1            8

 20 12   1.000      -2.504       .417      47.920      50.996           -1            1     0011 004
 20 12   2.000      -2.502       .377      47.920      50.956            1            0
 20 12   3.000      -2.500       .336      47.920      50.916            3           -0
 20 12   4.000      -2.498       .295      47.930      50.876           -5           -1
 20 12   5.000      -2.496       .254      47.930      50.830           -3            4
 20 12   6.000      -2.494       .212      47.930      50.786           -1            6
 20 12   7.000      -2.491       .171      47.930      50.746            1            4
 20 12   8.000      -2.489       .128      47.940      50.706           -6            2
 20 12   9.000      -2.486       .086      47.940      50.666           -3           -0
 20 12  10.000      -2.483       .044      47.940      50.626           -0           -3
```

Figure C-29.—Sample tabular output of medium-antenna data for DR01K.

```
      STATION        KCA          KCB          FREQ        CALIB DATE          COARSE

       0011        50.193       49.850        136514          710302

         UT              AC          BC          AM          BM          RA          RB  P STA   PLATE

   20   0 32.000       -.075        .377      50.120      50.230          -2          -4    0011 002
   20   0 33.000       -.074        .332      50.128      50.184          -9          -3
   20   0 34.000       -.072        .286      50.130      50.140         -10          -4
   20   0 35.000       -.071        .241      50.130      50.090          -8           1
   20   0 36.000       -.068        .196      50.130      50.044          -6           1
   20   0 37.000       -.066        .150      50.130      50.000          -3          -0
   20   0 38.000       -.062        .105      50.130      49.954           1           0
   20   0 39.000       -.059        .059      50.138      49.910          -5          -1
   20   0 40.000       -.055        .013      50.140      49.864          -3          -1
   20   0 41.000       -.052       -.032      50.140      49.820           2          -3
   20   0 42.000       -.048       -.078      50.148      49.774          -4          -2
   20   0 43.000       -.044       -.123      50.150      49.730          -2          -4
   20   0 44.000       -.041       -.169      50.150      49.684           3          -3

   20   6  1.000        .250        .074      50.442      49.914           2           9    0011 003
   20   6  2.000        .238        .129      50.432      49.970          -1           8
   20   6  3.000        .226        .183      50.422      50.024          -3           9
   20   6  4.000        .214        .238      50.404      50.080           4           8
   20   6  5.000        .202        .292      50.392      50.134           3           8
   20   6  6.000        .189        .346      50.382      50.190           0           6
   20   6  7.000        .177        .400      50.372      50.244          -3           6

   20  12  1.000      -2.193        .361      48.000      50.218           0          -7    0011 004
   20  12  2.000      -2.191        .326      48.000      50.184           2          -8
   20  12  3.000      -2.189        .290      48.008      50.148          -5          -8
   20  12  4.000      -2.188        .254      48.010      50.114          -5         -10
   20  12  5.000      -2.186        .218      48.010      50.078          -4         -10
   20  12  6.000      -2.184        .182      48.010      50.044          -2         -13
   20  12  7.000      -2.182        .145      48.018      50.004          -8          -9
   20  12  8.000      -2.180        .108      48.020      49.968          -7         -10
   20  12  9.000      -2.177        .071      48.020      49.934          -5         -13
   20  12 10.000      -2.175        .034      48.028      49.894         -10         -11
```

Figure C-30.–Sample tabular output of coarse-antenna data for DR01K.

No more than nine listings of fine data may be printed in one program execution; 7500 fine data points, 900 ambiguity data points, and 30 plates of fine data in one file can be processed.

Problem Statement

Program DR01E determines the coefficients in nine sets of equations relating fine-antenna Minitrack data A_m and B_m to the corresponding values A_c and B_c computed from camera data. The coefficients are determined by the method of least squares, and residuals are computed to serve as indicators of the reliability of the equation sets. An option is provided to generate a tape for the CalComp plotter to display residuals that are not computed from the equations but which are simple differences.

Program Description

The program can be used either for a regular station calibration to compute new coefficients or for a historical comparison; that is, to show what sort of residuals would result from applying coefficients from a previous calibration to the data from the present calibration. One Lead card and a Minitrack data disk generated by the PROOFREAD program (DR01K) are used in either case. For historical runs, the input also includes cards that give the coefficients to be used. The flowchart in figure C-31 describes the program fully. Equation numbers 1 through 9 in the flowchart refer to equations (C-28) to (C-36), respectively.

Following is a list of symbols used in DR01E.

Symbol	*Definition*
$\frac{\Delta a_0}{\Delta \mathrm{UT}}$	Rate of change of A_c
A_c	East-west camera value
A_m	East-west Minitrack value
$\frac{\Delta b_0}{\Delta \mathrm{UT}}$	Rate of change of B_c
B_c	North-south camera value
B_m	North-south Minitrack value
C	Course of calibration aircraft measured clockwise from east
KC_A	East-west calibration constant
KC_B	North-south calibration constant
N'_A	East-west fine antenna baseline, in wavelengths
N'_B	North-south fine antenna baseline, in wavelengths
R_2	Range of calibration aircraft from center of fine-antenna system
UT	Time of data point in Greenwich mean time

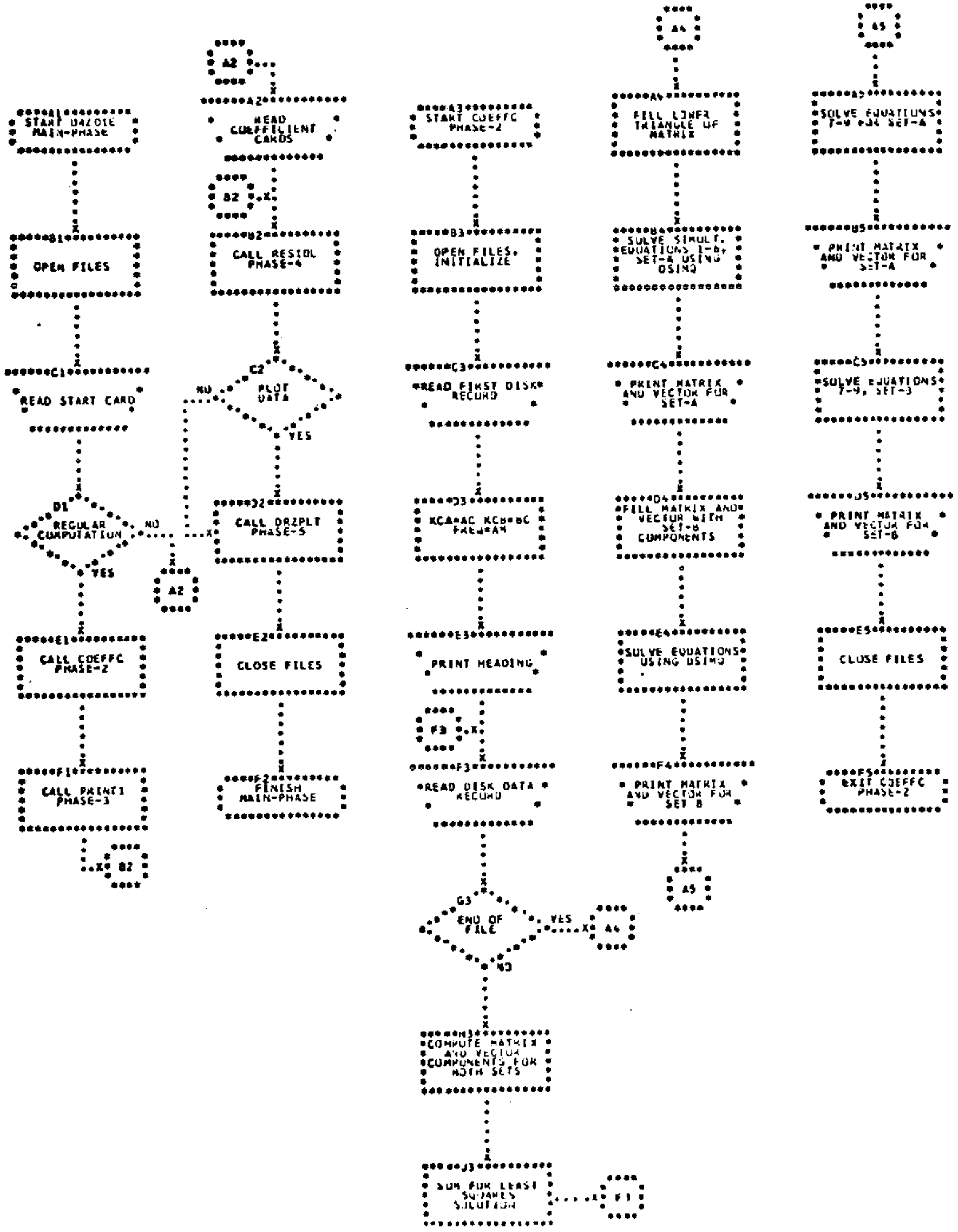


Figure C-31.—Flowchart for DR01E.

C2

C2

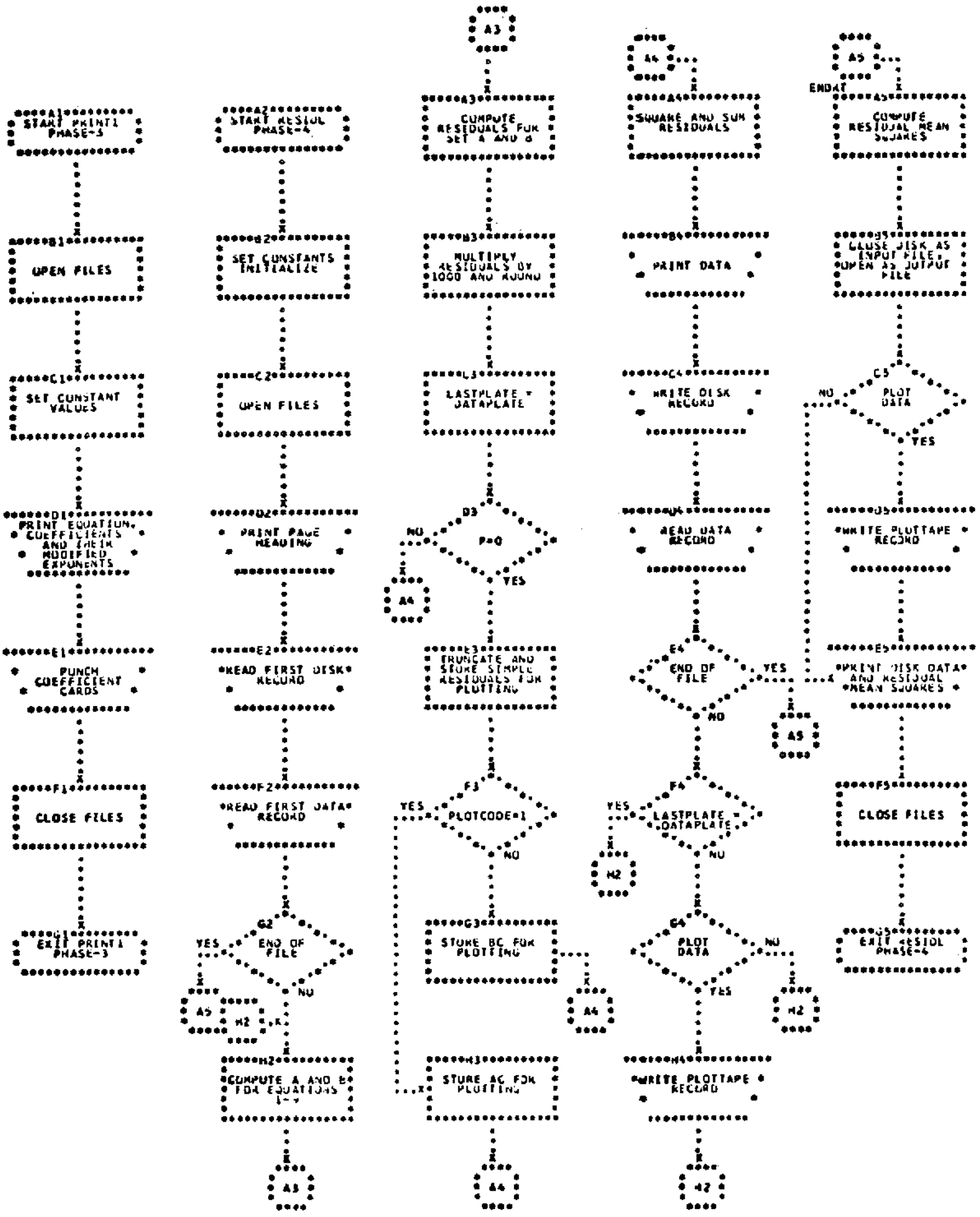


Figure C-31 (continued).—Flowchart for DR01E.

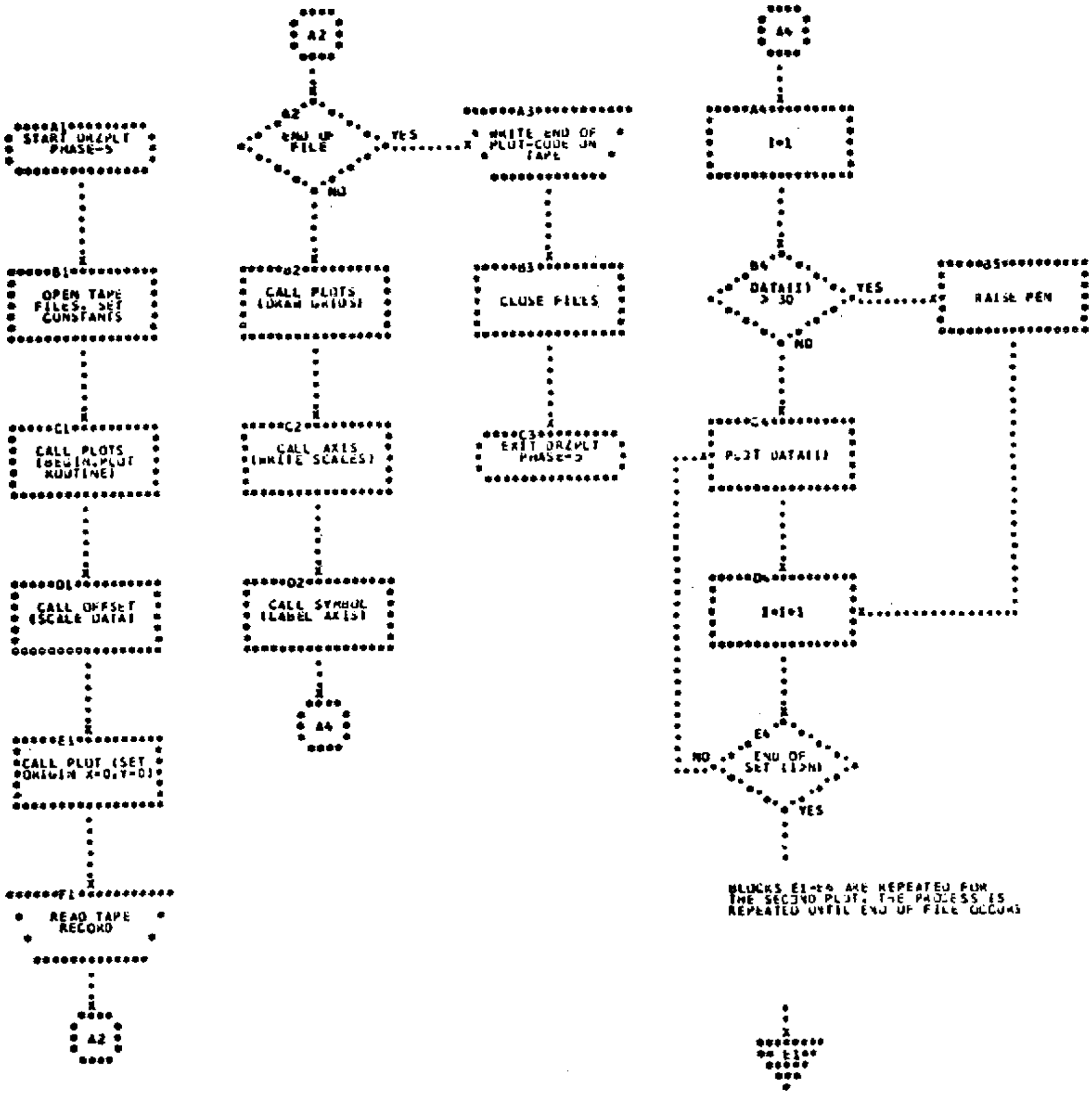


Figure C-31 (concluded).—Flowchart for DR01E.

The variables KC_A and KC_B are given in the first record of the input disk file for a regular calibration or on the Lead card for a historical run. All other items are given in the data records of the ınput disk file.

Computation of Calibration Coefficients

A least-squares solution using all data points is obtained for the following equations:

$$A = C_0 + C_1 X + C_2 Y \tag{C-28a}$$

$$B = C_0' + C_1'X + C_2'Y \tag{C-28b}$$

$$A = C_0 + C_1X + C_2Y + C_3XY + C_4X^2 + C_5Y^2 \tag{C-29a}$$

$$B = C_0' + C_1'X + C_2'Y + C_3'XY + C_4'X^2 + C_5'Y^2 \tag{C-29b}$$

$$A = C_0 + C_1X + C_2Y + C_3XY + C_4X^2 + C_5Y^2 + C_6Y^3 \tag{C-30a}$$

$$B = C_0' + C_1'X + C_2'Y + C_3'XY + C_4'X^2 + C_5'Y^2 + C_6'Y^3 \tag{C-30b}$$

$$A = C_0 + C_1X + C_2Y + C_7 \sin(X) + C_8 \cos(X) \tag{C-31a}$$

$$B = C_0' + C_1'X + C_2'Y + C_7' \sin(Y) + C_8' \cos(Y) \tag{C-31b}$$

$$A = C_0 + C_1X + C_2Y + C_3XY + C_4X^2 + C_5Y^2 + C_7 \sin(X) + C_8 \cos(X) \tag{C-32a}$$

$$B = C_0' + C_1'X + C_2'Y + C_3'XY + C_4'X^2 + C_5'Y^2 + C_7' \sin(Y) + C_8' \cos(Y) \tag{C-32b}$$

$$A = C_0 + C_1X + C_2Y + C_3XY + C_4X^2 + C_5Y^2 + C_6Y^3 + C_7 \sin(X) + C_8 \cos(X) \tag{C-33a}$$

$$B = C_0' + C_1'X + C_2'Y + C_3'XY + C_4'X^2 + C_5'Y^2 + C_6'Y^3 + C_7' \sin(Y) + C_8' \cos(Y) \tag{C-33b}$$

$$A = C_0 + C_9 \left[\frac{EN}{R_2}(-\sin\beta) - \frac{1-E^2}{R_2}\cos\beta\right] + C_{10}\left(\frac{1-E^2}{R_2}\sin\beta\,\frac{EN}{R_2}\cos\beta\right) + C_{11}\frac{\Delta a_0}{\Delta \mathrm{UT}} \tag{C-34a}$$

$$B = C_0' + C_9'\left(\frac{EN}{R_2}\cos\beta + \frac{1-N^2}{R_2}\sin\beta\right) + C_{10}'\left(-\frac{EN}{R}\sin\beta + \frac{1-N^2}{R}\cos\beta\right) + C_{11}'\frac{\Delta b_0}{\Delta \mathrm{UT}} \tag{C-34b}$$

$$A = C_0 + C_9 \left[\frac{EN}{R_2}(-\sin\beta) - \frac{1-E^2}{R_2}\cos\beta\right] + C_{10}\left(\frac{1-E^2}{R_2}\sin\beta - \frac{EN}{R_2}\cos\beta\right) \tag{C-35a}$$

$$B = C_0' + C_9'\left(\frac{EN}{R_2}\cos\beta + \frac{1-N^2}{R_2}\sin\beta\right) + C_{10}'\left(\frac{EN}{R_2}\sin\beta + \frac{1-N^2}{R_2}\cos\beta\right) \tag{C-35b}$$

$$A = C_0 + C_{11}\frac{\Delta a_0}{\Delta \mathrm{UT}} \tag{C-36a}$$

$$B = C_0' + C_{11}'\frac{\Delta b_0}{\Delta \mathrm{UT}} \tag{C-36b}$$

For equations (C-28) to (C-33), set

$$A = A_c$$

$$B = B_c$$

For equations (C-34) to (C-36), set

$$A = A_c + KC_A - A_m$$

$$B = B_c + KC_B - B_m$$

Other terms in the equations are defined as

$$X = A_m - KC_A$$

$$Y = B_m - KC_B$$

$$(X) = 2\pi X$$

$$(Y) = 2\pi Y$$

$$E = \frac{A_c}{N'_A}$$

$$N = \frac{B_c}{N'_B}$$

$$B = \pi/2 - C$$

The above equations are applicable only to an equatorial calibration. For a polar calibration, X^3 must be substituted for Y^3 in equations (C-30) and (C-33).

Residuals

The residuals are computed for each data point. They are defined for equations (C-28) to (C-33) as

$$R_{A_i} = A_{c_i} - A_i$$

$$R_{B_i} = B_{c_i} - B_i$$

where A_{c_i} and B_{c_i} are the camera values for data point i and A_i and B_i are computed from the jth set of equations, using the predetermined coefficients and the Minitrack values for point i.

For equations (C-34) to (C-36), the residuals are computed by the equations

$$R_{A_i} = ra_i - A_i$$

$$R_{B_i} = rb_i - B_i$$

where ra_i and rb_i, the simple residuals, are defined as

$$ra_i = A_{c_i} - KC_A - A_{m_i}$$

$$rb_i = B_{c_i} - KC_B - B_{m_i}$$

The rms of all residuals is computed as

$$\text{rms}_i = \left(\frac{\sum R_i^2}{N}\right)^{1/2}$$

where N = number of data points and the summation is over all points.

Plotting

A CalComp tape is generated to produce line plots for 10 by 15-in. paper, one page per plate with each page containing two graphs; simple residuals *ra* and *rb* versus A_c or B_c are produced as indicated by a plot code (fig. C-32). The plot code may vary for different plates but is constant for single plate with a numerical 1 indicating A_c and a numeral 2 indicating B_c is to be used as the abscissa. The abscissa scale (A_c or B_c) is 1 in. per unit with the scale going from -5 to 5 wavelengths. The ordinate scale (*ra* or *rb*) goes from -30 to +30 counts (1 count = 0.001 wavelength) with 1 in. equal to 10 counts. A data point whose residual exceeds 30 counts in absolute value is omitted and the plot line broken to indicate a missing point. Axes are labeled, and scales printed. Plate number, frequency, and calibration date are also written on each page.

Program Phases

DRZ01E, the main phase, handles the card input and calls the other phases.

The Minitrack disk is input for COEFFC, the second phase, in which the least-squares equations are solved. The matrices and vectors used in the solution of the A and B sets are printed here.

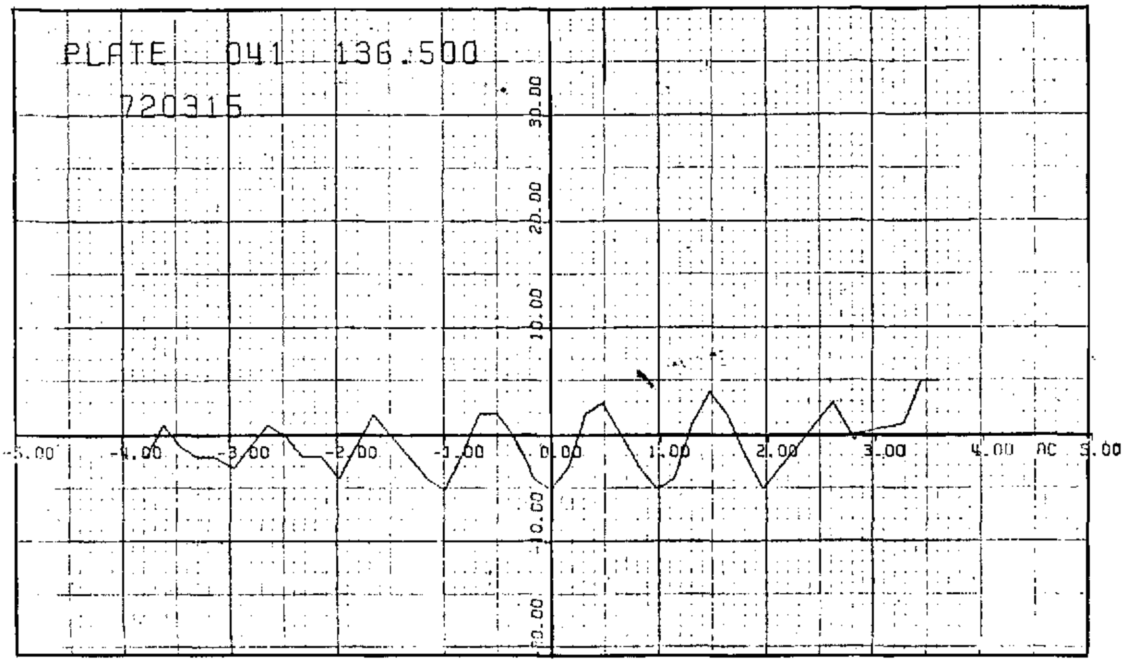


Figure C-32.—Example of CalComp plot of *ra* and *rb* versus A_c or B_c for DR01E.

The equations and their coefficients are printed in the third phase, PRINT. Coefficients cards are also punched here.

In the fourth phase, RESIDL, the coefficients computed in COEFFC or entered on the Lead card and the data from the Minitrack disk are used as input to compute the residuals for all sets for each data point. Equations (C-28) to (C-33) are printed as computed. Equations (C-34) to (C-36) are saved on disk and printed after all points have been processed. Plot data are also saved on the disk if plotting is to be done.

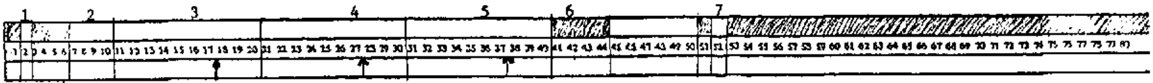
In the fifth, DRZPLT, a plot tape is generated from the disk file written in RESIDL.

Input

The input for a regular calibration is a Lead card and one Minitrack disk file of fine data (output from the PROOFREAD program). For a historical comparison, the input is a Lead card, a deck of coefficients from the previous calibration, and the disk file previously mentioned.

Card Deck

Input cards are fully explained in figure C-33. The input deck layout is shown in figure C-34.



FIELD(S)

1 C, 1 - Polar, 0 = Equatorial

2 Station Number

3 KCA, 3 assumed decimals

4 KCB, 3 assumed decimals

For Historical Comparisons, these fields must contain the KCA and KCB of the previous calibration.

5 Frequency, 3 assumed decimals

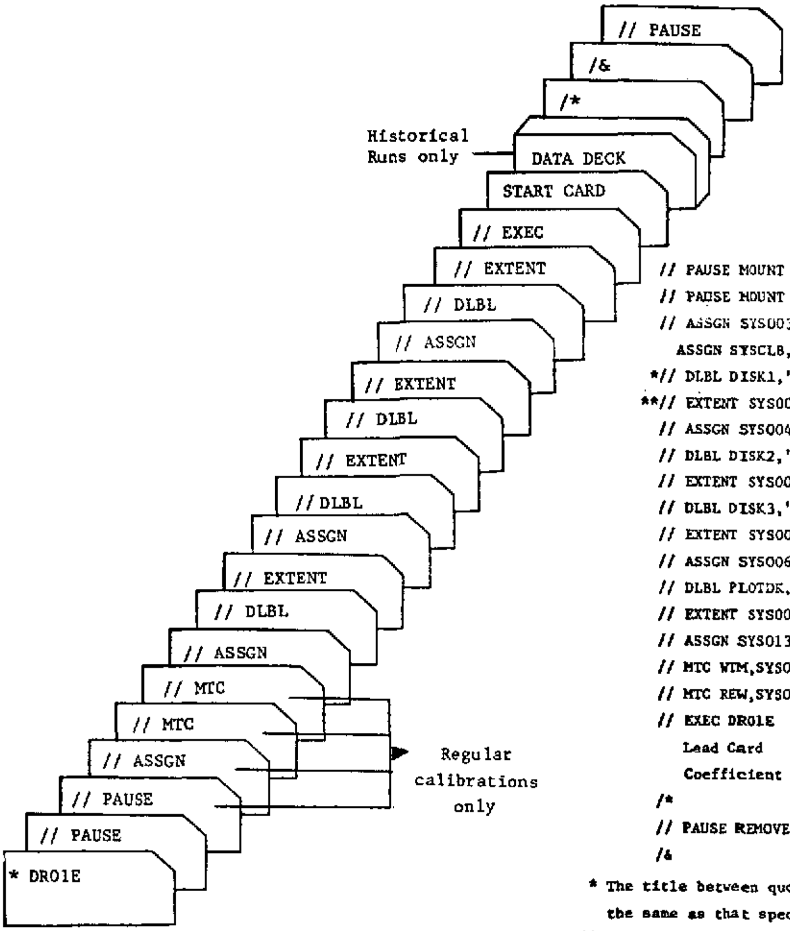
6 Calibration Date

7 P, 1 - Historical, 0 - Regular Computation

8 SPECIAL = blank KCA, KCB and FREQ will be taken from the first disk record otherwise start card values will be used. This is a special option to be used with regular calibrations for testing purposes only.

All values must be right justified.

Figure C-33.—Input card for DR01E.



```
   // PAUSE MOUNT SCRATCH ON 283
   // PAUSE MOUNT DRZ001 ON 133
   // ASSGN SYS003,X'133'
     ASSGN SYSCLB,X'135'
  *// DLBL DISK1,'FINE DATA TITLE       136.500',68/001,SD
 **// EXTENT SYS003,DRZ001,,,2700,25
   // ASSGN SYS004,X'131'
   // DLBL DISK2,'TEMP STORAGE-RESIDUALS',69/100,SD
   // EXTENT SYS004,111111,,,2400,300
   // DLBL DISK3,'TEMP STORAGE-RESIDUALS',69/100,SD
   // EXTENT SYS004,111111,,,2400,300
   // ASSGN SYS006,X'131'
   // DLBL PLOTDK,'DR201E RESID PLOT DATA',69/100,SD
   // EXTENT SYS006,111111,,,2700,300
   // ASSGN SYS013,X'283',X'28'
   // MTC WTM,SYS013,01
   // MTC REW,SYS013
   // EXEC DR01E
      Lead Card
      Coefficient Card Deck - for historical comparison only
   /*
   // PAUSE REMOVE AND LABEL PLOT TAPE
   /&
```

* The title between quotes (') on the card shown must be exactly the same as that specified for the DR01K Program

** The extent on this card must be exactly the same as that for the DR01K Program.

Figure C-34.—Input deck layout for DR01E.

Minitrack Disk File

The first record contains general information:

Location (bytes)	*Item*
105 to 112	KC_A
113 to 120	KC_B
121 to 128	Frequency
129 to 136	Calibration date

The succeeding records contain data, one data point per record:

Location (bytes)	*Item*	
1 to 4	Hours	Data point time
5 to 8	Minutes	
9 to 12	Seconds	
17 to 24	A_c	
25 to 32	B_c	
33 to 40	Decimal portion of A_m	
41 to 48	Decimal portion of B_m	
49 to 56	C	
57 to 64	R_2	
65 to 72	$\Delta a_0/\Delta \mathrm{UT}$	
73 to 80	$\Delta b_0/\Delta \mathrm{UT}$	
81 to 88	N'_A	
89 to 96	N'_B	
97 to 100	Station number	
101 to 103	Plate number	
104	Plot code	
105 to 112	ra	
113 to 120	rb	
121 to 128	A_m	
129 to 136	B_m	

Output

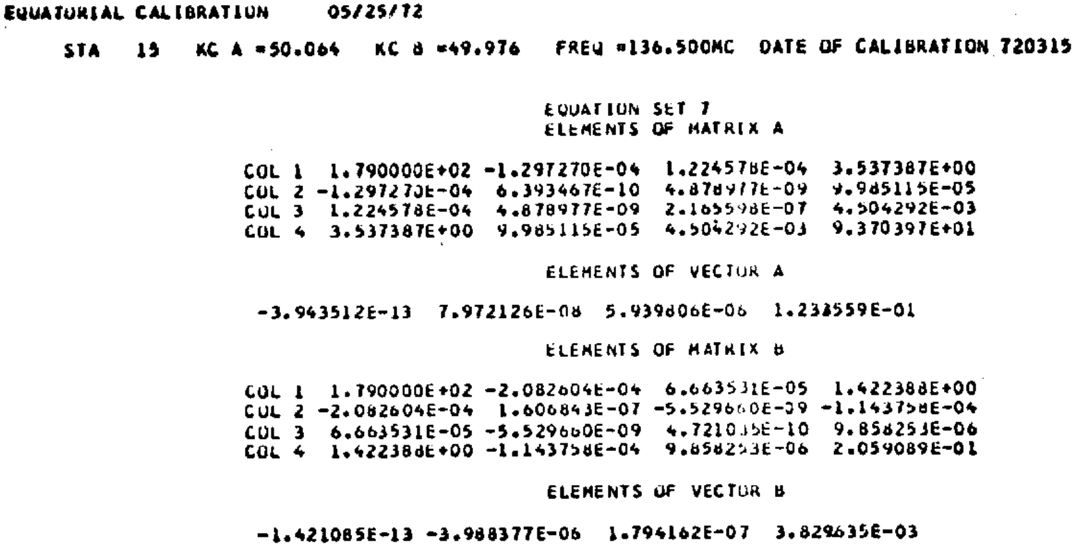
The regular output consists of a tabular listing, a card deck, and, if requested, a plot tape. The historical comparison consists of a tabular listing only.

Tabular Listing

The station number, KC_A, KC_B, frequency, and calibration data are printed at the top of each page for identification (fig. C-35). For a regular calibration, the coefficient matrices and the data vectors used in solving equations (C-33) and (C-34) are listed. Then for each

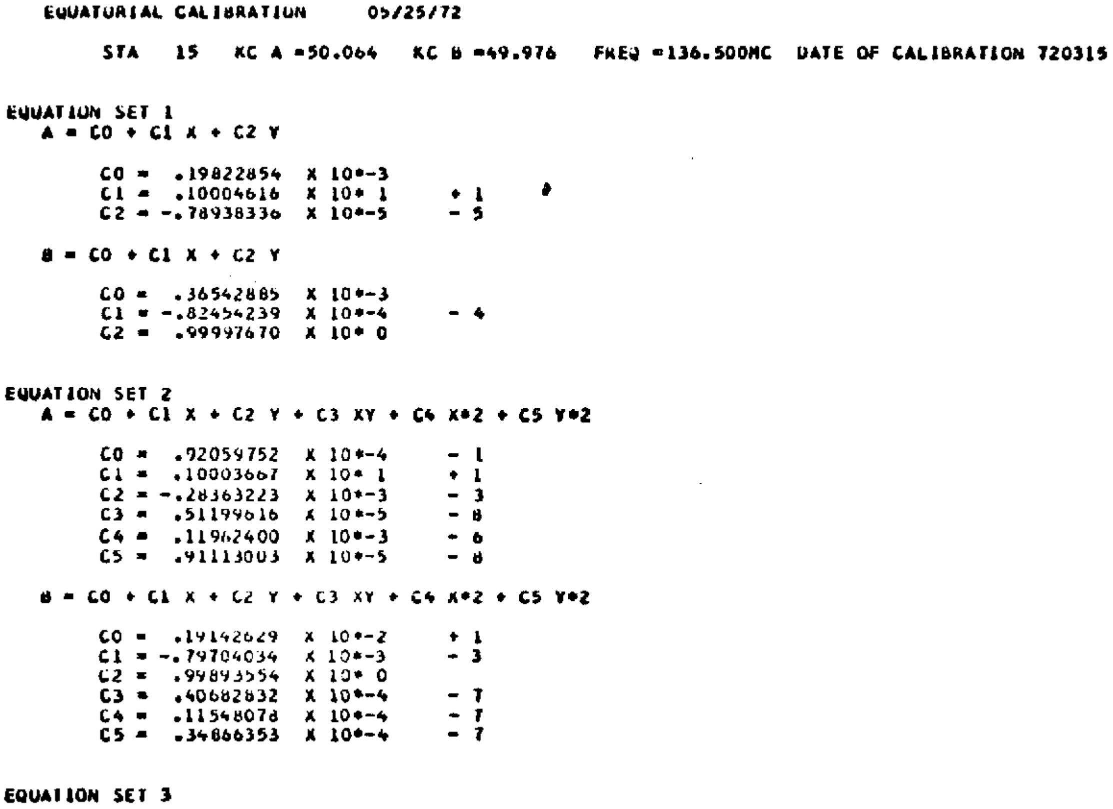
```
     EQUATORIAL CALIBRATION     05/25/72

          STA   15   KC A =50.064   KC B =49.976   FREQ =136.500MC  DATE OF CALIBRATION 720315

                                                        EQUATION SET 6
                                                        ELEMENTS OF MATRIX A

 COL 1  1.790000E+02 -2.719376E+01  2.903734E+03 -1.241827E+02  8.910018E+02  7.805102E+04  2.196401E+06 -6.538439E+00  1.137760E+00
 COL 2 -2.719376E+01  8.910018E+02 -1.241827E+02  1.518018E+04 -1.076547E+02 -1.916117E+03 -3.765252E+04 -1.569079E+01  3.841688E+00
 COL 3  2.903734E+03 -1.241827E+02  7.805102E+04 -1.916117E+03  1.518018E+04  2.196401E+06  6.270065E+07 -1.512380E+02  1.101518E+02
 COL 4 -1.241827E+02  1.518018E+04 -1.916117E+03  4.152593E+05  2.583123E+03 -3.765252E+04 -6.194838E+05 -2.954414E+02  1.383315E+02
 COL 5  8.910018E+02 -1.076547E+02  1.518018E+04  2.583123E+03  8.148668E+03  4.152593E+05  1.184637E+07 -2.609935E+01  2.464640E+00
 COL 6  7.805102E+04 -1.916117E+03  2.196401E+06 -3.765252E+04  4.152593E+05  6.270065E+07  1.809593E+09 -4.408600E+03  2.836436E+03
 COL 7  2.196401E+06 -3.765252E+04  6.270065E+07 -6.194838E+05  1.184637E+07  1.809593E+09  5.271607E+10 -1.322763E+05  7.072885E+04
 COL 8 -6.538439E+00 -1.569079E+01 -1.512380E+02 -2.954414E+02 -2.609935E+01 -4.408600E+03 -1.372763E+05  8.926766E+01 -2.496430E+00
 COL 9  1.137760E+00  3.841688E+00  1.101518E+02  1.383315E+02  2.464640E+00  2.836436E+03  7.072885E+04 -2.496430E+00  8.973234E+01

                                                        ELEMENTS OF VECTOR A

  -2.719376E+01  8.914090E+02 -1.242806E+02  1.518838E+04 -1.071462E+02 -1.911838E+03 -3.730289E+04 -1.564223E+01  3.568005E+00

                                                        ELEMENTS OF MATRIX B

 COL 1  1.790000E+02 -2.719376E+01  2.903734E+03 -1.241827E+02  8.910018E+02  7.805102E+04  2.196401E+06  1.088651E-01 -6.392321E+01
 COL 2 -2.719376E+01  8.910018E+02 -1.241827E+02  1.518018E+04 -1.076547E+02 -1.916117E+03 -3.765252E+04 -1.154716E+01 -5.018416E+01
 COL 3  2.903734E+03 -1.241827E+02  7.805102E+04 -1.916117E+03  1.518018E+04  2.196401E+06  6.270065E+07  1.834634E+02 -1.290626E+03
 COL 4 -1.241827E+02  1.518018E+04 -1.916117E+03  4.152593E+05  2.583123E+03 -3.765252E+04 -6.194838E+05 -4.914026E+02 -2.426315E+03
 COL 5  8.910018E+02 -1.076547E+02  1.518018E+04  2.583123E+03  8.148668E+03  4.152593E+05  1.184637E+07  9.331661E+01 -3.029851E+02
 COL 6  7.805102E+04 -1.916117E+03  2.196401E+06 -3.765252E+04  4.152593E+05  6.270065E+07  1.809593E+09  1.027697E+04 -3.765430E+04
 COL 7  2.196401E+06 -3.765252E+04  6.270065E+07 -6.194838E+05  1.184637E+07  1.809593E+09  5.271607E+10  4.475478E+05 -1.106716E+06
 COL 8  1.088651E-01 -1.154716E+01  1.834634E+02 -4.914026E+02  9.331661E+01  1.027697E+04  4.475478E+05  9.500968E+01  3.633724E+00
 COL 9 -6.392321E+01 -5.018416E+01 -1.290626E+03 -2.426315E+03 -3.029851E+02 -3.765430E+04 -1.106716E+06  3.633724E+00  8.399032E+01

                                                        ELEMENTS OF VECTOR B

   2.903734E+03 -1.242632E+02  7.805027E+04 -1.910803E+03  1.518049E+04  2.196403E+06  6.270127E+07  1.834830E+02 -1.290847E+03
```

Figure C-35.—Sample output listing of regular calibration for DR01E.



```
EQUATORIAL CALIBRATION     05/25/72

     STA   15   KC A =50.064   KC B =49.976   FREQ =136.500MC  DATE OF CALIBRATION 720315

                                          EQUATION SET 7
                                          ELEMENTS OF MATRIX A

                 COL 1  1.790000E+02 -1.297270E-04  1.224578E-04  3.537387E+00
                 COL 2 -1.297270E-04  6.393467E-10  4.878977E-09  9.985115E-05
                 COL 3  1.224578E-04  4.878977E-09  2.165598E-07  4.504292E-03
                 COL 4  3.537387E+00  9.985115E-05  4.504292E-03  9.370397E+01

                                          ELEMENTS OF VECTOR A

                  -3.943512E-13  7.972126E-08  5.939806E-06  1.238559E-01

                                          ELEMENTS OF MATRIX B

                 COL 1  1.790000E+02 -2.082604E-04  6.663531E-05  1.422388E+00
                 COL 2 -2.082604E-04  1.606843E-07 -5.529660E-09 -1.143758E-04
                 COL 3  6.663531E-05 -5.529660E-09  4.721035E-10  9.858253E-06
                 COL 4  1.422388E+00 -1.143758E-04  9.858253E-06  2.059089E-01

                                          ELEMENTS OF VECTOR B

                  -1.421085E-13 -3.988377E-06  1.794162E-07  3.829635E-03
```

Figure C-35 (continued).–Sample output listing of regular calibration for DR01E.



```
    EQUATORIAL CALIBRATION     05/25/72

         STA   15   KC A =50.064   KC B =49.976   FREQ =136.500MC  DATE OF CALIBRATION 720315

EQUATION SET 1
   A = C0 + C1 X + C2 Y

        C0 =  .19822854  X 10*-3
        C1 =  .10004616  X 10* 1      + 1
        C2 = -.78938336  X 10*-5      - 5

   B = C0 + C1 X + C2 Y

        C0 =  .36542885  X 10*-3
        C1 = -.82454239  X 10*-4      - 4
        C2 =  .99997670  X 10* 0


EQUATION SET 2
   A = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2

        C0 =  .92059752  X 10*-4      - 1
        C1 =  .10003667  X 10* 1      + 1
        C2 = -.28363223  X 10*-3      - 3
        C3 =  .51199616  X 10*-5      - 8
        C4 =  .11962400  X 10*-3      - 6
        C5 =  .91113003  X 10*-5      - 8

   B = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2

        C0 =  .19142629  X 10*-2      + 1
        C1 = -.79704034  X 10*-3      - 3
        C2 =  .99893554  X 10* 0
        C3 =  .40682832  X 10*-4      - 7
        C4 =  .11548078  X 10*-4      - 7
        C5 =  .34866353  X 10*-4      - 7

EQUATION SET 3
```

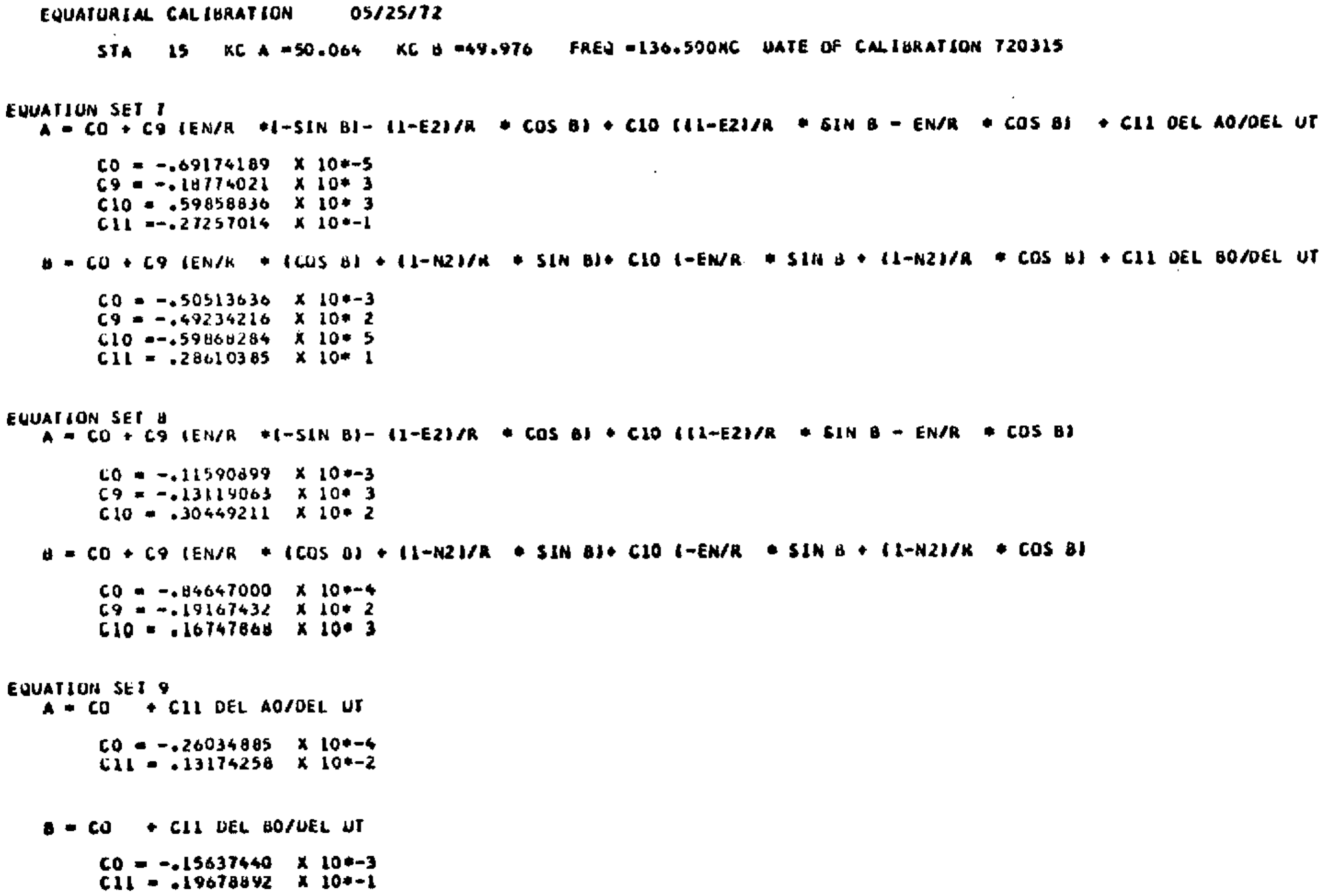
Figure C-35 (continued).—Sample output listing of regular calibration for DR01E.

```
    A = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C6 Y*3

        C0 = -.18664674  X 10*-2      + 1
        C1 =  .10004117  X 10* 1      + 1
        C2 =  .14373360  X 10*-2      - 2
        C3 =  .30547584  X 10*-5      - 8
        C4 =  .11377204  X 10*-3      - 6
        C5 = -.11464754  X 10*-3      - 6
        C6 =  .22851145  X 10*-5      -11

    B = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C6 Y*3

        C0 =  .56877944  X 10*-2      + 1
        C1 = -.88370764  X 10*-3      - 3
        C2 =  .99561971  X 10* 0
        C3 =  .44661898  X 10*-4      - 7
        C4 =  .22823162  X 10*-4      - 7
        C5 =  .27331486  X 10*-3      - 6
        C6 = -.44027736  X 10*-5      -11


  EQUATION SET 4
    A = C0 + C1 X + C2 Y + C7 SIN(X) + C8 COS(X)

        C0 =  .84369102  X 10*-4      - 1
        C1 =  .10004823  X 10* 1      + 1
        C2 =  .17620471  X 10*-5      - 5
        C7 =  .55244553  X 10*-3
        C8 = -.30585235  X 10*-2      + 1

    B = C0 + C1 X + C2 Y + C7 SIN(Y) + C8 COS(Y)

        C0 = -.62495176  X 10*-3
        C1 = -.34972694  X 10*-3      - 3
        C2 =  .99994224  X 10* 0
        C7 =  .43807431  X 10*-3
        C8 = -.42241919  X 10*-2      + 1

  EQUATION SET 5
```

Figure C-35 *(continued)*.—Sample output listing of regular calibration for DR01E.

```
A = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C7 SIN (X) + C8 COS (X)

     C0 = -.89943038  X 10*-4     - 1
     C1 =  .10003556  X 10* 1     + 1
     C2 = -.22482505  X 10*-3     - 3
     C3 =  .70342403  X 10*-5     - 8
     C4 =  .11467065  X 10*-3     - 6
     C5 =  .74876756  X 10*-5     - 8
     C7 =  .56109095  X 10*-3
     C8 = -.30231744  X 10*-2     + 1

B = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C7 SIN (Y) + C8 CUS (Y)

     C0 =  .12869701  X 10*-2     + 1
     C1 = -.54032534  X 10*-3     - 3
     C2 =  .99877502  X 10* 0
     C3 =  .15931412  X 10*-4     - 7
     C4 =  .78238507  X 10*-4     - 7
     C5 =  .39823781  X 10*-4     - 7
     C7 = -.17155679  X 10*-2     + 1
     C8 = -.21296327  X 10*-2     + 1


EQUATION SET 6
  A = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C6 Y*3 + C7 SIN (X) + C8 COS (X)

     C0 = -.19635039  X 10*-2     + 1
     C1 =  .10003995  X 10* 1     + 1
     C2 =  .14239020  X 10*-2     - 2
     C3 =  .50476605  X 10*-5     - 8
     C4 =  .10897120  X 10*-3     - 6
     C5 = -.11110931  X 10*-3     - 6
     C6 =  .21903153  X 10*-5     -11
     C7 =  .60585725  X 10*-3
     C8 = -.29983505  X 10*-2     + 1

  B = C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2 + C6 Y*3 + C7 SIN (Y) + C8 COS (Y)

     C0 = -.56294114  X 10*-3
     C1 = -.41546325  X 10*-3     - 3
     C2 =  .10001667  X 10* 1     + 1
     C3 =  .57652946  X 10*-5     - 8
     C4 =  .98477153  X 10*-4     - 7
     C5 = -.64269339  X 10*-4     - 7
     C6 =  .19828796  X 10*-5     -11
     C7 = -.25140077  X 10*-2     + 1
     C8 = -.27973224  X 10*-2     + 1
```

Figure C-35 (continued).—Sample output listing of regular calibration for DR01E.



```
   EQUATORIAL CALIBRATION     05/25/72

        STA   15   KC A =50.064   KC B =49.976   FREQ =136.590MC  DATE OF CALIBRATION 720315


EQUATION SET 7
   A = C0 + C9 (EN/R  *(-SIN B)- (1-E2)/R  * COS B) + C10 ((1-E2)/R  * SIN B - EN/R  * COS B)  + C11 DEL A0/DEL UT

        C0 = -.69174189  X 10*-5
        C9 = -.18774021  X 10* 3
        C10 = .59858836  X 10* 3
        C11 =-.27257014  X 10*-1

   B = C0 + C9 (EN/R  * (COS B) + (1-N2)/R  * SIN B)+ C10 (-EN/R  * SIN B + (1-N2)/R  * COS B) + C11 DEL B0/DEL UT

        C0 = -.50513636  X 10*-3
        C9 = -.49234216  X 10* 2
        C10 =-.59868284  X 10* 5
        C11 = .28610385  X 10* 1


EQUATION SET 8
   A = C0 + C9 (EN/R  *(-SIN B)- (1-E2)/R  * COS B) + C10 ((1-E2)/R  * SIN B - EN/R  * COS B)

        C0 = -.11590899  X 10*-3
        C9 = -.13119063  X 10* 3
        C10 = .30449211  X 10* 2

   B = C0 + C9 (EN/R  * (COS B) + (1-N2)/R  * SIN B)+ C10 (-EN/R  * SIN B + (1-N2)/R  * COS B)

        C0 = -.84647000  X 10*-4
        C9 = -.19167432  X 10* 2
        C10 = .16747868  X 10* 3


EQUATION SET 9
   A = C0   + C11 DEL A0/DEL UT

        C0 = -.26034885  X 10*-4
        C11 = .13174258  X 10*-2


   B = C0   + C11 DEL B0/DEL UT

        C0 = -.15637440  X 10*-3
        C11 = .19678892  X 10*-1
```

Figure C-35 (concluded).—Sample output listing of regular calibration for DR01E.

set, the equations are given along with the computed coefficients written in scientific notation (powers of ten) and with a number that represents a modified exponent determined as follows:

Coefficient	*Modified exponent*
C_0, C'_0	$y + 3$
C_1, C'_1	y
C_2, C'_2	y
C_3, C'_3	$y - 3$
C_4, C'_4	$y - 3$
C_5, C'_5	$y - 3$
C_6, C'_6	$y - 6$
C_7, C'_7	$y + 3$
C_8, C'_8	$y + 3$

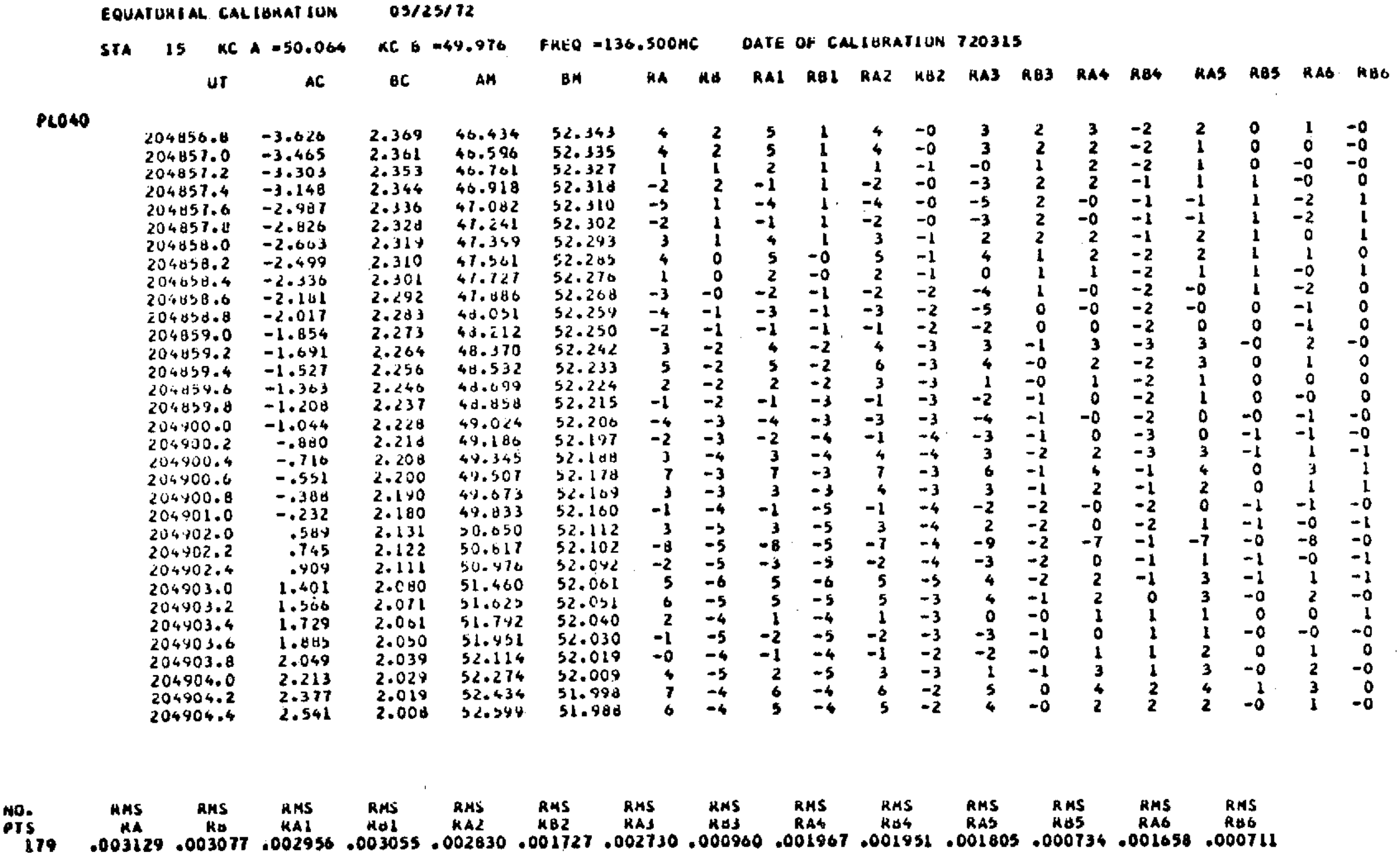
y represents the actual exponent. For both a regular calibration and a historical run, two residual listings are generated, the first containing residuals from equations (C-28) to (C-33) (fig. C-36) and the second containing residuals from equations (C-34) to (C-36) (fig. C-37). In addition, the time, A_c, B_c, A_m, B_m, *ra,* and *rb* are listed for each data point on each of the two sets of residual listings. The points are further identified by plate number. Summary lines give the rms value of the residuals.

Coefficient Cards

For each set, 10 coefficient cards are punched. Coefficients are punched in the *E* format with eight decimals in a field 15 columns wide (NN.NNNNNNNNE±NN). The first card contains the station number, calibration date, set type (*A* or *B*), and the first three coefficients of equation (C-28) (fig. C-38). Each succeeding card contains five consecutive coefficients and the card number in columns 76-80 (fig. C-39). The last card contains a dummy value in the last field.

Plot Tape

The low-density tape (fig. C-32) for the CalComp plotter described earlier is generated with a regular calibration run.

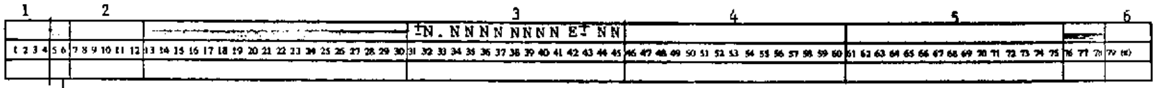


```
        EQUATORIAL CALIBRATION     05/25/72

        STA   15   KC A =50.064   KC B =49.976   FREQ =136.500MC    DATE OF CALIBRATION 720315
                  UT         AC        BC        AM        BM      RA   RB   RA1  RB1  RA2  RB2  RA3  RB3  RA4  RB4  RA5  RB5  RA6  RB6

PL040
             204856.8    -3.626     2.369    46.434    52.343     4    2    5    1    4   -0    3    2    3   -2    2    0    1   -0
             204857.0    -3.465     2.361    46.596    52.335     4    2    5    1    4   -0    3    2    2   -2    1    0    0   -0
             204857.2    -3.303     2.353    46.761    52.327     1    1    2    1    1   -1   -0    1    2   -2    1    0   -0   -0
             204857.4    -3.148     2.344    46.918    52.318    -2    2   -1    1   -2   -0   -3    2    2   -1    1    1   -0    0
             204857.6    -2.987     2.336    47.082    52.310    -5    1   -4    1   -4   -0   -5    2   -0   -1   -1    1   -2    1
             204857.8    -2.826     2.328    47.241    52.302    -2    1   -1    1   -2   -0   -3    2   -0   -1   -1    1   -2    1
             204858.0    -2.663     2.319    47.399    52.293     3    1    4    1    3   -1    2    2    2   -1    2    1    0    1
             204858.2    -2.499     2.310    47.561    52.285     4    0    5   -0    5   -1    4    1    2   -2    2    1    1    0
             204858.4    -2.336     2.301    47.727    52.276     1    0    2   -0    2   -1    0    1    1   -2    1    1   -0    1
             204858.6    -2.181     2.292    47.886    52.268    -3   -0   -2   -1   -2   -2   -4    1   -0   -2   -0    1   -2    0
             204858.8    -2.017     2.283    48.051    52.259    -4   -1   -3   -1   -3   -2   -5    0   -0   -2   -0    0   -1    0
             204859.0    -1.854     2.273    48.212    52.250    -2   -1   -1   -1   -1   -2   -2    0    0   -2    0    0   -1    0
             204859.2    -1.691     2.264    48.370    52.242     3   -2    4   -2    4   -3    3   -1    3   -3    3   -0    2   -0
             204859.4    -1.527     2.256    48.532    52.233     5   -2    5   -2    6   -3    4   -0    2   -2    3    0    1    0
             204859.6    -1.363     2.246    48.699    52.224     2   -2    2   -2    3   -3    1   -0    1   -2    1    0    0    0
             204859.8    -1.208     2.237    48.858    52.215    -1   -2   -1   -3   -1   -3   -2   -1    0   -2    1    0   -0    0
             204900.0    -1.044     2.228    49.024    52.206    -4   -3   -4   -3   -3   -3   -4   -1   -0   -2    0   -0   -1   -0
             204900.2     -.880     2.218    49.186    52.197    -2   -3   -2   -4   -1   -4   -3   -1    0   -3    0   -1   -1   -0
             204900.4     -.716     2.208    49.345    52.188     3   -4    3   -4    4   -4    3   -2    2   -3    3   -1    1   -1
             204900.6     -.551     2.200    49.507    52.178     7   -3    7   -3    7   -3    6   -1    4   -1    4    0    3    1
             204900.8     -.388     2.190    49.673    52.169     3   -3    3   -3    4   -3    3   -1    2   -1    2    0    1    1
             204901.0     -.232     2.180    49.833    52.160    -1   -4   -1   -5   -1   -4   -2   -2   -0   -2    0   -1   -1   -0
             204902.0      .589     2.131    50.650    52.112     3   -5    3   -5    3   -4    2   -2    0   -2    1   -1   -0   -1
             204902.2      .745     2.122    50.817    52.102    -8   -5   -8   -5   -7   -4   -9   -2   -7   -1   -7   -0   -8   -0
             204902.4      .909     2.111    50.976    52.092    -2   -5   -3   -5   -2   -4   -3   -2    0   -1    1   -1   -0   -1
             204903.0     1.401     2.080    51.460    52.061     5   -6    5   -6    5   -5    4   -2    2   -1    3   -1    1   -1
             204903.2     1.566     2.071    51.625    52.051     6   -5    5   -5    5   -3    4   -1    2    0    3   -0    2   -0
             204903.4     1.729     2.061    51.792    52.040     2   -4    1   -4    1   -3    0   -0    1    1    1    0    0    1
             204903.6     1.885     2.050    51.951    52.030    -1   -5   -2   -5   -2   -3   -3   -1    0    1    1   -0   -0   -0
             204903.8     2.049     2.039    52.114    52.019    -0   -4   -1   -4   -1   -2   -2   -0    1    1    2    0    1    0
             204904.0     2.213     2.029    52.274    52.009     4   -5    2   -5    3   -3    1   -1    3    1    3   -0    2   -0
             204904.2     2.377     2.019    52.434    51.998     7   -4    6   -4    6   -2    5    0    4    2    4    1    3    0
             204904.4     2.541     2.008    52.599    51.988     6   -4    5   -4    5   -2    4   -0    2    2    2   -0    1   -0



NO.       RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS     RMS
PTS       RA      RB      RA1     RB1     RA2     RB2     RA3     RB3     RA4     RB4     RA5     RB5     RA6     RB6
 179   .003129 .003077 .002956 .003055 .002830 .001727 .002730 .000960 .001967 .001951 .001805 .000734 .001658 .000711
```

Figure C-36.–Sample output listing of first residual set for regular calibration and historical run for DR01E.

```
EQUATORIAL CALIBRATION        05/25/72

 STA    15   KC A =50.064   KC B =49.976   FREQ =136.500MC    DATE OF CALIBRATION 720315

                         UT          AC          BC          AM           BH          RA    RB   RA7   RB7   RA8   RB8   RA9   RB9

      PL040           204856.8     -3.626       2.369      46.434       52.343       4     2     3     3     3     3     3     3
                      204857.0     -3.465       2.361      46.596       52.335       4     2     3     3     3     3     3     3
                      204857.2     -3.303       2.353      46.761       52.327       1     1    -0     2    -0     2    -0     2
                      204857.4     -3.148       2.344      46.918       52.318      -2     2    -3     3    -3     3    -3     3
                      204857.6     -2.987       2.336      47.082       52.310      -5     1    -6     2    -6     3    -6     2
                      204857.8     -2.826       2.328      47.241       52.302      -2     1    -3     2    -3     2    -3     2
                      204858.0     -2.663       2.319      47.399       52.293       3     1     2     3     2     2     2     2
                      204858.2     -2.499       2.310      47.561       52.285       4     0     3     2     3     1     3     1
                      204858.4     -2.336       2.301      47.727       52.276       1     0     0     1     0     1    -0     1
                      204858.6     -2.181       2.292      47.886       52.268      -3    -0    -4     1    -4     1    -4     1
                      204858.8     -2.017       2.283      48.051       52.259      -4    -1    -5     1    -5     0    -5     0
                      204859.0     -1.854       2.273      48.212       52.250      -2    -1    -3     0    -3     0    -3     0
                      204859.2     -1.691       2.264      48.370       52.242       3    -2     3    -1     3    -1     2    -1
                      204859.4     -1.527       2.256      48.532       52.233       5    -2     4    -0     4    -1     4    -1
                      204859.6     -1.363       2.246      48.699       52.224       2    -2     1    -0     1    -1     1    -1
                      204859.8     -1.208       2.237      48.858       52.215      -1    -2    -2    -1    -2    -1    -2    -1
                      204900.0     -1.044       2.228      49.024       52.206      -4    -3    -5    -2    -5    -2    -5    -2
                      204900.2      -.880       2.218      49.186       52.197      -2    -3    -3    -2    -3    -2    -3    -2
                      204900.4      -.716       2.208      49.345       52.188       3    -4     2    -2     2    -3     2    -3
                      204900.6      -.551       2.200      49.507       52.178       7    -3     6     0     6    -2     6    -2
                      204900.8      -.388       2.190      49.673       52.169       3    -3     2    -3     3    -2     2    -2
                      204901.0      -.232       2.180      49.833       52.160      -1    -4    -2    -3    -2    -3    -2    -3
                      204902.0       .589       2.131      50.650       52.112       3    -5     2    -4     2    -4     2    -4
                      204902.2       .745       2.122      50.817       52.102      -8    -5    -8    -3    -8    -3    -9    -4
                      204902.4       .909       2.111      50.976       52.092      -2    -5    -3    -3    -3    -4    -3    -4
                      204903.0      1.401       2.080      51.460       52.061       5    -6     5    -4     5    -4     4    -4
                      204903.2      1.566       2.071      51.625       52.051       6    -5     5    -2     5    -3     5    -4
                      204903.4      1.729       2.061      51.792       52.040       2    -4     1    -3     1    -3     1    -3
                      204903.6      1.885       2.050      51.951       52.030      -1    -5    -2    -2    -2    -3    -2    -3
                      204903.8      2.049       2.039      52.114       52.019      -0    -4    -1    -2    -1    -3    -1    -3
                      204904.0      2.213       2.029      52.274       52.009       4    -5     3    -3     3    -3     2    -3
                      204904.2      2.377       2.019      52.434       51.998       7    -4     7    -2     7    -2     6    -2
                      204904.4      2.541       2.008      52.599       51.988       6    -4     5    -2     5    -3     5    -3
                      204904.6      2.704       1.998      52.765       51.977       4    -3     3    -2     3    -2     2    -2

                          NO.      RMS        RMS        RMS        RMS        RMS        RMS
                          PTS      RA7        RB7        RA8        RB8        RA9        RB9
                          179   .002966    .002678    .002973    .002979    .002981    .003008
```

Figure C-37.—Sample output listing of second residual set for regular calibration and historical run for DR01E.



FIELD(S)

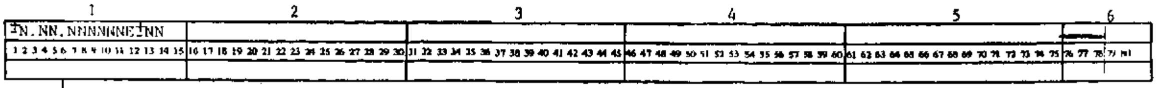
1 Station Number

2 Calibration Date (YY MM DD)

3-5 Coefficients 1-3 (Equations C-28 to C-33 for *A* and *B*, eight decimals, decimal point punched in third column of each field and exponent)

6 Equation number *A* or *B*

Figure C-38.—First coefficient card (*A* and *B* sets) for DR01E.



FIELD(S)

1-5 Coefficients in consecutive order beginning with equation (C-29)

6 Card No.

Figure C-39.—Coefficient cards 2 to 9 (*A* and *B* sets) for DR01E.

```
// JOB DR01J HAAS,3314,D,15001;NASA
// OPTION SYM,CATAL,NODECK
  PHASE DR01J,ROOT
// EXEC PL/I
* PROCESS LISTO,NOOPT,STMT
 DRZ01J: PROCEDURE OPTIONS(MAIN);
         DECLARE START FILE STREAM INPUT ENVIRONMENT (F(80) MEDIUM
                      (SYSIPT,2540)),
                 FINE FILE RECORD OUTPUT ENVIRONMENT (F(3400,136)MEDIUM
                      (SYS018,2314)),
                 AMB  FILE RECORD OUTPUT ENVIRONMENT (F(1680,56) MEDIUM
                      (SYS018,2314)),
                 LST FILE PRINT ENVIRONMENT(F(130) MEDIUM (SYSLST,1403)
                                                                       );
         DECLARE STA CHARACTER(4) EXTERNAL, TYPE CHARACTER(10)EXTERNAL,
                 PLT CHARACTER(3) EXTERNAL,
                (XC,YC,ALC,DLC,MR,EWC,NSC,VC)FLOAT DECIMAL(15) EXTERNAL
                ,NR  FIXED BINARY EXTERNAL;
         DECLARE (AL(3),DL(3),FILL,PC(6),PCP(6))FLOAT DECIMAL(15);
         DECLARE                     PLC CHARACTER (4),DM CHARACTER(1),
                 RR      CHARACTER(7),SEQ CHARACTER (8),
                 MRR(100) CHARACTER(3) EXTERNAL,NMRR EXTERNAL;
         NMRR=0;
         OPEN FILE (START),FILE(LST);
         GET FILE(START)EDIT (SEQ,TYPE,DM)(A(8),X(2),A(10),X(59),A(1));
         IF SEQ='ADDITION' THEN DO;
                                RR='START   ';
                                CALL OVERLAY('DR01J04');
                                CALL OUTPUT (SEQ,RR);
                                END;
         OPEN FILE(FINE),FILE(AMB);
   LOOP: ON ENDFILE (START) GO TO ALLOVR;
         GET FILE(START) EDIT (STA,PLT,XC,YC,AL,DL,NR,MR,RR,PLC,DM)
             (X(1),A(4),X(1),A(3),2(X(1),F(7,3)),X(1),2F(2,0),F(5,3),
                   X(1),2F(3,0),F(6,3),X(1),F(1),X(1),F(6,3),X(1),
                   A(7),X(1),A(4),X( 9),A(1));
         ALC=AL(3)/3.6E03;
         ALC=ALC+AL(2)/6.0E01;
         ALC=(ALC+ABS(AL(1)))*0.26179938779E00;
         IF  AL(1)<0.0E0 THEN ALC=-ALC;
         DLC=DL(3)/3.6E03;
         DLC=DLC+DL(2)/6.0E01;
         DLC=(DLC+DL(1))*0.1745329252E-01;
         IF RR='REPLACE' THEN DO;
                            NMRR=NMRR+1;
                            MRR(NMRR)=PLT;
                            END;
         IF PLC='PACT' THEN GO TO CALL3;
         IF PLC='PLAT' THEN DO;
                            GET EDIT (PC,DM)(6E(13,0),X(1),A(1));
                            GET EDIT (PCP,DM)(6E(13,0),X(1),A(1));
                            GO TO CALL3;
                            END;
                       ELSE DO;
                            CALL OVERLAY ('DR01J02');
                           CALL STRSOL (PC,PCP);
                            IF PCP(1)=9999999.9 THEN DO;
                                                 ON ENDFILE(START)GO TO
                                                          LOOP;
                                           FLUSH:GET FILE(START)EDIT(DM)
```

```
                                                           (X(79),A(1));
                                                  GO TO FLUSH;
                                                        END;
                             END;
  CALL3: CALL OVERLAY ('DR01J03');
         CALL MTRACK (PC,PCP,PLC);
         GO TO LOOP;
 ALLOVR: CALL OVERLAY ('DR01J04');
         CALL OUTPUT(SEQ,RR);
         END;
/*   END OF DATA
  INCLUDE IJKVCEM
  INCLUDE IJKVBCM
  INCLUDE IJKVTBM
  INCLUDE IJKVICM
  INCLUDE IJKQNLD
  INCLUDE IJKTCBM
  INCLUDE IJKTGDI
  INCLUDE IJKQSLD
  INCLUDE IJGFIEZZ
  PHASE DR01J02,*
// EXEC PL/1
* PROCESS LISTO,NOOPT,STMT
 STRSOL: PROCEDURE (PC,PCP);
         DECLARE (DSIMQ)ENTRY;
         DECLARE PNCH FILE STREAM OUTPUT ENVIRONMENT (F(78)  MEDIUM
                                      (SYSPCH,2540));
         DECLARE LST FILE PRINT ENVIRONMENT(F(130) MEDIUM (SYSLST,1403)
                                                                        );
         DECLARE (PC(6),PCP(6))FLOAT DECIMAL (15);
         DECLARE STA CHARACTER(4) EXTERNAL,TYPE CHARACTER(10) EXTERNAL,
                 PLT CHARACTER (3) EXTERNAL, NR FIXED BINARY EXTERNAL,
                 (XC,YC,ALC,DLC,MR) FLOAT DECIMAL(15) EXTERNAL,
                 1 DAY DEFINED DAYE,(2 YR,2 MO,2 DY)CHARACTER (2);
         DECLARE DAYE CHARACTER(6);
         DECLARE(STAR(75 ),STR) FIXED BINARY;
         DECLARE  RJOUT(8) CHARACTER (84);
         DECLARE IDENT(75 ) CHARACTER(8),CODE(75 )CHARACTER(1);
         DECLARE (LX(6),NX(6),IAS(2),IDS(2))FIXED BINARY (20);
         DECLARE  NG(9) FIXED BINARY;
         DECLARE  COPY FIXED BINARY;
         DECLARE REJSW CHARACTER (2), FD CHARACTER(1);
         DECLARE SG CHARACTER(1);
         DECLARE 1 SUMS,(2 SX,    2 SY,      2 SXR,     2 SYR,     2 SE,
                          2 SX2,   2 SY2,     2 SX2R,    2 SY2R,    2 SN,
                          2 SX3,   2 SY3,     2 SX3R,    2 SY3R,    2 SXE,
                          2 SX4,   2 SY4,     2 SXYR,    2 SXY2R,   2 SYE,
                          2 SXY ,  2 SX2Y2,   2 SX2YR,   2 SY2R2,   2 SXN,
                          2 SX2Y,  2 SXY2,    2 SX2R2,   2 SX2E,    2 SYN,
                          2 SX3Y,  2 SXY3,    2 SXRE,    2 SYRN,    2 SXYE,
                                                         2 SY2N,    2 SXYN)
                                                       FLOAT DECIMAL (15);
         DECLARE    (AS(75 ),  N(74),    XI(5),    WX(6,6),  WXV(6),
                     DS(75 ),  X(74),    YI(5),               XTX(6,6),
                    E(74),    Y(74))    FLOAT DECIMAL(15);
         DECLARE    (AC,  ADF, ASM, CADF,CDC, CDDF,CDS, D,   DA,  DC,
                     DDF, DM,  DSM, EC,  FNR, NC,  R,   RA,  RD,
                    SDC,  SDDF,XD,  X2,  X3,  YD,  Y2,  Y3,   SADF,
                    RFM,       RMSRA,    RMSRD)    FLOAT DECIMAL(15);
         DECLARE (IN,IM) FIXED BINARY (30);
         OPEN FILE(PNCH);
         IN=6;
         IM=1;
         SUMS=0.0E0;
         DAYE=DATE;
         RMSRA=0.0E00;          RMSRD=0.0E00;
         NCD=0;
         NRJ=0;
         K=0;
         CDC=COS(DLC);
         SDC=SIN(DLC);
GETSTAR:NCD=NCD+1;
         GET EDIT (STAR(NCD),(XI(I),YI(I)DO I=1 TO 5)) (X(7),F(3),
                                                                10F(7,3));
         IF STAR(NCD)>74  THEN DO;
                                   NCD=NCD-1;
                                   FNR=0.0;
                                   GO TO GETCAT;
                                   END;
         K=STAR(NCD);
         CODE(K)='M';
         X(K)=0.0E0;
         Y(K)=0.0E0;
         DO I=1 TO NR;
         X(K)=X(K)+XI(I);
         Y(K)=Y(K)+YI(I);
         END;
         FNR=NR;
         IF NR>3 THEN DO;
                         J=1;
                         DM=0.0;
                         DO I=1 TO NR;
                         XD=ABS(FNR*XI(I)-X(K));
                         IF XD>DM THEN DO;
                                         DM=XD;
                                         J=I;
                                         END;
                         YD=ABS(FNR*YI(I)-Y(K));
                         IF YD>DM THEN DO;
                                         DM=YD;
                                         J=I;
                                         END;
                         END;
                         X(K)=(X(K)-XI(J))/(FNR-1.0E00);
                         Y(K)=(Y(K)-YI(J))/(FNR-1.0E00);
                         END;
                    ELSE DO;
                         X(K)=X(K)/FNR;
                         Y(K)=Y(K)/FNR;
                         END;
         X(K)=X(K)-XC;
         Y(K)=Y(K)-YC;
         GO TO GETSTAR;
GETCAT:  GET EDIT (STR,IDENT(STR),AS(STR),DS(STR),CODE(STR))(X(7),F(3),
                  X(2),A(8),F(10,8),X(1),E(10,8),X(38),A(1));
         IF STR>74  THEN GO TO ITRATE;
         DO K=1 TO NCD;
         IF STR=STAR(K) THEN GO TO MATCH;
         END;
         GO TO GETCAT;
 MATCH:  K=STR;
         FNR=FNR+1.0;
         DDF=DS(K)-DLC;
         ADF=AS(K)-ALC;
         CDS=COS(DS(K));
         SDDF=SIN(DDF);
         CDDF=COS(DDF);
         CADF=COS(ADF);
         D=CDDF+CDC*CDS*(CADF-1.0E0);
         SADF=SIN(ADF);
         E(K)=(CDS*SADF)/D;
         N(K)=(SDDF-SDC*CDS*(CADF-1.0E0))/D;
         X2=X(K)*X(K);
         Y2=Y(K)*Y(K);
         X3=X2*X(K);
         Y3=Y2*Y(K);
         R=X2+Y2;
         SX=SX+X(K);
         SY=SY+Y(K);
         SX2=SX2+X2;
         SY2=SY2+Y2;
         SX3=SX3+X3;
         SY3=SY3+Y3;
         SX4=SX4+X2*X2;
         SY4=SY4+Y2*Y2;
         SXY=SXY+X(K)*Y(K);
         SX2Y=SX2Y+X2*Y(K);
         SXY2=SXY2+X(K)*Y2;
         SX2Y2=SX2Y2+X2*Y2;
         SX3Y=SX3Y+X3*Y(K);
         SXY3=SXY3+X(K)*Y3;
         SXR=SXR+X(K)*R;
         SYR=SYR+Y(K)*R;
         SX2R=SX2R+X2*R;
         SY2R=SY2R+Y2*R;
         SX3R=SX3R+X3*R;
         SY3R=SY3R+Y3*R;
         SXYR=SXYR+X(K)*Y(K)*R;
         SX2YR=SX2YR+X2*Y(K)*R;
         SXY2R=SXY2R+X(K)*Y2*R;
         SX2R2=SX2R2+X2*R*R;
         SY2R2=SY2R2+Y2*R*R;
         SE=SE+E(K);
         SN=SN+N(K);
         SXE=SXE+X(K)*E(K);
         SYE=SYE+Y(K)*E(K);
         SXN=SXN+X(K)*N(K);
         SYN=SYN+Y(K)*N(K);
         SXYE=SXYE+X(K)*Y(K)*E(K);
         SXYN=SXYN+X(K)*Y(K)*N(K);
         SX2E=SX2E+X2*E(K);
         SY2N=SY2N+Y2*N(K);
         SXRE=SXRE+X(K)*R*E(K);
         SYRN=SYRN+Y(K)*R*N(K);
         CODE(K)=' ';
         GO TO GETCAT;
ITRATE:  NSOL=0;
MATRIX:  XTX(1,1)=FNR;
         XTX(1,2)=SX;
```

```
        XTX(1,3)=SY;
        XTX(1,4)=SXY;
        XTX(1,5)=SX2;
        XTX(1,6)=SXR;
        XTX(2,2)=SX2;
        XTX(2,3)=SXY;
        XTX(2,4)=SX2Y;
        XTX(2,5)=SX3;
        XTX(2,6)=SX2R;
        XTX(3,3)=SY2;
        XTX(3,4)=SXY2;
        XTX(3,5)=SX2Y;
        XTX(3,6)=SXYR;
        XTX(4,4)=SX2Y2;
        XTX(4,5)=SX3Y;
        XTX(4,6)=SX2YR;
        XTX(5,5)=SX4;
        XTX(5,6)=SX3R;
        XTX(6,6)=SX2R2;
        DO J=1 TO 5;
        DO L=J+1 TO 6;
        XTX(L,J)=XTX(J,L);
        END;
        END;
        WX=XTX;
        WXV(1)=SE;
        WXV(2)=SXE;
        WXV(3)=SYE;
        WXV(4)=SXYE;
        WXV(5)=SX2E;
        WXV(6)=SXRE;
        CALL LINKPF(DSIMQ,WX,WXV,IN,IM);
        PC=WXV;
        XTX(1,5)=SY2;
        XTX(1,6)=SYR;
        XTX(2,5)=SXY2;
        XTX(2,6)=SXYR;
        XTX(3,5)=SY3;
        XTX(3,6)=SY2R;
        XTX(4,5)=SXY3;
        XTX(4,6)=SXY2R;
        XTX(5,5)=SY4;
        XTX(5,6)=SY3R;
        XTX(6,6)=SY2R2;
        DO J=1 TO 5;
        DO L=5 TO 6;
        XTX(L,J)=XTX(J,L);
        END;
        END;
        WX=XTX;
        WXV(1)=SN;
        WXV(2)=SXN;
        WXV(3)=SYN;
        WXV(4)=SXYN;
        WXV(5)=SY2N;
        WXV(6)=SYRN;
        CALL LINKPF(DSIMQ,WX,WXV,IN,IM);
        PCP=WXV;
        NSOL=NSOL+1; COPY=1;
PLATOUT:PUT FILE(LST)EDIT ('SOLUTION NO.',NSOL,'STATION',STA,'PLATE',
                PLT,TYPE,MO,'/',DY,'/',YR)(PAGE,X(37),A,F(2),X(4),A,
                X(1),A(4),X(4),A,X(1),A(3),X(2),A(10),X(29),A(2),
                2(A,A(2)));
        PUT FILE(LST)EDIT ('PLATE CONSTANTS',PC,PCP)(SKIP,SKIP,A(15),
                  2(SKIP,X(10),6(X(3),E(15,8))));
        PUT FILE(LST)EDIT ('STAR NO.','CATALOG NO.','RIGHT ASCENSION',
                 'DECLINATION','RESIDUAL A','RESIDUAL D ','CODE')
                (SKIP,SKIP,A(8),X(4),A(11),X(4),A(15),X(6),A(11),
                 X(6),A(10),X(4),A(10),X(5),A(4));
 COPY2: KR=0;/*KR KEEPS THE INDEX OF THE WORST RESIDUAL  */
        KS=1; /* KS COUNTS THE NUMBER OF STARS THAT HAVE BEEN DONE */
        REJSW='  ';  /* THIS SWITCH IS TURNED 'ON' IF ANY REJECTS
                      SHOW UP IN THIS SOLUTION  */
        KK=0;  /*KK COUNTS THE REJECTS FROM THIS SOLUTION  */
        NRET=0;    /* NRET COUNTS REJECTS PUT BACK IN       */
        RFM=0.0;  /* RFM IS THE LARGEST RESIDUAL   */
        RMSRA=0.0;       RMSRD=0.0;
RESID:  K=STAR(KS);
        IF CODE(K)='M' THEN DO;
                            KS=KS+1;
                            IF KS>NCD THEN GO TO TSTEND;
                            GO TO RESID;
                            END;
        X2=X(K)*X(K);    X3=X(K)*X2;
        Y2=Y(K)*Y(K); Y3=Y(K)*Y2;
        R=X2+Y2;
        EC=PC(1)+PC(2)*X(K)+PC(3)*Y(K)+PC(4)*X(K)*Y(K)+PC(5)*X2+PC(6)*
                X(K)*R;
        NC=PCP(1)+PCP(2)*X(K)+PCP(3)*Y(K)+PCP(4)*X(K)*Y(K)+PCP(5)*Y2+
                PCP(6)*Y(K)*R;
        DA=ATAN(EC/(CDC-NC*SDC));
        IF SIGN((EC*SIN(DS(K)))/(CDC*CDS*(NC+(SDC/CDC))))¬=SIGN(DA)
                      THEN DA=DA+3.141592653589;
        AC=ALC+DA;
        ADF=(SDC+NC*CDC)*COS(DA);
        DDF=(CDC-NC*SDC);
        DC=ATAN(ADF/DDF);
        IF ABS(AC-AS(K))>1.04720 THEN IF AC>AS(K) THEN AS(K)=AS(K)+
                                                           6.2831853;
                                                  ELSE AC=AC+
                                                           6.2831853;

        RA=(AC-AS(K))*COS(DS(K));
        RD=DC-DS(K);
        RA=RA*2.0626480625E05;
        RD=RD*2.0626480625E05;
        ASM=(AS(K)*5.7295779513E01)/1.5E01;
        IAS(1)=ASM;
        IAS(2)=(ASM-IAS(1))*6.0E01;
        ASM=((ASM-IAS(1))*6.0E01-IAS(2))*6.0E01;
        DSM=ABS(DS(K))*5.7295779513E01;
        IDS(1)=DSM;
        IDS(2)=(DSM-IDS(1))*6.0E01;
        DSM=((DSM-IDS(1))*6.0E01-IDS(2))*6.0E01;
        IDS(1)=IDS(1)*SIGN(DS(K));
        IF IDS(1)=0 & SIGN(DS(K))<0.0 THEN SG='-';
                                   ELSE SG=' ';
        IF CODE(K)¬='R' THEN GO TO REGLN;
        KK=KK+1;
        IF KK>8 THEN DO;
                  PUT FILE(LST)EDIT('MORE THAN 8 REJECTS FROM THIS'
                            ,' SOLUTION')(SKIP,2A);
                    GO TO ERROUT;
                    END;
       IF SG='-' THEN PUT STRING(RJOUT(KK))EDIT(K,IDENT(K),IAS,ASM,SG
                                 ,IDS,DSM,RA,RD)(F(3),X(8),A(8),X(6),
                                 F(3),X(1),F(2),X(1),F(6,3),X(7),
                                 A(1),F(1),X(1),F(2),X(1),F(6,3),
                                 X(4),F(9,3),X(5),F(9,3));
                 ELSE
       PUT STRING (RJOUT(KK))EDIT(K,IDENT(K),IAS,ASM,IDS,DSM,RA,RD)
                  (F(3),X(8),A(8),2(X(6),F(3),X(1),F(2),X(1),F(6,3)),
                   X(4),F(9,3),X(5),F(9,3));
       IF ABS(RA)<MR & ABS(RD)<MR THEN DO;
                                       CODE(K)=' ';
                                       NRET=NRET+1;
                                       NG(NRET)=K;
                                       FNR=FNR+1.0;
                                       SX=SX+X(K);
                                       SY=SY+Y(K);
                                       SX2=SX2+X2;
                                       SY2=SY2+Y2;
                                       SX3=SX3+X3;
                                       SY3=SY3+Y3;
                                       SX4=SX4+X2*X2;
                                       SY4=SY4+Y2*Y2;
                                       SXY=SXY+X(K)*Y(K);
                                       SX2Y=SX2Y+X2*Y(K);
                                       SXY2=SXY2+X(K)*Y2;
                                       SX3Y=SX3Y+X3*Y(K);
                                       SXY3=SXY3+X(K)*Y3;
                                       SX2Y2=SX2Y2+X2*Y2;
                                       SXR=SXR+X(K)*R;
                                       SYR=SYR+Y(K)*R;
                                       SX2R=SX2R+X2*R;
                                       SY2R=SY2R+Y2*R;
                                       SX3R=SX3R+X3*R;
                                       SY3R=SY3R+Y3*R;
                                       SXYR=SXYR+X(K)*Y(K)*R;
                                       SX2YR=SX2YR+X2*Y(K)*R;
                                       SXY2R=SXY2R+X(K)*Y2*R;
                                       SX2R2=SX2R2+X2*R*R;
                                       SY2R2=SY2R2+Y2*R*R;
                                       SE=SE+E(K);
                                       SN=SN+N(K);
                                       SXE=SXE+X(K)*E(K);
                                       SYE=SYE+Y(K)*E(K);
                                       SXN=SXN+X(K)*N(K);
                                       SYN=SYN+Y(K)*N(K);
                                       SXYE=SXYE+X(K)*Y(K)*E(K);
                                       SXYN=SXYN+X(K)*Y(K)*N(K);
                                       SX2E=SX2E+X2*E(K);
                                       SY2N=SY2N+Y2*E(K);
                                       SXRE=SXRE+X(K)*R*E(K);
                                       SYRN=SYRN+Y(K)*R*N(K);
                                       END;
       KS=KS+1;
       IF KS>NCD THEN GO TO TSTEND;
                 ELSE GO TO RESID;
REGLN: IF SG='-' THEN PUT FILE(LST)EDIT(K,IDENT(K),IAS,ASM,SG,IDS,DSM
                           ,RA,RD)(SKIP,X(3),F(3),X(8),A(8),X(6),F(3)
```

```
                            ,X(1),F(2),X(1),F(6,3),X(7),A(1),F(1),X(1)
                           ,F(2),X(1),F(6,3),X(4),F(9,3),X(5),F(9,3));
                  ELSE
        PUT FILE(LST)EDIT(K,IDENT(K),IAS,ASM,IDS,DSM,RA,RD)(SKIP,X(3),
                 F(3),X(8),A(8),2(X(6),2F(3),F(7,3)),
                 X(4),F(9,3),X(5),F(9,3));
        RMSRA=RMSRA+RA*RA;          RMSRD=RMSRD+RD*RD;
        IF ABS(RA)>MR|ABS(RD)>MR THEN DO;
                                      REJSW='ON';
                                      IF ABS(RA)>RFM THEN DO;
                                                          RFM=ABS(RA);
                                                          KR=K;
                                                          END;
                                      IF ABS(RD)>RFM THEN DO;
                                                          RFM=ABS(RD);
                                                          KR=K;
                                                          END;
                                      END;
        KS=KS+1;
        IF KS<=NCD THEN GO TO RESID;
TSTEND: IF KK>0 THEN DO K=1 TO KK;
                     PUT FILE(LST)EDIT(RJOUT(K),'REJECT')(SKIP,X(3),
                             A(84),X(6),A);
                     END;
        RMSRA=SQRT(RMSRA/FNR);
        RMSRD=SQRT(RMSRD/FNR);
        PUT FILE(LST)EDIT('ROOT MEAN SQUARES:',RMSRA,RMSRD)(SKIP(2),
                     A,X(49),F(6,3),X(8),F(6,3));
        IF COPY=2 THEN GO TO RETURN;
        IF REJSW='  ' THEN DO;
                           COPY=2;
                           IF NRET¬=0 THEN DO K=1 TO NRET;
                                           J=NG(K);
                                           CODE(J)=' ';
                                           END;
                           GO TO PLATOUT;
                           END;
        CODE(KR)='R';
        K=KR;
        X2=X(K)*X(K);
        Y2=Y(K)*Y(K);
        X3=X2*X(K);
        Y3=Y2*Y(K);
        R=X2+Y2;
        FNR=FNR-1.0;
        SX=SX-X(K);
        SY=SY-Y(K);
        SX2=SX2-X2;
        SY2=SY2-Y2;
        SX3=SX3-X3;
        SY3=SY3-Y3;
        SX4=SX4-X2*X2;
        SY4=SY4-Y2*Y2;
        SXY=SXY-X(K)*Y(K);
        SX2Y=SX2Y-X2*Y(K);
        SXY2=SXY2-X(K)*Y2;
        SX2Y2=SX2Y2-X2*Y2;
        SX3Y=SX3Y-X3*Y(K);
        SXY3=SXY3-X(K)*Y3;
        SXR=SXR-X(K)*R;
        SYR=SYR-Y(K)*R;
        SX2R=SX2R-X2*R;
        SY2R=SY2R-Y2*R;
        SX3R=SX3R-X3*R;
        SY3R=SY3R-Y3*R;
        SXYR=SXYR-X(K)*Y(K)*R;
        SX2YR=SX2YR-X2*Y(K)*R;
        SXY2R=SXY2R-X(K)*Y2*R;
        SX2R2=SX2R2-X2*R*R;
        SY2R2=SY2R2-Y2*R*R;
        SE=SE-E(K);
        SN=SN-N(K);
        SXE=SXE-X(K)*E(K);
        SYE=SYE-Y(K)*E(K);
        SXN=SXN-X(K)*N(K);
        SYN=SYN-Y(K)*N(K);
        SXYE=SXYE-X(K)*Y(K)*E(K);
        SXYN=SXYN-X(K)*Y(K)*N(K);
        SX2E=SX2E-X2*E(K);
        SY2N=SY2N-Y2*N(K);
        SXRE=SXRE-X(K)*R*E(K);
        SYRN=SYRN-Y(K)*R*N(K);
        GO TO MATRIX;
 CNVOUT: PUT FILE(LST)EDIT('CONVERSION ERROR IN STAR SOLUTION DATA')
                  (SKIP,A);
 ERROUT: PCP=9999999.9;
        CLOSE FILE(PNCH);
        RETURN;
 RETURN: PUT FILE(LST)EDIT ('SOLUTION NUMBER ',NSOL,
                  ' MEETS THE REQUIREMENTS')(SKIP,A(16),F(2),A(23));
        PUT FILE (PNCH) EDIT (PC)(6E(13,0,8));
        PUT FILE (PNCH) EDIT (PCP)(6E(13,0,8));
        CLOSE FILE(PNCH);
        RETURN;
        END;
/*   END OF DATA
  INCLUDE ILFDSIMQ
  INCLUDE AAPC6A
  PHASE DR01J03,DR01J02
// EXEC PL/I
* PROCESS LISTO,NOOPT,STMT
 MTRACK: PROCEDURE (PC,PCP,PACT);
         DECLARE CARD FILE STREAM INPUT ENVIRONMENT (F(80) MEDIUM
                          (SYSIPT,2540)),
                  DIROUT FILE RECORD OUTPUT ENVIRONMENT (F(3200,80)
                             MEDIUM (SYS019,2314)),
                  DIRIN  FILE RECORD INPUT  ENVIRONMENT (F(3200,80)
                             MEDIUM (SYS019,2314)),
                 FINE FILE RECORD OUTPUT ENVIRONMENT (F(3400,136)MEDIUM
                      (SYS018,2314)),
                 AMB  FILE RECORD OUTPUT ENVIRONMENT (F(1680,56) MEDIUM
                      (SYS018,2314)),
                 LST FILE PRINT ENVIRONMENT(F(130)MEDIUM(SYSLST,1403));
         DECLARE STA CHARACTER (4) EXTERNAL,
                 PLT CHARACTER (3) EXTERNAL,
                (XC,YC,ALC,DLC,EWC,NSC,VC) FLOAT DECIMAL(15) EXTERNAL,
                 NR FIXED BINARY EXTERNAL, TYPE CHARACTER(10) EXTERNAL,
                 1 DAY DEFINED DAYE,(2 YR,2 MO,2 DA)CHARACTER (2);
         DECLARE SG CHARACTER(1);
         DECLARE DAYE CHARACTER(6);
```

```
      DECLARE (PC(6),PCP(6)) FLOAT DECIMAL (15);
      DECLARE 1 FINDATA,2 UT1(3) FLOAT DECIMAL (6),
                        2 NIL CHARACTER(4),
                       (2 AC,2 BC,2 AM,2 BM,2 C,2 R2, 2 DAO,2 DBO,
                        2 NA,2 NB) FLOAT DECIMAL (15),
                        2 STN CHARACTER (4),2 PT CHARACTER (3),
                        2 CDE CHARACTER (1),
                        2 IDLE(4) FLOAT DECIMAL (15),
              1 AMBDATA,2 UT2(3) FLOAT DECIMAL,
                        2 NL CHARACTER(4),
                       (2 ACM,2 ACC,2 BCM,2 BCC) FLOAT DECIMAL(15),
                        2 STT CHARACTER(4),2 PLTT CHARACTER (3),
                        2 COD CHARACTER(1),
              1 DIRDATA,2 IUT(2) FIXED BINARY,2 UT FLOAT DECIMAL,
                        2 IAL(2) FIXED BINARY,2 ALM FLOAT DECIMAL,
                        2 IDL(2) FIXED BINARY,2 DLM FLOAT DECIMAL,
                        2 BLANK CHARACTER (4),
                       (2 A,2 EP,2 DRCO (3)) FLOAT DECIMAL (15);
DECLARE VPES(91) DECIMAL FLOAT(6)   INITIAL
(0.169,0.187,0.210,0.232,0.259,0.286,0.317,0.351,0.389,0.430,0.476,
0.526,0.580,0.640,0.705,0.776,0.854,0.939,1.031,1.132,1.241,1.361,
1.490,1.632,1.785,1.950,2.131,2.326,2.537,2.765,3.013,3.280,3.568,
3.880,4.217,4.579,4.926,5.294,5.685,6.101,6.543,7.013,7.513,8.045,
8.609,9.209,9.844,10.518,11.231,11.987,12.788,13.634,14.530,15.477,
16.477,17.535,18.650,19.827,21.068,22.377,23.756,25.209,26.739,
28.349,30.043,31.824,33.695,35.663,37.729,39.898,42.175,44.563,
47.067,49.692,52.442,55.324,58.340,61.500,64.800,68.260,71.880,
75.650,79.600,83.710,88.020,92.510,97.220,102.09,107.20,112.51,
118.04);
      DECLARE   (PI,  CDL, SDL, F,   P,   STG, LON, LAT, MC,  NNO,
                 MUB, UTN, UTO, RL,  HS,  HA,  HD,  TEMP,PRES,
                 RLH, STP, RATIO,    ES,  NS,  DN,  ENO, CLA,
                 SLA, RF,  XM,  YM,  FNR, DM,  DF,  X2,  Y2,
                 R,   E,   N,   AL,  DL,  H,   CH,  SH,  CD,  HPI,
                 SD,  EN,  NN,  EO,  NO,  CC,  SC,  E1,  N1,  ATEN,
                 R2U, LAO, MUU, NUO, AR,  RBS, AOU, BOO, GAM,
                 AON, BON, FEW, FNS, VEW, VNS, ASV, BSV, R2S, UTSO,
                 XA,  YA,  ZA,  XX,  YY,  RP,  LA,  MU,  NU,  VVL,
                 RA,  UTST,UST, AMO, BMO, DLTA) FLOAT DECIMAL (15);
      DECLARE (COPY,IUTO(2)) FIXED BINARY;
      DECLARE   (SA(3),     LO(3),    LT(3),     BSLN(6),  TC(8),
                            NP(6),    X(5),      Y(5),     AUP(4),
                 AUPO(4),   NPO(6),   ACB(4))FLOAT DECIMAL (15);
      DECLARE STRTC(4) CHARACTER(80);
      DECLARE (DUM,PACT)CHARACTER(4);
      DECLARE( IST,INUT) FIXED BINARY(31);
      DECLARE (UTRD,ROUND) DECIMAL FLOAT(15);
      OPEN FILE(CARD),                FILE (DIROUT);
      ON ENDFILE(CARD)GO TO GOBACK;
      ON CONVERSION GO TO FLUSH;
      DAYE=DATE;
      ROUND=0.00005;
      PT=PLT; PLTT=PLT;
      STN=STA; STT=STA;
      COD='3'; CDE='1';
      GAM=3.2007048E-05;
      PI=3.14159265359E00;
      VVL=9.83569229E08;
      KTR=0;  /* THIS INDICATES A STARTING POINT IN DATA */
      COPY=1;
```

```
        CDL=COS(DLC);
        SDL=SIN(DLC);
STCD1:  GET FILE(CARD)EDIT(STRTC)(4A(80));
        GET STRING(STRTC(1))EDIT(SA,LO,LT,F,P,DUM)(X(10),2(F(3),F(2),
                            F(5,3)),2F(3),F(6,3),X(4),2F(8,3),X(14),A(4));
        STG=SA(3)/3.6E03;
        STG=STG+SA(2)/6.0E01;
        STG=(STG+SA(1))*0.26179938779E00;
        LON=LO(3)/3.6E03;
        LON=LON+LO(2)/6.0E01;
        LON= (LON+ABS(LO(1)))*1.745329252E-02;
        IF LO(1)<0.0E0 THEN LON=-LON;
        LAT=LT(3)/3.6E03;
        LAT=LAT+LT(2)/6.0E01;
        LAT= (LAT+LT(1))*1.745329252E-02;
        CLA=COS(LAT);
        SLA=SIN(LAT);
STCD2:  GET STRING(STRTC(2))EDIT (MC,MUB,VC,EWC,NSC,HS,HA)(X(10),
                           F(10,4),F(10,8),3F(10,7),2F(10,2));
        HD=HA-HS; RBS=MC*HD*1.0E06/VVL;
 REG1:  RF=(1.0-EXP(-GAM*HD))/(GAM*HD);
STCD3:  GET STRING(STRTC(3))EDIT (BSLN,TC)   (X(4),6E(6,3),8E(5,2));
        DO I=1 TO 6;
        BSLN(I)=(BSLN(I)*MC*1.0E06)/VVL;
        END;
STCD4:  GET STRING(STRTC(4))EDIT (TEMP,PRES,RLH,DUM)(X(10),F(10,3),
                                2F(10,5) ,X(36),A(4));
        PUT FILE(LST)EDIT('START CARDS:',STRTC)(PAGE,A,4(SKIP(2),A(80)
                  ));
        PRES=PRES/0.02953144;     /* IN HG TO MB   */
        TEMP=(5.0E0/9.0E0)*(TEMP-32.0E0);
        KTP=TEMP;
        STP=KTP;
        IF TEMP>0 THEN DO;
                           IX=KTP+35;
                           RATIO=ABS(TEMP-STP);
                           END;
                  ELSE DO;
                           IX=KTP+36;
                           RATIO=1.0-ABS(TEMP-STP);
                           END;
        ES=VPES(IX)+RATIO*(VPES(IX+1)-VPES(IX));
        ES=ES*0.0393701;       /* MM TO INCHES HG   */
        ES=ES/0.02953144;      /* INCHES HG TO MB */
        TEMP=TEMP+2.7316E02;
        IF PACT='PACT' THEN RL=0.0;
                           ELSE RL=(77.6*PRES*RF*1.0E-06)/TEMP;
        NS=(7.76E01/TEMP)*(PRES+(4.810E03*ES*RLH/TEMP))*1.0E-06;
        DN= -7.32E0*1.0E-06*EXP(5577.0*NS);
        DN=LOG(NS/(NS+DN))*HD*3.048E-04;
        MUB=1.0E0+NS*(1.0E0-EXP(-DN))/DN;
        DO I=1 TO 6;
        NP(I)=(1.0E0-RL)*MUB*BSLN(I);
        END;
HEAD1:  PUT FILE (LST)  EDIT ('DIRECTION DATA FOR STATION ',STA,
               ', PLATE NUMBER ',PLT,TYPE,MO,'/',DA,'/',YR)(PAGE,X(33)
               ,A,A(4),A,A(3),X(2),A(10),X(25),A(2),2(A,A(2)));
        IF PACT='PACT'  THEN GO TO PCAPT;
                            ELSE GO TO RCAPT;
PCAPT:  PUT FILE (LST)   EDIT ('FLASH TIME','AZIMUTH','ELEVATION',
                  'LAMBDA','MU','NU')(SKIP(2),X(4),A(10),X(7),A(7),
                   X(7) ,A(9),X(8),A(6),X(11),A(2),X(13),A(2));
       LNKT=3;
       IF COPY>2 THEN GO TO BOTH2;
                 ELSE IF COPY=2 THEN GO TO RDDIR;
       GO TO BOTH1;
RCAPT: PUT FILE (LST)   EDIT ('FLASH TIME','RIGHT ASCENSION',
                'DECLINATION','AZIMUTH','ELEVATION','LAMBDA','MU',
                'NU')(SKIP(2),X(2),A(10),X(4),A(15),X(3),A(11),X(7),
                 A(7),X(7),A(9),X(8),A(6),X(11),A(2),X(13),A(2));
       LNKT=3;
       IF COPY>2 THEN GO TO BOTH2;
                 ELSE IF COPY=2 THEN GO TO RDDIR;
BOTH1: GET FILE(CARD)EDIT(AM,BM,IUT,UT,DUM)(X(10),2E(10,3),X(10),
                       2F(2),F(6,4),X(26),A(4));
       COPY=4;
       KTR=KTR+1;
       UTN= UT/3.6E03;
       UTN=UTN+IUT(2)/6.0E01;
       UTN=UTN+IUT(1);
       H=(9.8565*UTN)/3.6E03;
       UTN=UTN*0.26179938779;
       IF PACT¬='PACT' THEN GO TO REG2;
       GET FILE(CARD)EDIT(DRCO,DUM)(X(10),3 F(10,9),X(36),A(4));
       A=ATAN(DRCO(2),DRCO(1));
       EP=SQRT(1.0E0-DRCO(3)**2);
       EP=ATAN(DRCO(3),EP);
       UTRD=UT;
       PUT FILE (LST) EDIT (IUT,UTRD,A,EP,DRCO)(SKIP,X(2),2(F(2),X(1)
                                ),F(7,4),5(X(4),F(11,8)));
       LNKT=LNKT+1;
       IF LNKT>39 THEN GO TO HEAD1;
       GO TO BOTH2;
 REG2: GET FILE(CARD)EDIT((X(I),Y(I) DO I=1 TO 5)) (X(10),10E(7,3));
       XM=0.0E0;
       YM=0.0E0;
       DO I=1 TO NR;
       XM=XM+X(I);
       YM=YM+Y(I);
       END;
       FNR=NR;
       XM=XM/FNR;
       YM=YM/FNR;
       IF NR>3 THEN DO;
                    K=1;
                    DM=0.0E00;
                    DO I=1 TO NR;
                    DF=ABS( X(I)-XM);
                    IF DF>DM THEN DO;
                                  DM=DF;
                                  END;
                    DF=ABS(Y(I)-YM);
                    IF DF>DM THEN DO;
                                  DM=DF ;
                                  K=I;
                                  END;
                     END;
                     XM=(XM*FNR-X(K))/(FNR-1.0E0);
                     YM=(YM*FNR-Y(K))/(FNR-1.0E0);
                     END;
```

```
        XM=XM-XC;
        YM=YM-YC;
        X2=XM*XM; Y2=YM*YM; R=K2+Y2;
        E=PC(1)+PC(2)*XM+PC(3)*YM+PC(4)*XM*YM+PC(5)*X2+PC(6)*XM*R;
        N=PCP(1)+PCP(2)*XM+PCP(3)*YM+PCP(4)*XM*YM+PCP(5)*Y2+PCP(6)*
                                                           YM*R;
        XM=CDL-N*SDL;
        DF=ATAN(E/XM);
        YM=(SDL+N*CDL)*COS(DF);
        DL=ATAN(YM/XM);
        IF  ABS(DL-DLC)>(PI/3.0) THEN DO;
                                 DL=-DL;
                                 DF=DF+PI;
                                 END;
        AL=ALC+DF;
        IF  AL<0.0  THEN  AL=PI*2.0+AL;
        H=H*0.26179938779E00;
        H=STG+H-LON-AL+UTN;
        CH=COS(H);
        SH=SIN(H);
        CD=COS(DL);
        SD=SIN(DL);
        DRCO(1)=SD*CLA-CD*SLA*CH;
        DRCO(2)=-CD*SH;
        DRCO(3)=SD*SLA+CD*CLA*CH;
        A=ATAN(DRCO(2),DRCO(1));
        IF  A<0.0  THEN  A=A+2.0*PI;
        EP=SQRT(1.0E0-DRCO(3)**2);
        EP=ATAN(DRCO(3),EP);
        ALM=(AL*5.7295779513E01)/1.5E01;
        IAL(1)=ALM;
        IAL(2)=(ALM-IAL(1))*6.0E01;
        ALM=((ALM-IAL(1))*6.0E01-IAL(2))*6.0E01;
        DLM=ABS(DL)*5.7295779513E01;
        IDL(1)=DLM;
        IDL(2)=(DLM-IDL(1))*6.0E01;
        DLM=((DLM-IDL(1))*6.0E01-IDL(2))*6.0E01;
        IDL(1)=IDL(1)*SIGN(DL);
        UTRO=UT;
        IF  IDL(1)=0 & DL<0.0 THEN PUT FILE(LST)EDIT(IUT,UTRO,IAL,ALM,
                                  '-',IDL,DLM,A,EP,DRCO)(SKIP,2(F(2),
                                 X(1)),F(7,4),X(4),2(F(2),X(1)),F(6,3)
                                  ,X(5),A,F(1),X(1),F(2),X(1),F(6,3),
                                   5(X(4),F(11,8)));
                              ELSE
        PUT  FILE (LST) EDIT(IUT,UTRO,IAL,ALM,IDL,DLM,A,EP,DRCO)
                   (SKIP,2(F(2),X(1)),F(7,4),X(4),2(F(2),X(1)),F(6,3)
                  ,X(4),F(3),X(1),F(2),X(1),F(6,3),5(X(4),F(11,8)));
        LNKT=LNKT+1;
        IF  LNKT>39 THEN GO TO HEAD1;
BOTH2:  WRITE FILE (DIROUT) FROM (DIRDATA);
        EN=DRCO(2)*HD/DRCO(3);
        NN=DRCO(1)*HD/DRCO(3);
        IF  KTR=1 THEN GO TO NEXT;
        IF  UTN-UTO>0.145444104E-03 THEN DO;
                                           KTR=1;
                                           GO TO NEXT; END;
RIGHT:  IF  ABS(NN-NO)<=ABS(EN-EO) THEN DO;
                                        C=ATAN((NN-NO)/(EN-EO));
                                        IF EO>EN THEN C=C+PI;
```

```
                                       ELSE IF C<0.0 THEN
                                           C=C+2.0E0*PI;
                                CEND;
                            ELSE DO;
                                EN0=EN-E0;
                                NN0=NN-N0;
                              ATEN=ATAN(EN0/NN0);
                              HPI=PI*0.5;
                              C=HPI-ATEN;
                              IF NN<N0 THEN C=C+PI;
                              END;
        CC=COS(C);
        SC=SIN(C);
        IF KTR¬=2 THEN GO TO SKIP;
        E1=F*CC-P*SC+E0;
        N1=P*CC+F*SC+N0;
        R20=SQRT(E1*E1+N1*N1+HD*HD);
        LA0=N1/R20;
        MU0=E1/R20;
        AR=RBS/NU0;
        AR=1.0/(4.00*AR*AR);
        A00=(NP0(1)*MU0)/SQRT(1.0E0+AR*BSLN(1)**2*(1.0E0-MU0*MU0));
        B00=(NP0(2)*LA0)/SQRT(1.0E0+AR*BSLN(2)**2*(1.0E0-LA0*LA0));
 SKIP:  E=F*CC-P*SC+EN;
        N=P*CC+F*SC+NN;
        R2=SQRT(E*E+N*N+HD*HD);
        DRCO(1)=N/R2;
        DRCO(2)=E/R2;
        AR=RBS/DRCO(3);
        AR=2.5E-01/(AR*AR);
        A0N=(NP(1)*DRCO(2))/SQRT(1.0E0+AR*BSLN(1)**2*(1.0E0-DRCO(2)**2
                                                                  ));
        B0N=(NP(2)*DRCO(1))/SQRT(1.0E0+AR*BSLN(2)**2*(1.0E0-DRCO(1)**2
                                                                  ));
        DA0=(A0N-A00)/((UTN-UT0)*13750.98709);
        DB0=(B0N-B00)/((UTN-UT0)*13750.98709);
        FEW=DA0*EWC;
        FNS=DB0*NSC;
        IF KTR¬=2 THEN GO TO SAVE1;
        UT1(1)=IUT0(1);
        UT1(2)=IUT0(2);
        UT1(3)=UTS0;
        VEW=-DA0*VC*AM0;
        VNS=-DB0*VC*BM0;
        AC=A00-FEW-VEW;
        BC=B00-FNS-VNS;
        ASV=AM;
        BSV=BM;
        AM=AM0;
        BM=BM0;
        R2S=R2;
        R2=R20;
        NA=NP0(1);
        NB=NP0(2);
        WRITE FILE (FINE) FROM (FINDATA);
        AM=ASV;
        BM=BSV;
        R2=R2S;
SAVE1:  VEW=-DA0*VC*AM;
        VNS=-DB0*VC*BM;
```

```
         UT1(1)=IUT(1);
         UT1(2)=IUT(2);
         UT1(3)=UT;
         AC=AON-FEW-VEW;
         BC=BON-FNS-VNS;
                                         K=I;
         NA=NP(1);
         NB=NP(2);
         WRITE FILE(FINE) FROM (FINDATA);
         XA=HD*DRCO(1)/DRCO(3);
         YA=HD*DRCO(2)/DRCO(3);
         ZA=HD;
         DO I=1 TO 4;
         XX=XA-TC(I);
         YY=YA-TC(I+4);
         RP=SQRT(XX*XX+YY*YY+ZA*ZA);
         LA=XX/RP;
         MU=YY/RP;
         NU=ZA/RP;
         RA=RBS/NU;
         RA=2.5E-01/(RA*RA);
         IF I=1|I=3 THEN GO TO AOC;
                    ELSE GO TO BOC;
    AOC: AOP(I)=(NP(I+2)*MU)/SQRT(1.0E0+RA*BSLN(I+2)**2*(1.0E0-MU*MU));
         GO TO RET1;
    BOC: AOP(I)=(NP(I+2)*LA)/SQRT(1.0E0+RA*BSLN(I+2)**2*(1.0E0-LA*LA));
   RET1: END;
         INUT=UT*10000.0;
         IST=UT;
         IST=IST*10000;
         IF IST≠INUT THEN GO TO COMPAMB;
  GETCN: AOPO=AOP;
         AOO=AON;
         BOO=BON;
         GO TO NORMAL;
   NEXT: AMO=AM;
         BMO=BM;
         IUTO=IUT;
         NUO=DRCO(3);
         NPO=NP;
 NORMAL:UTO=UTN;
         UTSO=UT;
         EO=EN;
         NO=NN;
         GO TO BOTH1;
COMPAMB:IF KTR¬=2 THEN GO TO OKAY;
         XA=HD*LAO/NUO;
         YA=HD*MUO/NUO;
         ZA=HD;
         DO I=1 TO 4;
         XX=XA-TC(I);
         YY=YA-TC(I+4);
         RP=SQRT(XX*XX+YY*YY+ZA*ZA);
         LA=XX/RP;
         MU=YY/RP;
         NU=ZA/RP;
         RA=RBS/NU;
         RA=2.5E-01/(RA*RA);
         IF I=1|I=3 THEN GO TO AOCO;
                    ELSE GO TO BOCO;
   ADCO: AOPO(I)=(NPO(I+2)*MU)/SQRT(1.0E0+RA*BSLN(I+2)**2*(1.0E0-MU*MU)
                                                                    );
         GO TO RET2;
   BOCU: AOPO(I)=(NPO(I+2)*LA)/SQRT(1.0E0+RA*BSLN(I+2)**2*(1.0E0-LA*LA)
                                                                    );
   RET2: END;
    OKAY:DO I=1 TO 4;
         DLTA=(AOP(I)¬AOPO(I))/(UT-UTSO);
         IF I=1|I=3 THEN FEW=EWC*DLTA;
                    ELSE FEW=NSC*DLTA;
         ACB(I)=AOP(I)-FEW;
         END;
         UT2(1)=IUT(1);
         UT2(2)=IUT(2);
         UT2(3)=UT;
         ACM=ACB(3);
         ACC=ACB(1);
         BCM=ACB(4);
         BCC=ACB(2);
         WRITE FILE (AMB) FROM (AMBDATA);
       GO TO GETON;
GOBACK:COPY=2;
       CLOSE FILE(DIROUT);
       OPEN FILE (DIRIN);
       PUT FILE(LST)EDIT('START CARDS:',STRTC)(PAGE,A,4(SKIP(2),A(80)
                 ));
       ON ENDFILE(DIRIN) GO TO FINIS;
       GO TO HEAD1;
RDDIR: READ FILE (DIRIN) INTO (DIRDATA);
       UTRD=UT;
       IF PACT='PACT' THEN PUT FILE (LST)EDIT(IUT,UTRD,A,EP,DRCO)
                     (SKIP,X(2),2(F(2),X(1)),F(7,4),5(X(4),F(11,8)));
                      ELSE IF IDL(1)=0 & DL<0.0 THEN
                                 PUT FILE(LST)EDIT(IUT,UTRD,IAL,ALM,
                                 '-',IDL,DLM,A,EP,DRCO)(SKIP,2(F(2),
                                X(1)),F(7,4),X(4),2(F(2),X(1)),F(6,3)
                                 ,X(5),A,F(1),X(1),F(2),X(1),F(6,3),
                                 5(X(4),F(11,8)));
                                                ELSE
                           PUT FILE (LST) EDIT(IUT,UTRD,IAL,ALM,IDL,
                             DLM,A,EP,DRCO)(SKIP,2(F(2),X(1)),F(7,4),
                             X(4),2(F(2),X(1)),F(6,3),X(4),F(3),X(1),
                             F(2),X(1),F(6,3),5(X(4),F(11,8)));
         LNKT=LNKT+1;
         IF LNKT>39 THEN GO TO HEAD1;
         GO TO RDDIR;
  FLUSH: ON ENDFILE(CARD) GO TO FINIT;
         PUT FILE(LST)EDIT('CONVERSION ERROR---PLEASE CHECK CARDS')
                           (SKIP,A);
 FLSHCD: GET FILE(CARD)EDIT(DUM)(X(76),A(4));
         GO TO FLSHCD;
  FINIT: CLOSE FILE(DIROUT);
  FINIS: CLOSE FILE (DIRIN),             FILE(CARD);
    EXIT:RETURN;
         END;
/*   END OF DATA
 PHASE DR01J04,DR01J02
// EXEC PL/I
* PROCESS NOOPT,LISTO,STMT
 OUTPUT: PROCEDURE (SEQ,RR);
```

```
DECLARE FINE FILE RECORD OUTPUT ENVIRONMENT (F(3400,136)MEDIUM
             (SYS018,2314)),
        AMB  FILE RECORD OUTPUT ENVIRONMENT (F(1680,56) MEDIUM
             (SYS018,2314)),
      FINEIN FILE RECORD  INPUT ENVIRONMENT (F(3400,136)MEDIUM
             (SYS018,2314)),
      AMBIN  FILE RECORD  INPUT ENVIRONMENT (F(1680,56) MEDIUM
             (SYS018,2314)),
      TFINE  FILE RECORD OUTPUT ENVIRONMENT (F(3400,136)MEDIUM
                            (SYS019,2314)),
      TAMB   FILE RECORD OUTPUT ENVIRONMENT (F(1680,56) MEDIUM
                            (SYS019,2314)),
      TFINEI FILE RECORD  INPUT ENVIRONMENT (F(3400,136)MEDIUM
                            (SYS019,2314)),
      TAMBIN FILE RECORD  INPUT ENVIRONMENT (F(1680,56) MEDIUM
                            (SYS019,2314));
DECLARE LST FILE PRINT ENVIRONMENT(F(130) MEDIUM (SYSLST,1403)
                                                             );
DECLARE  1 FINDATA,2 UT1(3) FLOAT DECIMAL,
                    2 NIL CHARACTER(4),
                   (2 AC,2 BC,2 AM,2 BM,2 C,2 RNG,2 DAO,2 DBO,
                    2 NA,2 NB) FLOAT DECIMAL (15),
                    2 ST CHARACTER (4),2 PT CHARACTER (3),
                    2 CD CHARACTER (1),
                    2 FINED(4) FLOAT DECIMAL (15),
         1 AMBDATA,2 UT2(3) FLOAT DECIMAL,
                   2 NL  CHARACTER (4),
                  (2 ACM,2 ACC,2 BCM,2 BCC) FLOAT DECIMAL(15),
                   2 STT CHARACTER (4),2 PLTT CHARACTER(3),
                   2 COD CHARACTER (1);
DECLARE IUT(2) FIXED BINARY;
DECLARE (FEW,FNS,VEW,VNS,PI,UT) FLOAT DECIMAL (15);
DECLARE                      (PLC,PLTO) CHARACTER (3),
RR CHARACTER (7),SEQ CHARACTER(8);
DECLARE NMRR EXTERNAL,MRR(100) CHARACTER (3) EXTERNAL,
        STA CHARACTER(4) EXTERNAL,TYPE CHARACTER(10) EXTERNAL,
        PLT CHARACTER (3) EXTERNAL,
        1 DAY DEFINED DAYE,(2 YR,2 MO,2 DA)CHARACTER (2),
        (EWC,NSC,VC) FLOAT DECIMAL (15) EXTERNAL;
DECLARE DAYE CHARACTER(6);
IF SEQ='ADDITION'& RR='START  ' THEN DO;
                       OPEN FILE(FINEIN),FILE(AMBIN),
                            FILE(TFINE), FILE(TAMB);
                       ON ENDFILE(FINEIN)GO TO AMBT;
                       READ FILE (FINEIN) INTO (FINDATA);
                       IF CD='9' THEN GO TO FINET;
                                 ELSE GO TO WRTFN;
                FINET: READ FILE(FINEIN) INTO (FINDATA);
                WRTFN: WRITE FILE(TFINE) FROM (FINDATA);
                       GO TO FINET;
                 AMBT: ON ENDFILE (AMBIN) GO TO EXIT1;
                 RAMB: READ FILE (AMBIN) INTO (AMBDATA);
                       WRITE FILE(TAMB) FROM (AMBDATA);
                       GO TO RAMB;
                EXIT1: CLOSE FILE(FINEIN),FILE(AMBIN),
                             FILE(TFINE), FILE(TAMB);
                       RETURN;
                                     END;
DAYE=DATE;
PI=3.141592653E00;
```

```
        IF SEQ='INITIAL ' THEN GO TO STPPUT;
        OPEN FILE(TFINEI),FILE(TAMBIN);
        ON ENDFILE(TFINEI) GO TO AMB1;
        ON ENDFILE(TAMBIN) GO TO STPPUT;
  FIN1: READ FILE(TFINEI) INTO(FINDATA);
        IF NMRR>0       THEN DO I=1 TO NMRR;
                             IF PT=MRR(I) THEN GO TO FIN1;
                             END;
        WRITE FILE(FINE) FROM (FINDATA);
        GO TO FIN1;
  AMB1: READ FILE (TAMBIN) INTO (AMBDATA);
        IF NMRR>0       THEN DO I=1 TO NMRR;
                             IF PLTT=MRR(I) THEN GO TO AMB1;
                             END;
        WRITE FILE(AMB) FROM (AMBDATA);
        GO TO AMB1;
STPPUT: CLOSE FILE(FINE),FILE(AMB);
        OPEN FILE(FINEIN),FILE(AMBIN);
        ON ENDFILE(FINEIN) GO TO PRTAMB;
        PLTO='000';
 INFINE:READ FILE(FINEIN) INTO(FINDATA);
        STA=ST;
        PLT=PT;
        IF PLT¬=PLTO | LNKT=40 THEN GO TO HED1;
                                    ELSE GO TO LINF;
  HED1: PUT FILE(LST)EDIT ('FINE ANTENNA DATA FOR STATION ',STA,
               ' PLATE NO.',PLT,TYPE,MO,'/',DA,'/',YR)(PAGE,X(35),
               A,A(4),A,A(3),X(2),A(10),X(27),A(2),2(A,A(2)));
        PUT FILE(LST) EDIT    ('FLASH TIME','AC','BC','AM','BM',
                 'DA/DT','DB/DT','NA','NB','COURSE','RANGE','FEW',
                'FNS','VEW','VNS')(SKIP(2),X(2),A,2(X(6),A),X(5),A,
                2(X(4),A),X(2),A,X(6),A,X(8),A,X(6),A,X(5),A,X(6),
                4(A,X(4)));
        LNKT=3;
  LINF: FEW=DAO*EWC;
        FNS=DBO*NSC;
        VEW=VC*AM*(-DAO);
        VNS=VC*BM*(-DBO);
        C=PI/2.0E0-C;
        IUT(1)=UT1(1);
        IUT(2)=UT1(2);
        UT=UT1(3);
        PUT  FILE (LST) EDIT  (IUT,UT,AC,BC,AM,BM,DAO,DBO,NA,NB,C,RNG,
                FEW, FNS, VEW, VNS)(SKIP,2(F(2),X(1)),F(7,4),X(1),
                2F(8,3),2F(6,3),X(1),2F(7,3),X(1),2F(10,6),F(10,5),
                F(11,2),X(1),4F(7,3));
        PLTO=PLT;
        LNKT=LNKT+1;
        GO TO INFINE;
PRTAMB: ON ENDFILE (AMBIN) GO TO STOP;
        PLTO='000';
INAMB:  READ FILE(AMBIN) INTO (AMBDATA);
        STA=STT;
        PLT=PLTT;
        IF PLT¬=PLTO | LNKT=40 THEN GO TO HED2;
                                   ELSE GO TO LINA;
  HED2: PUT FILE (LST)    EDIT ('AMBIGUITY ANTENNA DATA FOR STATION ',
               STA,' PLATE NO. ',PLT,TYPE,MO,'/',DA,'/',YR)(PAGE,X(4)
               ,A,A(4),A,A(3),X(2),A(10),X(50),A(2),2(A,A(2)));
        PUT FILE (LST)   EDIT ('FLASH TIME','AC(MED)','AC(COURSE)',
                  'BC(MED)','BC(COURSE)')(SKIP(2),X(4),A(10),X(9),A(7),
                   X(5),A(10),X(6),A(7),X(5),A(10));
         LNKT=3;
   LINA: IUT(1)=UT2(1);
         IUT(2)=UT2(2);
         UT=UT2(3);
         PUT FILE (LST)    EDIT (IUT,UT,ACM,ACC,BCM,BCC)(SKIP,X(2),
                           2(F(2),X(1)),F(7,4),4(X(4),F(10,3)));
         PLTO=PLT;
         LNKT=LNKT+1;
         GO TO INAMB;
   STOP: CLOSE FILE (FINEIN),FILE (AMBIN),FILE(LST);
   DONE: RETURN;
         END;
/*   END OF DATA
/&   END OF JOB
```

```
* DR01K;        M.KOPP   AAP      ;14401,DR01K
// OPTION CATAL
   PHASE DR01K,ROOT
// EXEC PL/I
* PROCESS LISTO,NOOPT
 DRZ01K:  PROCEDURE OPTIONS(MAIN);                                     DR0001
               DECLARE                                                 DR0002
          INPUT FILE STREAM INPUT ENVIRONMENT                          DR0003
              (F(80) MEDIUM(SYSIPT,2540)),                             DR0004
          TFINE  FILE RECORD OUTPUT ENVIRONMENT                        DR0005
             (F(3400,136) MEDIUM (SYS019,2314)),
          TFINE1 FILE RECORD INPUT  ENVIRONMENT                        DR0007
             (F(3400,136) MEDIUM (SYS019,2314)),
          M_OUT  FILE RECORD OUTPUT ENVIRONMENT                        DR0009
             (F(1800,72)  MEDIUM (SYS019,2314)),
          M_IN   FILE RECORD INPUT  ENVIRONMENT                        DR0011
             (F(1800,72)  MEDIUM (SYS019,2314)),
          C_OUT  FILE RECORD OUTPUT ENVIRONMENT                        DR00121
             (F(1800,72)  MEDIUM (SYS019,2314)),
          C_IN   FILE RECORD INPUT  ENVIRONMENT                        DR00123
             (F(1800,72)  MEDIUM (SYS019,2314)),
          FINEIN FILE RECORD INPUT  ENVIRONMENT                        DR00125
             (F(3400,136) MEDIUM (SYS018,2314)),
          FINE   FILE RECORD OUTPUT ENVIRONMENT                        DR00127
             (F(3400,136) MEDIUM (SYS018,2314)),
          AMBIN  FILE RECORD INPUT  ENVIRONMENT                        DR00129
             (F(1680,56)  MEDIUM (SYS018,2314)),
          1 FINE_SAVE,                                                 DR0013
           (2 HR,                                                      DR0014
            2 MN,                                                      DR0015
            2 SEC)         FLOAT,                                      DR0016
            2 EX           CHARACTER(4),                               DR00165
           (2 AC,                                                      DR0017
            2 BC,                                                      DR0018
            2 AMQ,                                                     DR0019
            2 BMQ,                                                     DR0020
            2 C,                                                       DR0021
            2 R,                                                       DR0022
            2 DA,                                                      DR0023
            2 DB,                                                      DR00234
            2 NA,                                                      DR00235
            2 NB)          FLOAT(10),                                  DR00236
            2 STA          CHARACTER(4),                               DR00237
            2 F_PLATE      CHARACTER(3),                               DR00238
            2 CODE         CHARACTER(1),                               DR00239
           (2 RA,                                                      DR0023A
            2 RB,                                                      DR0023B
            2 AM,                                                      DR0023C
            2 BM)          FLOAT(10),                                  DR0023D
          1 AMB_SAVE,                                                  DR0024
           (2 AC_AMB,                                                  DR0026
            2 BC_AMB,                                                  DR0027
            2 AM_AMB,                                                  DR0028
            2 BM_AMB)      FLOAT(10),                                  DR0029
           (2 AMBHR,                                                   DR00295
            2 AMBMN)       FIXED(4),                                   DR00296
            2 AMBSC        FIXED(6,3),                                 DR00297
            2 RES2         CHARACTER(20),                              DR0030
            2 CONSTANTS4   CHARACTER(5),                               DR0031
            2 APLATE       CHARACTER(3),                               DR0032
```

```
  2 CONSTANTS5 CHARACTER(2),                              DR0033
 1 AMB_IN,                                                DR00333
  (2 AHR,                                                 DR00334
   2 AMN,                                                 DR00335
   2 ASC)          FLOAT,                                 DR00336
   2 EXT           CHARACTER(4),                          DR003365
  (2 AC_M,                                                DR00337
   2 AC_C,                                                DR00338
   2 BC_M,                                                DR00339
   2 BC_C)         FLOAT(10),                             DR0033A
   2 ASTA          CHARACTER(4),                          DR0033B
   2 PLATE         CHARACTER(3),                          DR0033C
   2 ACODE         CHARACTER(1),                          DR0033D
 BLANKS                                                   DR0034
      CHARACTER(20),                                      DR0035
 PLOT_CODE                                                DR0036
      CHARACTER(1),                                       DR0037
 CDE               FIXED(1),                              DR00373
(HRS,                                                     DR00374
 CORECT,                                                  DR003745
 MIN)                                                     DR00375
      FIXED(2),                                           DR00376
(CPLATE,                                                  DR00377
 DPLATE)           FIXED(3),                              DR00378
 PREV_PLATE                                               DR0038
      CHARACTER(3),                                       DR0039
 SSTA                                                     DR00395
      CHARACTER(4),                                       DR00396
 CONSTANTS1                                               DR0040
      CHARACTER(11),                                      DR0041
 CONSTANTS2                                               DR0042
      CHARACTER(28),                                      DR0043
(EOP,                                                     DR00432
 CDATE,                                                   DR00433
 CORAM(50),             CORBM(50),                        DR004335
 CONSTANTS3)                                              DR00434
      CHARACTER(10),                                      DR00435
 CONSTANTS6                                               DR0044
      CHARACTER(30),                                      DR0045
(CONSTANTS,                                               DR0046
 FREQ)                                                    DR0047
      CHARACTER(10),                                      DR0048
(DEC_A,    DEC_B,    IA,        IB,        DIFF_AC,       DR0049
 DEC_AM,   DEC_BM,   IA_M,      IB_M,      DIFF_BM,       DR0050
 DEC_AC,   DEC_BC,   IA_C,      IB_C,      DIFF_BC,       DR00505
 CHR,          CMN,             CSEC,                     DR0051
 HCD(50),      MCD(50),         SCD(50),                  DR00511
 UH,                 UM,                   US,            DR00512
 KSAM,                    KSBM,                           DR00513
 KSAC,                    KSBC,                           DR00514
 KS1A,                    KS1B)                           DR00515
      FLOAT,                                              DR0052
(FINE_FLAG,                                               DR0053
 AMB_FLAG)                                                DR0054
      BIT(1),                                             DR0055
(AMB_UT,                  NUT,                            DR0056
 UT_S,                                                    DR0057
 AM_M,                    BM_M,                           DR0064
 AM_C,                    BM_C,                           DR0065
 M_QAM,                   M_QBM,                          DR0066
```

```
           C_QAM,                   C_QBM,                      DR0067
           QAM_C,                                               DR0068
           QBM_M,                                               DR0069
           QBM_C,                                               DR0070
           KCA_SUM,                 KCB_SUM,                    DR0072
           KCAM_SUM,                KCBM_SUM,                   DR0073
           KCAC_SUM,                KCBC_SUM,                   DR0074
           SAVE_KCA,                SAVE_KCB,                   DR0075
           SAVE_KCAM,               SAVE_KCBM,                  DR0076
           SAVE_KCAC,               SAVE_KCBC,                  DR0077
           KCA,                     KCB,                        DR0078
           KCAM,                    KCBM,                       DR0079
           KCAC,                    KCBC,                       DR0080
           FINE_KCA,                FINE_KCB,                   DR0081
           AVE_KCAM,                AVE_KCBM,                   DR0082
           AVE_KCAC,                AVE_KCBC,                   DR0083
           VAL_AC,                                              DR0084
           VAL_BM,                                              DR0085
           VAL_BC)                                              DR0086
                FLOAT(10);                                      DR0087
DECLARE KF(6) FIXED BINARY;
      DECLARE                                                   DR0 870
          (CORCT,                                               DR008704
           CT)            BINARY FIXED(15),                     DR008705
           ST(30)         CHARACTER(4),                         DR0 871
           PLT(30)        CHARACTER(3),                         DR0 872
           KSB(30)        FIXED(5,3),                           DR0 873
           KSA(30)        FIXED(5,3),                           DR0 8735
           PC(30)         CHARACTER(1);                         DR0 8736
FORMATF: FORMAT(A(10),4F(10,3),A(19),A(1),A(5),A(3),A(2));      DR00874
FORMATC: FORMAT(A(10),4F(10,3),A(20),A(5),A(3),A(2));           DR00875
CORRECT: PROCEDURE;                                             DR 1*
         IF CORAM(CORCT)=' ' & CORBM(CORCT)=' '                 DR 2*
              THEN DO;                                          DR 3*
                     CORCT=CORCT+1;                             DR 4*
                     CSEC=SCD(CORCT);                           DR 5*
                     CMN=MCD(CORCT);                            DR 6*
                    CHR=HCD(CORCT);                             DR 7*
                    GO TO RDREC;                                DR 8*
                  END;                                          DR 9*
      IF CORAM(CORCT)¬=' '                                      DR10*
           THEN GET STRING(CORAM(CORCT)) EDIT(AMQ) (F(10,3));   DR11*
      IF CORBM(CORCT)¬=' '                                      DR12*
           THEN GET STRING(CORBM(CORCT)) EDIT(BMQ) (F(10,3));   DR13*
      CORCT=CORCT+1;                                            DR14*
      CSEC=SCD(CORCT);    CMN=MCD(CORCT);     CHR=HCD(CORCT);   DR15*
 END;           /* END CORRECT PROCEDURE */                     DR16*
 PRINT:    PROCEDURE;                                           DR0088
 /* CONTINUED PLATE */                                          DR0089
   IF PLTE=1 THEN DO;                                           DR0090
 /* CONTINUED PAGE */                                           DR0091
      IF PAGE=1 THEN DO;                                        DR0092
         PUT EDIT(HRS,' ',MIN,' ',SEC,AC,BC,AM,BM,RA,RB)        DR0093
               (SKIP(1),F(2),A(1),F(2),A(1),F(6,3),             DR0094
                4F(12,3),2F(12));                               DR00945
         LINE=LINE+1;                                           DR0095
                     END;                                       DR0096
 /* NEW PAGE */                                                 DR0097
      ELSE DO;                                                  DR0098
      PUT EDIT('UT','AC','BC','AM','BM','RA','RB','P STA',      DR0099
                  'PLATE')(PAGE,X(6),A(14),3A(12),A(14),A(12),      DR0100
                  A(4),2A(7));                                      DR0101
            PUT EDIT(HRS,' ',MIN,' ',SEC,AC,BC,AM,BM,RA,RB)         DR0102
                  (SKIP(2),F(2),A(1),F(2),A(1),F(6,3),              DR0103
                   4F(12,3),2F(12));                                DR01035
            PAGE=1;     LINE=LINE+3;                                DR0104
              END;                                                  DR0105
           END;                                                     DR0106
    /* NEW PLATE */                                                 DR0107
      ELSE DO;                                                      DR0108
    /* CONTINUED PAGE */                                            DR0109
         IF PAGE=1 THEN DO;                                         DR0110
            PUT EDIT(HRS,' ',MIN,' ',SEC,AC,BC,AM,BM,               DR0111
                     RA,RB,CONSTANTS3)                              DR01115
                  (SKIP(2),F(2),A(1),F(2),A(1),F(6,3),              DR0112
                   4F(12,3),2F(12),X(2),A(12));                     DR01125
            PLTE=1;     LINE=LINE+2;                                DR0113
                        END;                                        DR0114
    /* NEW PAGE */                                                  DR0115
         ELSE DO;                                                   DR0116
         PUT EDIT('UT','AC','BC','AM','BM','RA','RB','P STA',       DR0117
                  'PLATE')(PAGE,X(6),A(14),3A(12),A(14),A(12),      DR0118
                  A(4),2A(7));                                      DR0119
            PUT EDIT(HRS,' ',MIN,' ',SEC,AC,BC,AM,BM,               DR0120
                     RA,RB,CONSTANTS3)                              DR01205
                  (SKIP(2),F(2),A(1),F(2),A(1),F(6,3),              DR0121
                   4F(12,3),2F(12),X(2),A(12));                     DR01215
            PAGE=1;     PLTE=1;    LINE=LINE+3;                     DR0122
              END;                                                  DR0123
                      END;                                          DR0124
END;                                                                DR0125
START:                                                              DR0126
   KF=0;
         ON ENDFILE(INPUT) GO TO FINISH;                            DR0127
    /* READ LEAD CARD */                                            DR0128
         GET FILE(INPUT)EDIT(FINE_FLAG,AMB_FLAG,CURECT,EX)          DR0129
                 (B(1),X(1),B(1),X(1),F(2),A(74));                  DR01295
    /* SAVE START CARD INFORMATION */                               DR0130
         GET FILE(INPUT)EDIT(CODE,SSTA,FREQ,CDATE,NO,EOP,EX)        DR0131
                 (X(1),A(1),X(4),A(4),X(20),2A(10),X(1),F(1),       DR0132
                  X(8),2A(10));                                     DR0133
/* PROCESS FINE DATA */                                             DR0134
         IF ¬FINE_FLAG THEN GO TO AMB;                              DR0135
         OPEN FILE(TFINE),FILE(FINEIN);                             DR01275
         ON ENDFILE (FINEIN)GO TO AVE_FINE;                         DR01276
         ON ENDFILE (INPUT) GO TO REC1;                             DR0136
         KCA_SUM=0;     KCB_SUM=0;     N=0;      J=0;      CT=0;    DR0137
   CURCT=1;  CSEC=999.9;  CMN=999.9; CHR=999.9;
         IF CORECT¬=0 THEN DO I=1 TO CORECT;                        DR01375
            GET FILE(INPUT)EDIT(HCD(I),MCD(I),SCD(I),CORAM(I),      DR01376
                      CORBM(I),EX) (2F(2),F(4,2),X(9),A(3),         DR
                      X(7),A(3),A(50));                             DR013775
         END;                                                       DR01378
LOOP:                                                               DR0138
    CT=CT+1;                                                        DR01381
    GET FILE (INPUT) EDIT (PC(CT),KSA(CT),KSB(CT),ST(CT),PLT(CT))   DR01382
                        (A(1),X(9),2F(5,3),X(49),X(4),A(4),A(3));
    GO TO LOOP;                                                     DR01384
    REC1:                                                           DR013845
       CT=CT-1;
```

```
           CSEC=SCD(1);    CMN=MCD(1);     CHR=HCD(1);     CORCT=1;        DR013846
FSTRD:     READ FILE (FINEIN)INTO (FINE_SAVE);                             DR01385
   IF CODE='9' THEN GO TO FSTRD;
    /* CORRECT DATA VALUES FOR TIME IF NECESSARY */                        DR013854
           IF SEC=CSEC THEN IF MN=CMN THEN IF HR=CHR                       DR013855
                  THEN CALL CORRECT;                                       DR013856
  DSKCHK:                                                                  DR01386
           DO I=1 TO CT;                                                   DR01387
             IF STA=ST(I) & F_PLATE=PLT(I) THEN DO;                        DR0139
              PLOT_CODE=PC(I);                                             DR0140
              KS1A=KSA(I);                                                 DR0141
              KS1B=KSB(I);                                                 DR0142
              PREV_PLATE=F_PLATE;                                          DR0143
              GO TO FINE_PLATE;                                            DR0144
             END;                                                          DR0145
           END;                                                            DR0146
  NOPLT:                                                                   DR0147
    PREV_PLATE=F_PLATE;                                                    DR01471
    DO WHILE (F_PLATE=PREV_PLATE);                                         DR01472
      READ FILE(FINEIN) INTO (FINE_SAVE);                                  DR01473
    END;                                                                   DR01474
    GO TO DSKCHK;                                                          DR01475
    /* EACH PLATE */                                                       DR0148
   FINE_PLATE:                                                             DR0149
         DO WHILE(F_PLATE=PREV_PLATE);                                     DR0150
           IF PLOT_CODE='1' THEN DO;                                       DR1504
              IF ABS(AC)>5 THEN GO TO RDREC;                               DR01505
           END;                                                            DR01506
           ELSE IF PLOT_CODE='2' THEN DO;                                  DR01507
              IF ABS(BC)>5 THEN GO TO RDREC;                               DR01508
           END;                                                            DR01509
    /* AM,BM, AND KC TOTALS */                                             DR0151
           DEC_A=MOD(100+AMQ-AC-KS1A,1);                                   DR0152
           DEC_B=MOD(100+BMQ-BC-KS1B,1);                                   DR0153
    /* IA AND IB FOR FIRST CARD */                                         DR0154
           IF J=0 THEN DO;                                                 DR0155
              IF DEC_A>.850 THEN IA=49;                                    DR0156
                                 ELSE IA=50;                               DR0157
              IF DEC_B>.850 THEN IB=49;                                    DR0158
                                 ELSE IB=50;                               DR0159
                         END;                                              DR0160
    /* FOR OTHER CARDS */                                                  DR0161
           ELSE DO;                                                        DR0162
              IF SAVE_KCA>=50.100 THEN IA=50;                              DR0163
              ELSE IF SAVE_KCA<50.100 & DEC_A<0.500 THEN IA=50;            DR0164
                   ELSE IA=49;                                             DR0165
              IF SAVE_KCB>=50.100 THEN IB=50;                              DR0166
              ELSE IF SAVE_KCB<50.100 & DEC_B<0.500 THEN IB=50;            DR0167
                   ELSE IB=49;                                             DR0168
                END;                                                       DR0169
           KCA=DEC_A+IA;                                                   DR0170
           KCB=DEC_B+IB;                                                   DR0171
    /* SAVE KC'S OF FIRST CARD */                                          DR0172
           IF J=0 THEN DO;                                                 DR0173
              SAVE_KCA=KCA;                                                DR0174
              SAVE_KCB=KCB;                                                DR0175
              J=1;                                                         DR0176
                         END;                                              DR0177
    /* TOTAL KC'S FOR AVERAGE */                                           DR0178
```

```
      KCA_SUM=KCA_SUM+KCA;                                          DR0179
      KCB_SUM=KCB_SUM+KCB;                                          DR0180
      N=N+1;                                                        DR0181
  /* COMPUTE AM AND BM */                                           DR0182
      CODE=PLOT_CODE;                                               DR01825
      AM=KCA+AC;                                                    DR0183
      BM=KCB+BC;                                                    DR0184
      CODE=PLOT_CODE;                                               DR01845
      WRITE FILE(TFINE) FROM(FINE_SAVE);                            DR0185
KOREC:                                                              DR01855
      READ FILE (FINEIN) INTO (FINE_SAVE);                          DR0186
      IF SEC=CSEC THEN IF MN=CMN THEN IF HR=CHR                     DR0187
               THEN CALL CORRECT;                                   DR0188
   END;                                                             DR0190
      GO TO DSKCHK;                                                 DR0197
AVE_FINE:                                                           DR0198
  /* AVERAGE KC'S */                                                DR0199
      FINE_KCA=KCA_SUM/N;                                           DR0200
      FINE_KCB=KCB_SUM/N;                                           DR0201
         CLOSE FILE(TFINE),FILE(FINEIN);                            DR0202
/* COMPUTE RESIDUALS AND LIST FINE DATA */                          DR0203
         OPEN FILE(TFINE1),FILE(FINE);          N=0;                DR0204
      ON ENDFILE(TFINE1) GO TO REPEAT;                              DR0205
FINELBL:                                                            DR0206
  /* START CARD INFORMATION */                                      DR0207
      PUT EDIT('STATION','KCA','KCB','FREQ','CALIB DATE')           DR0208
              (PAGE,X(5),A(7),X(6),4A(12));                         DR0209
      PUT EDIT(SSTA,FINE_KCA,FINE_KCB,FREQ,CDATE,EOP)
              (SKIP(2),X(6),A(4),2F(12,3),X(3),A(19),A(20),         DR0211
               A(10));                                              DR02115
      PUT EDIT('UT','AC','BC','AM','BM','RA','RB','P STA',          DR0212
               'PLATE')(SKIP(3),X(6),A(14),3A(12),A(14),A(12),      DR0213
               A(4),2A(7));                                         DR0214
      PAGE=1;          PLTE=0;          LINE=6;                     DR0215
  /* FOR DISK OUTPUT */                                             DR0216
      IF N=0 THEN DO;                                               DR0217
         CODE='9';
         AC=FINE_KCA;          BC=FINE_KCB;                         DR0218
         GET STRING(CDATE) EDIT (HR,MN,SEC) (X(4),3F(2));           DR0219
         GET STRING(FREQ) EDIT (AM) (F(10,3));                      DR02194
         WRITE FILE(FINE) FROM(FINE_SAVE);                          DR02195
      END;                                                          DR02196
/* LIST FINE DATA */                                                DR0220
      READ FILE(TFINE1)INTO(FINE_SAVE);                             DR0221
      PREV_PLATE=F_PLATE;                                           DR0222
FINE_LIST:                                                          DR0223
      DO WHILE(LINE<39);                 /* EACH PAGE  */           DR0225
      IF F_PLATE¬=PREV_PLATE THEN DO;                               DR02254
         PLTE=0;                     PREV_PLATE=F_PLATE;            DR02255
      END;                                                          DR02256
  /* COMPUTE RESIDUALS */                                           DR0226
         RA=(AC+FINE_KCA-AM);                                       DR0227
         RB=(BC+FINE_KCB-BM);                                       DR0228
         CONSTANTS3=CODE||' '||STA||' '||F_PLATE;                   DR0229
  /* FOR DISK OUTPUT */                                             DR0231
         HRS=HR;                   MIN=MN;                          DR0232
         IF N=0 THEN WRITE FILE(FINE) FROM(FINE_SAVE);              DR0233
         KA=RA*1000.0;        RB=RB*1000.0;
      CALL PRINT;                                                   DR0235
         READ FILE(TFINE1)INTO(FINE_SAVE);                          DR02357
      END;                               /* END OF PAGE  */         DR0236
```

```
             LINE=0;        PAGE=0;                                         DR0237
             GO TO FINE_LIST;                                               DR0240
       /* FOR SECOND LISTING IF NECESSARY */                                DR0241
   REPEAT:                                                                  DR0242
             CLOSE FILE(TFINE1);                                            DR0243
             N=N+1;                                                         DR0244
             IF N<NO THEN DO;                                               DR0245
                OPEN FILE(TFINE1);                                          DR0246
                GO TO FINELBL;                                              DR0247
                          END;                                              DR0248
   /* AMBIGUITY DATA */                                                     DR0249
       /* CHECK FOR DATA */                                                 DR0250
   AMB:                                                                     DR0251
             IF ¬AMB_FLAG THEN GO TO START;                                 DR0252
             OPEN FILE(M_OUT), FILE(C_OUT),FILE(AMBIN);                     DR0253
             ON ENDFILE (AMBIN) GO TO NEXTC;                                DR02535
             ON ENDFILE(INPUT) GO TO AVE_AMB;                               DR0254
             KCAM_SUM=0;         KCBM_SUM=0;         N=0;                   DR0255
             KCAC_SUM=0;         KCBC_SUM=0;         J=0;                   DR0256
 CRDRD:      PROCEDURE;          /* READ MINITRACK CARD */                  DR0257
       GET FILE (INPUT) EDIT (AMB_UT,M_QAM,QAM_C,QBM_M,QBM_C,CDE,           DR0258
       CPLATE)  (F(6),X(4),        4F(10,3),X(9),F(1),X(17),F(3)); D
   END;      /* END CRDRD */                                                DR0260
 DSKRD:      PROCEDURE;          /* READ AMBIGUITY CAMERA DISK */           DR0261
       READ FILE (AMBIN) INTO (AMB_IN);   CONSTANTS4=ASTA;                  DR0262
       UH=AHR;         UM=AMN;          US=ASC;                             DR02625
       NUT=AHR*10000+AMN*100+ASC;         CONSTANTS5=ACODE;                 DR0263
       GET STRING (PLATE) EDIT (DPLATE) (F(3));                             DR0264
   END;      /* END DSKRD */                                                DR0265
 KS1S:       PROCEDURE;          /* SET KS1 VALUES */                       DR0267
                            /* SET KS1 VALUES */                            DR0268
       KSAM=M_QAM;     KSAC=QAM_C;     KSBM=QBM_M;     KSBC=QBM_C;          DR0269
       CALL CRDRD;                                                          DR0270
                            /* SAVE VALUES FOR INTERPOLATION */             DR02701
       UT_S=AMB_UT;    VAL_AC=QAM_C;   VAL_BM=QBM_M;   VAL_BC=QBM_C;        DR02702
       CALL CRDRD;                                                          DR02703
  END;       /* END KS1S   */                                               DR02704
 NEWPLT:     PROCEDURE;          /* FIND MATCHING PLATES */                 DR02705
       DO WHILE (DPLATE¬=CPLATE);                                           DR02706
          IF DPLATE<CPLATE                                                  DR02707
             THEN CALL DSKRD;                                               DR02708
          IF CPLATE<DPLATE                                                  DR02709
             THEN CALL CRDRD;                                               DR0271
       END;                                                                 DR02711
       GO TO TMECHK;                                                        DR02712
   END;     /* END NEWPLT */                                                DR02713
AMB_PLATE:                                                                  DR02714
   CALL DSKRD;                                                              DR02715
   NEXTC:                                                                   DR027155
   CALL CRDRD;                                                              DR02716
 TMECHK:                                                                    DR02717
       DO WHILE (AMB_UT¬=NUT);          /* FIND MATCHING TIMES */           DR02718
          IF CDE=1                                                          DR02719
                      THEN CALL KS1S;                                       DR0272
          IF DPLATE¬=CPLATE                                                 DR0273
                      THEN CALL NEWPLT;                                     DR0274
          IF AMB_UT<NUT                                                     DR0275
                      THEN CALL CRDRD;                                      DR02752
          IF NUT<AMB_UT                                                     DR02753
                      THEN CALL DSKRD;                                      DR02754
       END;                                                             DR02755
   IF CDE=1 THEN CALL KSIS;                                             DR02756
       /* INTERPOLATE */                                                DR0276
      IF(AMB_UT-UT_S)>4000.0 THEN DO;
                                UT_S=UT_S+4040.0;
                                GO TO FXED;
                                END;
      IF(AMB_UT-UT_S)>39.0 THEN DO;
                              UT_S=UT_S+40.0;
                              END;
FXED: IF ABS(QAM_C-VAL_AC)>0.25*(AMB_UT-UT_S)THEN IF QAM_C>VAL_AC
                                              THEN DO;
                                                   VAL_AC=VAL_AC+1.0;
                                                   KF(1)=1;
                                                   END;
                                              ELSE DO;
                                                   QAM_C=QAM_C+1.0;
                                                   KF(2)=1;
                                                   END;
      IF ABS(QBM_M-VAL_BM)>0.25*(AMB_UT-UT_S)THEN IF QBM_M>VAL_BM
                                              THEN DO;
                                                   VAL_BM=VAL_BM+1.0;
                                                   KF(3)=1;
                                                   END;
                                              ELSE DO;
                                                   QBM_M=QBM_M+1.0;
                                                   KF(4)=1;
                                                   END;
      IF ABS(QBM_C-VAL_BC)>0.25*(AMB_UT-UT_S)THEN IF QBM_C>VAL_BC
                                              THEN DO;
                                                   VAL_BC=VAL_BC+1.0;
                                                   KF(5)=1;
                                                   END;
                                              ELSE DO;
                                                   QBM_C=QBM_C+1.0;
                                                   KF(6)=1;
                                                   END;
            C_QAM=QAM_C-(.2*(QAM_C-VAL_AC)/(AMB_UT-UT_S));              DR0284
            M_QBM=QBM_M-(.4*(QBM_M-VAL_BM)/(AMB_UT-UT_S));              DR0285
            C_QBM=QBM_C-(.6*(QBM_C-VAL_BC)/(AMB_UT-UT_S));              DR0286
                     IF KF(1)=1 THEN VAL_AC=VAL_AC-1.0;
                     IF KF(2)=1 THEN QAM_C=QAM_C-1.0;
                     IF KF(3)=1 THEN VAL_BM=VAL_BM-1.0;
                     IF KF(4)=1 THEN QBM_M=QBM_M-1.0;
                     IF KF(5)=1 THEN VAL_BC=VAL_BC-1.0;
                     IF KF(6)=1 THEN QBM_C=QBM_C-1.0;
                     KF=0;
        /* AM,BM, AND KC TOTALS */                                      DR0287
DECP:        DEC_AM=MOD(100+M_QAM-AC_M-KSAM,1);                         DR0288
             DEC_AC=MOD(100+C_QAM-AC_C-KSAC,1);                         DR0289
             DEC_BM=MOD(100+M_QBM-BC_M-KSBM,1);                         DR0290
             DEC_BC=MOD(100+C_QBM-BC_C-KSBC,1);                         DR0291
        /* IA AND IB FOR FIRST CARD */                                  DR0292
             IF J=0 THEN DO;                                            DR0293
                IF DEC_AM>.850 THEN IA_M=49;                            DR0294
                               ELSE IA_M=50;                            DR0295
                IF DEC_AC>.850 THEN IA_C=49;                            DR0296
                               ELSE IA_C=50;                            DR0297
                IF DEC_BM>.850 THEN IB_M=49;                            DR0298
                               ELSE IB_M=50;                            DR0299
```

```
          IF DEC_BC>.850 THEN IB_C=49;                           DR0300
                         ELSE IB_C=50;                           DR0301
                   END;                                          DR0302
  /* FOR OTHER CARDS */                                          DR0303
       ELSE DO;                                                  DR0304
          IF SAVE_KCAM>=50.100 THEN IA_M=50;                     DR0305
          ELSE IF SAVE_KCAM<50.100 & DEC_AM<0.500 THEN IA_M=50;  DR0306
               ELSE IA_M=49;                                     DR0307
          IF SAVE_KCAC>=50.100 THEN IA_C=50;                     DR0308
          ELSE IF SAVE_KCAC<50.100 & DEC_AC<0.500 THEN IA_C=50;  DR0309
               ELSE IA_C=49;                                     DR0310
          IF SAVE_KCBM>=50.100 THEN IB_M=50;                     DR0311
          ELSE IF SAVE_KCBM<50.100 & DEC_BM<0.500 THEN IB_M=50;  DR0312
               ELSE IB_M=49;                                     DR0313
          IF SAVE_KCBC>=50.100 THEN IB_C=50;                     DR0314
          ELSE IF SAVE_KCBC<50.100 & DEC_BC<0.500 THEN IB_C=50;  DR0315
               ELSE IB_C=49;                                     DR0316
            END;                                                 DR0317
       KCAM=DEC_AM+IA_M;   KCBM=DEC_BM+IB_M;                     DR0318
       KCAC=DEC_AC+IA_C;   KCBC=DEC_BC+IB_C;                     DR0319
  /* SAVE KC'S OF FIRST CARD */                                  DR0320
       IF J=0 THEN DO;                                           DR0321
          SAVE_KCAM=KCAM;  SAVE_KCBM=KCBM;                       DR0322
          SAVE_KCAC=KCAC;  SAVE_KCBC=KCBC;                       DR0323
          J=1;                                                   DR0324
                   END;                                          DR0325
  /* TOTAL KC'S FOR AVERAGE */                                   DR0326
       KCAM_SUM=KCAM_SUM+KCAM;        KCBM_SUM=KCBM_SUM+KCBM;    DR0327
       KCAC_SUM=KCAC_SUM+KCAC;        KCBC_SUM=KCBC_SUM+KCBC;    DR0328
       N=N+1;                                                    DR0329
  /* COMPUTE AM AND BM */                                        DR0330
       AM_M=KCAM+AC_M;     BM_M=KCBM+BC_M;                       DR0331
       AM_C=KCAC+AC_C;     BM_C=KCBC+BC_C;                       DR0332
       AMBHR=UH;           AMBMN=UM;            AMBSC=US;       DR03325
       APLATE=PLATE;                                            DR03326
  /* FOR MEDIUM */                                               DR0333
       AC_AMB=AC_M;   AM_AMB=AM_M;                               DR0334
       BC_AMB=BC_M;   BM_AMB=BM_M;                               DR0335
       WRITE FILE(M_OUT)FROM(AMB_SAVE);                          DR0336
  /* FOR COARSE */                                               DR0337
       AC_AMB=AC_C;   AM_AMB=AM_C;                               DR0338
       BC_AMB=BC_C;   BM_AMB=BM_C;                               DR0339
       WRITE FILE(C_OUT)FROM(AMB_SAVE);                          DR0340
            /* SAVE VALUES FOR INTERPOLATION */                  DR0341
  UT_S=AMB_UT;   VAL_AC=QAM_C;   VAL_BM=QBM_M;  VAL_BC=QBM_C;    DR0342
  UH=AHR;        UM=AMN;         US=ASC;                         DR0343
       GO TO AMB_PLATE;                    /* END OF PLATE */    DR0350
AVE_AMB:                                                         DR0351
  /* AVERAGE KC'S */                                             DR0352
       AVE_KCAM=KCAM_SUM/N;           AVE_KCBM=KCBM_SUM/N;       DR0353
       AVE_KCAC=KCAC_SUM/N;           AVE_KCBC=KCBC_SUM/N;       DR0354
       KCAC=DEC_AC+IA_C;   KCBC=DEC_BC+IB_C;                     DR0319
       CLOSE FILE(M_OUT), FILE(C_OUT);                           DR0355
/* COMPUTE RESIDUALS AND LIST DATA FOR MEDIUM */                 DR0356
       OPEN FILE (M_IN);                        N=0;             DR0357
       ON ENDFILE (M_IN)  GO TO COARSE;                          DR0358
MEDIUM:                                                          DR0359
       READ FILE(M_IN)INTO(AMB_SAVE);                            DR0360
       HRS=AMBHR;          MIN=AMBMN;           SEC=AMBSC;      DR03604
       PREV_PLATE=APLATE;                                        DR0361
```

```
 /* START CARD INFORMATION */                                         DR0362
     PUT EDIT('STATION','KCA','KCB','FREQ','CALIB DATE',              DR0363
            'MEDIUM')                                                 DR0364
           (PAGE,X(5),A(7),X(6),4A(12),X(10),A(6));                   DR0365
     PUT EDIT(SSTA,AVE_KCAM+.0005*SIGN(AVE_KCAM),                     DR0366
            AVE_KCBM+.0005*SIGN(AVE_KCBM),FREQ,CDATE)                 DR03665
           (SKIP(2),X(6),A(4),2F(12,3),X(3),A(19),A(10));             DR0367
     PUT EDIT('UT','AC','BC','AM','BM','RA','RB','P STA',             DR0368
           'PLATE')(SKIP(3),X(6),A(14),3A(12),A(14),A(12),            DR0369
            A(4),2A(7));                                              DR0370
     PAGE=1;          PLTE=0;          LINE=6;                        DR0371
MEDIUM_LIST:                                                          DR0372
     DO WHILE(LINE<39);                        /* EACH PAGE  */       DR0374
     IF  APLATE¬=PREV_PLATE THEN DO;                                  DR03744
        PLTE=0;                        PREV_PLATE=APLATE;             DR03745
     END;                                                             DR03746
 /* COMPUTE RESIDUALS */                                              DR0375
        RA=(AC_AMB+AVE_KCAM-AM_AMB)*1000;
        RB=(BC_AMB+AVE_KCBM-BM_AMB)*1000;
        AM=AM_AMB;
        BM=BM_AMB;
        AC=AC_AMB;
        BC=BC_AMB;
        CONSTANTS3='   '||CONSTANTS4||APLATE||CONSTANTS5;             DR0381
     CALL PRINT;                                                      DR0382
        READ FILE(M_IN)INTO(AMB_SAVE);                                DR0383
     HRS=AMBHR;            MIN=AMBMN;              SEC=AMBSC;         DR03834
     END;                                      /* END OF PAGE  */     DR0384
     LINE=0;                   PAGE=0;                                DR0385
     GO TO MEDIUM_LIST;                                               DR0388
COARSE:                                                               DR0389
     CLOSE FILE (M_IN);  OPEN FILE(C_IN);                             DR0390
     ON ENDFILE(C_IN) GO TO REPEAT2;                                  DR0391
     READ FILE(C_IN)INTO(AMB_SAVE);                                   DR0392
     HRS=AMBHR;            MIN=AMBMN;              SEC=AMBSC;         DR03925
     PREV_PLATE=APLATE;                                               DR0393
 /* START CARD INFORMATION */                                         DR0394
     PUT EDIT('STATION','KCA','KCB','FREQ','CALIB DATE',              DR0395
            'COARSE')                                                 DR0396
           (PAGE,X(5),A(7),X(6),4A(12),X(10),A(6));                   DR0397
     PUT EDIT(SSTA,AVE_KCAC,AVE_KCBC,FREQ,CDATE)
           (SKIP(2),X(6),A(4),2F(12,3),X(3),A(19),A(10));             DR0393
     PUT EDIT('UT','AC','BC','AM','BM','RA','RB','P STA',             DR0400
           'PLATE')(SKIP(3),X(6),A(14),3A(12),A(14),A(12),            DR0401
            A(4),2A(7));                                              DR0402
     PAGE=1;          PLTE=0;          LINE=6;                        DR0403
COARSE_LIST:                                                          DR0404
     DO WHILE(LINE<39);                        /* EACH PAGE  */       DR0406
     IF  APLATE¬=PREV_PLATE THEN DO;                                  DR04064
        PLTE=0;                        PREV_PLATE=APLATE;             DR04065
     END;                                                             DR04066
 /* COMPUTE RESIDUALS */                                              DR0407
        RA=(AC_AMB+AVE_KCAC-AM_AMB)*1000; RA=RA+(.5*SIGN(RA));DR0408
        RB=(BC_AMB+AVE_KCBC-BM_AMB)*1000;
        AM=AM_AMB;
        BM=BM_AMB;
        AC=AC_AMB;
        BC=BC_AMB;
        CONSTANTS3='   '||CONSTANTS4||APLATE||CONSTANTS5;             DR0413
        CALL PRINT;                                                        DR0414
           READ FILE(C_IN)INTO(AMB_SAVE);                                  DR0415
        HRS=AMBHR;          MIN=AMBMN;          SEC=AMBSC;                 DR04154
        END;                               /* END OF PAGE  */              DR0416
        LINE=0;             PAGE=0;                                        DR0417
        GO TO COARSE_LIST;                                                 DR0420
   /* FOR SECOND LISTING */                                                DR0421
REPEAT2:                                                                   DR0422
        CLOSE FILE(C_IN);                                                  DR0423
        N=N+1;                                                             DR0424
        IF N<2 THEN DO;                                                    DR0425
           OPEN FILE (M_IN);                                               DR0426
           GO TO MEDIUM;                                                   DR0427
                    END;                                                   DR0428
   /* FOR NEXT SET OF DATA */                                              DR0429
        GO TO START;                                                       DR0430
      FINISH:                                                              DR0431
 END;                                                                      DR0432
/*   END OF DATA
/&   END OF JOB
```

```
*                     M.HARRISON;14401DDRZ01E,L,018
// OPTION CATAL,DECK,SYM
 PHASE DRC1E,ROOT
// EXEC PL/I
* PROCESS LISTC,NCOPT
      DRZ01E:PROCEDURE OPTIONS(MAIN);
                                   /*ANTENNA CORRECTIONS */
 DECLARE
 CARDS FILE INPUT
      ENVIRONMENT(MEDIUM(SYSIPT,2540) F(80)),
LISTNG FILE OUTPUT PRINT
     ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),
      PUNCH FILE OUTPUT STREAM
      ENVIRONMENT(MEDIUM(SYSPCH,2540) F(80)),
 1 START EXTERNAL,
  2 C  FIXED DECIMAL(1),
  2 STA FIXED DECIMAL(4),
 (2 KCA,
  2 KCB,
    2 FREQ) DECIMAL FLOAT(7),
  2 CALIBD FIXED DECIMAL(6),
  2 P FIXED DECIMAL (2),
  2 BLANK CHARACTER (1) ;
DECLARE
NPLOTS(30) FIXED BINARY(31),
  CARDNO FIXED DECIMAL (5),
  SETNO CHARACTER(2),
  (CARDSTA1,CARDSTA2) FIXED DECIMAL(4),
  (CARDD1,CARDD2) FIXED DECIMAL(6),
 (ACO(47),
  SAVEA(47)) DECIMAL FLOAT(16);
DECLARE MATRIX(48) DECIMAL FLOAT(16);
      OPEN FILE(CARDS), FILE(LISTNG),FILE(PUNCH);
      GET FILE(CARDS) EDIT (START)
           (X(1),F(1),X(4),F(4),3F(10,3),X(4),F(6),F(2),
            X(27),A(1));
                                        /* P=0 DO REGULAR COMPUTATION
                                           P=1 DO HISTORICAL   */
      IF START.P=0 THEN GO TO PROCEED;
 HISTOR:
                                        /* READ COEFF CARDS */
      DO K1=1 TO 2;
      GET FILE(CARDS) EDIT
      (CARDSTA1,CARDD1,(MATRIX(I) DO I=1 TO 3),SETNO)
      (F(4),X(2),F(6),X(18),3E(15,8,9),X(3),A(2));
      I1=4; I2=8;
      DO J=1 TO 9;
      GET FILE(CARDS) EDIT
      ((MATRIX(I) DO I=I1 TO I2),CARDNO)
      (5E(15,8,9),F(5));
           I2=I2+5;
           I1=I1+5;
      END;
      IF K1=1 THEN DO J1=1 TO 47;SAVEA(J1)=MATRIX(J1); END;
      ELSE DO J1=1 TO 47; ACO(J1)=MATRIX(J1); END;
      END;                 /*HISTORICAL    */
      GO TO CONTINUE;
  PROCEED:
     CALL OVERLAY ('DR01E1');
      CALL COEFFC(ACO,SAVEA);
     CALL OVERLAY ('DR01E2');                                          0058
     CALL PRINT1(ACO,SAVEA);                                           0059
CONTINUE:                                                              0060
     CALL OVERLAY ('DR01E3');                                          0061
     CALL RESIDL(ACO,SAVEA,NPLOTS);                                    0062
     IF START.P=0 THEN DO;                                           MA  63
     CALL OVERLAY ('DR01E4');                                          0064
     CALL DRZPLT(NPLOTS);                                              0065
     END;                                                              0066
     CLOSE FILE(CARDS),FILE(LISTNG),FILE(PUNCH);                       0067
       END; /* DRZ01E */                                               0068
/*   END OF DATA
 PHASE DR01E1,*                                                        0070
 INCLUDE AAP06A
 INCLUDE ILFDSIMQ
// EXEC PL/I
* PROCESS LISTO,NOOPT
COEFFC:PROCEDURE(ACO,SAVEA);                                           0072
DECLARE                                                                0073
     LISTNG FILE OUTPUT PRINT                                          0074
          ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),                     0075
     CARDS FILE INPUT                                                  0076
          ENVIRONMENT(MEDIUM(SYSIPT,2540) F(80)),                      0077
     DISK1 FILE INPUT RECORD                                           0078
          ENVIRONMENT(MEDIUM(SYS003,2314) F(3400,136) ),
                                    /* CARD INPUT  */                  0080
 1 START EXTERNAL,                                                     0081
  2 C FIXED DECIMAL(1),                                                0082
  2 STA FIXED DECIMAL(4),                                              0083
 (2 KCA,                                                               0084
  2 KCB,                                                               0085
    2 FREQ) DECIMAL FLOAT(7),
  2 CALIBD FIXED DECIMAL(6),                                           0087
  2 P FIXED DECIMAL (2),                                               0088
     2 SPECIAL           CHAR (1),                                  1006071
 1 DATA,                                                               0090
  2 DT (3) DECIMAL FLOAT,                                              0091
  2 BLANKS CHARACTER(4),                                               0092
 (2 AC,                                                                0093
  2 BC,                                                                0094
 2 AM1,                                                                0095
 2 BM1,                                                                0096
  2 BETA,                                                              0097
  2 R2,                                                                0098
  2 DELA,                                                              0099
  2 DELB,                                                              0100
  2 NA,                                                               ,0101
  2 NB) DECIMAL FLOAT(16),                                             0102
  2 STA CHARACTER(4),                                                  0103
  2 PLATE CHARACTER(3),                                                0104
  2 PLOTCODE CHARACTER(1),                                             0105
 (2 RA,                                                                0106
 2 RB,                                                                 0107
  2 AM,                                                                0108
 2 BM) DECIMAL FLOAT(16),                                              0109
 RETRNH LABEL;                                                         0110
DECLARE                                                                0111
 ROUND,                                                                0112
    (DECXA,DECYA) FIXED DECIMAL(11,4),                                 0113
                              /* COEFFICIENTS */                       0114
```

```
(SINYCOSY,                                                                  0115
 SINY,COSY,                                                                 0116
 XSINY,XCOSY,YSINY,YCOSY,XYSINY,XYCOSY,X2SINY,X2COSY,Y2SINY,                0117
 Y2COSY,X3SINY,X3COSY,Y3SINY,Y3COSY,SINY2,COSY2,                            0118
 SIN1,SIN2,COS1,COS2,                                                       0119
 XA,YA,X2A,Y2A,X3A,Y3A,XYA,                                                 0120
 ACO (47),                                                                  0121
 SAVEA(47),                                                                 0122
 Q71,Q72,Q73,Q81,Q82,Q83,                                                   0123
 COSB,Q9A,Q10A,Q11A,SINB,ENRS,ENRC,                                         0124
D11,D21,Q10Q11,DN,                                                          0125
ENR2,Q9B,Q10B,Q11B,NSQAR,ESQAR,                                             0126
      EVALUE,NVALUE,                                                        0127
 ANGLX,ANGLY,                                                               0128
 AQ(4),BQ(4),                                                               0129
 CA,                                                                        0130
 CB,                                                                        0131
 Q92A,Q9Q10A,Q9Q11A,Q102A,Q10Q11A,Q112A,                                    0132
 Q92B,Q9Q10B,Q9Q11B,Q102B,Q10Q11B,Q112B,                                    0133
 DECY,                                                                      0134
 DECX,                                                                      0135
 MATRIX(81),                                                                0136
 MATRX6 (9,9),                                                              0137
 ANSWER (9),                                                                0138
COEFF,                                                                      0139
 COLOG,                                                                     0140
 BVECTOR(9),                                                                0141
 SVECTOR(9),                                                                0142
 LEAST(18),                                                                 0143
 VECTOR(9)) DECIMAL FLOAT(16),                                              0144
RETRNP LABEL;                                                               0145
DECLARE                                                                     0146
 (WHOLEX,WHOLEY) DECIMAL FIXED(7),                                          0149
 SETNO CHARACTER(2),                                                        0150
 HMATRX CHARACTER(12),                                                      0151
DSIMQ ENTRY,                                                                0152
 (D,E,F) FIXED BINARY(15);                                                  0153
                                      /* BEGINNING OF PROGRAM */            0154
      SETNO='A'; LOOP=0;  COLNO=0;                                          0155
      HMATRX='ELEMENTS OF ';                                                0156
      OPEN FILE(CARDS),FILE(LISTNG),FILE(DISK1);                            0157
      J2=0;                                                                 0158
 STARTP:                                                                    0159
                                 /*INITIALIZE FOR SUMS */                   0160
      READ FILE(DISK1) INTO(DATA);                                          0161
      IF SPECIAL = ' ' THEN DO ;                                         1/06071
      KCA=AC; KCB=BC; FREQ=AM;                                              0162
      END ;                                                              1/06071
      AQ=0; BQ=0; SVECTOR=0; BVECTOR=0;  MATRX6=0;                           163
      XCOSY=0;XSINY=0;YCOSY=0;YSINY=0;XYSINY=0;XYCOSY=0;                    0171
                         X2SINY=0;X2COSY=0;Y2SINY=0;Y2COSY=0;               0172
      Y3SINY=0;Y3COSY=0;SINYCOSY=0;SINY2=0;COSY2=0;                         0173
      SINY=0;COSY=0;                                                        0174
      Q9A=0;Q10A=0;Q11A=0;                                                  0175
      Q9B=0;Q10B=0;Q11B=0;                                                  0176
      Q92A=0; Q9Q10A=0; Q9Q11A=0; Q102A=0; Q10Q11A=0;Q112A=0;               0177
      Q92B=0; Q9Q10B=0; Q9Q11B=0; Q102B=0; Q10Q11B=0;Q112B=0;               0178
      ROUND=.00005;                                                         0179
      CALL PAGEHD;                                                          0180
      PUT FILE(LISTNG) EDIT
```

```
        ('EQUATION SET 6') (SKIP,X(50),A(14));
READDATA:                                                                         0181
     READ FILE(DISK1) INTO (DATA);                                                0182
     ON ENDFILE(DISK1) GO TO MATRIX_6;                                           4 183
     XA=DATA.AM-START.KCA;                                                        0185
     YA=DATA.BM-START.KCB;                                                        0186
                                   /*COMPUTE X,Y,(X),(Y) */                       0187
     DECXA=XA+ROUND*SIGN(XA); DECYA=YA+ROUND*SIGN(YA);                            0188
     WHOLEX=TRUNC(DECXA);                                                         0189
     WHOLEY=TRUNC(DECYA);                                                         0190
     DECX=DECXA-WHOLEX;                                                           0191
     DECY=DECYA-WHOLEY;                                                           0192
     ANGLX=DECX*6.2831854;                                                        0193
     ANGLY=DECY*6.2831854;                                                        0194
     SIN1=SIN(ANGLX);                                                             0195
     SIN2=SIN(ANGLY);                                                             0196
     COS1=COS(ANGLX);                                                             0197
     COS2=COS(ANGLY);                                                             0198
     X2A=XA*XA;                                                                   0199
     Y2A=YA*YA;                                                                   0200
     XYA=XA*YA;                                                                   0201
     X3A=XA*X2A;                                                                  0202
     Y3A=YA*Y2A;                                                                  0203
     D11=DATA.AC;
     D21=DATA.BC;                                                                 0204
                                   /* SUMMATIONS  */                              0205
     MATRX6(1,1)=MATRX6(1,1)+1;                                                   0206
     MATRX6(1,2)=MATRX6(1,2)+XA;                                                  0207
     MATRX6(1,3)=MATRX6(1,3)+YA;                                                  0208
     MATRX6(1,4)=MATRX6(1,4)+XYA;                                                 0209
     MATRX6(1,5)=MATRX6(1,5)+X2A;                                                 0210
     MATRX6(1,6)=MATRX6(1,6)+Y2A;                                                 0211
     MATRX6(1,8)=MATRX6(1,8)+SIN1;                                                0212
     MATRX6(1,9)=MATRX6(1,9)+COS1;                                                0213
     MATRX6(2,2)=MATRX6(2,2)+X2A;                                                 0214
     MATRX6(2,3)=MATRX6(2,3)+XYA;                                                 0215
     MATRX6(2,4)=MATRX6(2,4)+X2A*YA;                                              0216
     MATRX6(2,5)=MATRX6(2,5)+X3A;                                                 0217
     MATRX6(2,6)=MATRX6(2,6)+XA*Y2A;                                              0218
     MATRX6(2,8)=MATRX6(2,8)+XA*SIN1;                                             0219
     MATRX6(2,9)=MATRX6(2,9)+XA*COS1;                                             0220
     MATRX6(3,3)=MATRX6(3,3)+Y2A;                                                 0221
     MATRX6(3,4)=MATRX6(3,4)+XA*Y2A;                                              0222
     MATRX6(3,5)=MATRX6(3,5)+X2A*YA;                                              0223
     MATRX6(3,6)=MATRX6(3,6)+Y3A;                                                 0224
     MATRX6(3,8)=MATRX6(3,8)+YA*SIN1;                                             0225
     MATRX6(3,9)=MATRX6(3,9)+YA*COS1;                                             0226
     MATRX6(4,4)=MATRX6(4,4)+X2A*Y2A;                                             0227
     MATRX6(4,5)=MATRX6(4,5)+X3A*YA;                                              0228
     MATRX6(4,6)=MATRX6(4,6)+XA*Y3A;                                              0229
     MATRX6(4,8)=MATRX6(4,8)+XYA*SIN1;                                            0230
     MATRX6(4,9)=MATRX6(4,9)+XYA*COS1;                                            0231
     MATRX6(5,5)=MATRX6(5,5)+XA*X3A;                                              0232
     MATRX6(5,6)=MATRX6(5,6)+X2A*Y2A;                                             0233
     MATRX6(5,8)=MATRX6(5,8)+X2A*SIN1;                                            0234
     MATRX6(5,9)=MATRX6(5,9)+X2A*COS1;                                            0235
     MATRX6(6,6)=MATRX6(6,6)+YA*Y3A;                                              0236
     MATRX6(6,8)=MATRX6(6,8)+Y2A*SIN1;                                            0237
     MATRX6(6,9)=MATRX6(6,9)+Y2A*COS1;                                            0238
     MATRX6(8,8)=MATRX6(8,8)+SIN1*SIN1;                                           0239
```

```
MATRX6(8,9)=MATRX6(8,9)+SIN1*COS1;                                    0240
MATRX6(9,9)=MATRX6(9,9)+COS1*COS1;                                    0241
COSY=COSY+COS2;                                                       0242
SINY=SINY+SIN2;                                                       0243
SINY2=SINY2+SIN2*SIN2;                                                0244
COSY2=COSY2+COS2*COS2;                                                0245
SINYCOSY=SINYCOSY+SIN2*COS2;                                          0246
XSINY=XSINY+XA*SIN2;                                                  0247
XCOSY=XCOSY+XA*COS2;                                                  0248
YSINY=YSINY+YA*SIN2;                                                  0249
YCOSY=YCOSY+YA*COS2;                                                  0250
XYSINY=XYSINY+XYA*SIN2;                                               0251
XYCOSY=XYCOSY+XYA*COS2;                                               0252
X2SINY=X2SINY+X2A*SIN2;                                               0253
X2COSY=X2COSY+X2A*COS2;                                               0254
Y2COSY=Y2COSY+Y2A*COS2;                                               0255
Y2SINY=Y2SINY+Y2A*SIN2;                                               0256
CA=DATA.AC+START.KCA-DATA.AM;                                         0257
CB=DATA.BC+START.KCB-DATA.BM;                                         0258
EVALUE=DATA.AC/DATA.NA;                                               0259
NVALUE=DATA.BC/DATA.NB;                                               0260
ENR2=(EVALUE*NVALUE)/DATA.R2;                                         0261
DATA.BETA=1.5707963-DATA.BETA;
COSB=COS(DATA.BETA);                                                  0262
SINB=SIN(DATA.BETA);                                                  0263
NSQAR=(1-NVALUE*NVALUE)/DATA.R2;                                      0264
ESQAR=(1-EVALUE*EVALUE)/DATA.R2;                                      0265
Q71=ENR2*(-1*SINB)-ESQAR*COSB;                                        0266
Q72=(ESQAR*SINB-ENR2*COSB) ;                                          0267
Q73= DATA.DELA;
Q81=ENR2*COSB+NSQAR*SINB;                                             0269
Q82=-ENR2*SINB+NSQAR*COSB;                                            0270
Q83=DATA.DELB;
Q9A=Q9A+Q71;                                                          0272
Q10A=Q10A+Q72;                                                        0273
Q11A=Q11A+Q73;                                                        0274
Q92A=Q92A+Q71*Q71;                                                    0275
Q9Q10A=Q9Q10A+Q71*Q72;                                                0276
Q9Q11A=Q9Q11A+Q71*Q73;                                                0277
Q102A =Q102A+Q72*Q72;                                                 0278
Q10Q11A=Q10Q11A+Q72*Q73;                                              0279
Q112A =Q112A+Q73*Q73;                                                 0280
Q9B=Q9B+Q81;                                                          0281
Q10B=Q10B+Q82;                                                        0282
Q11B=Q11B+Q83;                                                        0283
Q92B=Q92B+Q81*Q81;                                                    0284
Q9Q10B=Q9Q10B+Q81*Q82;                                                0285
Q9Q11B=Q9Q11B+Q81*Q83;                                                0286
Q102B=Q102B+Q82*Q82;                                                  0287
Q10Q11B=Q10Q11B+Q82*Q83;                                              0288
Q112B=Q112B+Q83*Q83;                                                  0289
AQ(1)=AQ(1)+CA;                                                       0290
AQ(2)=AQ(2)+CA*Q71;                                                   0291
AQ(3)=AQ(3)+CA*Q72;                                                   0292
AQ(4)=AQ(4)+CA*Q73;                                                   0293
BQ(1)=BQ(1)+CB;                                                       0294
BQ(2)=BQ(2)+CB*Q81;                                                   0295
BQ(3)=BQ(3)+CB*Q82;                                                   0296
BQ(4)=BQ(4)+CB*Q83;                                                   0297
SVECTOR(1)=SVECTOR(1)+D11;                                            0298
```

```
     SVECTOR(2)=SVECTOR(2)+D11*XA;                                          0299
     SVECTOR(3)=SVECTOR(3)+D11*YA;                                          0300
     SVECTOR(4)=SVECTOR(4)+D11*XYA;                                         0301
     SVECTOR(5)=SVECTOR(5)+D11*X2A;                                         0302
     SVECTOR(6)=SVECTOR(6)+D11*Y2A;                                         0303
     SVECTOR(8)=SVECTOR(8)+D11*SIN1;                                        0305
     SVECTOR(9)=SVECTOR(9)+D11*COS1;                                        0306
     BVECTOR(1)=BVECTOR(1)+D21;                                             0307
     BVECTOR(2)=BVECTOR(2)+D21*XA;                                          0308
     BVECTOR(3)=BVECTOR(3)+D21*YA;                                          0309
     BVECTOR(4)=BVECTOR(4)+D21*XYA;                                         0310
     BVECTOR(5)=BVECTOR(5)+D21*X2A;                                         0311
     BVECTOR(6)=BVECTOR(6)+D21*Y2A;                                         0312
     BVECTOR(8)=BVECTOR(8)+D21*SIN2;                                        0314
     BVECTOR(9)=BVECTOR(9)+D21*COS2;                                        0315
   IF START.C=1 THEN DO;                                                    0316
     SVECTOR(7)=SVECTOR(7)+D11*X3A;
     BVECTOR(7)=BVECTOR(7)+D21*X3A;                                         0313
     MATRX6(1,7)=MATRX6(1,7)+X3A;                                           0317
     MATRX6(2,7)=MATRX6(2,7)+XA*X3A;                                        0318
     MATRX6(3,7)=MATRX6(3,7)+YA*X3A;                                        0319
     MATRX6(4,7)=MATRX6(4,7)+X2A*X2A*YA;                                    0320
     MATRX6(5,7)=MATRX6(5,7)+X3A*X2A;
     MATRX6(6,7)=MATRX6(6,7)+X3A*Y2A;
     MATRX6(7,7)=MATRX6(7,7)+X3A*X3A;                                       0323
     MATRX6(7,8)=MATRX6(7,8)+X3A*SIN1;                                      0324
     MATRX6(7,9)=MATRX6(7,9)+X3A*COS1;                                      0325
     Y3SINY=Y3SINY+X3A*SIN2;                                                0326
     Y3COSY=Y3COSY+X3A*COS2;                                                0327
                END;                                                        0328
   ELSE DO;                                                                 0329
     SVECTOR(7)=SVECTOR(7)+D11*Y3A;                                         0304
     BVECTOR(7)=BVECTOR(7)+D21*Y3A;                                         0313
     MATRX6(1,7)=MATRX6(1,7)+Y3A;                                           0330
     MATRX6(2,7)=MATRX6(2,7)+XA*Y3A;                                        0331
     MATRX6(3,7)=MATRX6(3,7)+Y3A*YA;                                        0332
     MATRX6(4,7)=MATRX6(4,7)+XA*Y3A*YA;                                     0333
     MATRX6(5,7)=MATRX6(5,7)+X2A*Y3A;                                       0334
     MATRX6(6,7)=MATRX6(6,7)+Y2A*Y3A;                                       0335
     MATRX6(7,7)=MATRX6(7,7)+Y3A*Y3A;                                       0336
     MATRX6(7,8)=MATRX6(7,8)+Y3A*SIN1;                                      0337
     MATRX6(7,9)=MATRX6(7,9)+Y3A*COS1;                                      0338
     Y3SINY=Y3SINY+Y3A*SIN2;                                                0339
     Y3COSY=Y3COSY+Y3A*COS2;                                                0340
         END;                                                               0341
     GO TO READDATA;                                                        0365
                               /* FILL  9X9 MATRIX AND VECTOR*/             0366
MATRIX_6:                                                                   0367
         IJ=0;                                                              0368
       DO I1=2 TO 9;                                                        0369
         IJ=1+IJ;                                                           0370
       DO J1=1 TO IJ;                                                       0371
         MATRX6(I1,J1)=MATRX6(J1,I1);                                       0372
         END; END; GO TO LOOP1;                                             0373
FILLM:PROCEDURE;                                                            0374
     I1=1;                                                                  0375
     DO I2=1 TO N;                                                          0376
     VECTOR(I2)=SVECTOR(I2);                                                0377
     DO J1=1 TO N;                                                          0378
     MATRIX(I1)=MATRX6(I2,J1);                                              0379
     I1=I1+1;
     END; END;
     J2=J2+1;
     CALL LINKPF(DSIMQ,MATRIX,VECTOR,N,IER);
     DO I=1 TO N;
     ACO(K1)=VECTOR(I);  K1=K1+1; END;
     IF IER=1 THEN DO;
          PUT FILE (LISTNG) EDIT
          ('NO SOLUTION SET',J2,SETNO,IER)
          (SKIP(2),A(15),F(2),A(1),F(2));
          END; ELSE;
END;    /* FILLM */
                                        /* FILL COEFFICIENTS FOR A */
LOOP1:
     DO IN=1 TO 2;
                                        /* SET 1  */
     K1=1; N=3; CALL FILLM;
                                        /* SET 2A  */
     N=6; CALL FILLM;
                                                  /* SET 3A */
     N=7; CALL FILLM;
                                        /* SET 4A */
     L=0;
     I1=0;
     DO I=1 TO 3,8 TO 9;
         DO J=1 TO 3,8 TO 9;
     I1=I1+1; MATRIX(I1)=MATRX6(I,J); END;
     L=L+1; VECTOR(L)=SVECTOR(I); END;
     N=5;
     CALL LINKPF(DSIMQ,MATRIX,VECTOR,N,IER);
     IF IER=1 THEN DO;
          PUT FILE (LISTNG) EDIT
          ('NO SOLUTION SET  4',SETNO,IER)
          (SKIP,A(17),A(1),F(2));
          END; ELSE;
SET4A:
     DO I=1 TO 5;
     ACO(K1)=VECTOR(I);
          K1=K1+1;
     END;
                                          /* SET 5A  */
     I1=0; L=0;
     DO I=1 TO 6, 8 TO 9;
          DO J=1 TO 6, 8 TO 9;
          I1=I1+1;
          MATRIX(I1)=MATRX6(I,J);
     END;
     L=L+1;
          VECTOR(L)=SVECTOR(I);
                    END;
     N=8;
     CALL LINKPF(DSIMQ,MATRIX,VECTOR,N,IER);
     IF IER=1 THEN DO;
          PUT FILE (LISTNG) EDIT
          ('NO SOLUTION SET 5',SETNO,IER)
          (SKIP,A(17),A(1), F(2));
          END; ELSE;
SET5A:
     DO I=1 TO 8;
          ACO(K1)=VECTOR(I);
```

```
      K1=K1+1;
   END;
                                   /* SET 6A */
   N=9; CALL FILLM;
                                   /* PRINT A MATRIX AND VECTOR */
   RETRNP=STORE;
   GO TO PRINT_MATRIX;
STORE:
   IF IN=2 THEN GO TO EQUATNS;
      ELSE DO;
                                   /* SAVE A COEFFICIENTS */
   DO I=1 TO 38;
   SAVEA(I)=ACO(I);
   END;
                                   /* INITIALIZE FOR B COEFFICNTS*/
         MATRX6(8,1)=SINY;
         MATRX6(8,2)=XSINY;
         MATRX6(8,4)=XYSINY;
         MATRX6(8,3)=YSINY;
         MATRX6(8,5)=X2SINY;
         MATRX6(8,6)=Y2SINY;
         MATRX6(8,7)=Y3SINY;
         MATRX6(8,8)=SINY2;
         MATRX6(8,9)=SINYCOSY;
         MATRX6(9,1)=COSY;
         MATRX6(9,2)=XCOSY;
         MATRX6(9,3)=YCOSY;
         MATRX6(9,4)=XYCOSY;
         MATRX6(9,5)=X2COSY;
         MATRX6(9,6)=Y2COSY;
         MATRX6(9,7)=Y3COSY;
         MATRX6(9,8)=SINYCOSY;
         MATRX6(9,9)=COSY2;
         DO I=1 TO 9;
            SVECTOR(I)=BVECTOR(I) ;
         DO J=8 TO 9;
            MATRX6(I,J)=MATRX6(J,I);
         END;
                     END;
  END;
   SETNC='B'; COLNO=0;
         END;
EQUATNS:
   MATRX6(1,2)=Q9A;
   MATRX6(1,3)=Q10A;
   MATRX6(1,4)=Q11A;
   MATRX6(2,1)=Q9A;
   MATRX6(2,2)=Q92A;
   MATRX6(2,3)=Q9Q10A;
   MATRX6(2,4)=Q9Q11A;
   MATRX6(3,1)=Q10A;
   MATRX6(3,2)=Q9Q10A;
   MATRX6(3,3)=Q102A;
   MATRX6(3,4)=Q10Q11A;
   MATRX6(4,1)=Q11A;
   MATRX6(4,2)=Q9Q11A;
   MATRX6(4,3)=Q10Q11A;
   MATRX6(4,4)=Q112A;
      DO I=1 TO 4;
        SVECTOR(I)=AQ(I);
```

```
           END;
      SETNO='A'; COLNO=0;
      CALL PAGEHD;
      PUT FILE(LISTNG) EDIT
           ('EQUATION SET 7') (SKIP,X(50),A(14));
ADDITIONS:
    DO IN=1 TO 2;
      RETRNP=SET7A;
      GO TO PRINTOUT;
                                /* SET 7A  */
 SET7A:
      K1=39; N=4; CALL FILLM;
                                /* SET  8A  */
      N=3; CALL FILLM;
                                /* SET 9A */
      MATRIX(1)=MATRX6(1,1);
      IF IN=1 THEN DO;
      MATRIX(2)=Q11A;
      MATRIX(3)=Q11A;
      MATRIX(4)=Q112A;
      VECTOR(1)=AQ(1);
      VECTOR(2)=AQ(4);
         END;
           ELSE DO;
                MATRIX(2)=Q11B;
                MATRIX(3)=Q11B;
                MATRIX(4)=Q112B;
                VECTOR(1)=BQ(1);
                VECTOR(2)=BQ(4);
                END;
      N=2;
      CALL LINKPF(DSIMQ,MATRIX,VECTOR,N,IER);
      ACO(46)=VECTOR(1);
      ACO(47)=VECTOR(2);
      IF IER=1   THEN DO;
           PUT FILE(LISTNG) EDIT
           ('NO SOLUTION SET 9',SETNO, IER)
           (SKIP,A(17),A(1), F(2)); END; ELSE;
         IF IN=1 THEN DO;
      MATRX6(1,2)=Q9B;
      MATRX6(1,3)=Q10B;
      MATRX6(1,4)=Q11B;
      MATRX6(2,1)=Q9B;
      MATRX6(2,2)=Q92B;
      MATRX6(2,3)=Q9Q10B;
      MATRX6(2,4)=Q9Q11B;
      MATRX6(3,1)=Q10B;
      MATRX6(3,2)=Q9Q10B;
      MATRX6(3,3)=Q102B;
      MATRX6(3,4)=Q10Q11B;
      MATRX6(4,1)=Q11B;
      MATRX6(4,2)=Q9Q11B;
      MATRX6(4,3)=Q10Q11B;
      MATRX6(4,4)=Q112B;
      DO I=1 TO 4;
           AQ(I)=BQ(I);
           SVECTOR(I)=BQ(I); END;
 STOREC:
                DO I=39 TO 47;
                SAVEA(I)=ACO(I);
                END;
     SETNO='B'; COLNO=0;
     END;
         ELSE;
   END;                  /* END OF ADDITIONS */
     GO TO ENDRT;
 PRINT_MATRIX:
     PUT FILE(LISTNG) EDIT
         (HMATRX,'MATRIX',SETNO)
         (SKIP,X(50),A(12),A(7),A(2));
     PUT FILE(LISTNG) SKIP;
   DO;
     DO I=1 TO 9;
         COLNO=COLNO+1;
     PUT FILE(LISTNG) EDIT
         ('COL',COLNO,(MATRX6(I,J) DO J=1 TO 9))
             (SKIP,A(3),F(2),9E(14,6));
     END;
     PUT FILE(LISTNG) EDIT
         (HMATRX,'VECTOR',SETNO)
         (SKIP(2),X(50),A(12),A(7),A(2));
     PUT FILE(LISTNG) EDIT
         ((SVECTOR(I) DO I=1 TO 9))
         (SKIP(2),9E(14,6));
     PUT FILE (LISTNG) SKIP;
     GO TO RETRNP;
     END;
PRINTOUT:
     PUT FILE(LISTNG) EDIT (HMATRX,'MATRIX',SETNO)
     (SKIP,X(50),A(12),A(7),A(2));
     PUT FILE(LISTNG) SKIP;
     DO J=1 TO 4;
     COLNO=COLNO+1;
     PUT FILE(LISTNG) EDIT
         ('COL',COLNO,(MATRX6(J,K) DO K=1 TO 4)) (SKIP,
         X(25),A(3),F(2),4E(14,6));
     END;
     PUT FILE(LISTNG) EDIT (HMATRX,'VECTOR', SETNO)
         (SKIP(2),X(50),A(12),A(7),A(2));
     PUT FILE(LISTNG) EDIT
     ((AQ(J) DO J=1 TO 4)) (SKIP(2),X(25), 4E(14,6));
     PUT FILE(LISTNG) SKIP;
     GO TO RETRNP;
ENDRT:
         CLOSE FILE(DISK1);
     CLOSE FILE(LISTNG), FILE(CARDS);
        END;     /* COEFFC */
/*   END OF DATA
// EXEC PL/I
* PROCESS LISTO,NOOPT
     PAGEHD:PROCEDURE;
        DECLARE
     LISTNG FILE OUTPUT PRINT
         ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),
    1 START EXTERNAL,
     2 C FIXED DECIMAL (1),
     2 STA FIXED DECIMAL(4),
    (2 KCA,
     2 KCB,
   2 FREQ) DECIMAL FLOAT(7),
      2 CALIBD FIXED DECIMAL(6),                                          11
      2 P FIXED DECIMAL(2),                                               12
      2 BLANK CHARACTER(1),                                               13
         TODAY CHARACTER(6),                                              14
.DATATYPE CHARACTER(10),
     1 DAY DEFINED TODAY,                                                 15
     (2 Y,                                                                16
      2 M,                                                                17
      2 D) CHARACTER(2);                                                  18
      TODAY=DATE;                                                         20
      IF START.C=1 THEN DATATYPE='      POLAR';
           ELSE DATATYPE='EQUATORIAL';
      PUT FILE(LISTNG) PAGE EDIT
      (DATATYPE,'CALIBRATION',DAY.M,'/',DAY.D,'/',DAY.Y)                  22
      (X(5),A(10),X(1),A(11),X(5),A(2),A(1),A(2),A(1),A(2));              23
      PUT FILE(LISTNG) EDIT ('STA',START.STA,'KC A =',                    24
      KCA, 'KC B =', KCB,'FREQ =', FREQ, 'MC ',                           25
      'DATE OF CALIBRATION',START.CALIBD)                                 26
      (SKIP(2),X(10),A(3),X(1),F(4),X(3),A(6),F(6,3),X(3),                27
      A(6),F(6,3),X(3),A(6),F(7,3),A(3),X(1),A(19),X(1),F(6));            28
      PUT FILE(LISTNG) SKIP(2);                                           29
    END;   /* PAGEHD */                                                   31
/*   END OF DATA
 PHASE DR01E2,DR01E1
// EXEC PL/I
* PROCESS LISTO,NOOPT
  PRINT1:PROCEDURE(ACO,SAVEA);                                            0626
DECLARE                                                                   0627
      LISTNG FILE OUTPUT PRINT                                            0628
           ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),                       0629
      PUNCH FILE OUTPUT STREAM
      ENVIRONMENT(MEDIUM(SYSPCH,2540) F(80)),                             0631
 1 START EXTERNAL,                                                        0632
  2 C  FIXED DECIMAL(1),                                                  0633
  2 STA FIXED DECIMAL(4),                                                 0634
 (2 KCA,                                                                  0635
  2 KCB,                                                                  0636
    2 FREQ) DECIMAL FLOAT(7),
  2 CALIBD FIXED DECIMAL(6),                                              0638
  2 P FIXED DECIMAL (2),                                                  0639
  2 BLANK CHARACTER (1),                                                  0640
      MATRIX(48) DECIMAL FLOAT(16),                                        641
      CARDSTA1 FIXED DECIMAL(4),                                           642
      CARDNO FIXED DECIMAL(5),                                             644
     (EXP,MODEXP) FIXED DECIMAL(2),                                        645
      CARDD1 FIXED DECIMAL(6);                                            0647
DECLARE (ACO(47), SAVEA(47),COEFF,COLOG) DECIMAL FLOAT(16);               0648
DECLARE                                                                    649
      SETTYPE(2) CHARACTER(4),                                            0643
      SETNO CHARACTER(2),                                                  650
      ALPHAC (9) CHARACTER(5),                                             651
     (HEAD6, HEAD7) CHARACTER(5),                                          652
      (HEAD8(2), HEAD9(2)) CHARACTER(41),                                  653
      HEAD10(2) CHARACTER(19),                                             654
      HEAD1(4) CHARACTER(42),                                             0655
      HEAD2(6) CHARACTER(35);                                             0656
      DECLARE  MOD1  PICTURE 'SZZ',                                       0657
               PRTMOD CHARACTER(3) DEFINED MOD1;                          0658
      OPEN FILE(PUNCH),FILE(LISTNG);                                      0659
HEAD1(1)='C0 + C1 X + C2 Y                           ';                   0660
```

```
HEAD1(2)='C0 + C1 X + C2 Y + C3 XY + C4 X*2 + C5 Y*2';
HEAD1(3)='C0 + C1 X + C2 Y + C7 SIN(X) + C8 COS(X)   ';
HEAD1(4)='C0 + C1 X + C2 Y + C7 SIN(Y) + C8 COS(Y)   ';
HEAD2(1)='                                         ';
HEAD2(3)=' + C7 SIN (X) + C8 COS (X)               ';
HEAD2(4)=' + C7 SIN (Y) + C8 COS (Y)               ';
IF START.C=1 THEN DO;
HEAD2(2)=' + C6 X*3                                ';
HEAD2(5)=' + C6 X*3 + C7 SIN (X) + C8 COS (X)';
HEAD2(6)=' + C6 X*3 + C7 SIN (Y) + C8 COS (Y)';
                 END;
ELSE DO;
HEAD2(2)=' + C6 Y*3                                ';
HEAD2(5)=' + C6 Y*3 + C7 SIN (X) + C8 COS (X)';
HEAD2(6)=' + C6 Y*3 + C7 SIN (Y) + C8 COS (Y)';
END;
      HEAD6='X 10*';
      HEAD7='C0 + ';
   HEAD8(1)='C9 (EN/R  *(-SIN B)- (1-E2)/R   * COS B) +';
   HEAD8(2)='C9 (EN/R  * (COS B) + (1-N2)/R   * SIN B)+';
   HEAD9(1)='C10 ((1-E2)/R   * SIN B - EN/R   * COS B)   ';
   HEAD9(2)='C10 (-EN/R   * SIN B + (1-N2)/R   * COS B) ';
      HEAD10(1)='+ C11 DEL A0/DEL UT';
      HEAD10(2)='+ C11 DEL B0/DEL UT';
  ALPHAC(1)='C0 =';
  ALPHAC(2)='C1 =';
  ALPHAC(3)='C2 =';
  ALPHAC(4)='C3 =';
  ALPHAC(5)='C4 =';
  ALPHAC(6)='C5 =';
  ALPHAC(7)='C6 =';
  ALPHAC(8)='C7 =';
  ALPHAC(9)='C8 =';
      SETTYPE(1)='A = ';
      SETTYPE(2)='B = ';
                              /* PRINT EQUATIONS AND COEF */
      GO TO EQUATIONS;
HEADNG:PROCEDURE;
      PUT FILE(LISTNG) EDIT (SETTYPE(K1),HEAD1(KK),HEAD2(JJ))
      (X(5),A(4),A(42),A(35));
     END;
                          /*ROUTINES FOR PRINTING EQUATIONS */
COMMON:PROCEDURE;
      PUT FILE (LISTNG) SKIP;
      DO J=1 TO NUM;
      I=I+1;
        IF K1=1 THEN DO;
      COLOG=ABS(SAVEA(I)); COEFF=SAVEA(I); END;
      ELSE DO;COLOG=ABS(ACO(I)); COEFF=ACO(I); END;
      IF COLOG=0 THEN GO TO PRINTC;
                               /* FIND EXPONENT */
 LOG1:EXP=LOG10(COLOG);
      IF COLOG >=1 THEN EXP=EXP+1;
           ELSE;
      COEFF=COEFF*10**(-EXP);
      IF (J=1)|(J=8) | (J=9) THEN MODEXP=EXP+3;
           ELSE IF J=7 THEN MODEXP=EXP-6;
                ELSE IF (J=2)|(J=3) THEN MODEXP=EXP;
                     ELSE MODEXP=EXP-3;
      IF I <39 THEN DO;
     MOD1=MODEXP;
               END;
     ELSE PRTMOD='  ';
PRINTC:
     PUT FILE(LISTNG) EDIT (ALPHAC(J),COEFF,HEAD6,EXP,PRTMOD)
              (SKIP,X(10),A(5),F(10,8),X(2),A(5),F(2),
               X(5),A(3));
     IF I=19 THEN J=7;
          ELSE IF I=27 THEN J=J+1;
            ELSE IF I=46 THEN J=J+2;
               ELSE;
   END;                       /* COMMNB */
     PUT FILE(LISTNG) SKIP(2);
   END;                                 /* COMMON */
  EQUATIONS:
     CALL PAGEHD;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 1')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     K1=1;
     I=0; NUM=3; KK=1; JJ=1; CALL HEADNG; CALL COMMON;
     I=0; K1=2; CALL HEADNG; CALL COMMON;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 2')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     NUM=6; K1=1; KK=2; CALL HEADNG; CALL COMMON;
     I=3; K1=2; CALL HEADNG; CALL COMMON;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 3')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     NUM=7; K1=1; JJ=2; CALL HEADNG; CALL COMMON;
     I=9; K1=2; CALL HEADNG; CALL COMMON;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 4')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     NUM=9; K1=1; KK=3; JJ=1; CALL HEADNG; CALL COMMON;
     I=16; K1=2; KK=4; CALL HEADNG; CALL COMMON;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 5')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     K1=1; KK=2; JJ=3; CALL HEADNG; CALL COMMON;
     I=21; K1=2; JJ=4; CALL HEADNG; CALL COMMON;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 6')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
     K1=1; JJ=5; CALL HEADNG; CALL COMMON;
     K1=2; I=29; JJ=6; CALL HEADNG; CALL COMMON;
                         /* PRINT HEADINGS AND COEFFICIENTS
                            FOR THE ADDITIONS  */
     ALPHAC(2)='C9 =';
     ALPHAC(3)='C10 =';
     ALPHAC(4)='C11 =';
     CALL PAGEHD;
     PUT FILE(LISTNG) EDIT ('EQUATION SET 7')
     (SKIP,X(2),A);
     PUT FILE(LISTNG) SKIP;
CHEAD7:
     NUM=4; K1=1; I=38;
     DO II=1 TO 2;
     PUT FILE (LISTNG) EDIT
```

```
       (SETTYPE(K1),HEAD7,HEAD8(K1),HEAD9(K1),HEAD10(K1))
    (X(5),A(4),A(5),A(41),X(1),A(41),A(19));
    CALL COMMON; I=38; K1=2;
    END;
    PUT FILE(LISTNG) EDIT ('EQUATION SET 8')
    (SKIP,X(2),A);
    PUT FILE(LISTNG) SKIP;
CHEAD8:
    NUM=3; K1=1; I=42;
    DO I1=1 TO 2;
    PUT FILE (LISTNG) EDIT
        (SETTYPE(K1),HEAD7,HEAD8(K1),HEAD9(K1))
        (X(5),A(4),A(5),A(41),X(1),A(41));
    CALL COMMON; I=42; K1=2;
    END;
    PUT FILE(LISTNG) EDIT ('EQUATION SET 9')
    (SKIP,X(2),A);
    PUT FILE(LISTNG) SKIP;
    NUM=4; K1=1; I=45;
CHEAD9:
    HEAD7='CO   ';
    DO I1=1 TO 2;
    PUT FILE (LISTNG) EDIT
        (SETTYPE(K1),HEAD7,HEAD10(K1))
            (X(5), A(4), A(5),A(19));
    CALL COMMON; I=45; K1=2;
    END;
PUNCHO:
                        /*COEFFICIENT CARDS */
    CARDSTA1=START.STA;
    CARDD1=START.CALIBD;
    SETNO=' A'; CARDNO=0; I1=1; MATRIX(48)=0;
    DO I=1 TO 47; MATRIX(I)=SAVEA(I); END;
    DO K=1 TO 2;
    PUT FILE(PUNCH) EDIT
        (CARDSTA1,CARDD1, (MATRIX(I) DO I=1 TO 3), SETNO)
    (F(4),X(2),F(6),X(18),3E(15,8,9),X(3),A(2));
        I1=4;
        I2=8;
    DO J=1 TO 9;
    CARDNO=CARDNO+1;
    PUT FILE(PUNCH) EDIT
        ((MATRIX(J1) DO J1=I1 TO I2), CARDNO)
    (5E(15,8,9),F(5));
    I1=I1+5; I2=I2+5;
    END;
    CARDNO=0; SETNO='B';
    IF K=1 THEN DO I2=1 TO 47;MATRIX(I2)=ACO(I2); END;
            END;
    CLOSE FILE(LISTNG),FILE(PUNCH);
       END;  /* PRINT1 */
/*   END OF DATA
// EXEC PL/I
* PROCESS LISTG,NOOPT
     PAGEHD:PROCEDURE;
       DECLARE
            LISTNG FILE OUTPUT PRINT
         ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),
    1 START EXTERNAL,
     2 C FIXED DECIMAL (1),
     2 STA FIXED DECIMAL(4),
    (2 KCA,
     2 KCB,
   2 FREQ) DECIMAL FLOAT(7),
     2 CALIBD FIXED DECIMAL(6),
     2 P FIXED DECIMAL(2),
     2 BLANK CHARACTER(1),
        TOUAY CHARACTER(6),
 UATATYPE CHARACTER(10),
    1 DAY DEFINED TODAY,
    (2 Y,
     2 M,
     2 D) CHARACTER(2);
     TUDAY=DATE;
     IF START.C=1 THEN DATATYPE='      POLAR';
          ELSE DATATYPE='EQUATORIAL';
     PUT FILE(LISTNG) PAGE EDIT
     (DATATYPE,'CALIBRATION',DAY.M,'/',DAY.D,'/',DAY.Y)
     (X(5),A(10),X(1),A(11),X(5),A(2),A(1),A(2),A(1),A(2));
     PUT FILE(LISTNG) EDIT ('STA',START.STA,'KC A =',
     KCA, 'KC B =', KCB,'FREQ =', FREQ, 'MC ',
     'DATE OF CALIBRATION',START.CALIBD)
     (SKIP(2),X(10),A(3),X(1),F(4),X(3),A(6),F(6,3),X(3),
     A(6),F(6,3),X(3),A(6),F(7,3),A(3),X(1),A(19),X(1),F(6));
     PUT FILE(LISTNG) SKIP(2);
   END:    /* PAGEHD */
/*   END OF DATA
 PHASE DR01E3,DR01E2
// EXEC PL/I
* PROCESS LISTO,NOOPT
 RESIDL:PROCEDURE(ACU,SAVEA,NPLOTS);
 DECLARE
      LISTNG FILE OUTPUT PRINT
           ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),
      DISK1 FILE INPUT RECORD
           ENVIRONMENT(MEDIUM(SYS003,2314) F(3400,136) ),
      DISK2 FILE OUTPUT RECORD
           ENVIRONMENT(MEDIUM(SYS004,2314) U(96) ),
    DISK3 FILE INPUT RECORD
           ENVIRONMENT(MEDIUM(SYS004,2314) U(96) ),
      PLOTDK FILE OUTPUT RECORD
           ENVIRONMENT(MEDIUM(SYS006,2314) F(1208) ),
 1 START EXTERNAL,
  2 C FIXED DECIMAL(1),
  2 STA FIXED DECIMAL(4),
 (2 KCA,
  2 KCB,
  2 FREQ) DECIMAL FLOAT(7),
  2 CALIBD FIXED DECIMAL(6),
  2 P FIXED DECIMAL(2),
  2 BLANK CHARACTER(1),
1 DATA,
 2 UT (3) DECIMAL FLOAT,
 2 BLANKS CHARACTER(4),
(2 AC,
 2 BC,
 2 AM1,
 2 BM1,
 2 BETA,
 2 R2,
 2 DELAO,
 2 DELBO,
 2 NAP,
 2 NBP) DECIMAL FLOAT(16),
 2 STA CHARACTER(4),
 2 PLATE CHARACTER(3),
 2 PLOTCODE CHARACTER(1),
(2 RA,
 2 RB,
 2 AM,
 2 BM) DECIMAL FLOAT(16),
1 DK,
 2 TBLNK CHARACTER(4),
 2 S1 CHARACTER(4),
 2 NAME CHARACTER(2),
 2 PL1 CHARACTER (3),
 2 BLNK CHARACTER(3),
 2 T1 DECIMAL FLOAT(16),
(2 C1,
 2 C2,
 2 M1,
 2 M2,
 2 R1,
 2 R2,
 2 R3(6)) DECIMAL FLOAT,
 1 PDATA,
  (2 PLOTX(100),
   2 PLOTY1(100),
   2 PLOTY2(100)) DECIMAL FLOAT(6),
   2 NPTS FIXED BINARY(31),
   2 PLATE CHARACTER(3),
   2 PLOTCODE CHARACTER(1);
 DECLARE                                  /* VARIABLES */
NPLOTS(30) FIXED BINARY(31),
      ROUND FIXED DECIMAL(5,4),
 (PRA(9),PRB(9),PRTRA,PRTRB) FLOAT(6),
(A1(9), B1(9), SAVE(47),SAVEA(47),ACO(47),RA(9),RB(9),
 RESSUM(20)) DECIMAL FLOAT(16),
 RETRNP LABEL;
 DECLARE
 (DUMMY1,
  TIME,
  XA,X2A,XYA,
  YA,Y2A,Y3A,
  SIN1,SIN2,COS1,COS2,
  ANGX,ANGY,
  EDATA,B9Q,B10Q,
  NDATA,
  ESQAR,
  NSQAR,
  ENR2,
  ONEE2,
  ONEN2,
  COSB,
  SINB,
  A9Q,A10Q,
  PKCA,PKCB,PFREQ,
  SUM) DECIMAL FLOAT(16),
 (WHOLEX,WHOLEY) FIXED DECIMAL(7),
  LAST_PLATE CHARACTER(3),
  TODAY CHARACTER(6),
 1 DAY DEFINED TODAY,
  2 Y CHARACTER(2),
  2 M CHARACTER(2),
  2 D CHARACTER(2),
 DATATYPE CHARACTER(10),
  RETRNC LABEL,
 RETRNH LABEL;
DECLARE (TRUNCRA,TRUNCRB) FIXED DECIMAL(3);
                                        /*COMPUTE RESIDUALS */
     DK.TBLNK='     ';
      TODAY=DATE;
     IF START.C=1 THEN DATATYPE='      POLAR';
         ELSE DATATYPE='EQUATORIAL';
     KK=1;
     RETRNH=DATAHDNG;
     ROUND=.0005;
     TEN3=1000.0;
     RETRNP=BEGIN1; GO TO PAGEHD;
 BEGIN1:
     NPLTCODE=0; NPLT=0; C01=0;
     BLNK='    '; DUMMY=0; N2=1;
     OPEN FILE(DISK2),FILE(DISK1),FILE(PLOTDK),FILE(LISTNG);
     READ FILE(DISK1) INTO(DATA);
     NPLOTS=0; RESSUM=0;
     DO I=1 TO 38; SAVE(I)=SAVEA(I); END;
     ON ENDPAGE(LISTNG) GO TO PAGEHD;
         ON ENDFILE(DISK1) GO TO DATAEND;
     CALL READD; GO TO DATAR;
READD:PROCEDURE;
     READ FILE(DISK1) INTO (DATA);
         ON ENDFILE(DISK1) GO TO DATAEND;
     END;     /* READD */
CFS:PROCEDURE;
     B1(1)=SAVEA(1)+XA*SAVEA(2)+YA*SAVEA(3);
     B1(2)=SAVEA(4)+XA*SAVEA(5)+YA*SAVEA(6)+XA*YA*SAVEA(7)
                  +X2A*SAVEA(8)+Y2A*SAVEA(9);
     B1(3)=SAVEA(10)+XA* SAVEA(11)+YA*SAVEA(12)+XYA*SAVEA(13)
           +X2A*SAVEA(14)+Y2A*SAVEA(15)+Y3A*SAVEA(16);
     B1(4)=SAVEA(17)+XA*SAVEA(18)+YA*SAVEA(19)+SIN1*SAVEA(20)
           +COS1*SAVEA(21);
     B1(5)=SAVEA(22)+XA*SAVEA(23)+YA*SAVEA(24)+XYA*SAVEA(25)
           +X2A*SAVEA(26)+Y2A*SAVEA(27)+SIN1*SAVEA(28)+
            COS1*SAVEA(29);
     B1(6)=SAVEA(30)+XA*SAVEA(31)+YA*SAVEA(32)+XYA*SAVEA(33)+
           X2A*SAVEA(34)+Y2A*SAVEA(35)+Y3A*SAVEA(36)+
           SIN1*SAVEA(37)+COS1*SAVEA(38);
     END;   /* COMPUTE_CFS */
 DATAR:
     XA=DATA.AM-START.KCA;
     YA=DATA.BM-START.KCB;
     WHOLEX=TRUNC(XA);
     WHOLEY=TRUNC(YA);
     XYA=XA*YA;
     X2A=XA*XA;
     Y2A=YA*YA;
     IF START.C=1 THEN Y3A=X2A*XA;
     ELSE Y3A=Y2A*YA;
     ANGX=(XA-WHOLEX)*6.2831854;
     ANGY=(YA-WHOLEY)*6.2831854;
```

```
SIN1=SIN(ANGX);
SIN2=SIN(ANGY);
COS1=COS(ANGX);
COS2=COS(ANGY);
BETA=1.5707963-BETA;
COSB=COS(BETA);
SINB=SIN(BETA);
EDATA=DATA.AC/DATA.NAP;
NDATA=DATA.BC/DATA.NBP;
ESQAR=EDATA*EDATA;
NSQAR=NDATA*NDATA;
ENR2=(EDATA*NDATA)/DATA.R2;
ONEN2=(1-NSQAR)/DATA.R2;
ONEE2=(1-ESQAR)/DATA.R2;
DO I=1 TO 38; SAVEA(I)=SAVE(I); END;
CALL CFS;
DO I=1 TO 6; A1(I)=B1(I); END;
A9Q=(ENR2*(-SINB)-(ONEE2*COSB));
A10Q=((ONEE2*SINB)-(ENR2*COSB));
A1(7)=SAVEA(39)+SAVEA(40)*A9Q+SAVEA(41)*A10Q
       +SAVEA(42)*DELA0;
A1(8)=SAVEA(43)+SAVEA(44)*A9Q+SAVEA(45)*A10Q;
A1(9)=SAVEA(46)+SAVEA(47)*DELA0;
B9Q=((ENR2*COSB)+(ONEN2*SINB));
B10Q=(-ENR2*SINB)+(ONEN2*COSB);
B1(7)=ACO(39)+ACO(40)*B9Q+ACO(41)*B10Q+ACO(42)*DELB0;
B1(8)=ACO(43)+ACO(44)*B9Q+ACO(45)*B10Q;
B1(9)=ACO(46)+ACO(47)*DELB0;
SIN1=SIN2; COS1=COS2;
DO I=1 TO 38; SAVEA(I)=ACO(I); END;
CALL CFS;
CO1=CO1+1;
DATA.RA=DATA.AC+START.KCA-DATA.AM;
DATA.RB=DATA.BC+START.KCB-DATA.BM;
DO I=1 TO 6;
      RA(I)=DATA.AC-A1(I);
      RB(I)=DATA.BC-B1(I);
END;
DO I=7 TO 9; RA(I)=DATA.RA-A1(I);RB(I)=DATA.RB-B1(I);
END;
DO I=1 TO 9;
PRA(I)=RA(I)*TEN3;
PRB(I)=RB(I)*TEN3;
END;
PRTRA=DATA.RA*TEN3;
PRTRB=DATA.RB*TEN3;
IF START.P=0 THEN DO;
PLOTRA =(DATA.RA+ROUND*SIGN(DATA.RA))*TEN3;
PLOTRB =(DATA.RB+ROUND*SIGN(DATA.RB))*TEN3;
TRUNCRA=TRUNC(PLOTRA);
TRUNCRB=TRUNC(PLOTRB);
NPLOTS(KK)=NPLOTS(KK)+1;
PDATA.PLOTY1(N2)=TRUNCRA;
PDATA.PLOTY2(N2)=TRUNCRB;
IF DATA.PLOTCODE='1' THEN PDATA.PLOTX(N2)=DATA.AC;
          ELSE PDATA.PLOTX(N2)=DATA.BC;
N2=N2+1; END; ELSE;
J=1;
DO I=2 TO 6 BY 2;
R3(I-1)=PRA(6+J); R3(I)=PRB(6+J); J=J+1; END;
```

```
      RESSUM(1) =RESSUM(1)+DATA.RA*DATA.RA;   /*SUM RESIDUAL */        1014
      RESSUM(2) =RESSUM(2)+DATA.RB*DATA.RB;   /*SQUARES (RMS)*/        1015
      J=3;                                                             1016
      DO I=1 TO 9;                                                     1017
      RESSUM(J)=RESSUM(J)+RA(I)*RA(I);                                 1018
      RESSUM(J+1)=RESSUM(J+1)+RB(I)*RB(I);                             1019
      J=J+2; END;                                                      1020
      PDATA.PLOTCODE=DATA.PLOTCODE;                                    1021
      LAST_PLATE=DATA.PLATE;                                           1022
      RETRNP=DATALIST;                                                 1023
 DATALIST:                                                             1024
      DATA.STA='    ';
      IF N2=1 | N2=2 THEN DO;                                          1025
      DK.PL1=DATA.PLATE;                                               1026
      DK.NAME='PL';                                                    1027
      DK.S1 =DATA.STA;                                                 1028
      PUT FILE(LISTNG) EDIT                                            1029
      (DATA.STA,'PL',DATA.PLATE)
      (SKIP,A(4),A(2),A(3));
      IF START.P=1 THEN N2=3; END;                                     1032
      ELSE DO;                                                         1033
      PL1='   '; S1='    '; NAME='  '; END;                            1034
      TIME= DATA.UT(1)*10000+DATA.UT(2)*100+ DATA.UT(3);               1035
      T1=TIME;                                                         1042
      DK.R1=PRTRA;                                                     1043
      DK.R2=PRTRB;                                                     1044
      DK.C1 =DATA.AC;                                                  1045
      DK.C2 =DATA.BC;                                                  1046
      DK.M1 =DATA.AM;                                                  1047
      DK.M2 =DATA.BM;                                                  1048
      RETRNP=PRINT1;                                                   1049
 PRINT1:                                                               1050
      PUT FILE(LISTNG) EDIT                                            1051
      (TIME, DATA.AC,DATA.BC,DATA.AM,DATA.BM,PRTRA,                    1052
             PRTRB,(PRA(I),PRB(I) DO I=1 TO 6))                        1053
      (SKIP,X(10), F(12,1),4F(9,3),14F(5));                            1054
      WRITE FILE(DISK2) FROM(DK);                                      1055
 CHECKPL:                                                              1056
      CALL READD;                                                      1057
      IF DATA.PLATE=LAST_PLATE THEN GO TO DATAR;                       1058
      ELSE DO;RETRNC=DATAR;GO TO PLOTCK;END;                           1059
 PAGEHD:                                                               1060
      PUT FILE(LISTNG) PAGE EDIT                                       1061
           (DATATYPE,'CALIBRATION',DAY.M,'/',DAY.D,'/',DAY.Y)          1062
           (X(10),A(10),X(1),A(11),X(5),A(2),A(1),A(2),A(1),           1063
            A(2));                                                     1064
      PUT FILE(LISTNG) EDIT                                            1065
      ('STA', START.STA,'KC A =',KCA,'KC B =',KCB,'FREQ =',            1065
       FREQ, 'MC ','DATE OF CALIBRATION', START.CALIBD)                1066
      (SKIP(2),X(10),A(3),X(1),F(4),X(3),A(6),F(6,3),X(3),             1067
      A(6),F(6,3),X(3),A(6),F(7,3),A(3),X(3),A(19),X(1),F(6));
      PUT FILE(LISTNG) SKIP;                                           1069
      GO TO RETRNH;                                                    1070
 DATAHDNG:                                                             1071
      PUT FILE(LISTNG) EDIT                                            1072
      ('UT         AC        BC         AM         BM        RA',      1073
           'RB    RA1   RB1   RA2   RB2   RA3   RB3   RA4   RB4',      1074
                 'RA5   RB5   RA6   RB6')                              1075
      (SKIP,X(20),A(43),X(3),A(43),X(3),A(18));                        1076
      PUT FILE (LISTNG) SKIP;                                          1077
```

```
     GO TO RETRNP;
DATAHDNG2: PUT FILE(LISTNG) EDIT
     ('UT       AC         BC          AM          BM',
      'RA   RB  RA7  RB7  RA8  RB8  RA9  RB9')
     (SKIP,X(33),A(40),X(7),A(37));
     PUT FILE (LISTNG) SKIP;
     GO TO RETRNP;
 PLOTCK:
     IF START.P=0 THEN DO;
     PDATA.PLATE=LAST_PLATE;
     WRITE FILE(PLOTDK) FROM(PDATA);
     KK=KK+1; N2=1; END; ELSE;
     N2=1;
     GO TO RETRNC;
 DATAEND:
     DO I=1 TO 20; RESSUM(I)=(SQRT(RESSUM(I)/CO1));END;
     CLOSE FILE(DISK2);
     IF START.P=0 THEN DO; RETRNC=LASTLINE;GO TO PLOTCK;END;
 LASTLINE:
     PUT FILE(LISTNG) PAGE EDIT
          ('NO.        RMS      RMS      RMS      RMS      RMS',
           'RMS      RMS      RMS      RMS      RMS      RMS',
           'RMS      RMS      RMS')
          (SKIP(2),A(45),X(5),A(43),X(5),A(19));
     PUT FILE(LISTNG) EDIT
          ('PTS        RA       RB       RA1      RB1      RA2',
           'RB2      RA3      RB3      RA4      RB4      RA5',
           'RB5      RA6      RB6')
          (SKIP,A(45),X(5),A(43),X(5),A);
     PUT FILE(LISTNG) EDIT
          (CO1,(RESSUM(I) DO I=1 TO 14))
          (SKIP,F(5),X(2),14F(8,6));
     OPEN FILE(DISK3);
     RETRNH=DATAHDNG2;
     RETRNP=SECONDLIST;
     GO TO PAGEHD;
SECONDLIST:
     READ FILE(DISK3) INTO (DK);
     RETRNP=PRINT2;
 PRINT2:
     PUT FILE(LISTNG) EDIT (DK)
     (SKIP,X(6),A(4), A(4),A(2),A(4),A(4),F(12,1),
          4F(10,3),8F(5));
     ON ENDFILE(DISK3) GO TO ENDRT;
     GO TO SECONDLIST;
 ENDRT:
     ON ENDPAGE(LISTNG);
     PUT FILE(LISTNG) EDIT
     ('NO.      RMS       RMS       RMS       RMS       RMS',
     'RMS') (SKIP(2),X(30),A(48),X(6),A(3));
     PUT FILE(LISTNG) EDIT
     ('PTS      RA7       RB7       RA8       RB8       RA9',
     'RB9') (SKIP,X(30),A(48),X(6),A(3));
     PUT FILE(LISTNG) EDIT (CO1,(RESSUM(I) DO I=15 TO 20))
     (SKIP,X(28),F(5),6F(9,6));
     CLOSE FILE(DISK1), FILE(DISK3),FILE(PLOTDK);
     CLOSE FILE(LISTNG);
       END; /* RESIDL */
/*   END OF DATA
 PHASE DR01E4,DR01E3
```

```
// EXEC PL/I
* PROCESS LISTO,NOOPT
DRZPLT:PROCEDURE(NPLOTS);                                                       1139
 DECLARE                        /* DRXO1E PLOTS   */                            1140
      LISTNG FILE OUTPUT PRINT                                                  1141
           ENVIRONMENT(MEDIUM(SYSLST,1403) F(132)),                             1142
      PLOTDK FILE INPUT RECORD                                                  1143
           ENVIRONMENT(MEDIUM(SYS006,2314) F(1208) ),
 1 START EXTERNAL,                                                              1145
  2 C FIXED DECIMAL(1),                                                         1146
  2 STA FIXED DECIMAL(4),                                                       1147
 (2 KCA,                                                                        1148
  2 KCB,                                                                        1149
  2 FREQ) DECIMAL FLOAT(7),                                                     1150
  2 CALIBD FIXED DECIMAL(6),                                                    1151
  2 P FIXED DECIMAL (2),                                                        1152
  2 BLANK CHARACTER(1),                                                         1153
 1 DATA,                                                                        1154
  (2 PLOTX(100),                                                                1155
   2 PLOTY1(100),                                                               1156
   2 PLOTY2(100)) DECIMAL FLOAT(6),                                             1157
   2 PTS FIXED BINARY(31),                                                      1158
   2 PLATE CHARACTER (3),                                                       1159
   2 PLOTCODE CHARACTER(1);                                                     1160
DECLARE                                                                         1161
      BUFFER(1000) FLOAT,                                                       1162
     (NPLOTS(30),NPTS) FIXBD BINARY (31),                                       1163
      FPL CHARACTER (3),                                                        1164
  GRIDSYM(3) CHARACTER(2),                                                      1165
 (Y1SC(5),                                                                      1166
  Y1S2(5),                                                                      1167
      X1SC(13)),                                                                1168
     (ALEN, CALIBD) FLOAT,                                                      1166
     (N,N12,NBUF,NC,SYS13,N1,N2,N3,N13,N22,N23,                                 1167
      N99) FIXED BINARY (31),                                                   1168
     (YSCALE,XSCALE) CHARACTER(1),                                              1169
      PA CHARACTER (5);                                                         1170
BEGINPROC:                                                                      1171
      OPEN FILE(PLOTDK);                                                        1185
      KK=0;                                                                     1186
      GRIDSYM(2)='RA';                                                          1187
      GRIDSYM(3)='RB';                                                          1188
   X1SC(1)=-5;                                                                  1189
   X1SC(2)=-4;                                                                  1190
   X1SC(3)=-3;                                                                  1191
   X1SC(4)=-2;                                                                  1192
   X1SC(5)=-1;                                                                  1193
   X1SC(6)=0;                                                                   1194
   X1SC(7)=1;                                                                   1195
   X1SC(8)=2;                                                                   1196
   X1SC(9)=3;                                                                   1197
  X1SC(10)=4;                                                                   1198
  X1SC(11)=5;                                                                   1199
      Y1SC(1)=-30;                                                              1200
      Y1SC(2)=-20;                                                              1201
      Y1SC(3)=-10;                                                              1202
      Y1S2(1)=10;                                                               1203
      Y1S2(2)=20;                                                               1
      Y1S2(3)=30;                                                               1204
      YSCALE=' ';                                                               1205
      XSCALE=' ';
      N1=1;
      N2=2;
      N3=3;
      N23=23;
      N22=22;
      N13=13;
      N12=12;
      NBUF=1000;
      SYS13=13;
      CALL PLOTS(BUFFER,NBUF,SYS13);
      FACT =0.5;
                                   /* OPERATOR ORIGIN*/
      FSTA=START.FREQ;
      PA='PLATE';
      CALIBD=START.CALIBD;
      XOFF=-5.0;
      XFCTR=1;
      YOFF=-30;
      YFCTR=10;
      CALL OFFSET(XOFF,XFCTR,YOFF,YFCTR);
BEGIN_PLOT:
      XP=0;
      YP=0;
      CALL PLOT (XP,YP,N23);
      KK=KK+1;
      NPTS=NPLOTS(KK);
      READ FILE(PLOTDK) INTO (DATA);
      ON ENDFILE(PLOTDK) GO TO PLOTEXIT;
      IF DATA.PLOTCODE='1' THEN
                     GRIDSYM(1)='AC';
                          ELSE GRIDSYM(1)='BC';
      FPL=DATA.PLATE;
                               /* FIRST PLOT GRIDS */
      XP=0;
      YP=3;
      CALL PLOT(XP,YP,N3);
      XP=10;
      CALL PLOT(XP,YP,N2);
      YP=0;
      XP=5;
      CALL PLOT(XP,YP,N3);
      YP=6;
      CALL PLOT(XP,YP,N2);
                               /* XAXIS */
      KP=1;
      ALEN=10;
      NC=11;
      CALL SCALE(X1SC,ALEN,NC,KP);
      XP=0;
      YP=3;
      CALL PLOT (XP,YP,N3);
      NC=-1;
      AOFF=X1SC(12);
      AFCTR=X1SC(13);
      ANG=0;
      CALL AXIS (XP,YP,XSCALE,NC,ALEN,ANG,AOFF,AFCTR);
      XP=5.1;
      YP=0.0;
      CALL PLOT (XP,YP,N3);
```

```
         HT=.1;
         N=2;
         CALL SYMBOL (XP,YP,HT,GRIDSYM(3),ANG,N);
                                       /* YAXIS */
         XP=5.0;
         YP=0.0;
         CALL PLOT(XP,YP,N3);
         ALEN=2;
         NC=3;
         CALL SCALE(Y1SC,ALEN,NC,KP);
         NC=1;
         ANG=90;
         AOFF=Y1SC(4);
         AFCTR=Y1SC(5);
         CALL AXIS(XP,YP,YSCALE,NC,ALEN,ANG,AOFF,AFCTR);
         YP=4;
         CALL PLOT (XP,YP,N3);
         NC=3;
         CALL SCALE(Y1S2,ALEN,NC,KP);
         NC=1;
         AOFF=Y1S2(4);
         AFCTR=Y1S2(5);
         CALL AXIS(XP,YP,YSCALE,NC,ALEN,ANG,AOFF,AFCTR);
         YP=2.8;
         XP=9.5;
         CALL PLOT (XP,YP,N3);
         N=2;
         ANG=0;
         CALL SYMBOL(XP,YP,HT,GRIDSYM(1),ANG,N);
                                              /* FIRST PLOT ORIGIN */
         XP=0;
         YP=3;
         CALL PLOT (XP,YP,N3);
         I=1;
CHECK1:
         IF ABS(PLOTY2(I)) >30 THEN DO;
         I=I+1; IF I >NPTS THEN GO TO ERROR;
         ELSE GO TO CHECK1; END;
         CALL PLOT(PLOTX(I),PLOTY2(I),N13);
         J=I+1;
         DO I=J TO NPTS;
         IF ABS(PLOTY2(I)) > 30 THEN GO TO PENUP2;
               ELSE DO;
                     CALL PLOT(PLOTX(I),PLOTY2(I),N12);
                     GO TO EXIT2;
                     END;
         PENUP2:
         I=I+1;
         IF I >  NPTS  THEN GO TO SECOND; ELSE
         IF ABS(PLOTY2(I)) >30 THEN GO TO PENUP2;
         CALL PLOT(PLOTX(I),PLOTY2(I),N13);
         EXIT2:
               END;
                                       /* 2ND PLOT GRID */
SECOND:
         XP=0;
         YP=10;
         CALL PLOT(XP,YP,N3);
         XP=10;
         CALL PLOT(XP,YP,N2);
```

```
XP=5;
YP=7;
CALL PLOT(XP,YP,N3);
YP=13;
CALL PLOT(XP,YP,N2);
                              /* DRAW GRIDS */
XP=0;
YP=10;
CALL PLOT (XP,YP,N3);
                              /* SCALE AXIS  */
ANG=0;
AOFF =X1SC(12);
AFCTR=X1SC(13);
ALEN=10;
NC=-1;
KP=1;
CALL AXIS(XP,YP,XSCALE,NC,ALEN,ANG,AOFF,AFCTR);
XP=9.5;
YP=9.8;
CALL PLOT (XP,YP,N3);
CALL SYMBOL(XP,YP,HT,GRIDSYM(1),ANG,N);
                         /* YAXIS SCALE */
XP=5.0;
YP=7;
CALL PLOT(XP,YP,N3);
ANG=90;
AOFF=Y1SC(4);
AFCTR=Y1SC(5);
ALEN=2;
NC=1;
CALL AXIS (XP,YP,YSCALE,NC,ALEN,ANG,AOFF,AFCTR);
AOFF=Y1S2(4);
AFCTR=Y1S2(5);
YP=11;
CALL PLOT(XP,YP,N3);
CALL AXIS (XP,YP,YSCALE,NC,ALEN,ANG,AOFF,AFCTR);
XP=5.1;
YP=7;
CALL PLOT (XP,YP,N3);
ANG=0;
N=2;
CALL SYMBOL(XP,YP,HT,GRIDSYM(2),ANG,N);
                         /* IDENTIFICATION, PLATE BOX */
N=5;
HT=0.2;
ANG=0;
XP=0.5;
YP=13.5;
CALL PLOT (XP,YP,N3);
CALL SYMBOL(XP,YP,HT,PA,ANG,N);
N=-1;
XP=2.0;
CALL PLOT(XP,YP,N3);
N=3;
CALL SYMBOL(XP,YP,HT,FPL,ANG,N);
XP=3.0;
CALL PLOT (XP,YP,N3);
N=3;
CALL NUMBER(XP,YP,HT,FSTA,ANG,N);
XP=1.0;
```

```
       YP=13.0;
       CALL PLOT(XP,YP,N3);
       N=-1;
       CALL NUMBER(XP,YP,HT,CALIBO,ANG,N);
       YP=13.3;
       XP=5.5;
       CALL PLOT(XP,YP,N3);
       XP=7.1;
       CALL PLOT(XP,YP,N2);
       YP=14.5;
       CALL PLOT(XP,YP,N2);
       XP=5.5;
       CALL PLOT(XP,YP,N2);
       YP=13.3;
       CALL PLOT(XP,YP,N2);
                                           /* 2ND PLOT ORIGIN */
       XP=0;
       YP=7;
       CALL PLOT(XP,YP,N23);
       I=1;
CHECK2:
       IF ABS(PLOTY1(I)) >30 THEN DO;
       I=I+1; IF I >NPTS THEN GO TO ERROR;
       ELSE GO TO CHECK2; END;
       CALL PLOT (PLOTX(I),PLOTY1(I),N13);
       J=I+1;
       DO I=J TO NPTS;
       IF ABS(PLOTY1(I)) > 30 THEN GO TO PENUP;
             ELSE DO;
               CALL PLOT (PLOTX(I),PLOTY1(I),N12);
               GO TO EXIT1;
                   END;
       PENUP:
       I=I+1;
       IF I > NPTS THEN GO TO FINISH; ELSE
       IF ABS(PLOTY1(I)) >30 THEN GO TO PENUP;
               CALL PLOT (PLOTX(I),PLOTY1(I),N13);
       EXIT1:
             END;
FINISH:
       YP=10;
       XP=0;
       CALL PLOT(XP,YP,-N23);
       GO TO BEGIN_PLOT;
ERROR:
       PUT FILE(LISTNG) EDIT ('INVALID DATA PLATE',DATA.PLATE)
       (SKIP(2),A(18),X(2),A(3));
       GO TO FINISH;
 PLOTEXIT:
       N99=99;
       YP=0;
       XP=0;
       CALL PLOT (XP,YP,N99);
       CLOSE FILE(PLOTDK);
            END;  /* DRZPLT */
 /*    END OF DATA
 /&    END OF JOB
 // PAUSE SEND CARDS TO ALLIE, LIST TO RAY CHAVES
```

Appendix D

QUADRATIC LISTING

The following pages display a listing of the computer preprocessing program used in the Minitrack system at the beginning of this effort. Functionally, it differs but little from the cubic preprocessing program now in operation. The quadratic program is presented for historical reasons and to assist in the analysis of the flow diagram shown in appendix E. We did not redraw the complete flow diagram for the present cubic preprocessing program because of excessive cost and the fact that only slight, insignificant changes would occur.

```
//G4DJCMIN JOB (G40042311F,P,G00043,030100),NCS,MSGLEVEL=1
// EXEC FORTRAN,PARM='OPT=2,MAP,BCD,DECK,LOAD,LIST'
//SOURCE.SYSLIN DD  DISP=(NEW,PASS)
//SUURCE.SYSIN DD  *
      INTEGER SIG1,SIG2,SIG3,SIG4,SIG5,HOUR,AMIN,SECN,BOUR,ASEC,INO,ISD
      INTEGER AMP,END,CSTA,CANT,SIG,ANTD,HORD,MIND,SECD,SAT,DATE,KFA,KFB
      INTEGER IGRADE
      REAL NSFD,NSCD,NSMD,PID,IFIT
      REAL*8 STATIO
      REAL IDIF1,IDIF2,IDIF3,IDIF4,IDIF5
      REAL NSM,NSC,NSFPO,NSFEQ,IDIF,      NSF1,NSF2,NSF3,NSF4,NSF5
      DIMENSION STATIO(12),KFA(12),KFB(12),EWM(12),CLEWM(12),EWC(12),CLE
     1WC(12),EWFEQ(12),NSM(12),CLNSM(12),NSC(12),CLNSC(12),NSFEQ(12),NSF
     2PO(12),ISTA(48),IANT(48),C1(48),C2(48),C3(48),C4(48),C5(48),C6(48)
     3,C7(48),C8(48),KSAID(50),FREQ(50),KSTA(12),EWFPO(12),TIM(31)
      DIMENSION SECD(31),EWMD(31),EWCD(31),EWFD(31),NSMD(31),NSCD(31),NS
     1FD(31),MIND(31),HORD(31),DAYD(31),ANTD(31),STAD(31),SIGD(31),EEWF
     2(31),ENSF(31),IDAYD(31),AST(31),DATE(12),CO(48)
      DIMENSION ASTA(3),ARMODA(7),AOUR(7),IALOBE(6),IARATE(5),IAACC(4),I
     XAWMER(3),IAWCER(3),IBLOBE(6),IBRATE(5),IBACC(4),IBNMER(3),IBNCER(3
     X),ICIGA(3),IOUR(5),ISEC(6),LCOS(8),MCOS(8),IENO(3),INNO(3)
      DIMENSION DATA(100)
      DIMENSION SLE(4),SLN(4)
      EQUIVALENCE (SLE,IEOVER)
      EQUIVALENCE (SLN,INOVER)
      LOGICAL*1 SLE,SLN
      LOGICAL*1 ASTA,ARMODA,AOUR,IALOBE,IARATE,IAACC,IAWMER,IAWCER,IBLOB
     XE,IBRATE,IBACC,IBNMER,IBNCER,ICIGA,IOUR,ISEC,LCOS,MCOS,IENO,INNO
      LOGICAL*1 DATA,PEZ,BIN,SPX,IAMP
      DATA PER,ASK,SPA,POL,EQ,F1,F2,F3,F4,F5,F6,F7,IAMP,SLA,PEZ,BIN,SPX/
     1Z48404040,Z5C404040,Z40404040,Z07404040,ZC5404040,ZC1404040,ZC2404
     2040,ZC3404040,ZC4404040,ZC5404040,ZC6404040,ZC7404040,Z50,Z6140404
     30,Z4B,ZF0,Z40/
      DATA A1,A2,A3,A4,A5,A6,A7,A8,A9/ZF1404040,ZF2404040,ZF3404040,ZF44
     X04040,ZF5404040,ZF6404040,ZF7404040,ZF8404040,ZF9404040/
      READ( 5,760)IGRADE
  760 FORMAT(9X,I1)
      JL=0
      JM=0
      WRITE( 6,760)IGRADE
      DO 35 J=1,10
C     INPUT STATION CONSTANTS
      READ( 5,80) STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
     1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
     2J),NSFPO(J),DATE(J)
      WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
     1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J).CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
     2J),NSFPO(J),DATE(J)
   80 FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
     X.3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)
  580 FORMAT(X,A6,X,I2,X,I3,X,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,
     X3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)
      DO 36 M=1,4
      JL=JM+M
C     INPUT STATION COEFF.
      READ( 5,81) IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
      WRITE(6,811)IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
  811 FORMAT(4X,A1,X,I2,5(X,E15.8))
      READ(5,581) C5(JL),C6(JL),C7(JL),C8(JL)
  581 FORMAT(8X,4(X,E12.8))
      WRITE(6,681)C5(JL),C6(JL),C7(JL),C8(JL)
  681 FORMAT(9X,4(X,E15.8))
   81 FORMAT(4X,A1,X,I2,5(X,E12.8))
   36 CONTINUE
```

```
      JM=JM+4
   35 CONTINUE
      WRITE(6,500)(KSTA(II),II=1,11)
  500 FORMAT(X,11(X,I2))
      DO 37 J=1,50
C     INPUT SATELLITE CONSTANTS
      READ( 5,82) KSAID(J),FREQ(J)
   82 FORMAT(I5,19X,F8.3)
      WRITE( 6,582) KSAID(J),FREQ(J)
  582 FORMAT(X,I5,19X,F8.3)
      IF(KSAID(J))37,38,37
   37 CONTINUE
   38 KSATCT=J-1
C     READ DATA   CHECK FOR AMPERSINE **********************************
   39 READ( 9,601,END=1,ERR=30)(DATA(I),I=1,65)
  601 FORMAT(65A1)
      IF(DATA(1).NE.IAMP)GO TO 39
      IF(DATA(7).EQ.SPX)GO TO 83
      IF(DATA(8).EQ.SPX)GO TO 83
      IF(DATA(4)-BIN.EQ.0)GO TO 85
      SAT=((DATA(3)-BIN)*10000.)+((DATA(4)-BIN)*1000.)+((DATA(5)-BIN)*10
     X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      GO TO 84
   85 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(5)-BIN)*10
     X0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
      GO TO 84
   83 SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(4)-BIN)*10
     X0)+((DATA(5)-BIN)*10)+(DATA(6)-BIN)
   84 DO 602 M=1,50
      IF(SAT.EQ.KSAID(M))GO TO 603
  602 CONTINUE
      WRITE( 6,635)
  635 FORMAT(19H SAID NOT IN TABLE )
      WRITE(6,183)IAMP,SAT
  183 FORMAT(X,A1,I5)
      GO TO 604
  603 WRITE(16,601)(DATA(I),I=1,65)
C     READ CAL.LINE WITH FORMAT CHECK ******************************
      READ( 9,601,END= 1,ERR= 39)(DATA(I),I=1,65)
      IF(DATA(1).EQ.IAMP)GO TO 83
      WRITE(16,601)(DATA(I),I=1,65)
C     CHECK PERIODS IN CAL. LINE
      IF(DATA(5).NE.PEZ)GO TO 604
      IF(DATA(13).NE.PEZ)GO TO 604
      IF(DATA(18).NE.PEZ)GO TO 604
      IF(DATA(26).NE.PEZ)GO TO 604
      IF(DATA(31).NE.PEZ)GO TO 604
      IF(DATA(39).NE.PEZ)GO TO 604
      IF(DATA(45).NE.PEZ)GO TO 604
      IF(DATA(53).NE.PEZ)GO TO 604
      IF(DATA(57).NE.PEZ)GO TO 604
      IF(DATA(65).NE.PEZ)GO TO 604
      DO 605 K=1,4
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  605 CONTINUE
      DO 606 K=6,12
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  606 CONTINUE
      DO 607 K=14,17
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  607 CONTINUE
      DO 608 K=19,25
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  608 CONTINUE
      DO 609 K=27,30
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  609 CONTINUE
      DO 610 K=32,38
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  610 CONTINUE
      DO 611 K=40,44
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  611 CONTINUE
      DO 612 K=46,52
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  612 CONTINUE
      DO 613 K=54,56
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  613 CONTINUE
      DO 614 K=58,64
      IF(DATA(K)-BIN.GT.9)GO TO 604
      IF(DATA(K)-BIN.LT.0)GO TO 604
      DATA(K)=DATA(K)-BIN
  614 CONTINUE
      X=DATA(9)+DATA(22)+DATA(35)+DATA(49)+DATA(61)
      IF(X.NE.45.)GO TO 604
      CSTA=(DATA(55)*10.)+(DATA(56))
      DO 616 L=1,12
      IF(KSTA(L).EQ.CSTA)GO TO 617
  616 CONTINUE
      WRITE( 6,618)
  618 FORMAT(27H WRONG STATION IN CAL.LINE )
      GO TO 604
  617 D=DATA(3)*10+DATA(4)
      CEWM=D/100.
      D=DATA(16)*10+DATA(17)
      CEWC=D/100.
      D=DATA(6)*100+DATA(7)*10+DATA(8)
      CEWF1=D/1000.
      D=DATA(19)*100+DATA(20)*10+DATA(21)
      CEWF2=D/1000.
      D=DATA(32)*100+DATA(33)*10+DATA(34)
      CEWF3=D/1000.
      D=DATA(46)*100+DATA(47)*10+DATA(48)
      CEWF4=D/1000.
      D=DATA(58)*100+DATA(59)*10+DATA(60)
      CEWF5=D/1000.
      D=DATA(10)*100+DATA(11)*10+DATA(12)
      CNSF1=D/1000.
      D=DATA(23)*100+DATA(24)*10+DATA(25)
      CNSF2=D/1000.
      D=DATA(36)*100+DATA(37)*10+DATA(38)
      CNSF3=D/1000.
      D=DATA(50)*100+DATA(51)*10+DATA(52)
      CNSF4=D/1000.
      D=DATA(62)*100+DATA(63)*10+DATA(64)
      CNSF5=D/1000.
      D=DATA(29)*10+DATA(30)
      CNSM=D/100.
      D=DATA(43)*10+DATA(44)
      CNSC=D/100.
      CSTA=DATA(55)*10+DATA(56)
      WRITE(11,501)IAMP,SAT,CSTA,STATIO(L)
  501 FORMAT(X,A1,I5,X,I2,X,A6)
      CANT=DATA(54)
      END=DATA(65)
   43 CEWM=EWM(L)+CEWM
      CEWC=EWC(L)+CEWC
      CNSM=NSM(L)+CNSM
      CNSC=NSC(L)+CNSC
      KM=5.
      RATE=0.
      IDIF1=CEWF2-CEWF1
      IDIF2=CEWF3-CEWF2
      IDIF3=CEWF4-CEWF3
      IDIF4=CEWF5-CEWF4
      CEWF1=CEWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
      IF(CANT-2.)162,160,161
C     NARROW BAND TRACKING FILTER *************************************
  160 IDIF=.0
      AST(1)=SPA
      GO TO 164
  161 IDIF=.0
      AST(1)=ASK
      GO TO 164
  162 IF(CANT.EQ.1.)GO TO 164
  163 IDIF=.120
C     2 CPS TRACKING FILTER *******************************************
      AST(1)=PER
  164 IDIF1=CNSF2-CNSF1
      IDIF2=CNSF3-CNSF2
      IDIF3=CNSF4-CNSF3
      IDIF4=CNSF5-CNSF4
      CNSF1=CNSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
C     CABLE LENGTH INEQUALITIES
      CEWM=((CLEWM(L)/.846)*(136.5-FREQ(M)))+CEWM
      CEWC=((CLEWC(L)/.846)*(136.5-FREQ(M)))+CEWC
      CNSM=((CLNSM(L)/.846)*(136.5-FREQ(M)))+CNSM
      CNSC=((CLNSC(L)/.846)*(136.5-FREQ(M)))+CNSC
      WRITE(11,636)
  636 FORMAT(55H CALIBRATED PHASE READINGS,5 POINT FITTED FINE READINGS)
C     CALIBRATED ZENITH ***********************************************
      WRITE(11,639)
  639 FORMAT(63H   CEWM        CEWC        CNSM        CNSC        CEWF
     X  CNSF )
      WRITE(11,150)CEWM,CEWC,CNSM,CNSC,CEWF1,CNSF1
  150 FORMAT(6(X,F10.6))
      WRITE(11,640)
  640 FORMAT(74H HRMNSC      EWFINE      EWMEDM      EWCORS      NSFINE
     XNSMEDM      NSCORS )
C     READ DATA WITH FORMAT CHECK************************************
      DO 41 JK=1,60
      K=JK
      IF(K.EQ.32.)GO TO 120
  621 READ( 9,601,END=1,ERR=32)(DATA(I),I=1,65)
      IF(DATA(1).EQ.IAMP)GO TO 83
      WRITE(16,601)(DATA(I),I=1,65)
      IF(DATA(10).EQ.SPX)GO TO 122
      GO TO 124
  122 IF(DATA(30).EQ.SPX)GO TO 123
      GO TO 124
  123 IF(DATA(50).EQ.SPX)GO TO 120
  124 IF(DATA(5).NE.PEZ)GO TO 621
C     CHECK PERIODS IN DATA LINE
      IF(DATA(13).NE.PEZ)GO TO 621
      IF(DATA(18).NE.PEZ)GO TO 621
      IF(DATA(26).NE.PEZ)GO TO 621
      IF(DATA(31).NE.PEZ)GO TO 621
      IF(DATA(39).NE.PEZ)GO TO 621
      IF(DATA(45).NE.PEZ)GO TO 621
      IF(DATA(53).NE.PEZ)GO TO 621
      IF(DATA(57).NE.PEZ)GO TO 621
      IF(DATA(65).NE.PEZ)GO TO 621
      DO 622 J=1,4
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  622 CONTINUE
      DO 623 J=6,12
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  623 CONTINUE
      DO 624 J=14,17
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  624 CONTINUE
      DO 625 J=19,25
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  625 CONTINUE
      DO 626 J=27,30
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  626 CONTINUE
      DO 627 J=32,38
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  627 CONTINUE
      DO 628 J=40,44
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  628 CONTINUE
      DO 629 J=46,52
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  629 CONTINUE
      DO 630 J=54,56
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  630 CONTINUE
```

```
      DO 631 J=58,64
      IF(DATA(J)-BIN.GT.9)GO TO 621
      IF(DATA(J)-BIN.LT.0)GO TO 621
      DATA(J)=DATA(J)-BIN
  631 CONTINUE
      SECD(K)=DATA(1)*10+DATA(2)
      MIND(K)=DATA(14)*10+DATA(15)
      HORD(K)=DATA(27)*10+DATA(28)
      ANTD(K)=DATA(54)
      IDAYD(K)=DATA(40)*100+DATA(41)*10+DATA(42)
      STAD(K)=DATA(55)*10+DATA(56)
      D=DATA(3)*10+DATA(4)
      EWMD(K)=D/100.
      D=DATA(16)*10+DATA(17)
      EWCD(K)=D/100.
      D=DATA(29)*10+DATA(30)
      NSMD(K)=D/100.
      D=DATA(43)*10+DATA(44)
      NSCD(K)=D/100.
      D=DATA(6)*100+DATA(7)*10+DATA(8)
      EWF1=D/1000.
      D=DATA(19)*100+DATA(20)*10+DATA(21)
      EWF2=D/1000.
      D=DATA(32)*100+DATA(33)*10+DATA(34)
      EWF3=D/1000.
      D=DATA(46)*100+DATA(47)*10+DATA(48)
      EWF4=D/1000.
      D=DATA(58)*100+DATA(59)*10+DATA(60)
      EWF5=D/1000.
      D=DATA(10)*100+DATA(11)*10+DATA(12)
      NSF1=D/1000.
      D=DATA(23)*100+DATA(24)*10+DATA(25)
      NSF2=D/1000.
      D=DATA(36)*100+DATA(37)*10+DATA(38)
      NSF3=D/1000.
      D=DATA(50)*100+DATA(51)*10+DATA(52)
      NSF4=D/1000.
      D=DATA(62)*100+DATA(63)*10+DATA(64)
      NSF5=D/1000.
      SIG1=DATA(9)
      SIG2=DATA(22)
      SIG3=DATA(35)
      SIG4=DATA(49)
      SIG5=DATA(61)
      END=DATA(65)
  100 IF(ANTD(K)-2)102,101,101
  101 ANT=57.
      GO TO 103
  102 ANT=46.
  103 TIM(K)=((HORD(K)*3600.)+(60.*MIND(K)))+SECD(K)
C     FIT FIVE FINES EACH LINE
      IDIF1=EWF2-EWF1
      CALL NORMAL(IDIF1)
      IDIF2=EWF3-EWF2
      CALL NORMAL(IDIF2)
      IDIF3=EWF4-EWF3
      CALL NORMAL(IDIF3)
      IDIF4=EWF5-EWF4
      CALL NORMAL(IDIF4)
      EEWF(K)=EWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
      IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
C     COUNTER DELAY (TIME) ********************************************
      EEWF(K)=(EEWF(K)-(.05*IDIF5*EWF3))
      IDIF1=NSF2-NSF1
      CALL NORMAL(IDIF1)
      IDIF2=NSF3-NSF2
      CALL NORMAL(IDIF2)
      IDIF3=NSF4-NSF3
      CALL NORMAL(IDIF3)
      IDIF4=NSF5-NSF4
      CALL NORMAL(IDIF4)
      ENSF(K)=NSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1))))/35.)
      IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
C     FILTER DELAY (TIME)
      ENSF(K)=ENSF(K)-(.05*IDIF5*NSF3)
      IDIF1=SIG2-SIG1
      IDIF2=SIG3-SIG2
      IDIF3=SIG4-SIG3
      IDIF4=SIG5-SIG4
      SIGD(K)=SIG3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1))))/35)
      WRITE(11,151)HORD(K),MIND(K),SECD(K),EEWF(K),EWMD(K),EWCD(K),ENSF(
     XK),NSMD(K),NSCD(K)
  151 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)))
   41 CONTINUE
C     DATA MSG.COMPLETE,START SMOOTHING ********************************
  120 SL=.01
      AST(6)=SPA
      K=K-1
      IF(K.LE.5)GO TO 780
      GO TO 782
  780 WRITE( 6,781)
  781 FORMAT(27H LESS THAN 5 LINES OF DATA )
      KZ=K-1
      GO TO 78
  782 JK=K
      IF(ANT.EQ.57.) GO TO 90
      GO TO 91
   90 CEWF1=CEWF1+EWFPO(L)
      CNSF1=CNSF1+NSFPO(L)
      AST(2)=POL
      GO TO 121
   91 CEWF1=CEWF1+EWFEQ(L)
      CNSF1=CNSF1+NSFEQ(L)
      AST(2)=EQ
  121 SM=2.0
      WRITE(11,638)
  638 FORMAT(32H KC-KS1 USING STATION CONSTANTS )
      WRITE(11,1152)CEWF1,EWFPO(L),EWFEQ(L),CNSF1,NSFPO(L),NSFEQ(L)
C     CHECK TIME SEQUENCE ***********************************************
      IB=0
      IC=0
      ID=0
      IE=0
      IF=0
      IG=0
      IH=0
      KQ=K-1
      DO 20 KS=1,KQ
  806 ITZM=TIM(KS+1)-TIM(KS)
      IF(ITZM)807,807,808
  807 TIM(KS+1)=TIM(KS+1)+86400.
      GO TO 806
  808 IF(ITZM.NE.1)GO TO 21
      IB=IB+1
      GO TO 20
   21 IF(ITZM.NE.2 )GO TO 22
      IC=IC+1
      GO TO 20
```

```
 22 IF(ITZM.NE.10 )GO TO 23
    ID=ID+1
    GO TO 20
 23 IF(ITZM.NE.20 )GO TO 24
    IE=IE+1
    GO TO 20
 24 IF(ITZM.NE.60 )GO TO 25
    IF=IF+1
    GO TO 20
 25 IF(ITZM.NE.120 )GO TO 26
    IG=IG+1
    GO TO 20
 26 IF(ITZM.NE.600 )GO TO 20
    IH=IH+1
    GO TO 20
 20 CONTINUE
    ITZM=MAXO(IB,IC,ID,IE,IF,IG,IH)
    IF(IB.EQ.ITZM)GO TO 200
    IF(IC.EQ.ITZM)GO TO 201
    IF(ID.EQ.ITZM)GO TO 202
    IF(IE.EQ.ITZM)GO TO 203
    IF(IF.EQ.ITZM)GO TO 204
    IF(IG.EQ.ITZM)GO TO 205
    IF(IH.EQ.ITZM)GO TO 206
200 AST(3)=F1
    TILI=1.
    GO TO 207
201 AST(3)=F2
    TILI=2.
    GO TO 207
202 AST(3)=F3
    TILI=10.
    GO TO 207
203 AST(3)=F4
    TILI=20.
    GO TO 207
204 AST(3)=F5
    TILI=60.
    GO TO 207
205 AST(3)=F6
    TILI=120.
    GO TO 207
206 AST(3)=F7
    TILI=600.
    GO TO 207
 72 WRITE( 6,73)
 73 FORMAT(22H TIME OUT OF SEQUANCE )
    KZ=K-1
 78 WRITE( 6,77)SAT,CSTA,STATIU(L),HORD(KZ),MIND(KZ),IDAYD(KZ)
 77 FORMAT(X,I5,X,I2,X,A6,X,I2,I2,X,I3)
    GO TO 65
 74 WRITE( 6,75)
 75 FORMAT(25H DATA EXCEEDS TIME CHECK )
    KZ=K-1
    GO TO 78
 50 WRITE(11,51)
 51 FORMAT(40H EAST MEDIUM CHANNEL EXCEEDS 100 COUNTS )
    AST(6)=F1
    RATE=0.
    GO TO 58
 52 WRITE(11,53)
 53 FORMAT(40H EAST COURSE CHANNEL EXCEEDS 100 COUNTS )
    AST(6)=F1
    RATE=0.
```

```
      GO TO 58
   54 WRITE(11,55)
   55 FORMAT(41H NORTH MEDIUM CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F2
      RATE=0.
      GO TO 59
   56 WRITE(11,57)
   57 FORMAT(41H NORTH COURSE CHANNEL EXCEEDS 100 COUNTS )
      AST(6)=F2
      RATE=0.
      GO TO 59
  804 WRITE(11,805)
  805 FORMAT(27H DATA WILL NOT LOBE ASSIGN )
      KZ=K-1
      GO TO 78
C     EW AMBIGUITY LOBE ASSIGN.
  207 SL=.015
      K=JK-1
      DO 70 N=1,K
      IF(TIM(N+1)-TIM(N))72,72,71
   71 IF(TIM(N+1)-TIM(N)-(5.*TILI))70,70,74
   70 CONTINUE
      K=JK
      CALL LOBASN(TIM,K,EWMD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LOBASN(TIM,K,EWCD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,EWMD,K,MID,SIA,SL,EWMB,EWMC,SM,ENA,EOA)
      EWMA=ALPHA+EWMD(MID)
      EWMT=TIM(MID)-.15
      SA=SIA
      WRITE(11,643)
  643 FORMAT(83H          ALPHA.EWM   MID.PT.   RATE              SIGMA
     X    BETA          GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWMB,EWMC
  152 FORMAT(X,F14.6,X,I6,X,4(F14.6,X))
      K=JK
      CALL LSQQUA(TIM,ALPHA,EWCD,K,MID,SIA,SL,EWCB,EWCC,SM,ENB,EOB)
      EWCA=ALPHA+EWCD(MID)
      EWCT=TIM(MID)+.05
      SB=SIA
      WRITE(11,644)
  644 FORMAT(83H          ALPHA.EWC   MID.PT.   RATE              SIGMA
     X    BETA          GAMMA    )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWCB,EWCC
      RATE=((EWMB*ANT/4.)+(EWCB*ANT/3.5))/2.
      IF(ABS(RATE).LE..05)RATE=0.0
C     EWF LOBE ASSIGN.
      IF(ABS(SB).GT..1)GO TO 52
      IF(ABS(SA).GT..1)GO TO 50
   58 SL=.01
      SM=2.5
      K=JK
      CALL LOBASN(TIM,K,EEWF,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,EEWF,K,MID,SIA,SL,EWFB,EWFC,SM,ENC,EOC)
      EWFT=TIM(MID)
      EWFA=ALPHA+EEWF(MID)
  224 SC=SIA
      INO=SIA*1000.+.5
  227 CALL ZERO(3,INO,IENO)
      IDAYD(1)=IDAYD(MID)
      WRITE(11,642)
  642 FORMAT(83H          ALPHA.EWF   MID.PT.   RATE              SIGMA
     X    BETA          GAMMA     )
      WRITE(11,152)ALPHA,MID,RATE,SIA,EWFB,EWFC
      WRITE(11,153)EWMA,EWMT,EWCA,EWCT,RATE
  153 FORMAT(6(F14.6,X))
C     NS AMBIGUITY LOBE ASSIGN.
      SL=.015
      RATE=0.
      K=JK
      SM=2.0
      CALL LOBASN(TIM,K,NSMD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LOBASN(TIM,K,NSCD,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,NSMD,K,MID,SIA,SL,SNMB,SNMC,SM,ENX,EOD)
      SNMA=ALPHA+NSMD(MID)
      SNMT=TIM(MID)+.25
      SD=SIA
      WRITE(11,646)
  646 FORMAT(83H       ALPHA.NSM   MID.PT.   RATE           SIGMA
     X    BETA          GAMMA     )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNMB,SNMC
      K=JK
      CALL LSQQUA(TIM,ALPHA,NSCD,K,MID,SIA,SL,SNCB,SNCC,SM,ENE,EOE)
      SNCA=ALPHA+NSCD(MID)
      SNCT=TIM(MID)+.45
      SE=SIA
      RATE=(((SNMB*ANT/4.)+(SNCB*ANT/3.5))/2.)
      IF(ABS(RATE).LE..05)RATE=0.0
      WRITE(11,647)
  647 FORMAT(83H       ALPHA.NSC   MID.PT.   RATE           SIGMA
     X    BETA          GAMMA     )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNCB,SNCC
C     NSF LOBE ASSIGN.
      IF(ABS(SD).GT..1)GO TO 54
      IF(ABS(SE).GT..1)GO TO 56
   59 SL=.01
      SM=2.5
      K=JK
      CALL LOBASN(TIM,K,ENSF,RATE,ITD)
      IF(ITD.GE.100)GO TO 804
      CALL LSQQUA(TIM,ALPHA,ENSF,K,MID,SIA,SL,SNFB,SNFC,SM,ENF,EOF)
      SNFT=TIM(MID)
      SNFA=ALPHA+ENSF(MID)
  230 SF=SIA
      ISO=SIA*1000.+.5
  229 CALL ZERO(3,ISO,INNO)
      IL=0
      IK=0
      IJ=50.
      IF(ABS(SC).GT..05)GO TO 231
  234 IF(ABS(SF).GT..05)GO TO 235
      GO TO 239
  231 AST(6)=F1
      IK=50
      GO TO 234
  235 AST(6)=F2
      IL=50
  239 WRITE(11,645)
  645 FORMAT(83H       ALPHA.NSF   MID.PT.   RATE           SIGMA
     X    BETA          GAMMA     )
      WRITE(11,152)ALPHA,MID,RATE,SIA,SNFB,SNFC
      WRITE(11,153)SNMA,SNMT,SNCA,SNCT,RATE
      WRITE(11,777)
```

```
C     LOBE ASSIGNED DATA AFTER THE FIT HAS BEEN APPLIED **************
      WRITE(11,641)
  641 FORMAT(99H FRA.TIME  EWFINE   O-C   EWMEDM   O-C   EWCORS   O-C
     XNSFINE   O-C   NSMEDM   O-C   NSCORS   O-C )
      K=JK
      DO 790 I=1,K
      T=TIM(I)-EWMT-.15
      AM=EWMA+(EWMB*T)+(EWMC*T**2)
      T=TIM(I)-EWCT+.05
      AN=EWCA+(EWCB*T)+(EWCC*T**2)
      T=TIM(I)-EWFT
      AO=EWFA+(EWFB*T)+(EWFC*T**2)
      T=TIM(I)-SNMT+.25
      AP=SNMA+(SNMB*T)+(SNMC*T**2)
      T=TIM(I)-SNCT+.45
      AQ=SNCA+(SNCB*T)+(SNCC*T**2)
      T=TIM(I)-SNFT
      AR=SNFA+(SNFB*T)+(SNFC*T**2)
      AO=EEWF(I)-AO
      AM=EWMD(I)-AM
      AN=EWCD(I)-AN
      AR=ENSF(I)-AR
      AP=NSMD(I)-AP
      AQ=NSCD(I)-AQ
      WRITE(11,791)TIM(I),EEWF(I),AO,EWMD(I),AM,EWCD(I),AN,ENSF(I),AR,NS
     XMD(I),AP,NSCD(I),AQ
  791 FORMAT(X,F7.1,6(X,F7.3,X,F6.3))
  790 CONTINUE
      WRITE(11,792)ENA,SA,EOA,ENB,SB,EOB,ENC,SC,EOC
      WRITE(11,793)
      WRITE(11,794)
      WRITE(11,792)ENX,SD,EOD,ENE,SE,EOE,ENF,SF,EOF
  792 FORMAT(3(X,I2,4X,F5.3,8X,I2))
  793 FORMAT( 67H IN    EWM SIGMA    OT IN    EWC SIGMA    OT IN    EWF
     XSIGMA    OT )
  794 FORMAT( 67H IN    NSM SIGMA    OT IN    NSC SIGMA    OT IN    NSF
     XSIGMA    OT )
C     ADJUST TIME TO EWFINE FITTED TIME
      XKFA=(KFA(L)/1000.)+IDIF
      XKFB=(KFB(L)/1000.)+IDIF
      EWFA=EWFA+EWFB*(-XKFA)+EWFC*((-XKFA)**2)
      EWFB=EWFB+2.*EWFC*(-XKFA)
      SNFA=SNFA+SNFB*(EWFT-SNFT-XKFB)+SNFC*((EWFT-SNFT-XKFB)**2)
      SNFB=SNFB+2.*SNFC*(-XKFB)
      EWMA=EWMA+EWMB*(EWFT-EWMT)+EWMC*((EWFT-EWMT)**2)
      EWCA=EWCA+EWCB*(EWFT-EWCT)+EWCC*((EWFT-EWCT)**2)
      SNMA=SNMA+SNMB*(EWFT-SNMT)+SNMC*((EWFT-SNMT)**2)
      SNCA=SNCA+SNCB*(EWFT-SNCT)+SNCC*((EWFT-SNCT)**2)
  118 EWFBB=EWFA-CEWF1
C     PHASE ANGLE (FITTED) MINUS KS2+KC-KS1
      EWMBB=EWMA-CEWM
      EWCBB=EWCA-CEWC
      SNFBB=SNFA-CNSF1
      SNMBB=SNMA-CNSM
      SNCBB=SNCA-CNSC
C     REMOVE LOBE INTERGER
      LDIF=EWFBB
      EWFBB=EWFBB-LDIF
      LDIF=EWMBB
      EWMBB=EWMBB-LDIF
      LDIF=EWCBB
      EWCBB=EWCBB-LDIF
      LDIF=SNFBB
      SNFBB=SNFBB-LDIF
      LDIF=SNMBB
      SNMBB=SNMBB-LDIF
      LDIF=SNCBB
      SNCBB=SNCBB-LDIF
      AB=EWMBB-EWCBB
      LDIF=AB
      AB=AB-LDIF
      CALL NORMAL(AB)
      BC=SNMBB-SNCBB
      LDIF=BC
      BC=BC-LDIF
      CALL NORMAL(BC)
      AB4=AB*8.
      BC4=BC*8.
      AB3=AB*7.
      BC3=BC*7.
      EAB4=AB4-EWMBB
      LDIF=EAB4
      EAB4=EAB4-LDIF
      CALL NORMAL(EAB4)
      EBC4=BC4-SNMBB
      LDIF=EBC4
      EBC4=EBC4-LDIF
      CALL NORMAL(EBC4)
      DE=AB4-EAB4
      FG=BC4-EBC4
      EAB3=AB3-EWCBB
      LDIF=EAB3
      EAB3=EAB3-LDIF
      CALL NORMAL(EAB3)
      EBC3=BC3-SNCBB
      LDIF=EBC3
      EBC3=EBC3-LDIF
      CALL NORMAL(EBC3)
      DE3=AB3-EAB3
      FG3=BC3-EBC3
      CORE=DE+DE3
      CORN=FG+FG3
      COREF=(ANT/7.5)*CORE
      CORNF=(ANT/7.5)*CORN
      FE=COREF-EWFBB
      LDIF=FE
      FE=FE-LDIF
      CALL NORMAL(FE)
      FN=CORNF-SNFBB
      LDIF=FN
      FN=FN-LDIF
      CALL NORMAL(FN)
      EWLOBE=COREF-FE
      SNLOBE=CORNF-FN
C     AMBIGUITY ERRORS
      EWMER=4./ANT*EWLOBE-DE
      CALL NORMAL(EWMER)
      EWCER=3.5/ANT*EWLOBE-DE3
      CALL NORMAL(EWCER)
      SNMER=4./ANT*SNLOBE-FG
      CALL NORMAL(SNMER)
      SNCER=3.5/ANT*SNLOBE-FG3
      CALL NORMAL(SNCER)
      ABB=AB*2
      ADE=DE/4
      ADE3=DE3/3.5
      ACORE=CORE/7.5
      AEWLOB=EWLOBE/ANT
      WRITE(11,1152)ABB,ADE3,ADE,ACORE,AEWLOB,EWFT
      ABC=BC*2
      AFG=FG3/3.5
      AFG3=FG3/3.5
      AFG=FG/4
      ACORN=CORN/7.5
      ASNLOB=SNLOBE/ANT
      WRITE(11,1152)ABC,AFG3,AFG,ACORN,ASNLOB,SNFT
 1152 FORMAT(6(X,F14.6))
      IF(ANTD(K)-2)650,651,651
  651 WRITE(11,648)
  648 FORMAT(12H POLOR PASS )
      GO TO 652
  650 WRITE(11,649)
  649 FORMAT(17H EQUATORIAL PASS )
      GO TO 652
  652 IAD=2
      IBD=3
      ACOS=EWLOBE/((FREQ(M)/136.)*ANT)
      BCOS=SNLOBE/((FREQ(M)/136.)*ANT)
C     SET UP FOR INTERGER OUTPUT
      IACOS=ACOS*1000000+.5
      IBCOS=BCOS*1000000+.5
      CALL ZERO(8,IACOS,LCOS)
      CALL ZERO(8,IBCOS,MCOS)
      SIGB=0.
      DO 105 JX=1,JK
      SIGS=SIGD(JX)+SIGD(JX+1)
      SIGB=SIGS+SIGB
  105 CONTINUE
      SIGA=SIGB/JK
      LDIF=SIGA
      SIGA=SIGA-LDIF
  226 IEWMER=EWMER*1000+.5
  260 IEWCER=EWCER*1000+.5
  261 ISNMER=SNMER*1000+.5
  262 ISNCER=SNCER*1000+.5
      IA=39.
      IF(ABS(IEWMER).GT.IA)GO TO 219
      IF(ABS(IEWCER).GT.IA)GO TO 219
      AST(4)=SPA
      GO TO 220
  219 AST(4)=SLA
  220 IF(ABS(ISNMER).GT.IA)GO TO 225
      IF(ABS(ISNCER).GT.IA)GO TO 225
      AST(5)=SPA
      GO TO 263
  225 AST(5)=SLA
  263 IELOBE=EWLOBE*1000+.5
      IF(ABS(IEWMER).GT.IJ)GO TO 221
      IF(ABS(IEWCER).GT.IJ)GO TO 221
      GO TO 240
  221 AST(6)=F1
      IK=50
  240 IF(ABS(ISNMER).GT.IJ)GO TO 222
      IF(ABS(ISNCER).GT.IJ)GO TO 222
      GO TO 241
  222 AST(6)=F2
      IL=50
  241 IX=IL+IK
      IXX=99
      IF(IX.GT.IXX)GO TO 265
      GO TO 264
  265 AST(6)=F3
```

```
264 INLOBE=SNLOBE*1000+.5
    IERATE=EWFB*10000.*TILI
    INRATE=SNFB*10000.*TILI
    IEACC=EWFC*100000.*(TILI**2)
    INACC=SNFC*100000.*(TILI**2)
    IEOVER=(IABS(IERATE/10000))*3+IABS(IEACC/1000)+64
    INOVER=(IABS(INRATE/10000))*3+IABS(INACC/1000)+64
    IERATE=IERATE-(IERATE/10000)*10000
    INRATE=INRATE-(INRATE/10000)*10000
    IEACC=IEACC-(IEACC/1000)*1000
    INACC=INACC-(INACC/1000)*1000
269 ISIGA=SIGA*100+.5
    ISIGA=ISIGA/5.
    IYR=68
    IF(EWFT-86400.)115,115,116
116 IDAYD(1)=IDAYD(1)+1
    EWFT=EWFT-86400.
115 HOUR=EWFT/3600.
    AMIN=((EWFT-(HOUR*3600.))/60.)
    SECN=(EWFT-((HOUR*3600.)+(AMIN*60.)))
    BOUR=HOUR*100
    BOUR=BOUR+AMIN
    CALL ZERO(5,BOUR,IOUR)
    ASEC=SECN*1000
    CALL ZERO(6,ASEC,ISEC)
    GO TO 117
117 CALL PYRD(IYR,IDAYD(1),YRMODA)
    CALL ZERO(3,CSTA,ASTA)
    CALL ZERO(7,YRMODA,ARMODA)
    HOUR=HOUR*10000
    HOUR=AMIN*100+HOUR+SECN
    CALL ZERO(7,HOUR,AOUR)
    CALL ZERO(6,IELOBE,IALOBE)
    CALL ZERO(5,IERATE,IARATE)
    CALL ZERO(4,IEACC,IAACC)
    CALL ZERO(3,IEWMER,IAWMER)
    CALL ZERO(3,IEWCER,IAWCER)
    CALL ZERO(6,INLOBE,IBLOBE)
    CALL ZERO(5,INRATE,IBRATE)
    CALL ZERO(4,INACC,IBACC)
    CALL ZERO(3,ISNMER,IBNMER)
    CALL ZERO(3,ISNCER,IBNCER)
    CALL ZERO(3,ISIGA,ICIGA)
    IF(IGRADE.EQ.1)GO TO 761
    AST(6)=SPA
761 PID=MID
    IF(PID.EQ.11.)GO TO 251
    IF(PID.EQ.12.)GO TO 252
    IF(PID.EQ.13.)GO TO 253
    IF(PID.EQ.14.)GO TO 254
    IF(PID.EQ.15.)GO TO 255
    IF(PID.EQ.16.)GO TO 256
    IF(PID.EQ.17.)GO TO 257
    IF(PID.EQ.10.)GO TO 250
    IF(PID.EQ.9.)GO TO 270
    IF(PID.EQ.8.)GO TO 271
    IF(PID.EQ.7.)GO TO 272
    IF(PID.EQ.6.)GO TO 273
    IF(PID.EQ.5.)GO TO 274
    IF(PID.EQ.4.)GO TO 275
    IF(PID.EQ.3.)GO TO 276
    IF(PID.EQ.2.)GO TO 277
    IF(PID.EQ.1.)GO TO 278
```

```
250 IFIT=SPA
    GO TO 215
251 IFIT=F1
    GO TO 215
252 IFIT=F2
    GO TO 215
253 IFIT=F3
    GO TO 215
254 IFIT=F4
    GO TO 215
255 IFIT=F5
    GO TO 215
256 IFIT=F6
    GO TO 215
257 IFIT=F7
    GO TO 215
270 IFIT=A9
    GO TO 215
271 IFIT=A8
    GO TO 215
272 IFIT=A7
    GO TO 215
273 IFIT=A6
    GO TO 215
274 IFIT=A5
    GO TO 215
275 IFIT=A4
    GO TO 215
276 IFIT=A3
    GO TO 215
277 IFIT=A2
    GO TO 215
278 IFIT=A1
    GO TO 215
215 WRITE(11,216)SAT,AST(1),(ASTA(I),I=2,3),AST(2),ARMODA,AOUR,SLE(4),
   XIALOBE,IARATE,IAACC,IAWMER,IAWCER,SLN(4),IBLOBE,IBRATE,IBACC,IBNME
   XR,IBNCER,(ICIGA(I),I=2,3),AST(3),IFIT,AST(6),(IENO(I),I=2,3),INNO,
   XAST(4),AST(5)
218 FORMAT(X,I5,A1,2A1,A1,7A1,7A1,A1,6A1,5A1,4A1,3A1,3A1,A1,6A1,5A1,4A
   X1,3A1,3A1,X,2A1,A1,A1,A1,2A1,3A1,A1,A1)
    WRITE(15,216)SAT,AST(1),(ASTA(I),I=2,3),AST(2),ARMODA,AOUR,SLE(4),
   XIALOBE,IARATE,IAACC,IAWMER,IAWCER,SLN(4),IBLOBE,IBRATE,IBACC,IBNME
   XR,IBNCER,(ICIGA(I),I=2,3),AST(3),IFIT,AST(6),(IENO(I),I=2,3),INNO,
   XAST(4),AST(5)
    WRITE(11,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,AST(2),IAO
    WRITE(11,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,AST(2),IBO
    GO TO 778
216 FORMAT(I5,A1,2A1,A1,7A1,7A1,A1,6A1,5A1,4A1,3A1,3A1,A1,6A1,5A1,4A1,
   X3A1,3A1,X,2A1,A1,A1,A1,2A1,3A1,A1,A1)
217 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,8A1,3X,A1,I1)
778 WRITE(11,777)
    GO TO 39
 30 WRITE( 6,31)
 31 FORMAT(17H CAL.LINE PARITY )
    GO TO 39
 32 WRITE(11,33)
 33 FORMAT(24H DATA PARITY,NO MSG.END )
    GO TO 120
604 DO 60 JZ=1,60
    JY=JZ
    IF(JY.EQ.32.)GO TO 39
 61 READ( 9,601,END=1,ERR=61)(DATA(I),I=1,65)
    IF(DATA(1).EQ.IAMP)GO TO 83
    IF(DATA(10).EQ.SPX)GO TO 62
```

```
      GO TO 64
   62 IF(DATA(30).EQ.SPX)GO TO 63
      GO TO 64
   63 IF(DATA(50).EQ.SPX)GO TO 39
   64 WRITE( 6,660)(DATA(I),I=1,65)
  660 FORMAT(X,65A1)
   60 CONTINUE
      GO TO 39
   65 K=JK-1
      DO 66 I=1,K
      WRITE( 6,67)HORD(I),MIND(I),SECD(I),EEWF(I),EWMD(I),EWCD(I),ENSF(I
     X),NSMD(I),NSCD(I),ANTD(I),IDAYD(I),STAD(I)
   67 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)),I2,X,I3,X,I2,X)
   66 CONTINUE
      GO TO 39
  777 FORMAT(1H1)
    1 WRITE( 6,990)
  990 FORMAT(16H JOB IS COMPLETE)
      END FILE 6
      END FILE 11
      END FILE 15
      STOP
      END
      SUBROUTINE LSQQUA(T,ALPHA,XY,ITO,L,SIGMA,F,BETA,GAMMA,SM,INN,IOT)
      DIMENSION B(3,4),A(3,4),T(32),XY(32)
      DO 1 I=1,3
      DO 1 J=1,4
    1 A(I,J)=0.0
      L=(ITO+1)/2
      A(1,1)=ITO
      INN=ITO
      DELSQ=0.
      DO 10 I=1,ITO
      RAPPA=XY(I)-XY(L)
      TAU=T(I)-T(L)
      A(1,2)=A(1,2)+TAU
      A(1,3)=A(1,3)+TAU**2
      A(1,4)=A(1,4)+RAPPA
      A(2,3) =A(2,3)+TAU**3
      A(2,4)=A(2,4)+RAPPA*TAU
      A(3,3) =A(3,3) +TAU**4
      A(3,4)=A(3,4)+RAPPA*TAU**2
   10 DELSQ=RAPPA**2+DELSQ
   67 A(2,2)=A(1,3)
      A(2,1)=A(2,4)
      A(3,1)=A(3,4)
      N=1
      DO20 K=1,3
      N=N+1
      DO15 J=N,4
   15 B(K,J)=A(K,J)/A(K,K)
      IF(N-4)12,21,11
   12 DO20I=N,3
      DO20J=I,4
   20 A(I,J)=A(I,J)-A(K,I)*B(K,J)
   21 GAMMA=B(3,4)
      BETA=B(2,4)-GAMMA*B(2,3)
      ALPHA     =B(1,4)-BETA*B(1,2)-GAMMA*B(1,3)
      N=A(1,1)
      SIGMA=((DELSQ-ALPHA*A(1,4)-BETA*A(2,1)-GAMMA*A(3,1))/A(1,1))
      IF(SIGMA)70,71,71
   70 SIGMA=0.
   71 SIGMA=SQRT(SIGMA)
```

```
      IF(SIGMA-F    )59,59,35
   35 DO61I=1,3
      DO61J=1,4
   61 A(I,J)=0.0
      DELSQ=0.0
      DO65I=1,ITO
      RAPPA=XY(I)-XY(L)
      TAU= T(I)-T(L)
      R=RAPPA-ALPHA-BETA*TAU-GAMMA*TAU**2
      IF(R)2,3,3
    2 R=-R
    3 IF(R-SM*SIGMA)32,32,65
   32 A(1,1)=A(1,1)+1.
      A(1,2)=A(1,2)+TAU
      A(1,3)=A(1,3)+TAU**2
      A(1,4)=A(1,4)+RAPPA
      A(2,3) =A(2,3)+TAU**3
      A(2,4)=A(2,4)+RAPPA*TAU
      A(3,3) =A(3,3) +TAU**4
      A(3,4)=A(3,4)+RAPPA*TAU**2
      DELSQ=RAPPA**2+DELSQ
   65 CONTINUE
      IF(A(1,1)-5)59,66,66
   66 IF(A(1,1)-N    )67,59,67
   59 ITO=A(1,1)
      IOT=ITO
      RETURN
   11 WRITE( 6,7)
    7 FORMAT(30H THIS MSG. HAS EXCESSIVE NOISE)
      RETURN
      END
      SUBROUTINE LOBASN(TEM,I,A,RATE,L)
      DIMENSION A(31),TEM(31)
      K=I-1
      IF(RATE.GT.0.)A(1)=A(1)+1.
      IF(RATE.LT.0.)A(1)=A(1)-1.
      DO 10 J=1,K
      L=0
      JJ=J+1
      DELTA=RATE*(TEM(JJ)-TEM(J))
   11 X=A(JJ)-A(J)
      L=L+1
      IF(L.GE.100)GO TO 12
      IF(ABS(DELTA-X)-.500)10,10,6
    6 IF(DELTA)4,5,3
    4 A(JJ)=A(JJ)-1
      GO TO 11
    3 A(JJ)=A(JJ)+1
      GO TO 11
    5 IF(X)3,10,4
   10 CONTINUE
   12 RETURN
      END
      SUBROUTINE NORMAL(X)
      IF(X)113,114,115
  113 IF(ABS (X)-.5)114,114,116
  116 X=X+1.0
      GO TO 114
  115 IF(ABS (X)-.5)114,114,117
  117 X=X-1.0
  114 RETURN
      END
      SUBROUTINE PYRD(NYR,JDAY,NYMODA)
      DIMENSION N(24)
      DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10),N(11),N(12
     1),N(13),N(14),N(15),N(16),N(17),N(18),N(19),N(20),N(21),N(22),N(23
     2),N(24)/0,31,59,90,120,151,181,212,243,273,304,334,0,31,60,91,121,
     3152,182,213,244,274,305,335/
      J=0
      A=NYR/4.
      L=A
      A=A-L
      IF(A)2,1,2
    1 J=12
    2 DO 5 K=1,12
      M=J+K
      IF(JDAY.LE.N(M))GO TO 4
    5 CONTINUE
    4 M=J+K-1
      NDAY=JDAY-N(M)
      K=K-1
      NYMODA=(NYR*10000)+(K*100)+NDAY
      RETURN
      END
      SUBROUTINE ZERO(N,IIN,AREA)
      DIMENSION DIV(7)
      INTEGER DIV
      LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)
      DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8
     X),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
     X/
      DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
     X00,10000,1000,100,10,1/
      AREA(1)=PLUS
      IF(IIN)1,2,2
    1 AREA(1)=MINUS
      IIN=IABS(IIN)
    2 IJ=9-N
      K=2
      J=N-1
      DO 4 I=1,J
      ITEMP=IIN/DIV(IJ)
      IF(ITEMP.LE.9.)GO TO 6
      AREA(K)=ASK
      GO TO 5
    6 AREA(K)=IC(ITEMP+1)
    5 IIN=IIN-(ITEMP*DIV(IJ))
      IJ=IJ+1
      K=K+1
    4 CONTINUE
      RETURN
      END
/*
// EXEC LINKGO
//GO.SYSUDUMP  DD  SYSOUT=A
//GO.FT06F001 DD SYSOUT=A
//GO.FT09F001 DD UNIT=2400-2,LABEL=(,BLP),VOLUME=SER=777777,             *
//               DCB=(,DEN=1,RECFM=U,BLKSIZE=80,TRTCH=ET)
//GO.FT11F001 DD SYSOUT=A
//GO.FT15F001  DD SYSOUT=B,DCB=(,RECFM=F,LRECL=80,BLKSIZE=80)
//GO.FT16F001  DD DUMMY,UNIT=2400,LABEL=(,BLP),VOLUME=SER=GEOSOT,        *
//               DCB=(,DEN=2,RECFM=FB,LRECL=65,BLKSIZE=2340),            *
//               DISP=(NEW,PASS)
//GO.DATA5 DD *
       2                       TWO=NO GRADE
FTMYR6 03 375 374    058 28 372 25  958 140  902 00 113 26  011 430   680205
```

```
FTM1P 03 -.14056888-3 +.10000418+1 -.52082086-4 +.00000000+0 +.00000000+0
FTM2P 03 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
FTM3P 03 -.36754459-3 +.87306887-4 +.10000007+1 +.00000000+0 +.00000000+0
FTM4P 03 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
FTM1E 03 -.53746323-3 +.99990131+0 -.76668863-4 +.00000000+0 +.00000000+0
FTM2E 03 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
FTM3E 03 +.82749814-4 -.52892510-6 +.10001207+1 +.00000000+0 +.00000000+0
FTM4E 03 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
QUIT06 05 386 386    952-00 871-00  821 753    078-00 727 00  030 002 680205
QUI1E 05 +.20940915-2 +.10008733+1 +.93641860-4 +.22458829-4 -.18011563-3
QUI2E 05 -.22970585-5 +.00000000+0 +.67678972-3 -.30191208-2
QUI3E 05 +.37754203-2 +.65052576-3 +.99998330+0 -.14789760-4 -.10148564-3
QUI4E 05 -.84940136-5 +.00000000+0 -.22913735-2 +.10156293-2
QUI1P 05 +.10066065-2 +.99996738-0 -.23974532-3 +.17607164-5 -.34724734-6
QUI2P 05 -.35060033-4 +.00000000+0 -.43782727-3 -.27184685-2
QUI3P 05 -.34792096-2 -.10548883-3 +.99976158+0 -.25381153-4 +.62359472-5
QUI4P 05 -.29230690-4 +.00000000+0 -.21747670-2 -.56144730-3
LIMAP6 06 390 390    443 28 696 52  950 019    962 00 981 00  257 882 680205
LIM1E 06 +.29623983-3 +.10003067+1 -.92405700-5 +.00000000-0 +.00000000+0
LIM2E 06 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
LIM3E 06 +.41370035-3 +.35355528-4 +.10002094+1 +.00000000+0 +.00000000+0
LIM4E 06 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
LIM1P 06 -.17288594-2 +.99924796+0 -.92915356-3 +.00000000+0 +.00000000+0
LIM2P 06 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
LIM3P 06 -.65251865-3 -.13613973-4 +.99854064+0 +.00000000+0 +.00000000+0
LIM4P 06 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
SNTAG6 08 397 397    983 00 112 00  076 945    003 00 971 00  059 965 680205
SNT1E 08 &.00000000&0 &.10000000&1 &.00000000&0 &.00000000&0 &.00000000&0
SNT2E 08 &.00000000&0 &.00000000&0 &.00000000&0 &.00000000&0
SNT3E 08 &.00000000&0 &.00000000&0 &.10000000&1 &.00000000&0 &.00000000&0
SNT4E 08 &.00000000&0 &.00000000&0 &.00000000&0 &.00000000&0
SNT1P 08 &.00000000&0 &.10000000&1 &.00000000&0 &.00000000&0 &.00000000&0
SNT2P 08 &.00000000&0 &.00000000&0 &.00000000&0 &.00000000&0
SNT3P 08 &.00000000&0 &.00000000&0 &.10000000&1 &.00000000&0 &.00000000&0
SNT4P 08 &.00000000&0 &.00000000&0 &.00000000&0 &.00000000&0
NEWFL6 12 377 376    457 29 052 25  933 168    887 00 503 28  942 921 680205
NEW1E 12 -.19170106-3 +.99974483+0 +.14185841-3 +.67414141-5 -.15397906-3
NEW2E 12 +.86701090-5 +.00000000+0 +.00000000+0 +.00000000+0
NEW3E 12 +.21653203-3 +.36391400-4 +.10002326+1 +.00000000+0 +.00000000+0
NEW4E 12 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
NEW1P 12 +.90977247-4 +.10001052+1 +.20317739-3 +.00000000+0 +.00000000+0
NEW2P 12 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
NEW3P 12 +.00000000+0 +.00000000+0 +.10000000+1 +.00000000+0 +.00000000+0
NEW4P 12 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
WNKFL6 15 390 390    540-29 222 25  067 870    107 00 779 28  976 011 680205
WNK1E 15 -.53815660-3 +.10007333+1 +.80645285-4 +.00000000+0 +.00000000+0
WNK2E 15 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
WNK3E 15 -.10372921-2 -.27364891-3 +.10001356+1 +.00000000+0 +.00000000+0
WNK4E 15 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
WNK1P 15 +.38055918-3 +.10000534+1 +.19731019-4 +.00000000+0 +.00000000+0
WNK2P 15 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
WNK3P 15 +.15497939-3 +.19661635-4 +.99907664+0 +.00000000+0 +.00000000+0
WNK4P 15 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
JOBUR6 16 417 417    410-29 201 00  051 092    586-29 850 00  937 866 680205
JOB1E 16 +.49278891-3 +.99962462+0 +.19324889-3 +.33577390-4 -.21329130-5
JOB2E 16 -.29092539-5 -.27345116-6 +.00000000+0 +.00000000+0
JOB3E 16 +.35652040-3 +.67227993-3 +.10002847+1 +.27899943-6 +.51953200-6
JOB4E 16 -.10846733-6 -.12167462-6 +.00000000+0 +.00000000+0
JOB1P 16 -.26924000-2 +.10001142+1 -.56369850-4 +.11091456-4 +.30864940-5
JOB2P 16 +.46664510-4 +.00000000+0 +.00000000+0 +.00000000+0
JOB3P 16 -.36107291-2 -.52875261-4 +.99965713+0 -.58391001-5 +.50927149-5
JOB4P 16 +.45694370-4 +.00000000+0 +.00000000+0 +.00000000+0
ULASK6 19 383 383    654 00 960 00  937 438    059 00 497 00  076 921 680205
```

```
ULA1E 19 &.20243968-2 &.10001606&1 -.10393016-3 &.48169120-6 &.11495417-4
ULA2E 19 -.40887407-5 &.00000000&0 &.17572682-2 -.10486416-2
ULA3E 19 &.18515669-3 -.44136583-4 &.10001553&1 -.54333988-5 -.16219868-4
ULA4E 19 -.74733460-7 &.00000000&0 &.69647932-3 -.14363958-2
ULA1P 19 -.95663269-3 &.10002807&1 -.36318935-3 -.13821753-4 &.27188637-5
ULA2P 19 &.22624058-4 &.00000000&0 -.14888439-2 -.20507426-4
ULA3P 19 -.11370458-2 &.67562270-5 &.10003729&1 -.18158790-4 &.38524111-5
ULA4P 19 -.41508788-4 &.00000000&0 -.17580689-3 -.17410220-2
ORORA6 21 401 401    015 00 780 00  787 031    770 00 298 00  117 929 680205
ORO1E 21 -.29778306-2 +.10009156+1 -.22943741-3 -.61053700-5 -.10163772-3
ORO2E 21 +.10541764-4 +.00000000+0 +.00000000+0 +.00000000+0
ORO3E 21 +.16105135-2 -.65773801-3 +.10002713+1 -.40021681-5 +.94035050-4
ORO4E 21 -.63904369-5 +.00000000+0 +.00000000+0 +.00000000+0
ORO1P 21 +.00000000+0 +.10000000+1 +.00000000+0 +.00000000+0 +.00000000+0
ORO2P 21 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
ORO3P 21 +.00000000+0 +.00000000+0 +.10000000+1 +.00000000+0 +.00000000+0
ORO4P 21 +.00000000+0 +.00000000+0 +.00000000+0 +.00000000+0
MADGA6 23 428 428    666 00 037 00  041 019    033 00 831 00  889 157 680205
MAD1E 23 +.35305277-3 +.10005870+1 +.99905285-4 -.49983774-5 -.10061742-3
MAD2E 23 +.16661472-5 +.00000000+0 -.13141496-2 -.21079430-3
MAD3E 23 +.29722606-2 +.33548552-3 +.10001234+1 +.93322190-6 -.91412230-4
MAD4E 23 -.89181924-5 +.00000000+0 +.47489629-3 +.19854451-2
MAD1P 23 +.29946892-2 +.99988668+0 -.12674803-3 -.44706464-5 -.65049924-5
MAD2P 23 +.67850790-4 +.00000000+0 -.24783701-2 -.61427310-4
MAD3P 23 +.23405773-2 +.80005626-4 +.10002585+1 -.82928921-5 -.54176817-5
MAD4P 23 -.57118971-4 +.00000000+0 -.16355670-2 -.17595554-2
64031 572599 462097-142  136.620      6400301
62491 574099 463308-566  136.978    136.595
63241 570979 460790 315  136.234
63541 570977 460788 315  136.233
64541 570838 460676 355  136.200
64641 570713 460575 390  136.170
64761 572976 462402-248  136.710
64861 571144 460923 268  136.273
65041 571130 460912 319  136.231
65091 573730 463010-461  136.890
65165 573521 462841-402  136.840
65321 574107 463315-567  136.980
65391 573730 463010-460  136.890
65421 570524 460423 044  136.125
65511 570964 460778 319  136.230
65601 572473 461995-106  136.590
65811 570838 460676 355  136.200
65931 572221 461793-035  136.530
65981 572473 461996-106  136.590    136.390
65982 571593 461285 142  136.380
51011 573353 462706-355  136.800
66081 570964 460778 319  136.230
66161 573227 462604-319  136.770
66401 572096 461691      136.500
66441 571341 461082 213  136.320    136.560
66491 570838 460676 355  136.200
66512 573353 462706-355  136.800
66772 573353 462706-355  136.800
66773 571836 461481 073  136.438
66871 573227 462604-319  136.770
67011 571844 461488 071  136.440    136.980
67061 573227 462604-319  136.770
67201 571232 460994 243  136.294
67261 571844 461488 071  136.440    136.980
67311 571970 461590 035  136.470
67361 573227 462604-319  136.770
67381 573081 462486-278  136.735
```

```
67403 572213 461786-033  136.528
67404 571086 460876 285  136.259
67405 571605 461295 138  136.383
//G4DJCOBS JOB (G40032311F,T,G00043,009040),MIN,MSGLEVEL=1
//  EXEC  CTLCARDS
//CTLCARDS.SYSUT2 DD SPACE=(CYL,(1,1))
//CTLCARDS.SYSUT1  DD DATA
68812 23E 681101 095222 -00141 1152 023-18-04 -10531 0104 008-11-32 008E 04 05
67421 06P 681101 150510  19333 0994-129 33-33 -09271 7554 179 19 19 10A4 10 06
/*
// EXEC FORTRAN,PARM='OPT=0,MAP,BCD,DECK,LOAD'
//SOURCE.SYSLIN DD  DISP=(NEW,PASS)
//SOURCE.SYSIN DD  *
      INTEGER IGRADE,IYEAR
      INTEGER CSTA,BOUR,MIN,SEC,CANT,ASEC,BDATE,TIM,TIMC,TIMA,T,TT
      INTEGER HOUR,SAT,SAID,HR,MN,BEG,END,HRR,MNN
      REAL*8 STATIO
      DIMENSION STATIO(12),KFA(12),KFB(12),EWM(12),CLEWM(12),EWC(12),CLE
     1WC(12),EWFEQ(12),NSM(12),CLNSM(12),NSC(12),CLNSC(12),NSFEQ(12),NSF
     2PO(12),ISTA(48),IANT(48),C1(48),C2(48),C3(48),C4(48),C5(48),C6(48)
     3,C7(48),C8(48),KSAID(50),FREQ(50),KSTA(12),EWFPO(12)
      DIMENSION AST(31),DATE(12),CO(48)
      DIMENSION ASTA(3),ARMODA(7),IOUR(5),ISEC(6),LCOS(8),MCOS(8)
      DIMENSION BLANK(80)
      DIMENSION SLZ(4),SLQ(4)
      EQUIVALENCE (SLZ,IEQVER)
      EQUIVALENCE (SLQ,INQVER)
      LOGICAL*1 BLANK
      LOGICAL*1 SLZ,SLQ
      LOGICAL*1 ARMODA,IOUR,ISEC,LCOS,MCOS,CANT,PO,SPA
      DATA PO,SPA,L1,L2,L3,L4,L5,L6,L7/ZD7,Z40,ZC1404040,ZC2404040,ZC340
     X4040,ZC4404040,ZC5404040,ZC6404040,ZC7404040/
      DATA F1,F2,F3,F4,F5,F6,F7/ZC1404040,ZC2404040,ZC3404040,ZC4404040,
     XZC5404040,ZC6404040,ZC7404040/
      DATA FO,SPX,M1,M2,M3,M4,M5,M6,M7,M8,M9/ZF0404040,Z40404040,ZF14040
     X40,ZF2404040,ZF3404040,ZF4404040,ZF5404040,ZF6404040,ZF7404040,ZF8
     X404040,ZF9404040/
      READ( 5,760)IYEAR,IGRADE
  760 FORMAT(X,I2,6X,I1)
      WRITE( 6,762)IYEAR,IGRADE
  762 FORMAT(X,I2,6X,I1)
      JL=0
      JM=0
      DO 35 J=1,10
C     INPUT STATION CONSTANTS
      READ( 5,80) STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
     1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
     2J),NSFPO(J)
   80 FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
     X.3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)
      WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
     1,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
     2J),NSFPO(J)
  580 FORMAT(X,A6,X,I2,I3,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,3X,F4.3,
     XF4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)
      DO 36 M=1,4
      JL=JM+M
C     INPUT STATION COEFF.
      READ( 5,81) IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
   81 FORMAT(4X,A1,X,I2,5(X,E12.8))
      WRITE(6,811)IANT(JL),ISTA(JL),CO(JL),C1(JL),C2(JL),C3(JL),C4(JL)
  811 FORMAT(4X,A1,X,I2,5(X,E15.8))
      READ(5,581) C5(JL),C6(JL),C7(JL),C8(JL)
  681 FORMAT(9X,4(X,E15.8))
  581 FORMAT(8X,4(X,E12.8))
      WRITE(6,681)C5(JL),C6(JL),C7(JL),C8(JL)
   36 CONTINUE
      JM=JM+4
   35 CONTINUE
      WRITE(6,500)(KSTA(II),II=1,11)
  500 FORMAT(X,11(X,I2))
      DO 37 J=1,50
C     INPUT SATELLITE CONSTANTS
      READ( 5,82) KSAID(J),FREQ(J)
   82 FORMAT(I5,19X,F8.3)
      WRITE( 6,582)KSAID(J),FREQ(J)
  582 FORMAT(X,I5,19X,F8.3)
      IF(KSAID(J))37,38,37
   37 CONTINUE
   38 KSATCT=J-1
  399 IEOVER=0
      INOVER=0
      READ( 2,499,END=219,ERR=399)SAT,AST(1),CSTA,CANT,BDATE,HOUR,MIN,SE
     XC,SLZ(4),EWLOBE,EWRATE,EWACC,EWMER,EWCER,SLQ(4),SNLOBE,SNRATE,SNAC
     XC,SNMER,SNCER,SIGA,AST(4),IFIT,AST(7),IENO,INNO,AST(5),AST(6)
  499 FORMAT(I5,A1,I2,A1,X,I6,X,I2,I2,I2,A1,F6.3,F5.4,F4.3,A3,A3,A1,F6.3
     X,F5.4,F4.3,A3,A3,X,I2,A1,A1,A1,2A1,X,2A1,A1,A1)
      EWACC=EWACC/100.
      SNACC=SNACC/100.
      IF(AST(7).NE.SPX)GO TO 399
      DO 490 L=1,12
      IF(KSTA(L).EQ.CSTA)GO TO 302
  490 CONTINUE
  302 IF(CANT.EQ.PO)GO TO 489
      ANT=46.
      GO TO 488
  489 ANT=57.
  488 DO 487 M=1,50
      IF(SAT.EQ.KSAID(M))GO TO 307
  487 CONTINUE
  307 IF(AST(4).EQ.F1)GO TO 401
      IF(AST(4).EQ.F2)GO TO 402
      IF(AST(4).EQ.F3)GO TO 403
      IF(AST(4).EQ.F4)GO TO 404
      IF(AST(4).EQ.F5)GO TO 405
      IF(AST(4).EQ.F6)GO TO 406
      IF(AST(4).EQ.F7)GO TO 407
      GO TO 399
  401 T=1.
      GO TO 400
  402 T=2.
      GO TO 400
  403 T=10.
      GO TO 400
  404 T=20.
      GO TO 400
  405 T=60.
      GO TO 400
  406 T=120.
      GO TO 400
  407 T=600.
  400 A=EWLOBE
      B=SNLOBE
      IGO=IFIT
  330 IF(IGO.EQ.M1)GO TO 310
      IF(IGO.EQ.M2)GO TO 311
      IF(IGO.EQ.M3)GO TO 312
      IF(IGO.EQ.M4)GO TO 313
      IF(IGO.EQ.M5)GO TO 314
      IF(IGO.EQ.M6)GO TO 315
      IF(IGO.EQ.M7)GO TO 316
      IF(IGO.EQ.M8)GO TO 317
      IF(IGO.EQ.M9)GO TO 318
      IF(IGO.EQ.L1)GO TO 321
      IF(IGO.EQ.L2)GO TO 322
      IF(IGO.EQ.L3)GO TO 323
      IF(IGO.EQ.L4)GO TO 324
      IF(IGO.EQ.L5)GO TO 325
      IF(IGO.EQ.L6)GO TO 326
      IF(IGO.EQ.L7)GO TO 327
      IF(IFIT.EQ.SPX)GO TO 320
      KT=10.
      GO TO 408
  310 KT=1.
      GO TO 408
  311 KT=2.
      GO TO 408
  312 KT=3.
      GO TO 408
  313 KT=4.
      GO TO 408
  314 KT=5.
      GO TO 408
  315 KT=6.
      GO TO 408
  316 KT=7.
      GO TO 408
  317 KT=8.
      GO TO 408
  318 KT=9.
      GO TO 408
  320 KT=10.
      GO TO 408
  321 KT=11.
      GO TO 408
  322 KT=12.
      GO TO 408
  323 KT=13.
      GO TO 408
  324 KT=14.
      GO TO 408
  325 KT=15.
      GO TO 408
  326 KT=16.
      GO TO 408
  327 KT=17.
  408 TT=T*KT
      IF(SLZ(4).EQ.SPA)GO TO 26
      IEOVER=IEOVER-64
      IER=IEOVER/3
      EWR=IER
      EWRATE=EWRATE+SIGN(EWR,EWRATE)
      EWA=(IEOVER-IER*3)/100.
      EWACC=EWACC+SIGN(EWA,EWACC)
   26 IF(SLQ(4).EQ.SPA)GO TO 27
      INOVER=INOVER-64
      INR=INOVER/3
      SNR=INR
      SNRATE=SNRATE+SIGN(SNR,SNRATE)
      SNA=(INOVER-INR*3)/100.
      SNACC=SNACC+SIGN(SNA,SNACC)
 27   IYR=BDATE/10000
      IDATE=BDATE
      CALL DCOUNT(BDATE)
      WRITE(6,498)SAT,AST(1),CSTA,CANT,BDATE,HOUR,MIN,SEC,SLZ(4),EWLOBE,
     XEWRATE,EWACC,EWMER,EWCER,SLQ(4),SNLOBE,SNRATE,SNACC,SNMER,SNCER,SI
     XGA,AST(4),IFIT
498   FORMAT(X,I5,A1,I2,A1,X,I6,X,I2,I2,I2,A1,F6.3,X,F5.4,X,F4.3,X,A3,A3
     X,A1,F6.3,X,F5.4,X,F4.3,X,A3,A3,X,I2,A1,A1)
      AA=A-(EWRATE*KT)+(EWACC*(KT**2))
      BB=B-(SNRATE*KT)+(SNACC*(KT**2))
      AC=A+(EWRATE*KT)+(EWACC*(KT**2))
      BC=B+(SNRATE*KT)+(SNACC*(KT**2))
      DO 21 IM=1,40
      IF(CSTA.EQ.ISTA(IM))GO TO 24
 21   CONTINUE
 24   IF(CANT.NE.PO)GO TO 25
      IM=IM+2
 25   PI=6.2831853
      AO=EWLOBE
      BO=SNLOBE
      CC=AO/((FREQ(M)/136.)*ANT)
      DD=BO/((FREQ(M)/136.)*ANT)
      IF(CANT.EQ.PO)GO TO 40
      IF(ABS(CC).GT..08716)GO TO 52
      IF(ABS(DD).GT..64279)GO TO 52
      EE=BO**3
      GO TO 41
 40   EE=AO**3
      IF(ABS(CC).GT..64279)GO TO 52
      IF(ABS(DD).GT..08716)GO TO 52
 41   A=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
     X(IM)*BO**2)+(C6(IM)*EE)+(C7(IM)*SIN(AO*PI))+(C8(IM)*COS(AO*PI))
      GO TO 53
 52   A=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(AO*PI)
     X)+(C8(IM)*COS(AO*PI))
 53   AO=AA
      BO=BB
      CC=AO/((FREQ(M)/136.)*ANT)
      DD=BO/((FREQ(M)/136.)*ANT)
      IF(CANT.EQ.PO)GO TO 42
      IF(ABS(CC).GT..08716)GO TO 54
      IF(ABS(DD).GT..64279)GO TO 54
      EE=BO**3
      GO TO 43
 42   EE=AO**3
      IF(ABS(CC).GT..64279)GO TO 54
      IF(ABS(DD).GT..08716)GO TO 54
 43   B=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
     X(IM)*BO**2)+(C6(IM)*EE)+(C7(IM)*SIN(AO*PI))+(C8(IM)*COS(AO*PI))
      GO TO 55
 54   B=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(AO*PI)
     X)+(C8(IM)*COS(AO*PI))
 55   AO=AC
      BO=BC
      CC=AO/((FREQ(M)/136.)*ANT)
      DD=BO/((FREQ(M)/136.)*ANT)
      IF(CANT.EQ.PO)GO TO 44
      IF(ABS(CC).GT..08716)GO TO 56
      IF(ABS(DD).GT..64279)GO TO 56
      EE=BO**3
      GO TO 45
```

```
44 EE=AO**3
   IF(ABS(CC).GT..64279)GO TO 56
   IF(ABS(DD).GT..08716)GO TO 56
45 C=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
  X(IM)*BO**2)+(C6(IM)*EE)+(C7(IM)*SIN(AO*PI))+(C8(IM)*COS(AO*PI))
   GO TO 57
56 C=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(AO*PI)
  X)+(C8(IM)*COS(AO*PI))
57 AO=EWLOBE
   BO=SNLOBE
   CC=AO/((FREQ(M)/136.)*ANT)
   DD=BO/((FREQ(M)/136.)*ANT)
   IM=IM+1
   IF(CANT.EQ.PO)GO TO 46
   IF(ABS(CC).GT..08716)GO TO 58
   IF(ABS(DD).GT..64279)GO TO 58
   FF=BO**3
   GO TO 47
46 FF=AO**3
   IF(ABS(CC).GT..64279)GO TO 58
   IF(ABS(DD).GT..08716)GO TO 58
47 D=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
  X(IM)*BO**2)+(C6(IM)*FF)+(C7(IM)*SIN(BO*PI))+(C8(IM)*COS(BO*PI))
   GO TO 59
58 D=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(BO*PI)
  X)+(C8(IM)*COS(BO*PI))
59 AO=AA
   BO=BB
   CC=AO/((FREQ(M)/136.)*ANT)
   DD=BO/((FREQ(M)/136.)*ANT)
   IF(CANT.EQ.PO)GO TO 48
   IF(ABS(CC).GT..08716)GO TO 60
   IF(ABS(DD).GT..64279)GO TO 60
   FF=BO**3
   GO TO 49
48 FF=AO**3
   IF(ABS(CC).GT..64279)GO TO 60
   IF(ABS(DD).GT..08716)GO TO 60
49 E=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
  X(IM)*BO**2)+(C6(IM)*FF)+(C7(IM)*SIN(BO*PI))+(C8(IM)*COS(BO*PI))
   GO TO 61
60 E=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(BO*PI)
  X)+(C8(IM)*COS(BO*PI))
61 AO=AC
   BO=BC
   CC=AO/((FREQ(M)/136.)*ANT)
   DD=BO/((FREQ(M)/136.)*ANT)
   IF(CANT.EQ.PO)GO TO 50
   IF(ABS(CC).GT..08716)GO TO 62
   IF(ABS(DD).GT..64279)GO TO 62
   FF=BO**3
   GO TO 51
50 FF=AO**3
   IF(ABS(CC).GT..64279)GO TO 62
   IF(ABS(DD).GT..08716)GO TO 62
51 F=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
  X(IM)*BO**2)+(C6(IM)*FF)+(C7(IM)*SIN(BO*PI))+(C8(IM)*COS(BO*PI))
   GO TO 63
62 F=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C7(IM)*SIN(BO*PI)
  X)+(C8(IM)*COS(BO*PI))
63 AA=B
   AC=C
   B=D
```

```
      BB=E
      BC=F
      TIM=((HOUR*3600)+(60*MIN))+SEC
      TIMC=TIM+TT
      TIMA=TIM-TT
      IF(TIMA)28,29,29
   28 TIMA=86400+TIMA
      BDATE=BDATE-1
   29 IAO=2
      IBO=3
C     MAXIUM OBSERVATIONS
  486 ACOS=AA/((FREQ(M)/136.)*ANT)
      BCOS=BB/((FREQ(M)/136.)*ANT)
      IACOS=ACOS*1000000+.5
      IBCOS=BCOS*1000000+.5
      CALL ZERO(8,IACOS,LCOS)
      CALL ZERO(8,IBCOS,MCOS)
      IF(TIMA-86400)303,303,304
  304 BDATE=BDATE+1
      TIMA=TIMA-86400
  303 HOUR=TIMA/3600.
      MIN=((TIMA-(HOUR*3600.))/60.)
      SEC=(TIMA-((HOUR*3600.)+(MIN*60.)))
      BOUR=HOUR*100.
      BOUR=BOUR+MIN
      ASEC=SEC*1000.
      CALL ZERO(5,BOUR,IOUR)
      CALL ZERO(6,ASEC,ISEC)
      CALL PYRD(IYR,BDATE,YRMODA)
      CALL ZERO(7,YRMODA,ARMODA)
      WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
      WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
      WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
      WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
  484 ACOS=A/((FREQ(M)/136.)*ANT)
      BCOS=B/((FREQ(M)/136.)*ANT)
      IACOS=ACOS*1000000+.5
      IBCOS=BCOS*1000000+.5
      CALL ZERO(8,IACOS,LCOS)
      CALL ZERO(8,IBCOS,MCOS)
      BDATE=IDATE
      CALL DCOUNT(BDATE)
      HOUR=TIM/3600.
      MIN=((TIM -(HOUR*3600.))/60.)
      SEC=(TIM -((HOUR*3600.)+(MIN*60.)))
      BOUR=HOUR*100.
      BOUR=BOUR+MIN
      ASEC=SEC*1000.
      CALL ZERO(5,BOUR,IOUR)
      CALL ZERO(6,ASEC,ISEC)
      CALL PYRD(IYR,BDATE,YRMODA)
      CALL ZERO(7,YRMODA,ARMODA)
      WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
      WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
      WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
      WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
      ACOS=AC/((FREQ(M)/136.)*ANT)
      BCOS=BC/((FREQ(M)/136.)*ANT)
      IACOS=ACOS*1000000+.5
      IBCOS=BCOS*1000000+.5
      CALL ZERO(8,IACOS,LCOS)
      CALL ZERO(8,IBCOS,MCOS)
      BDATE=IDATE
```

```
    CALL DCOUNT(BDATE)
    IF(TIMC-86400)305,305,306
306 BDATE=BDATE+1
    TIMC=TIMC-86400
305 HOUR=TIMC/3600.
    MIN=((TIMC-(HOUR*3600.))/60.)
    SEC=(TIMC-((HOUR*3600.)+(MIN*60.)))
    BOUR=HOUR*100.
    BOUR=BOUR+MIN
    ASEC=SEC*1000.
    CALL ZERO(5,BOUR,IOUR)
    CALL ZERO(6,ASEC,ISEC)
    CALL PYRD(IYR,BDATE,YRMODA)
    CALL ZERO(7,YRMODA,ARMODA)
    WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
    WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
    WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
    WRITE(15,218)SAT,STATIO(L),ARMODA,IOUR,ISEC,MCOS,CANT,IBO
    IAO=9
    IF(CANT.EQ.PO)GO TO 353
    ZOBE=EWLOBE
    RATE=EWRATE
    ZLOB=SNLOBE
    ZRAT=SNRATE
    GO TO 354
353 ZOBE=SNLOBE
    RATE=SNRATE
    ZLOB=EWLOBE
    ZRAT=EWRATE
354 ZB=((-ZOBE)/RATE)*T
    IB=ZB
    TIMA=TIM+IB
    ZA=(ZRAT*ZB)/T
    ZA=ZA+ZLOB
    ACOS=ZA/((FREQ(M)/136.)*ANT)
    IACOS=ACOS*1000000+.5
    CALL ZERO(8,IACOS,LCOS)
    BDATE=IDATE
    CALL DCOUNT(BDATE)
    IF(TIMA-86400)350,350,351
351 BDATE=BDATE+1
    TIMA=TIMA-86400
350 HOUR=TIMA/3600
    MIN=((TIMA-(HOUR*3600.))/60.)
    SEC=(TIMA-((HOUR*3600.)+(MIN*60)))
    BOUR=HOUR*100.
    BOUR=BOUR+MIN
    ASEC=SEC*1000.
    CALL ZERO(5,BOUR,IOUR)
    CALL ZERO(6,ASEC,ISEC)
    CALL PYRD(IYR,BDATE,YRMODA)
    CALL ZERO(7,YRMODA,ARMODA)
    WRITE( 6,352)
352 FORMAT(49H PASS CROSSOVER TIME AND ZENITH ANGLE IN DIR.COS
    WRITE( 6,217)SAT,STATIO(L),ARMODA,IOUR,ISEC,LCOS,CANT,IAO
217 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,8A1,3X,A1,I1)
218 FORMAT(X,I5,X,A6,7A1,5A1,6A1,28X,8A1,3X,A1,I1)
    GO TO 399
219 DO 340 I=1,80
    BLANK(I)=SPA
340 CONTINUE
    WRITE(15,220)(BLANK(I),I=1,80)
    WRITE(15,220)(BLANK(I),I=1,80)
```

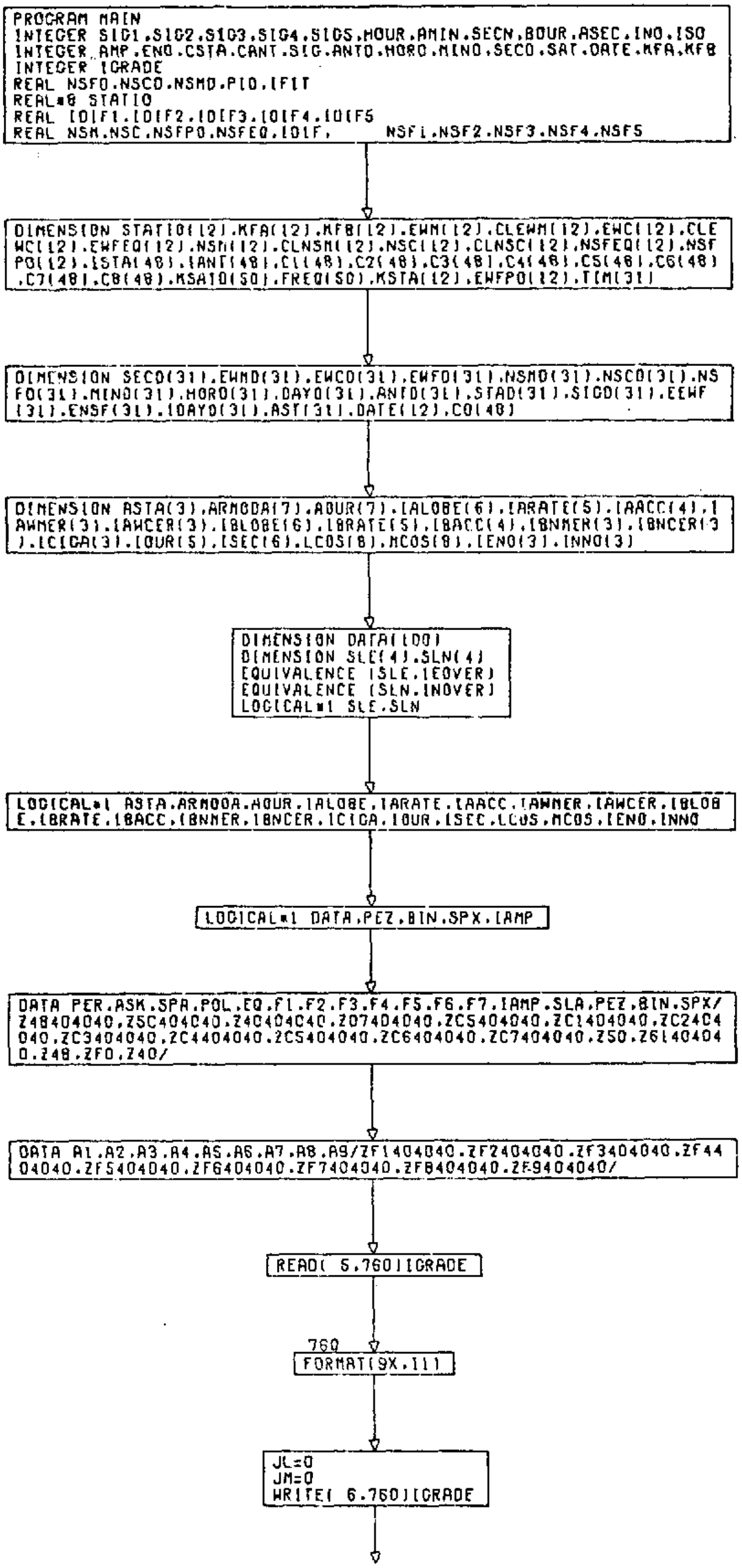
```
  220 FORMAT(80A1)
      END FILE 15
      REWIND 15
    1 STOP
      END
      SUBROUTINE PYRD(NYR,JDAY,NYMODA)
      DIMENSION N(24)
      DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10),N(11),N(12
     1),N(13),N(14),N(15),N(16),N(17),N(18),N(19),N(20),N(21),N(22),N(23
     2),N(24)/0,31,59,90,120,151,181,212,243,273,304,334,0,31,60,91,121,
     3152,182,213,244,274,305,335/
      J=0
      A=NYR/4.
      L=A
      A=A-L
      IF(A)2,1,2
    1 J=12
    2 DO 5 K=1,12
      M=J+K
      IF(JDAY.LE.N(M))GO TO 4
    5 CONTINUE
    4 M=J+K-1
      NDAY=JDAY-N(M)
      K=K-1
      NYMODA=(NYR*10000)+(K*100)+NDAY
      RETURN
      END
      SUBROUTINE DCOUNT%IDATE<
C
C
      DIMENSION IA%12<
C
      DATA IA%1<,IA%2<,IA%3<,IA%4<,IA%5<,IA%6<,IA%7<,IA%8<,IA%9<,
     1IA%10<,IA%11<,IA%12<
     1/0,31,59,90,120,151,181,212,243,273,304,334/
      JSUM#0
      IYR#IDATE/10000
      IMO#%IDATE-%IYR*10000<</100
      IDA#IDATE-%IYR*10000<-%IMO*100)
      JSUM#JSUM&IA%IMO<&IDA
      ISUM#0
      ICO=0
      J#1
      IRDA#365
    1 IF%ICO.EQ.0<GO TO 4
      ISUM#ISUM&IRDA
      J#J&1
      IF%J.EQ.4<GO TO 2
      IF%J.LE.4<GO TO 3
      J#1
      IRDA#364
    2 IRDA#IRDA&1
    3 ICO#ICO-1
      GO TO 1
    4 LYR#0
      FYR#IYR
      SYR#IYR/4
      IF%FYR/4.0.NE.SYR<GO TO 5
      IF(IMO.LT.3) GO TO 5
      LYR#LYR&1
    5 IDATE#ISUM&JSUM&LYR
      RETURN
    6 STOP
      END
      SUBROUTINE ZERO(N,IIN,AREA)
      DIMENSION DIV(7)
      INTEGER DIV
      LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)
      DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8
     X),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
     X/
      DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
     X00,10000,1000,100,10,1/
      AREA(1)=PLUS
      IF(IIN)1,2,2
    1 AREA(1)=MINUS
      IIN=IABS(IIN)
    2 IJ=9-N
      K=2
      J=N-1
      DO 4 I=1,J
      ITEMP=IIN/DIV(IJ)
      IF(ITEMP.LE.9.)GO TO 6
      AREA(K)=ASK
      GO TO 5
    6 AREA(K)=IC(ITEMP+1)
    5 IIN=IIN-(ITEMP*DIV(IJ))
      IJ=IJ+1
      K=K+1
    4 CONTINUE
      RETURN
      END
/*
// EXEC LINKGO
//GO.SYSUDUMP  DD   SYSOUT=A
//GO.FT02F001  DD DSNAME=&CTL,DISP=(OLD,DELETE)
//GO.DATA5 DD DSNAME=MINCON,UNIT=2314,VOLUME=SER=G2NCS9,               *
//             DISP=(OLD,KEEP),DCB=(,RECFM=F,LRECL=80,BLKSIZE=80)
//GO.FT15F001  DD SYSOUT=B,DCB=(,RECFM=F,LRECL=80,BLKSIZE=80)
/*
```

1814 CARDS

Appendix E

QUADRATIC FLOW DIAGRAM

Presented on the following pages is a flow diagram of the quadratic preprocessing program. Both main program sections corresponding to the cubic MIN-B and OBS-B programs are displayed as are the subroutine flow diagrams. The listings from which these diagrams were taken are printed in appendix D. The descriptions in sections 4, 5, 6, and 7 of volume 1 will serve to explain the functions of this, the quadratic, as well as the cubic preprocessing program.



```
PROGRAM MAIN
INTEGER SIG1.SIG2.SIG3.SIG4.SIG5.HOUR.AMIN.SECN.BOUR.ASEC.INO.ISO
INTEGER AMP.ENO.CSTA.CANT.SIG.ANTO.HORO.MINO.SECO.SAT.DATE.KFA.KFB
INTEGER IGRADE
REAL NSFO.NSCO.NSMD.PIO.IFIT
REAL*8 STATIO
REAL IDIF1.IDIF2.IDIF3.IDIF4.IDIF5
REAL NSM.NSC.NSFPO.NSFEQ.IDIF,     NSF1.NSF2.NSF3.NSF4.NSF5
```

```
DIMENSION STATIO(12).KFA(12).KFB(12).EWM(12).CLEWM(12).EWC(12).CLE
WC(12).EWFEQ(12).NSM(12).CLNSM(12).NSC(12).CLNSC(12).NSFEQ(12).NSF
PO(12).ISTA(48).IANT(48).C1(48).C2(48).C3(48).C4(48).C5(48).C6(48)
.C7(48).C8(48).KSAID(50).FREQ(50).KSTA(12).EWFPO(12).TIM(31)
```

```
DIMENSION SECO(31).EWMD(31).EWCD(31).EWFD(31).NSMD(31).NSCD(31).NS
FD(31).MINO(31).HORO(31).DAYO(31).ANTD(31).STAD(31).SIGD(31).EEWF
(31).ENSF(31).IDAYO(31).AST(31).DATE(12).CO(48)
```

```
DIMENSION ASTA(3).ARMODA(7).AOUR(7).IALOBE(6).IARATE(5).IAACC(4).I
AWMER(3).IAWCER(3).IBLOBE(6).IBRATE(5).IBACC(4).IBNMER(3).IBNCER(3
).ICIGA(3).IOUR(5).ISEC(6).LCOS(8).MCOS(8).IENO(3).INNO(3)
```

```
DIMENSION DATA(100)
DIMENSION SLE(4).SLN(4)
EQUIVALENCE (SLE.IEOVER)
EQUIVALENCE (SLN.INOVER)
LOGICAL*1 SLE.SLN
```

```
LOGICAL*1 ASTA.ARMODA.AOUR.IALOBE.IARATE.IAACC.IAWMER.IAWCER.IBLOB
E.IBRATE.IBACC.IBNMER.IBNCER.ICIGA.IOUR.ISEC.LCOS.MCOS.IENO.INNO
```

```
LOGICAL*1 DATA.PEZ.BIN.SPX.IAMP
```

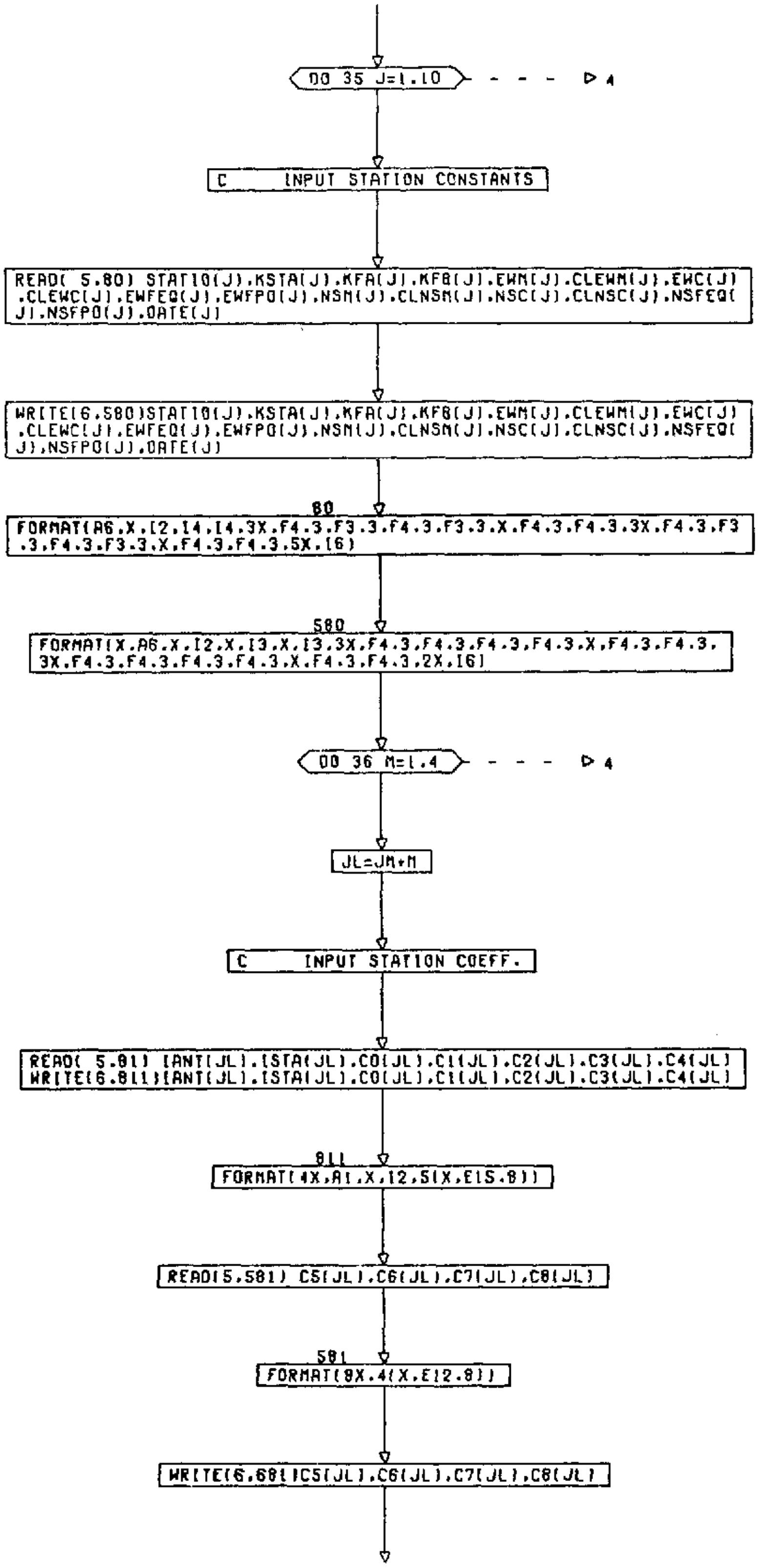
```
DATA PER.ASK.SPA.POL.EQ.F1.F2.F3.F4.F5.F6.F7.IAMP.SLA.PEZ.BIN.SPX/
Z48404040.Z5C404040.Z40404040.Z07404040.ZC5404040.ZC1404040.ZC2404
040.ZC3404040.ZC4404040.ZC5404040.ZC6404040.ZC7404040.Z50.Z6140404
0.Z48.ZF0.Z40/
```

```
DATA A1.A2.A3.A4.A5.A6.A7.A8.A9/ZF1404040.ZF2404040.ZF3404040.ZF44
04040.ZF5404040.ZF6404040.ZF7404040.ZF8404040.ZF9404040/
```

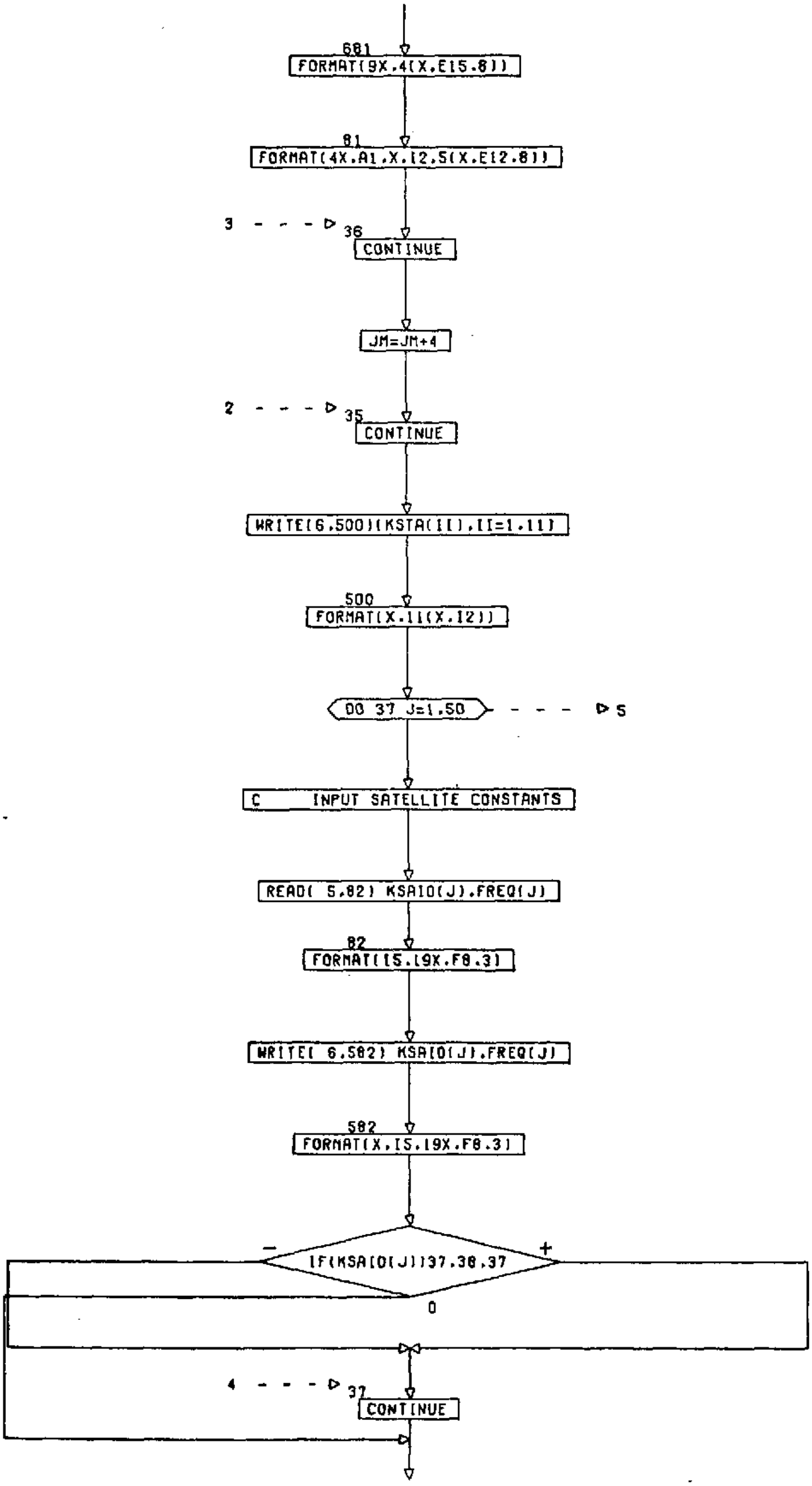
```
READ( 5.760)IGRADE
```

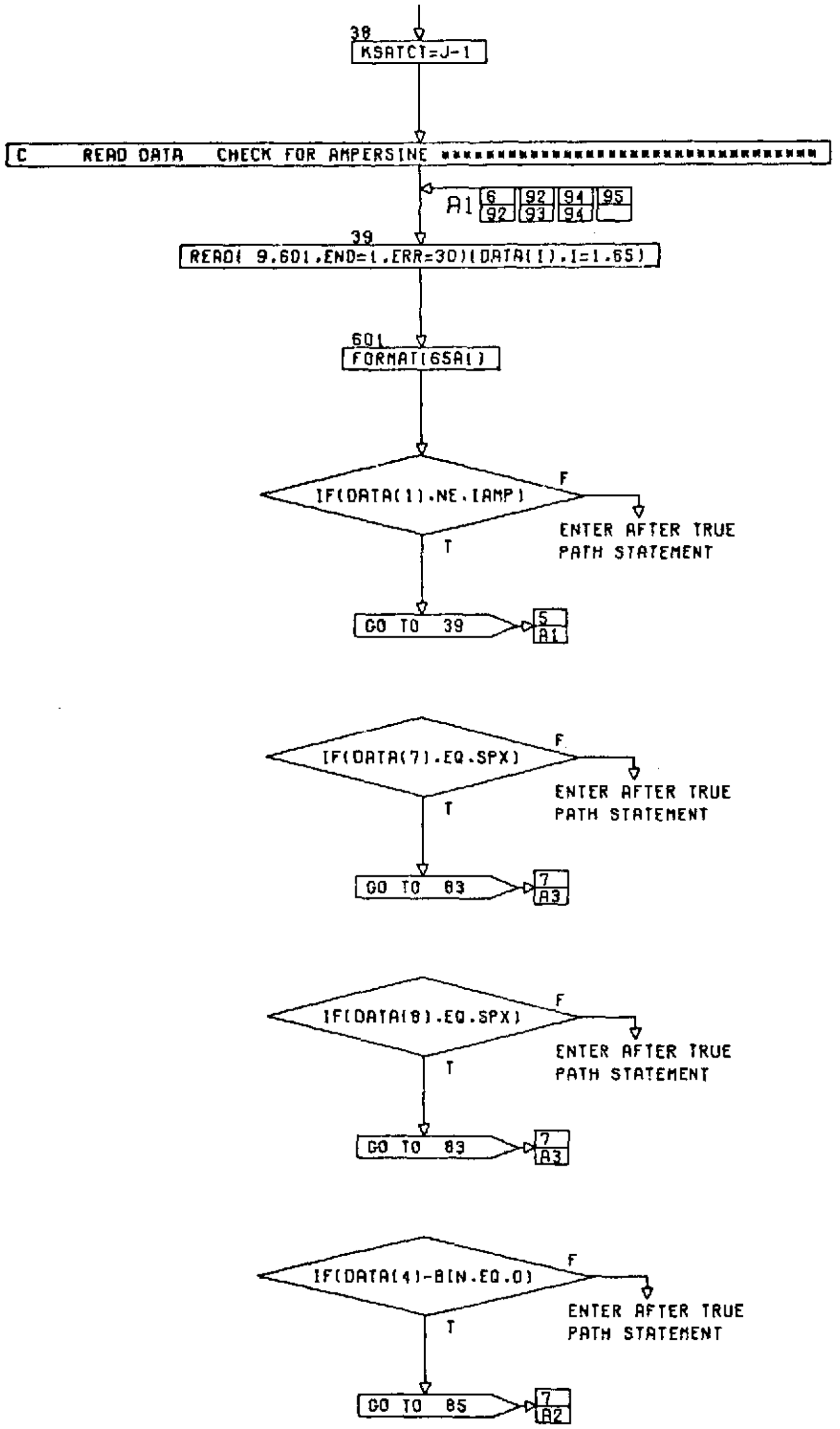
```
760 FORMAT(9X.I1)
```

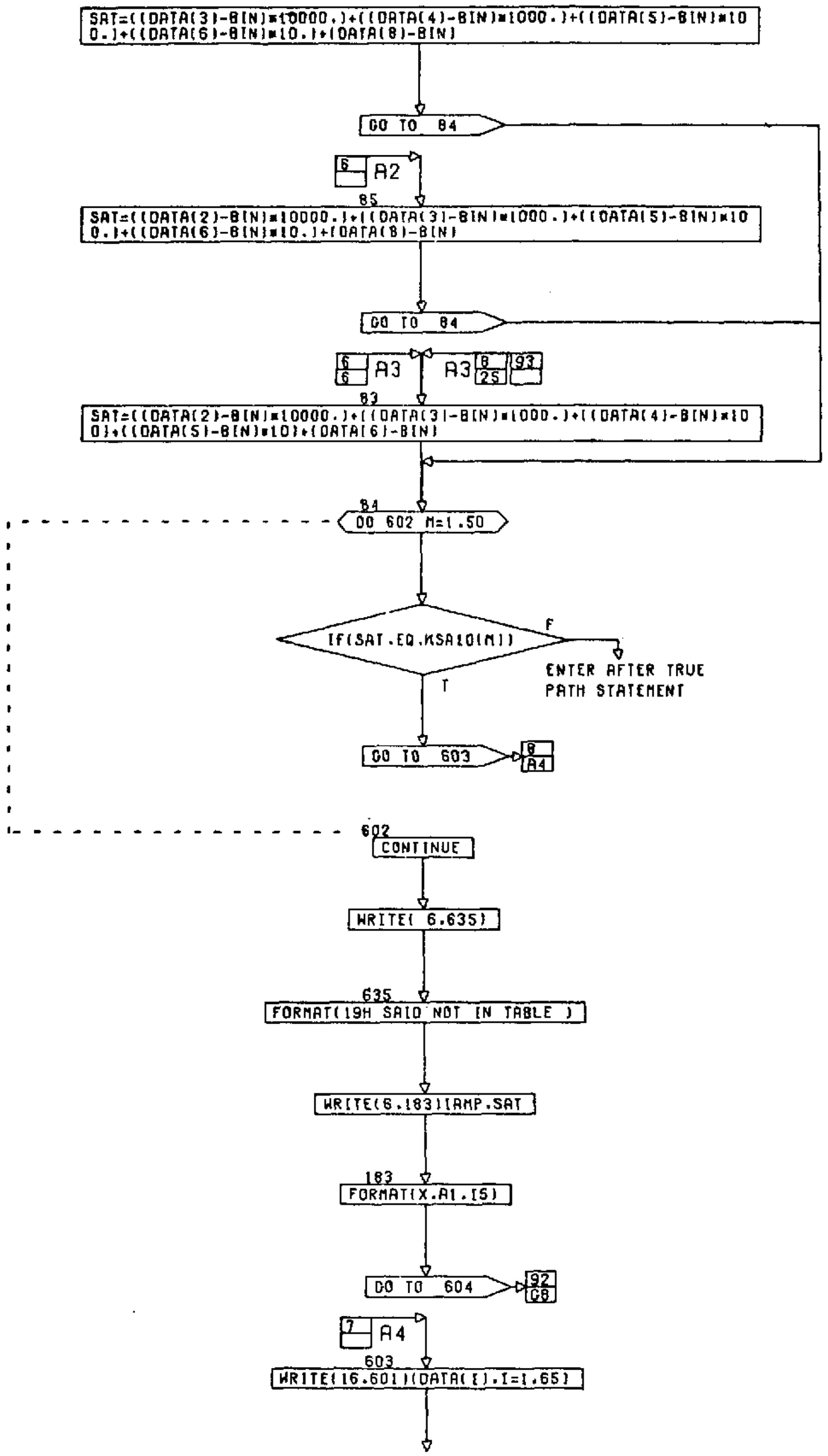
```
JL=0
JM=0
WRITE( 6.760)IGRADE
```

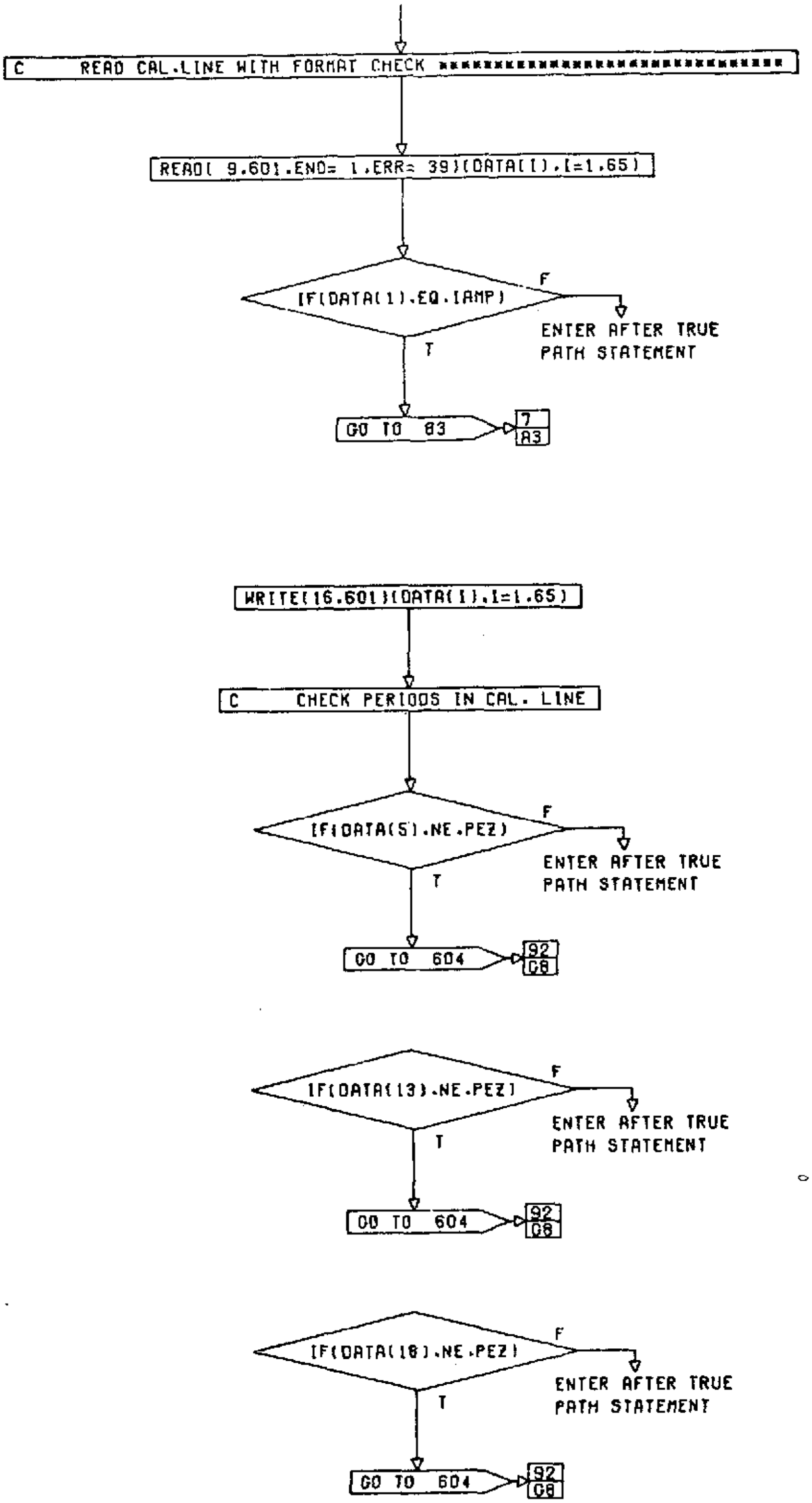


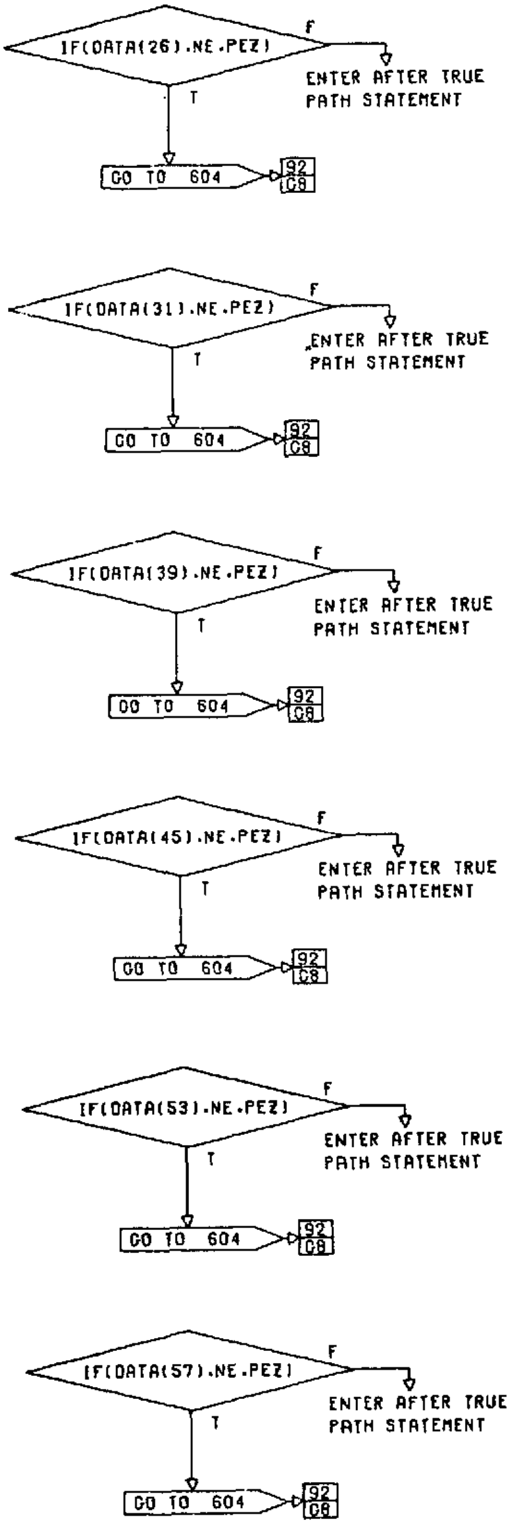
```
DO 35 J=1.10

C      INPUT STATION CONSTANTS

READ( 5.80) STATIO(J).KSTA(J).KFA(J).KFB(J).EWM(J).CLEWM(J).EWC(J)
.CLEWC(J).EWFEQ(J).EWFPO(J).NSM(J).CLNSM(J).NSC(J).CLNSC(J).NSFEQ(
J).NSFPO(J).DATE(J)

WRITE(6.580)STATIO(J).KSTA(J).KFA(J).KFB(J).EWM(J).CLEWM(J).EWC(J)
.CLEWC(J).EWFEQ(J).EWFPO(J).NSM(J).CLNSM(J).NSC(J).CLNSC(J).NSFEQ(
J).NSFPO(J).DATE(J)

80
FORMAT(A6.X.I2.I4.I4.3X.F4.3.F3.3.F4.3.F3.3.X.F4.3.F4.3.3X.F4.3.F3
.3.F4.3.F3.3.X.F4.3.F4.3.5X.I6)

580
FORMAT(X.A6.X.I2.X.I3.X.I3.3X.F4.3.F4.3.F4.3.F4.3.X.F4.3.F4.3.
3X.F4.3.F4.3.F4.3.F4.3.X.F4.3.F4.3.2X.I6)

DO 36 M=1.4

JL=JM+M

C      INPUT STATION COEFF.

READ( 5.811) IANT(JL).ISTA(JL).C0(JL).C1(JL).C2(JL).C3(JL).C4(JL)
WRITE(6.811)IANT(JL).ISTA(JL).C0(JL).C1(JL).C2(JL).C3(JL).C4(JL)

811
FORMAT(4X.A1.X.I2.5(X.E15.8))

READ(5.581) C5(JL).C6(JL).C7(JL).C8(JL)

581
FORMAT(8X.4(X.E12.8))

WRITE(6.681)C5(JL).C6(JL).C7(JL).C8(JL)
```

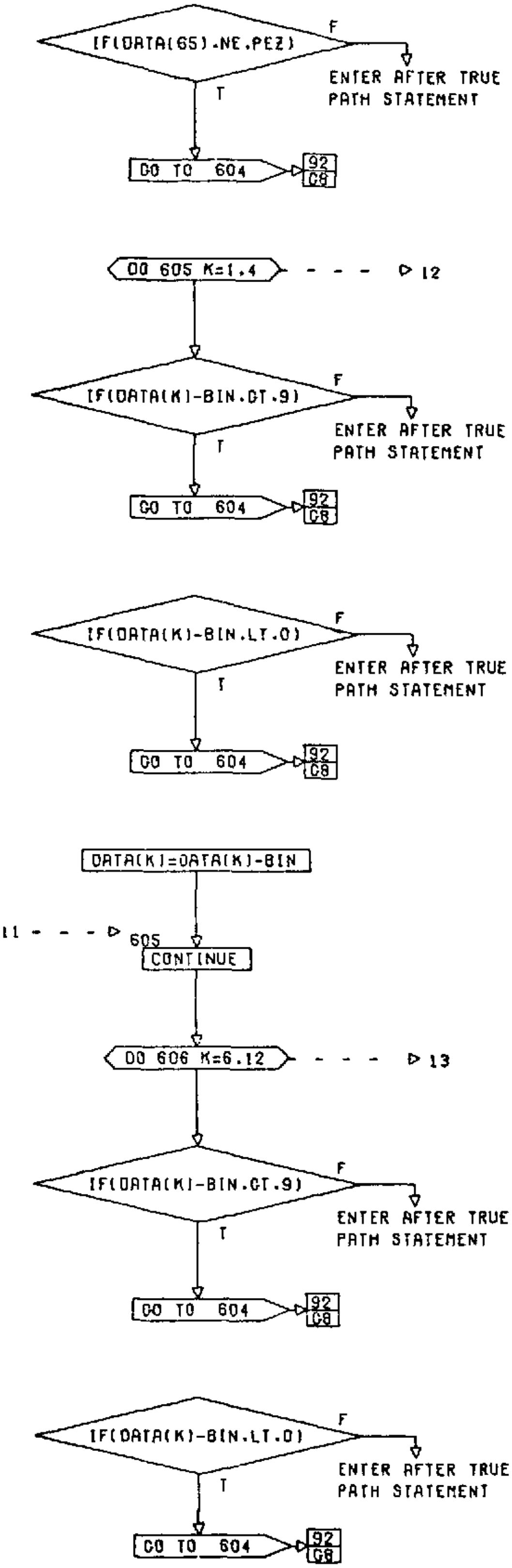

681
FORMAT(9X.4(X.E15.8))
81
FORMAT(4X.A1.X.I2.5(X.E12.8))
3
36
CONTINUE
JM=JM+4
2
35
CONTINUE
WRITE(6.500)(KSTA(II).II=1.11)
500
FORMAT(X.11(X.I2))
DO 37 J=1.50
5
C INPUT SATELLITE CONSTANTS
READ(5.82) KSAIO(J).FREQ(J)
82
FORMAT(I5.19X.F8.3)
WRITE(6.582) KSAIO(J).FREQ(J)
582
FORMAT(X.I5.19X.F8.3)
−
IF(KSAIO(J))37.38.37
+
0
4
37
CONTINUE

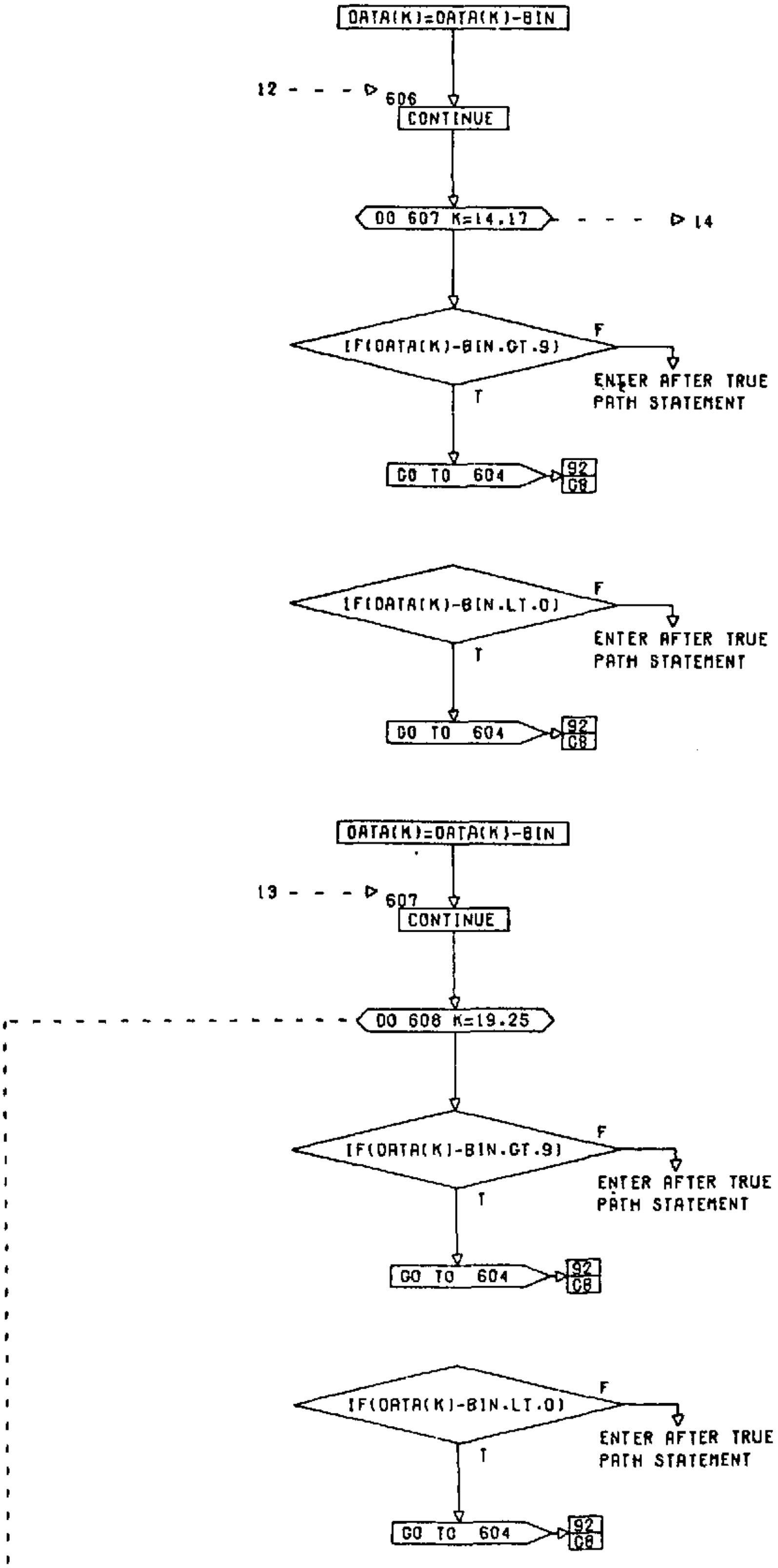

38
KSATCT=J-1
C READ DATA CHECK FOR AMPERSINE ************************************
A1
6
92
92
93
94
94
95
39
READ(9,601,END=1,ERR=30)(DATA(I),I=1,65)
601
FORMAT(65A1)
IF(DATA(1).NE.IAMP)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 39
5
A1
IF(DATA(7).EQ.SPX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 83
7
A3
IF(DATA(8).EQ.SPX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 83
7
A3
IF(DATA(4)-BIN.EQ.0)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 85
7
A2

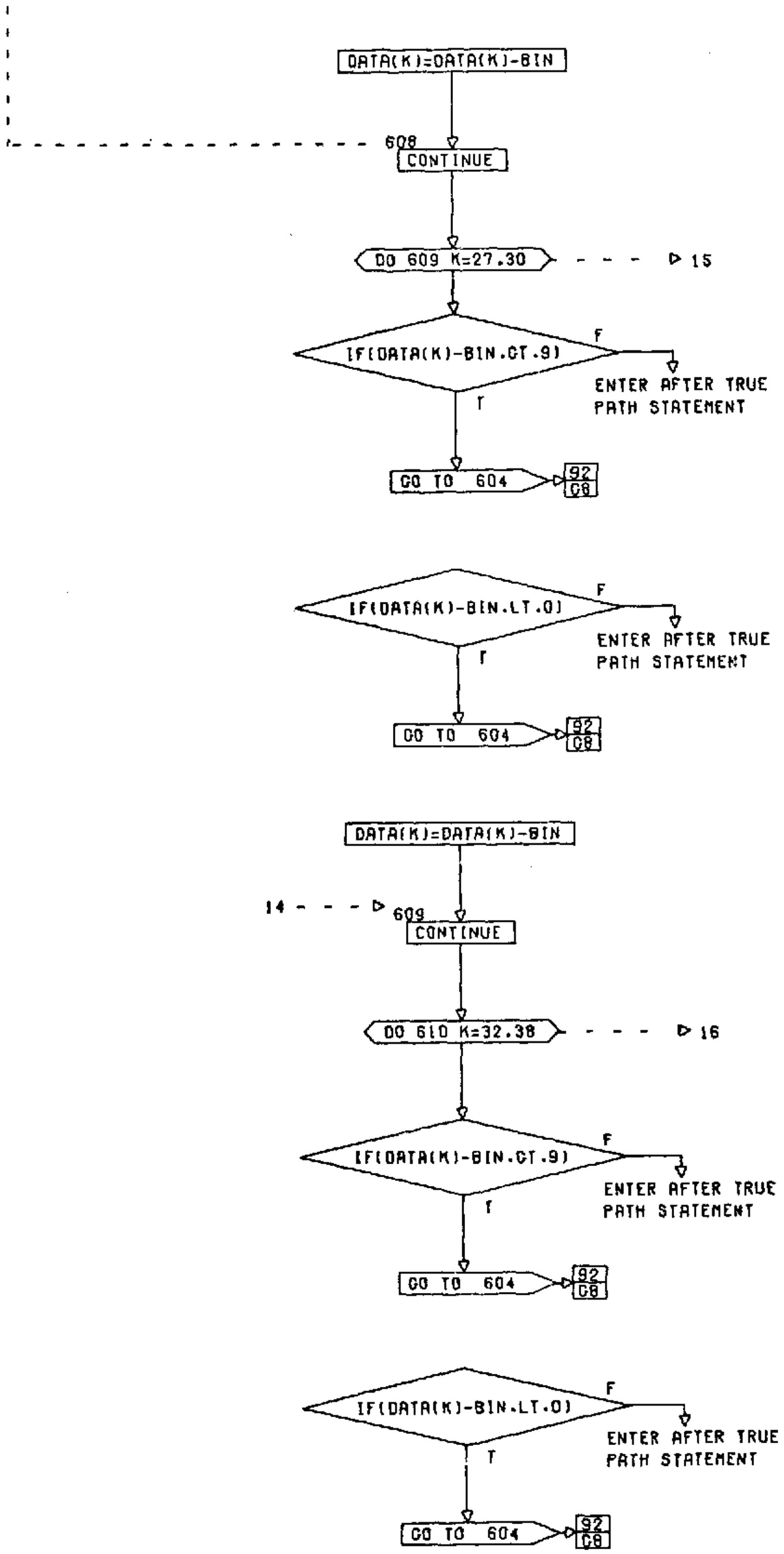

SAT=((DATA(3)-BIN)*10000.)+((DATA(4)-BIN)*1000.)+((DATA(5)-BIN)*10
0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
GO TO 84
6
A2
85
SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(5)-BIN)*10
0.)+((DATA(6)-BIN)*10.)+(DATA(8)-BIN)
GO TO 84
6
6
A3
A3
8
25
93
83
SAT=((DATA(2)-BIN)*10000.)+((DATA(3)-BIN)*1000.)+((DATA(4)-BIN)*10
0)+((DATA(5)-BIN)*10)+(DATA(6)-BIN)
84
DO 602 M=1.50
IF(SAT.EQ.KSAID(M))
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 603
8
A4
602
CONTINUE
WRITE(6.635)
635
FORMAT(19H SAID NOT IN TABLE)
WRITE(6.183)IAMP.SAT
183
FORMAT(X.A1.I5)
GO TO 604
92
C8
7
A4
603
WRITE(16.601)(DATA(I).I=1.65)

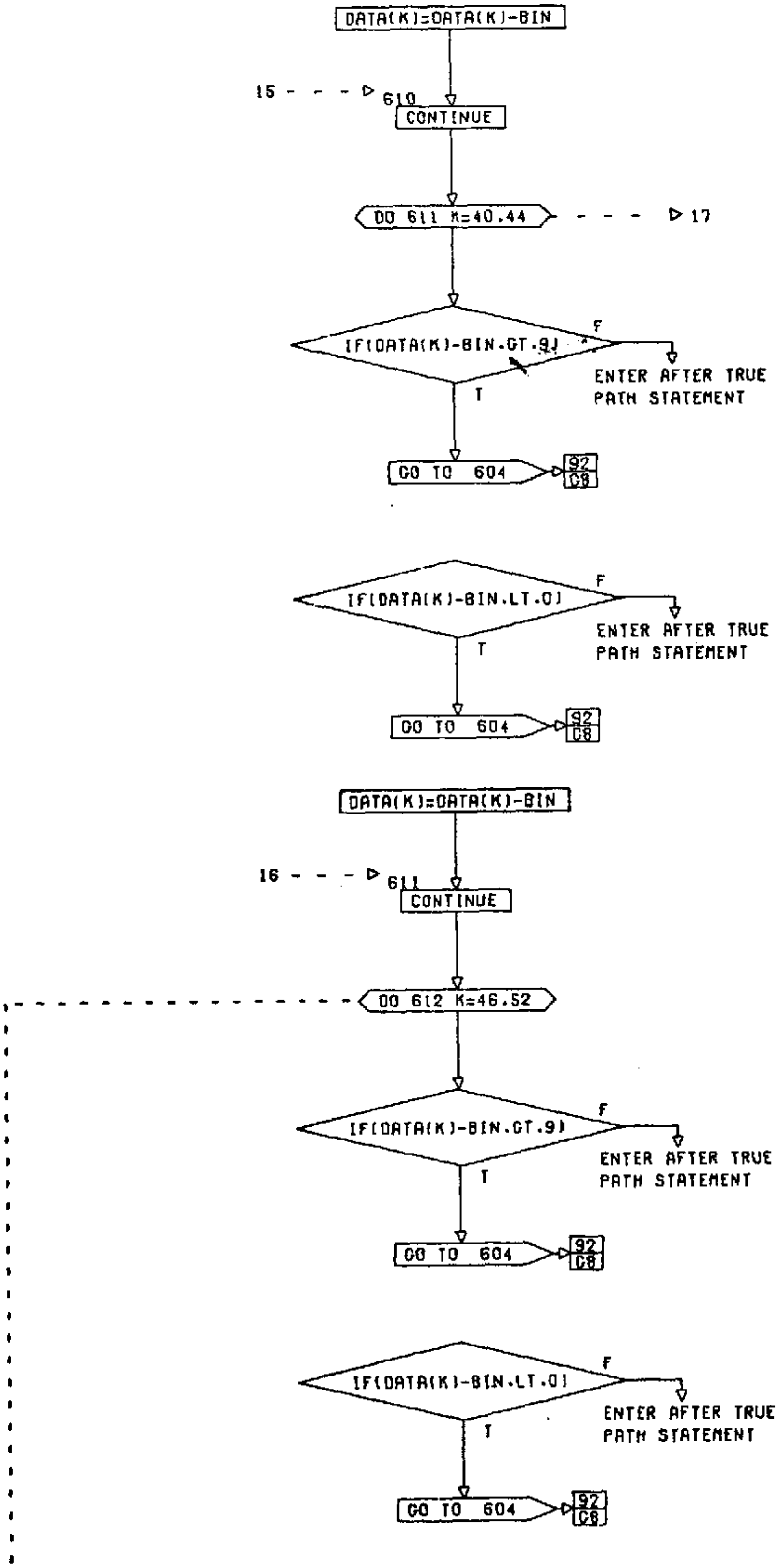

C READ CAL.LINE WITH FORMAT CHECK ***********************************
READ(9.601.END= 1.ERR= 39)(DATA(I).I=1.65)
IF(DATA(1).EQ.IAMP)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 83
7
A3
WRITE(16.601)(DATA(I).I=1.65)
C CHECK PERIODS IN CAL. LINE
IF(DATA(5).NE.PEZ)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 604
92
C8
IF(DATA(13).NE.PEZ)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 604
92
C8
IF(DATA(18).NE.PEZ)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 604
92
C8

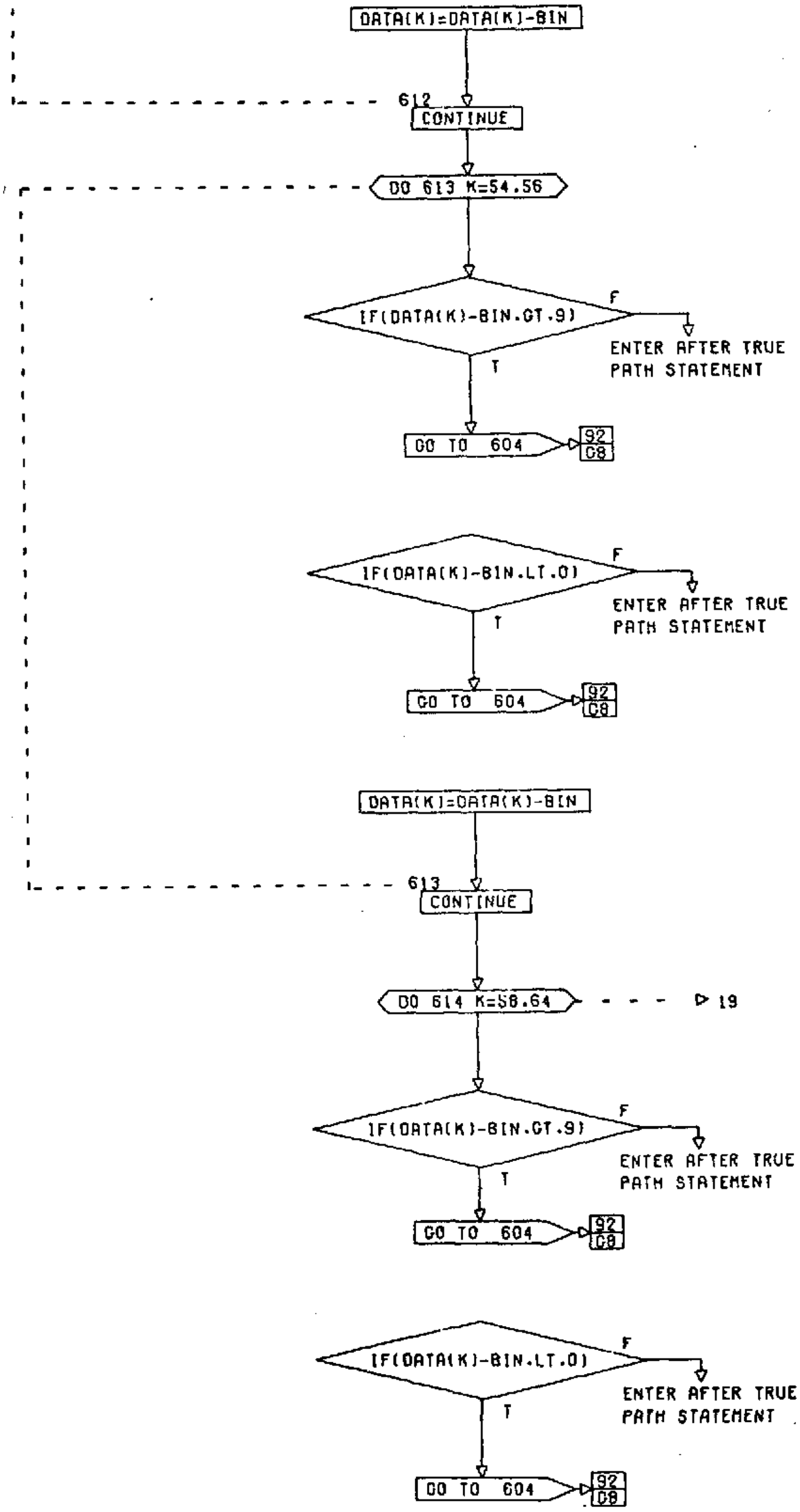

IF(DATA(26).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(31).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(39).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(45).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(53).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(57).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8

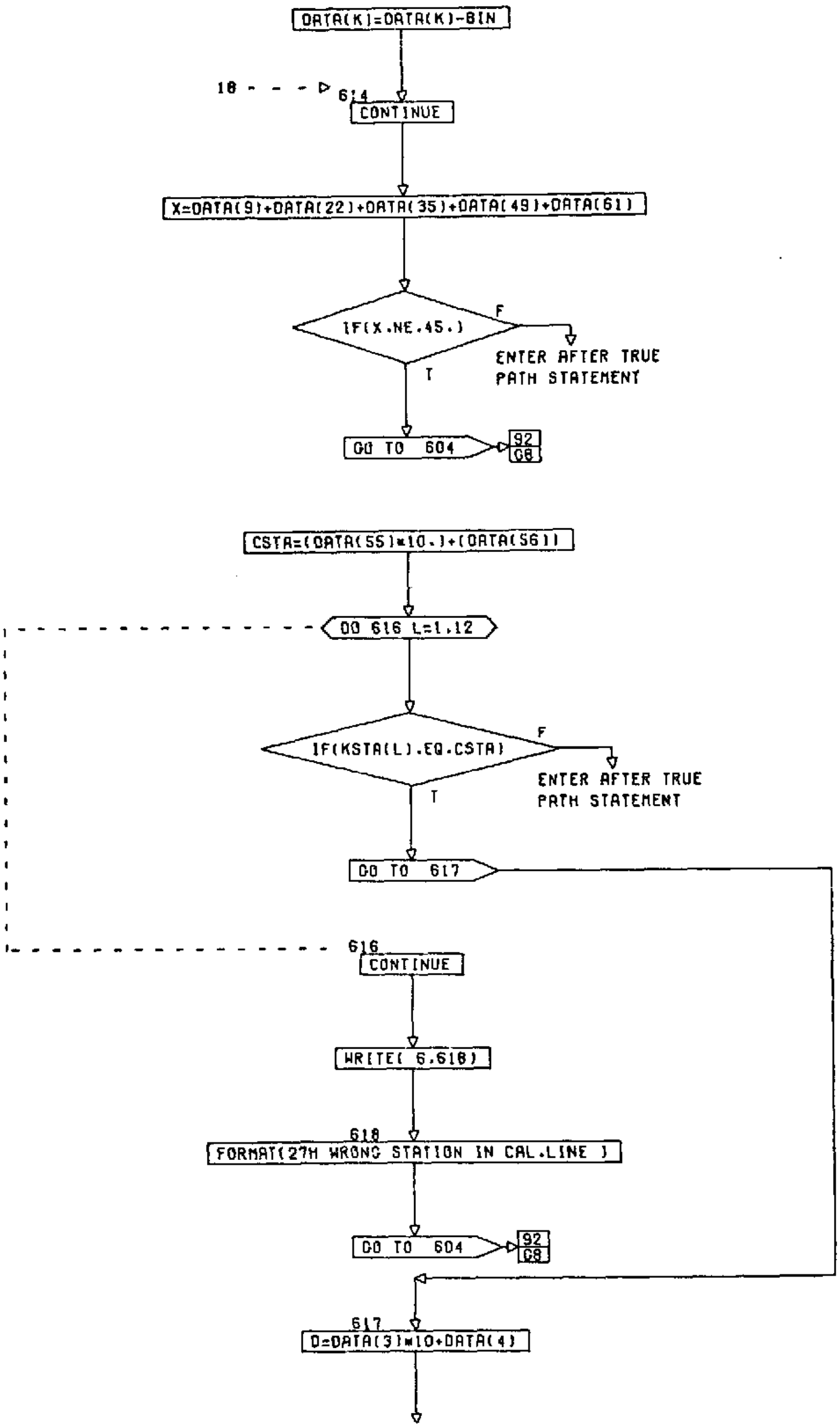

IF(DATA(65).NE.PEZ)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
DO 605 K=1,4
12
IF(DATA(K)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
DATA(K)=DATA(K)-BIN
11
605
CONTINUE
DO 606 K=6,12
13
IF(DATA(K)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8

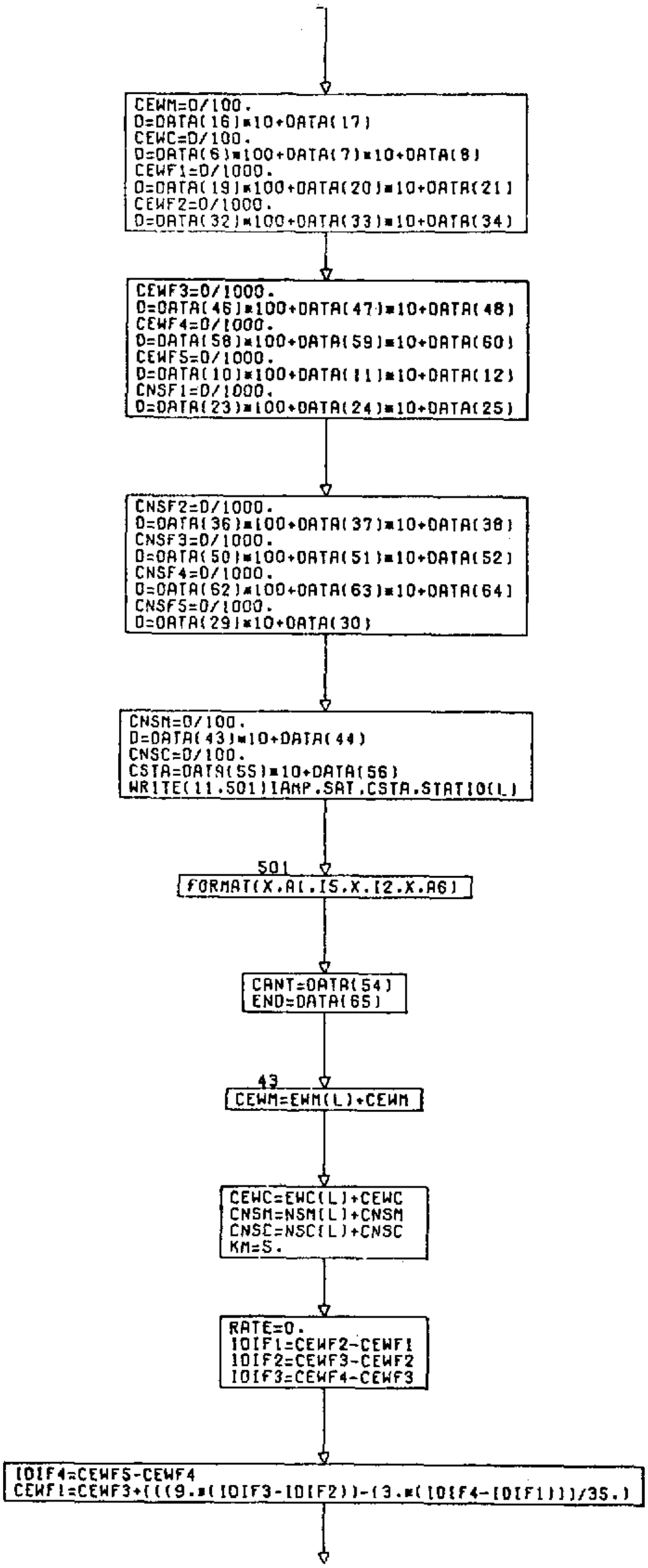

DATA(K)=DATA(K)-BIN
12
606
CONTINUE
DO 607 K=14,17
14
IF(DATA(K)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
DATA(K)=DATA(K)-BIN
13
607
CONTINUE
DO 608 K=19,25
IF(DATA(K)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 604
92
C8


DATA(K)=DATA(K)-BIN
608
CONTINUE
DO 609 K=27,30
15
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
DATA(K)=DATA(K)-BIN
14
609
CONTINUE
DO 610 K=32,38
16
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8

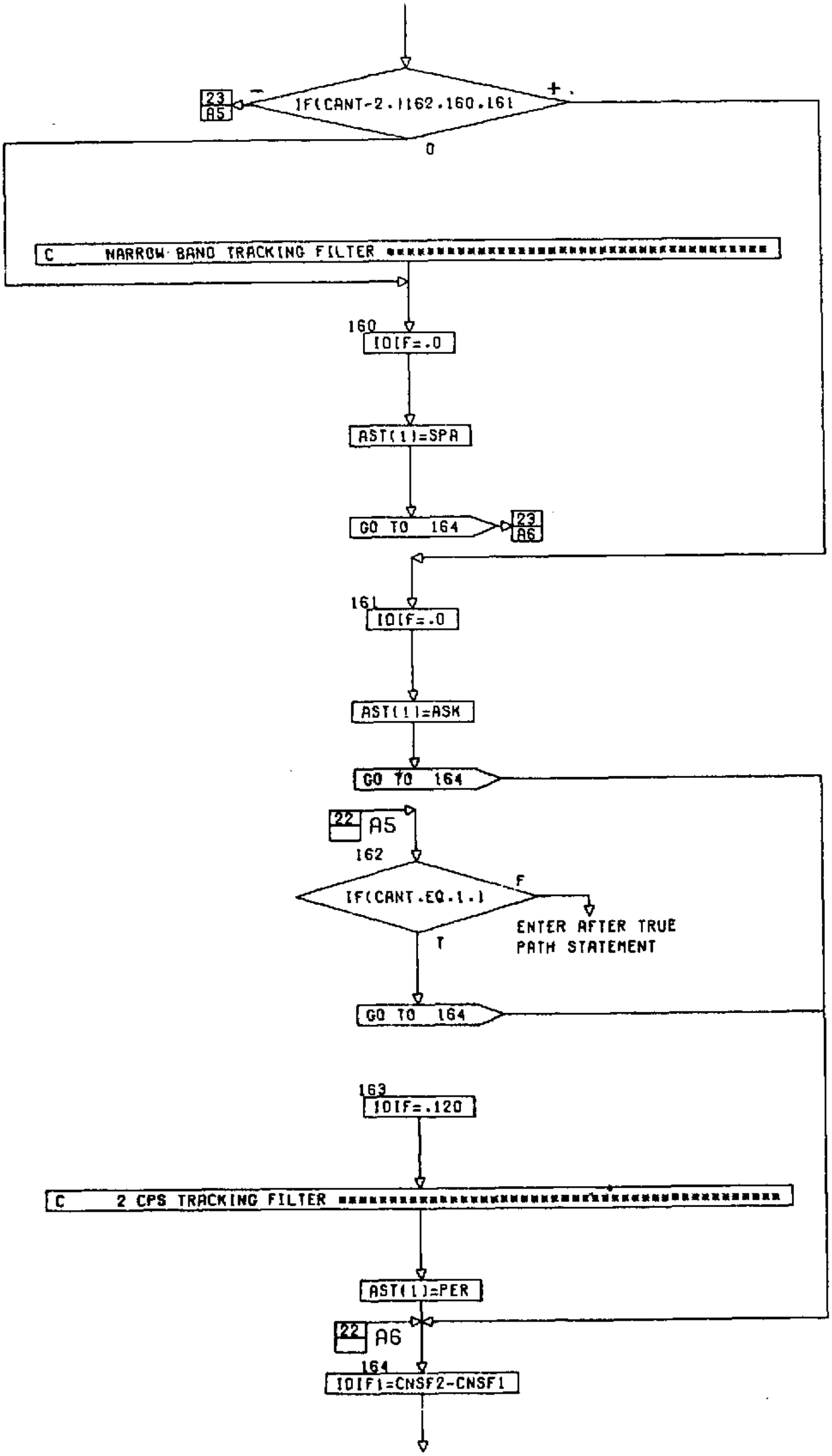

DATA(K)=DATA(K)-BIN
15 - - - ▷ 610
CONTINUE
DO 611 K=40,44
- - - ▷ 17
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
DATA(K)=DATA(K)-BIN
16 - - - ▷ 611
CONTINUE
DO 612 K=46,52
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8

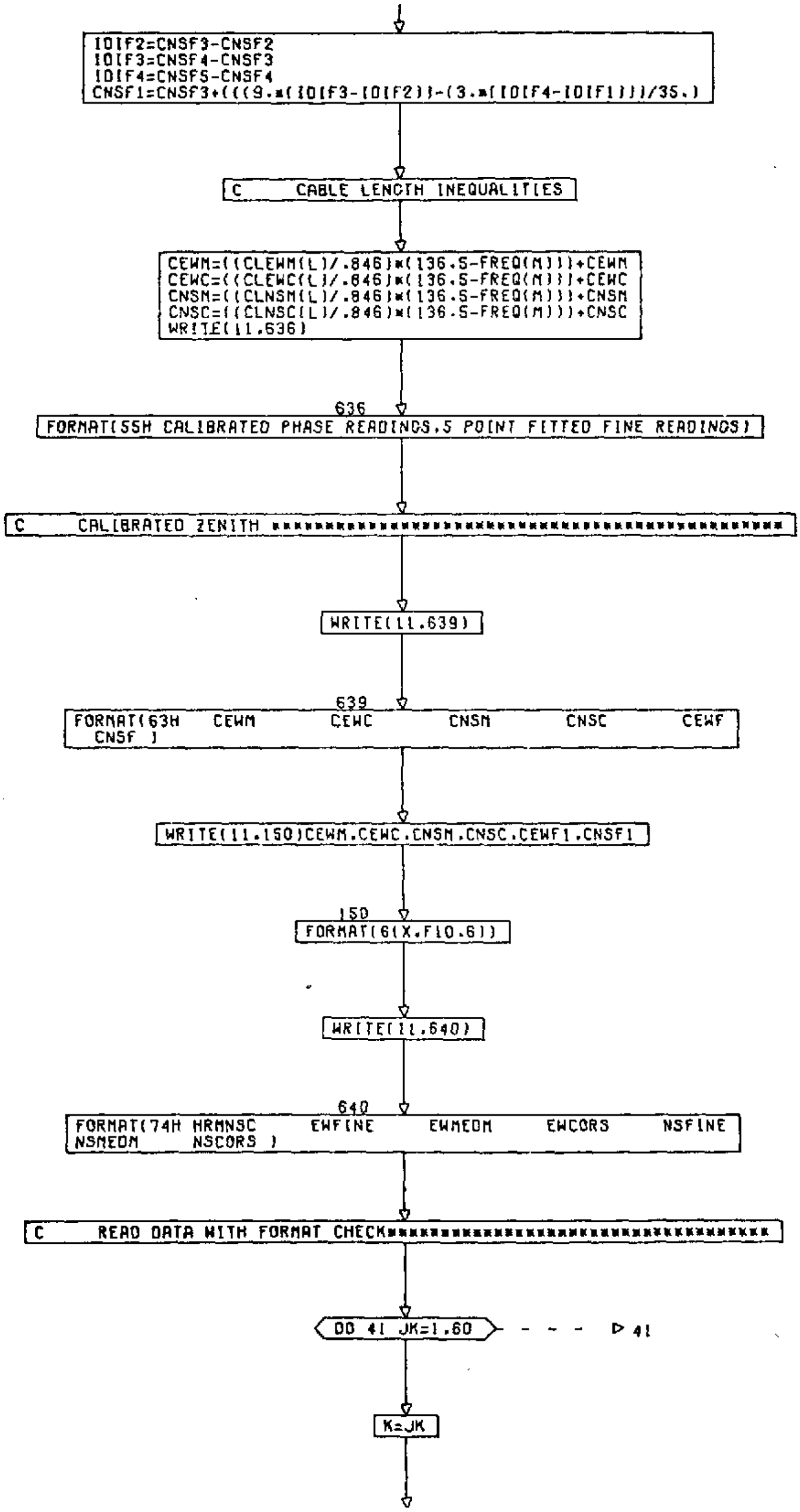

DATA(K)=DATA(K)-BIN
612
CONTINUE
DO 613 K=54,56
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
DATA(K)=DATA(K)-BIN
613
CONTINUE
DO 614 K=58,64
19
IF(DATA(K)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8
IF(DATA(K)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 604
92
C8


DATA(K)=DATA(K)-BIN
18
614
CONTINUE
X=DATA(9)+DATA(22)+DATA(35)+DATA(48)+DATA(61)
IF(X.NE.45.)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 604
92
C8
CSTA=(DATA(55)*10.)+(DATA(56))
DO 616 L=1,12
IF(KSTA(L).EQ.CSTA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 617
616
CONTINUE
WRITE(6,618)
618
FORMAT(27H WRONG STATION IN CAL.LINE)
GO TO 604
92
C8
617
D=DATA(3)*10+DATA(4)

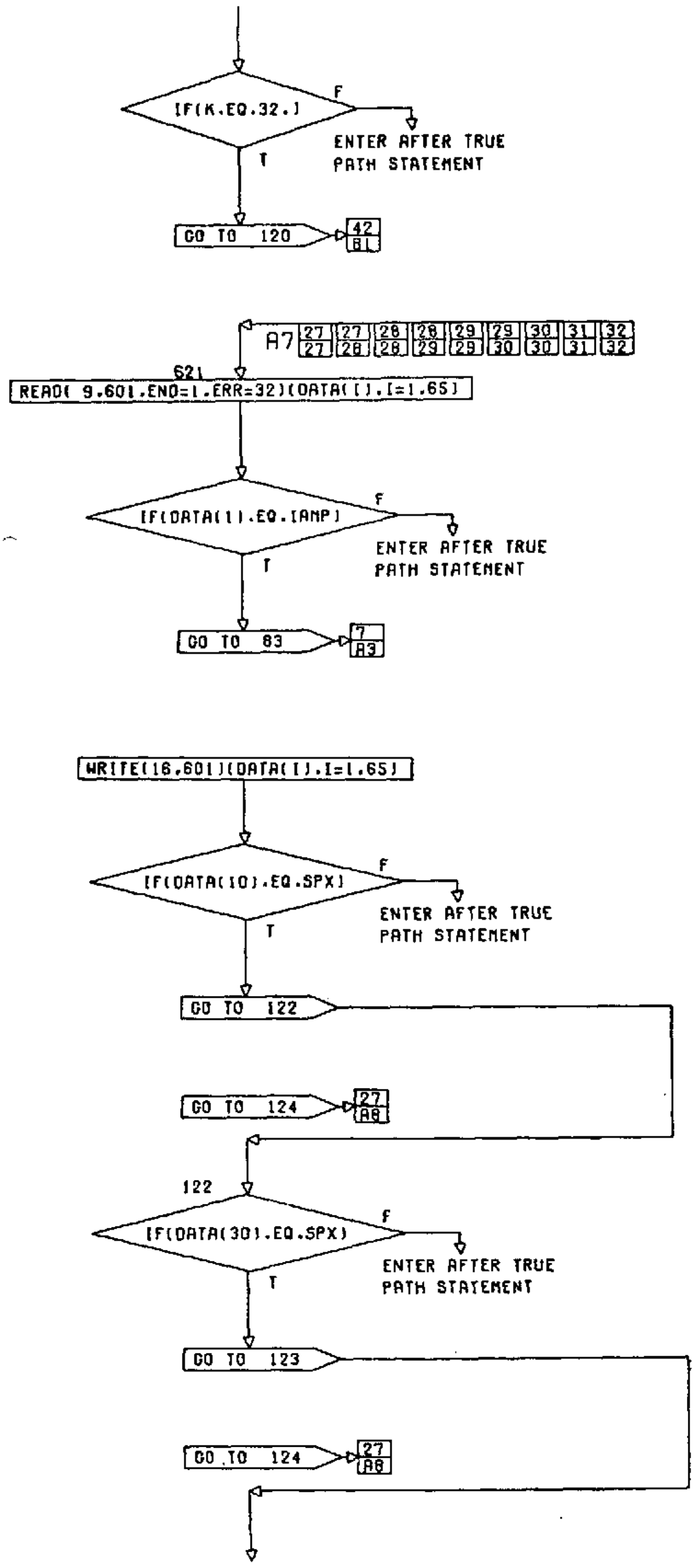


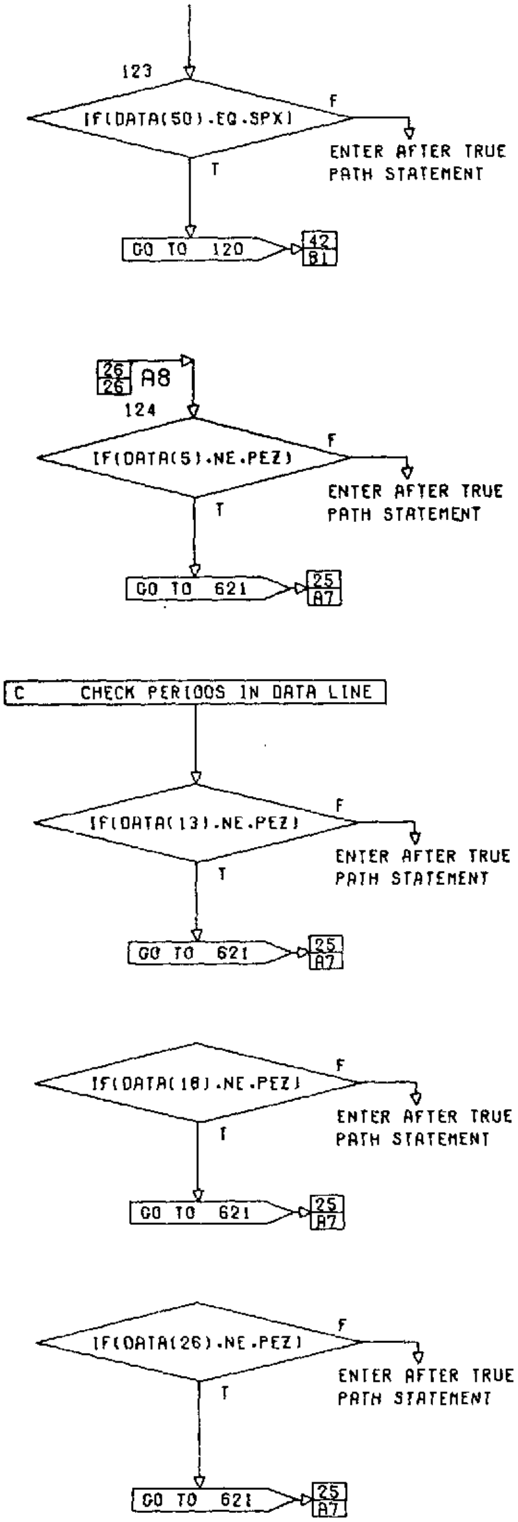
```
CEWM=D/100.
D=DATA(16)*10+DATA(17)
CEWC=D/100.
D=DATA(6)*100+DATA(7)*10+DATA(8)
CEWF1=D/1000.
D=DATA(19)*100+DATA(20)*10+DATA(21)
CEWF2=D/1000.
D=DATA(32)*100+DATA(33)*10+DATA(34)

CEWF3=D/1000.
D=DATA(46)*100+DATA(47)*10+DATA(48)
CEWF4=D/1000.
D=DATA(58)*100+DATA(59)*10+DATA(60)
CEWF5=D/1000.
D=DATA(10)*100+DATA(11)*10+DATA(12)
CNSF1=D/1000.
D=DATA(23)*100+DATA(24)*10+DATA(25)

CNSF2=D/1000.
D=DATA(36)*100+DATA(37)*10+DATA(38)
CNSF3=D/1000.
D=DATA(50)*100+DATA(51)*10+DATA(52)
CNSF4=D/1000.
D=DATA(62)*100+DATA(63)*10+DATA(64)
CNSF5=D/1000.
D=DATA(29)*10+DATA(30)

CNSM=D/100.
D=DATA(43)*10+DATA(44)
CNSC=D/100.
CSTA=DATA(55)*10+DATA(56)
WRITE(11.501)IAMP.SAT.CSTA.STATIO(L)

501
FORMAT(X.A1.I5.X.I2.X.A6)

CANT=DATA(54)
END=DATA(65)

43
CEWM=EWM(L)+CEWM

CEWC=EWC(L)+CEWC
CNSM=NSM(L)+CNSM
CNSC=NSC(L)+CNSC
KM=5.

RATE=0.
IDIF1=CEWF2-CEWF1
IDIF2=CEWF3-CEWF2
IDIF3=CEWF4-CEWF3

IDIF4=CEWF5-CEWF4
CEWF1=CEWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
```

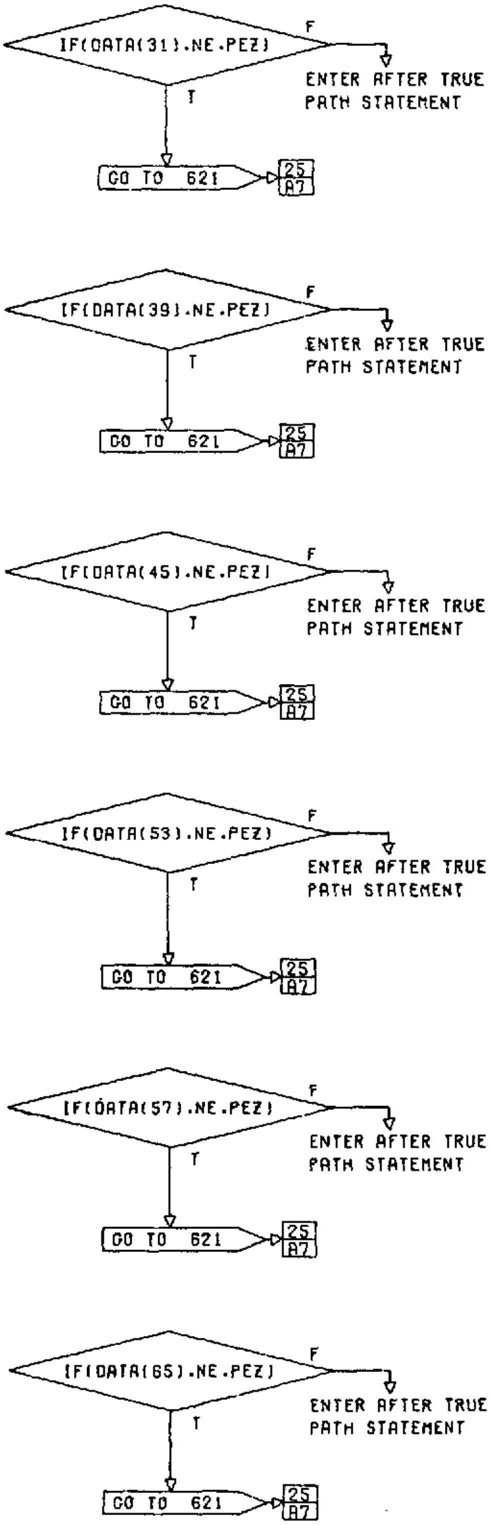

23
A5
-
IF(CANT-2.)162.160.161
+
0
C NARROW BAND TRACKING FILTER **
160
IDIF=.0
AST(1)=SPA
GO TO 164
23
A6
161
IDIF=.0
AST(1)=ASK
GO TO 164
22
A5
162
IF(CANT.EQ.1.)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 164
163
IDIF=.120
C 2 CPS TRACKING FILTER **
AST(1)=PER
22
A6
164
IDIF1=CNSF2-CNSF1

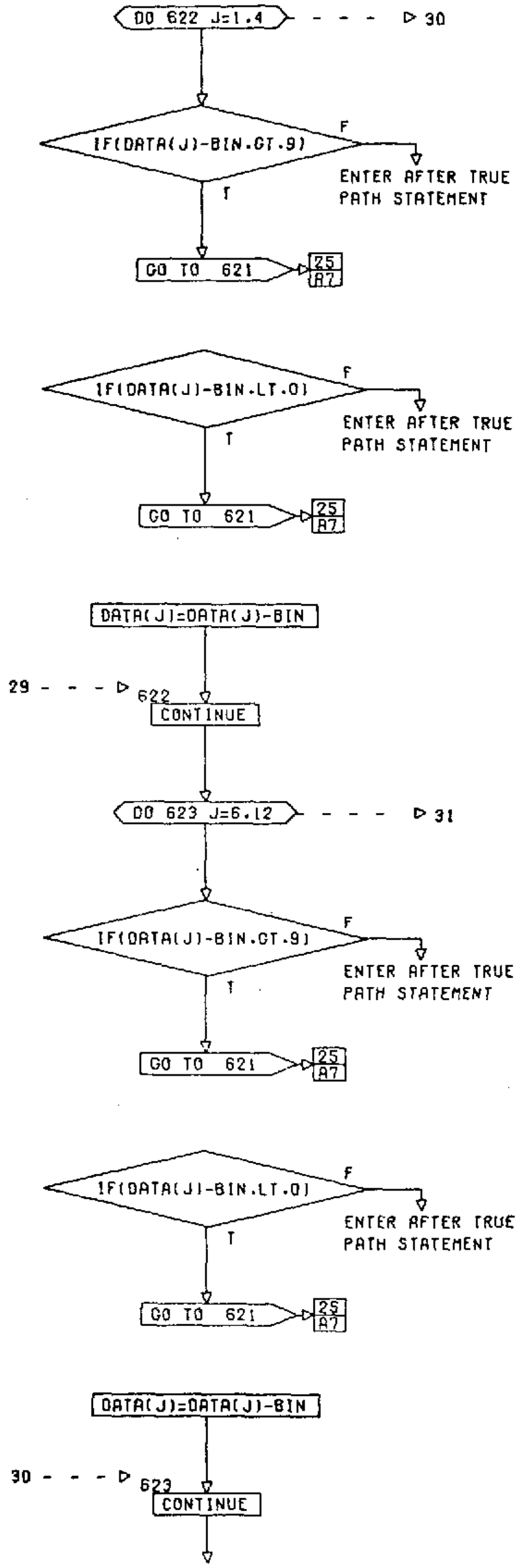


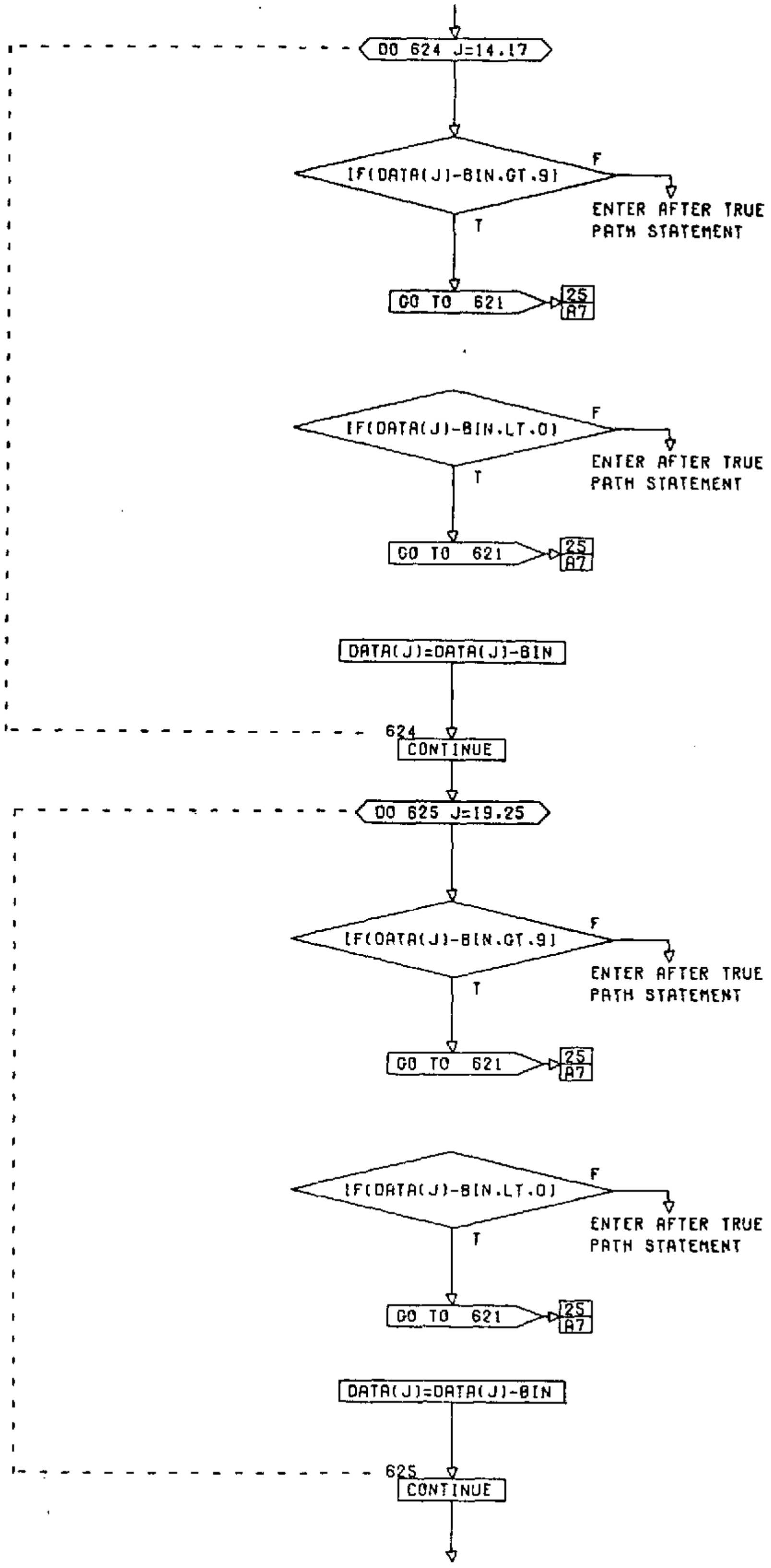
```
IDIF2=CNSF3-CNSF2
IDIF3=CNSF4-CNSF3
IDIF4=CNSF5-CNSF4
CNSF1=CNSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)

C     CABLE LENGTH INEQUALITIES

CEWM=((CLEWM(L)/.846)*(136.5-FREQ(M)))+CEWM
CEWC=((CLEWC(L)/.846)*(136.5-FREQ(M)))+CEWC
CNSM=((CLNSM(L)/.846)*(136.5-FREQ(M)))+CNSM
CNSC=((CLNSC(L)/.846)*(136.5-FREQ(M)))+CNSC
WRITE(11.636)

636
FORMAT(55H CALIBRATED PHASE READINGS,5 POINT FITTED FINE READINGS)

C     CALIBRATED ZENITH **************************************************

WRITE(11.639)

639
FORMAT(63H   CEWM         CEWC         CNSM         CNSC         CEWF
  CNSF )

WRITE(11.150)CEWM.CEWC.CNSM.CNSC.CEWF1.CNSF1

150
FORMAT(6(X.F10.6))

WRITE(11.640)

640
FORMAT(74H HRMNSC      EWFINE       EWMEOM       EWCORS       NSFINE
NSMEOM      NSCORS )

C     READ DATA WITH FORMAT CHECK*************************************

DO 41 JK=1.60  - - - -  ▷ 41

K=JK
```

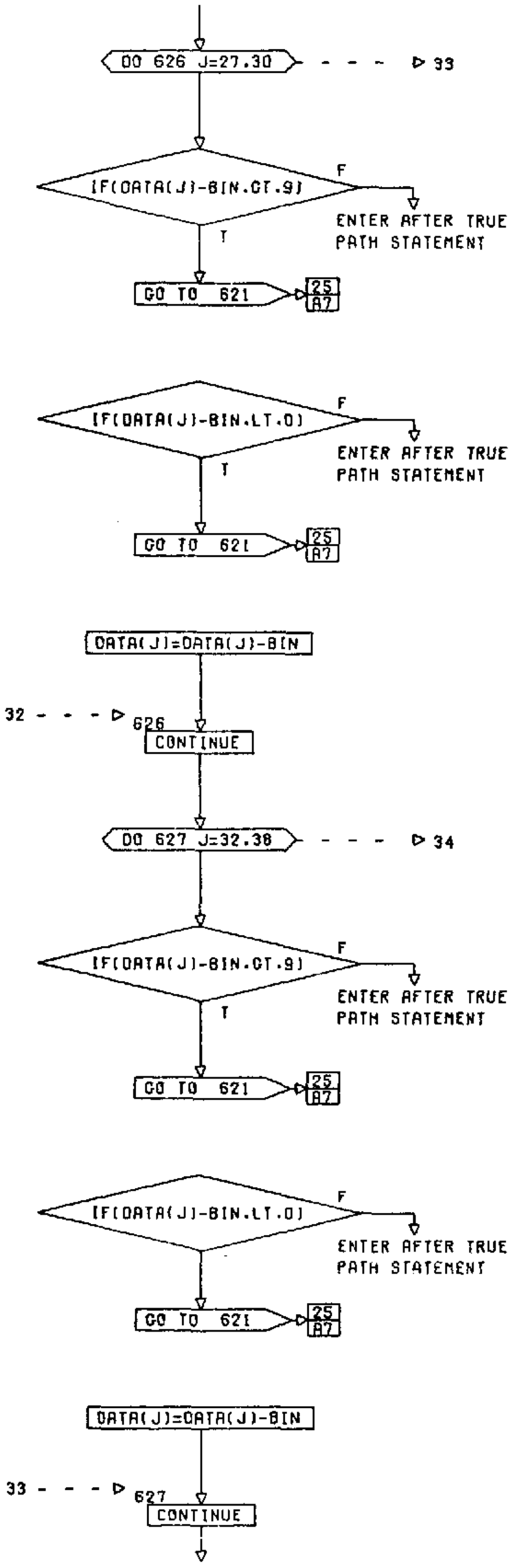

IF(K.EQ.32.)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 120
42
B1
A7
27
27
27
28
28
28
28
29
29
29
29
30
30
30
31
31
32
32
621
READ(9.601.END=1.ERR=32)(DATA(I).I=1.65)
IF(DATA(1).EQ.IAMP)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 83
7
A3
WRITE(16.601)(DATA(I).I=1.65)
IF(DATA(10).EQ.SPX)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 122
GO TO 124
27
A8
122
IF(DATA(30).EQ.SPX)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 123
GO TO 124
27
A8

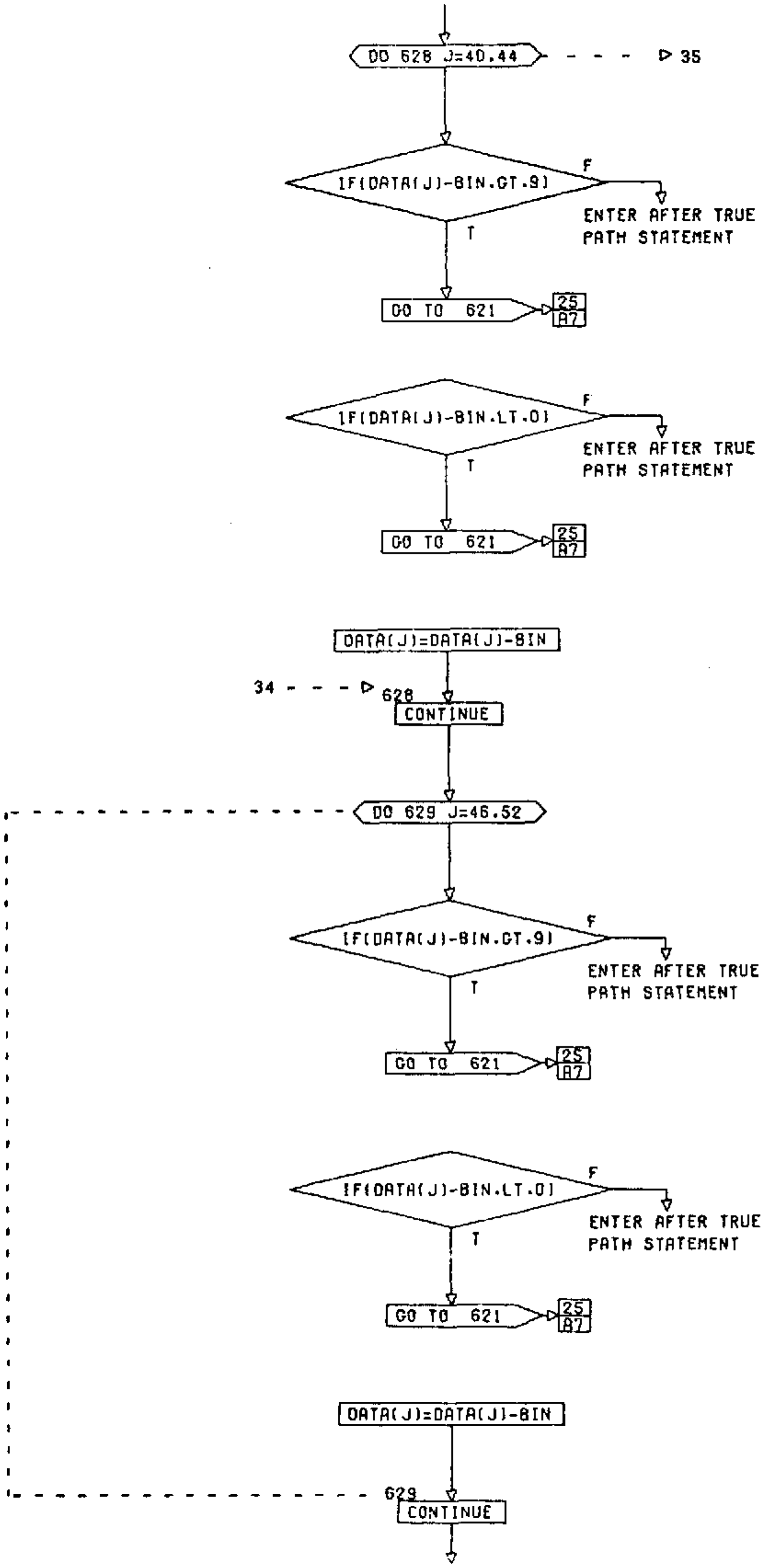

123
IF(DATA(50).EQ.SPX)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 120
42
B1
26
26
A8
124
IF(DATA(5).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
C CHECK PERIODS IN DATA LINE
IF(DATA(13).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(18).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(26).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7

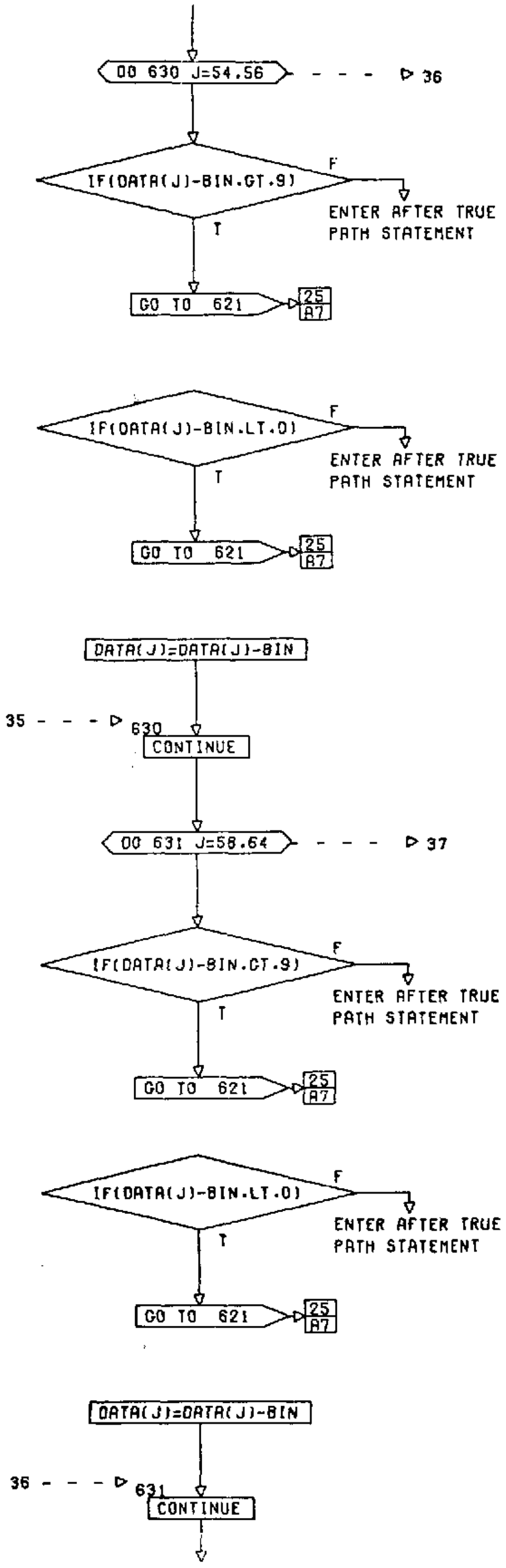

IF(DATA(31).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(39).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(45).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(53).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(57).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(65).NE.PEZ)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7

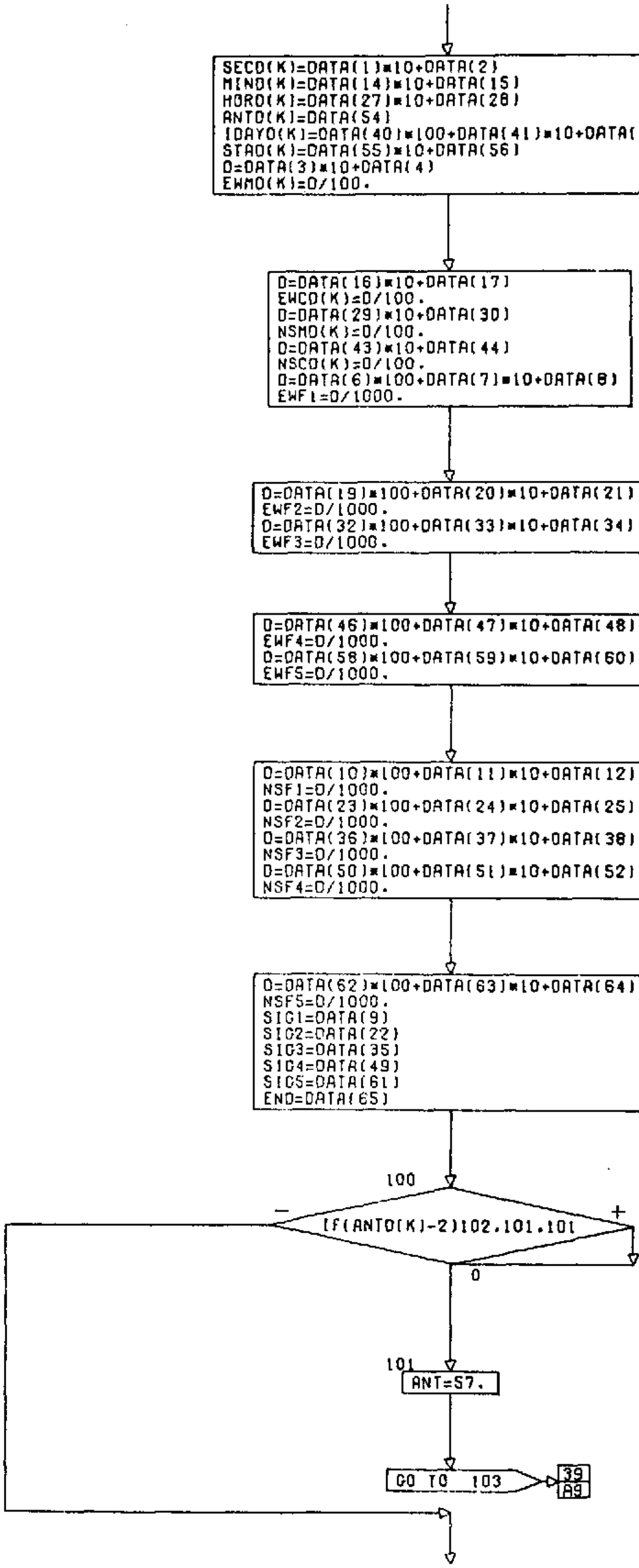

DO 622 J=1,4
▷ 30
IF(DATA(J)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
29 ▷ 622
CONTINUE
DO 623 J=6,12
▷ 31
IF(DATA(J)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
30 ▷ 623
CONTINUE


DO 624 J=14,17
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
624
CONTINUE
DO 625 J=19,25
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
625
CONTINUE

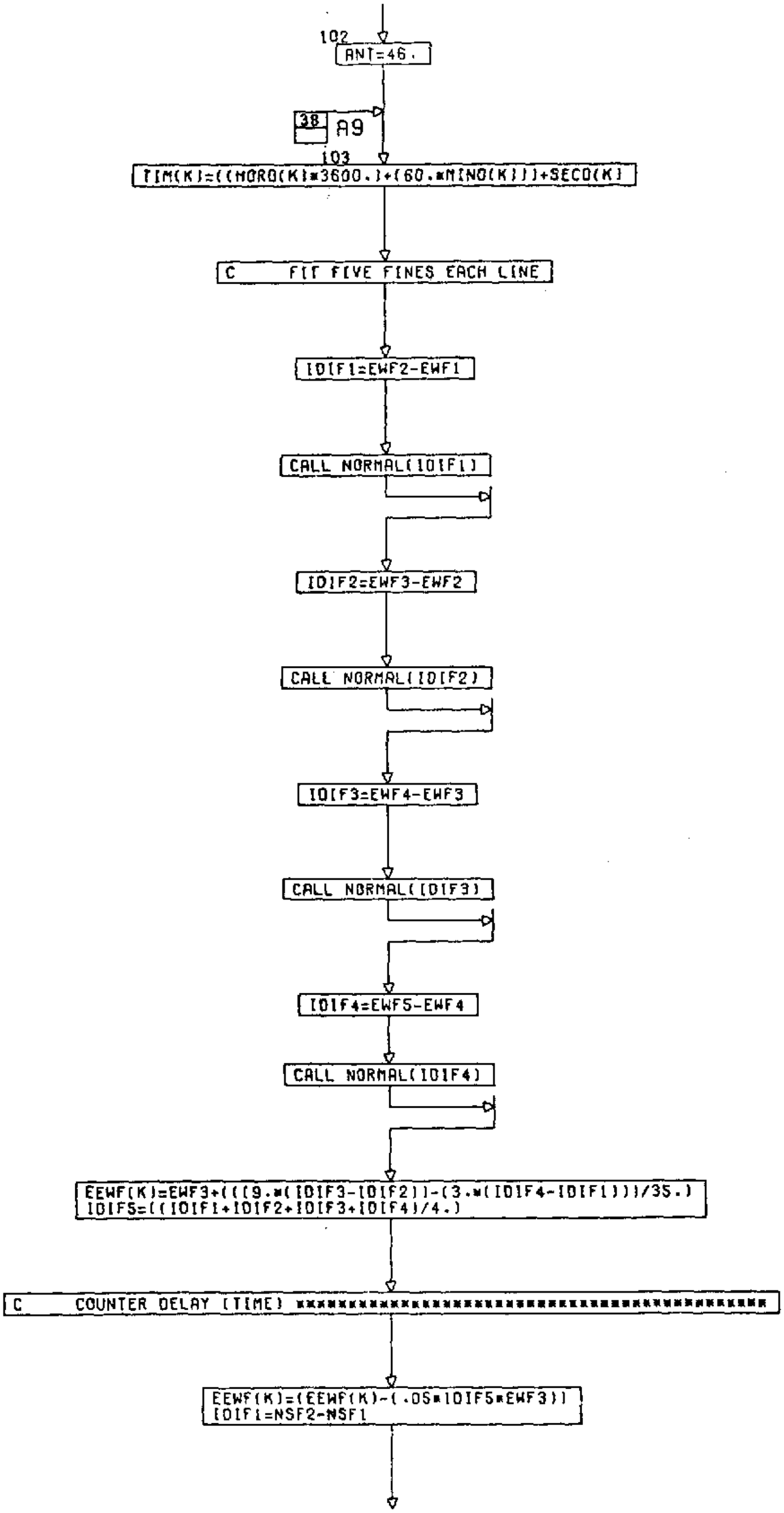

DO 626 J=27,30
33
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
32
626
CONTINUE
DO 627 J=32,38
34
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
33
627
CONTINUE


DO 628 J=40,44
35
IF(DATA(J)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
34
628
CONTINUE
DO 629 J=46,52
IF(DATA(J)-BIN.GT.9)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
629
CONTINUE

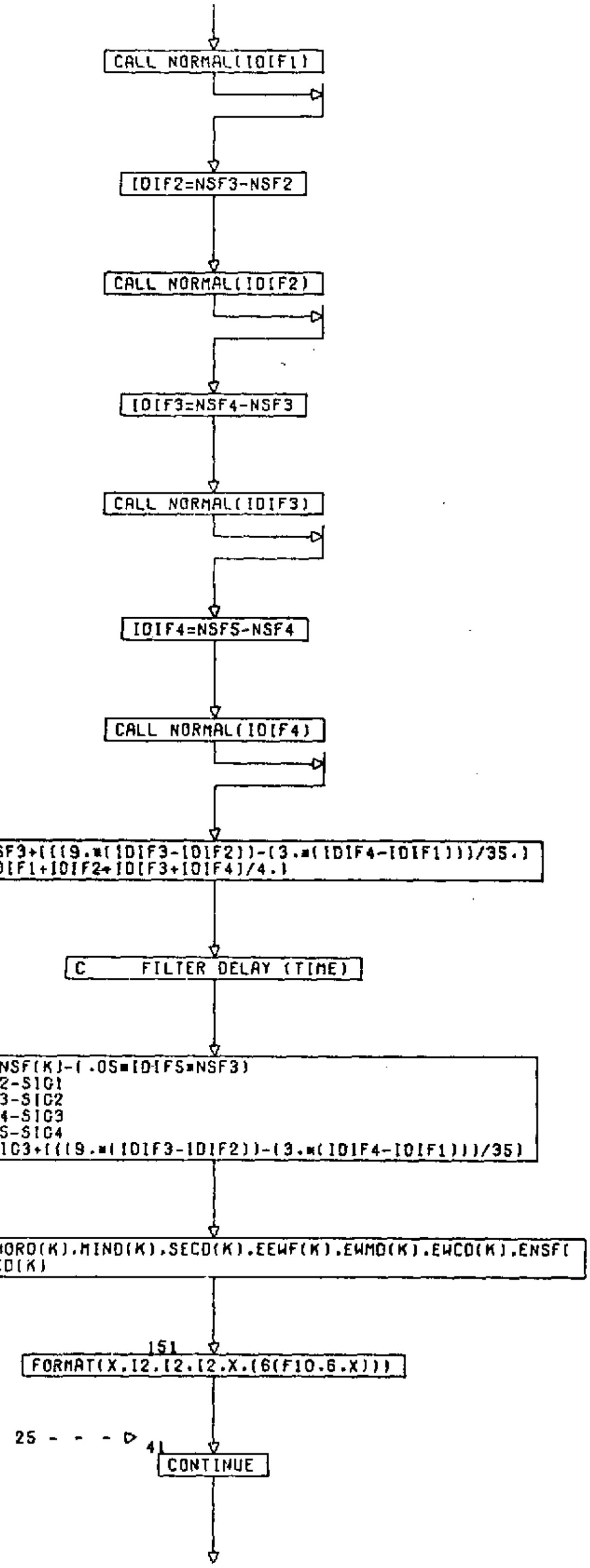

DO 630 J=54,56
36
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
35
630
CONTINUE
DO 631 J=58,64
37
IF(DATA(J)-BIN.GT.9)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
IF(DATA(J)-BIN.LT.0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 621
25
A7
DATA(J)=DATA(J)-BIN
36
631
CONTINUE



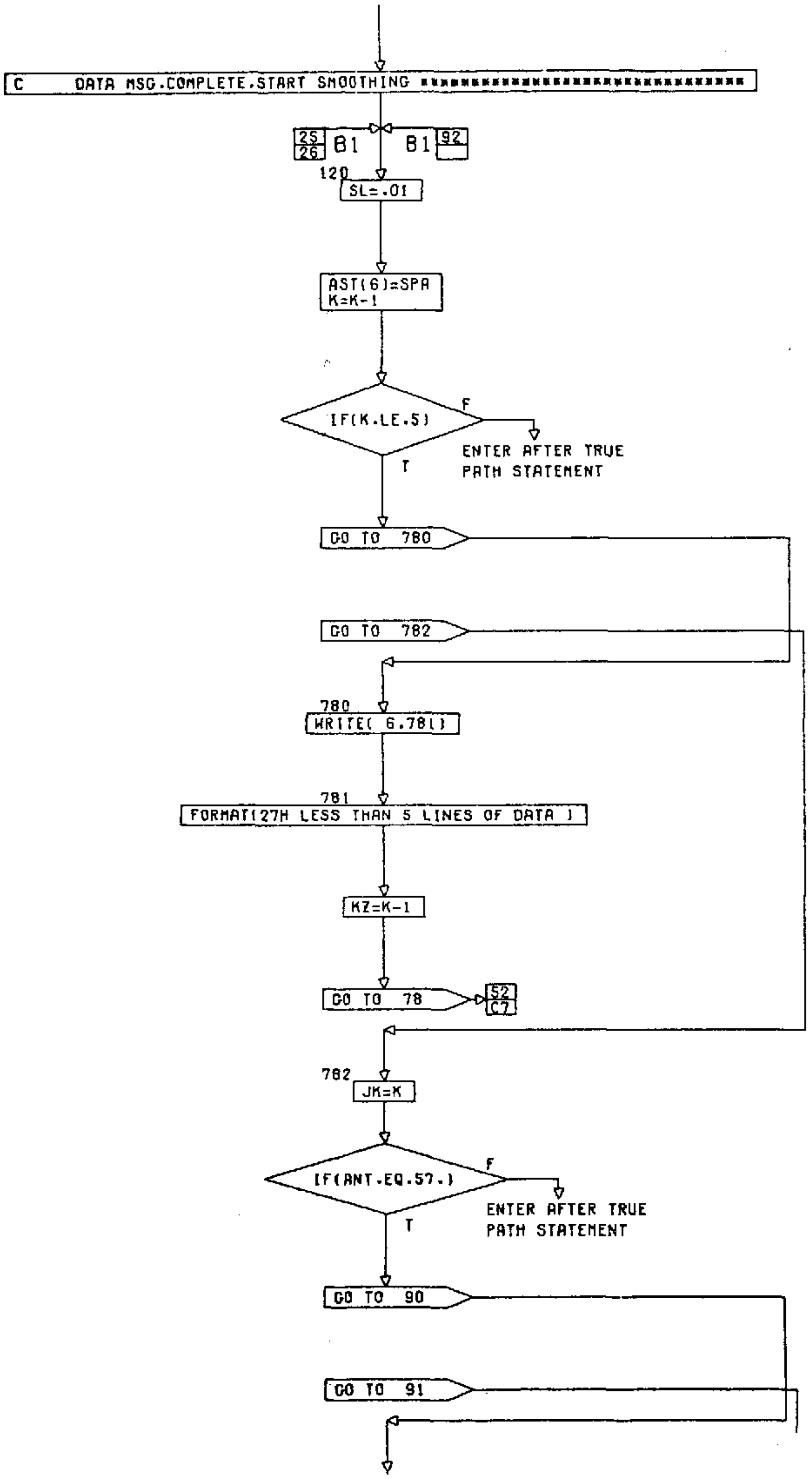
```
SECO(K)=DATA(1)*10+DATA(2)
MIND(K)=DATA(14)*10+DATA(15)
HORO(K)=DATA(27)*10+DATA(28)
ANTO(K)=DATA(54)
IDAYO(K)=DATA(40)*100+DATA(41)*10+DATA(42)
STAO(K)=DATA(55)*10+DATA(56)
D=DATA(3)*10+DATA(4)
EWMO(K)=D/100.

D=DATA(16)*10+DATA(17)
EWCO(K)=D/100.
D=DATA(29)*10+DATA(30)
NSMO(K)=D/100.
D=DATA(43)*10+DATA(44)
NSCO(K)=D/100.
D=DATA(6)*100+DATA(7)*10+DATA(8)
EWF1=D/1000.

D=DATA(19)*100+DATA(20)*10+DATA(21)
EWF2=D/1000.
D=DATA(32)*100+DATA(33)*10+DATA(34)
EWF3=D/1000.

D=DATA(46)*100+DATA(47)*10+DATA(48)
EWF4=D/1000.
D=DATA(58)*100+DATA(59)*10+DATA(60)
EWF5=D/1000.

D=DATA(10)*100+DATA(11)*10+DATA(12)
NSF1=D/1000.
D=DATA(23)*100+DATA(24)*10+DATA(25)
NSF2=D/1000.
D=DATA(36)*100+DATA(37)*10+DATA(38)
NSF3=D/1000.
D=DATA(50)*100+DATA(51)*10+DATA(52)
NSF4=D/1000.

D=DATA(62)*100+DATA(63)*10+DATA(64)
NSF5=D/1000.
SIG1=DATA(9)
SIG2=DATA(22)
SIG3=DATA(35)
SIG4=DATA(49)
SIG5=DATA(61)
END=DATA(65)

100 IF(ANTO(K)-2)102.101.101

101 ANT=57.

GO TO 103
```

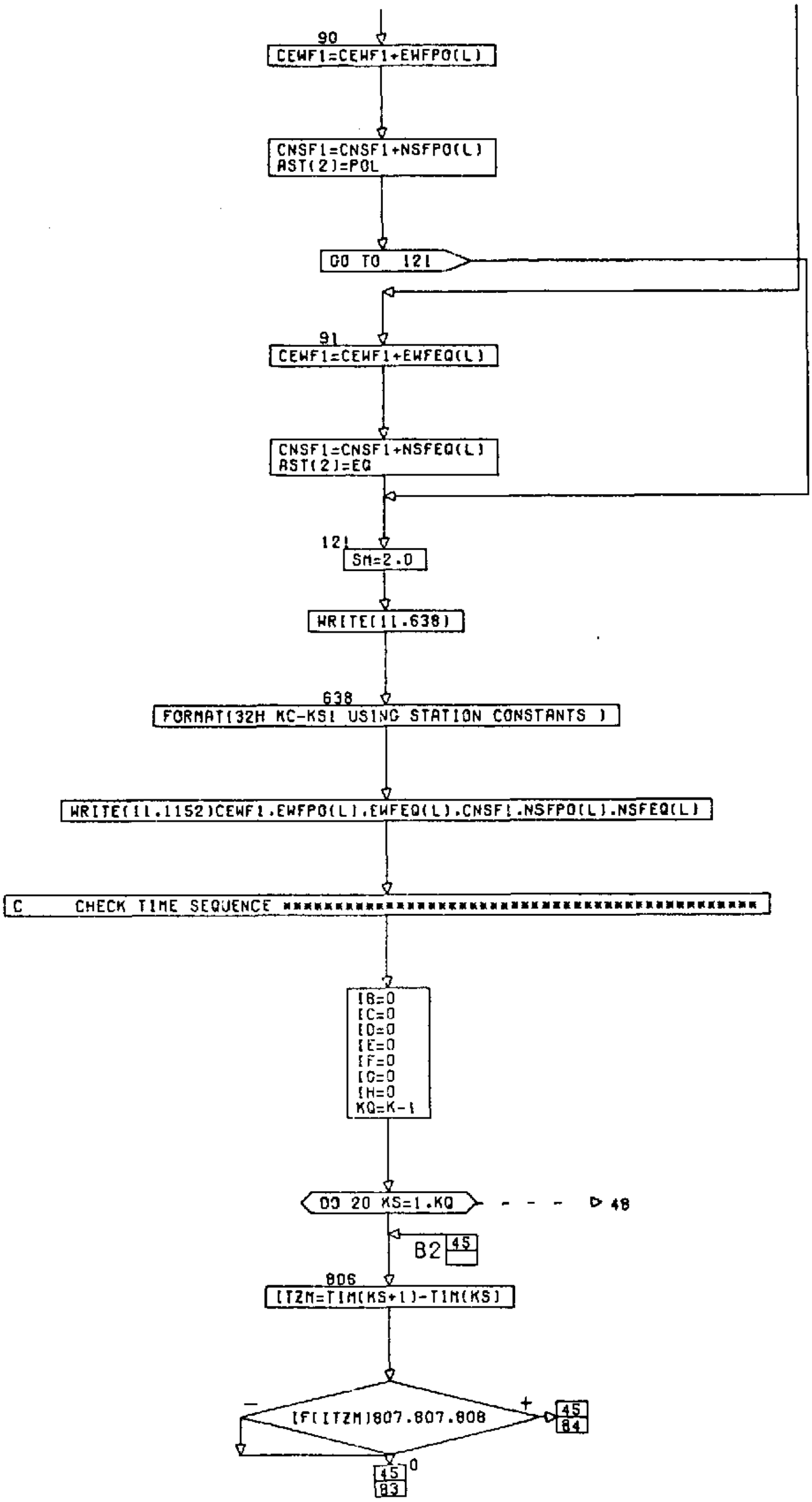


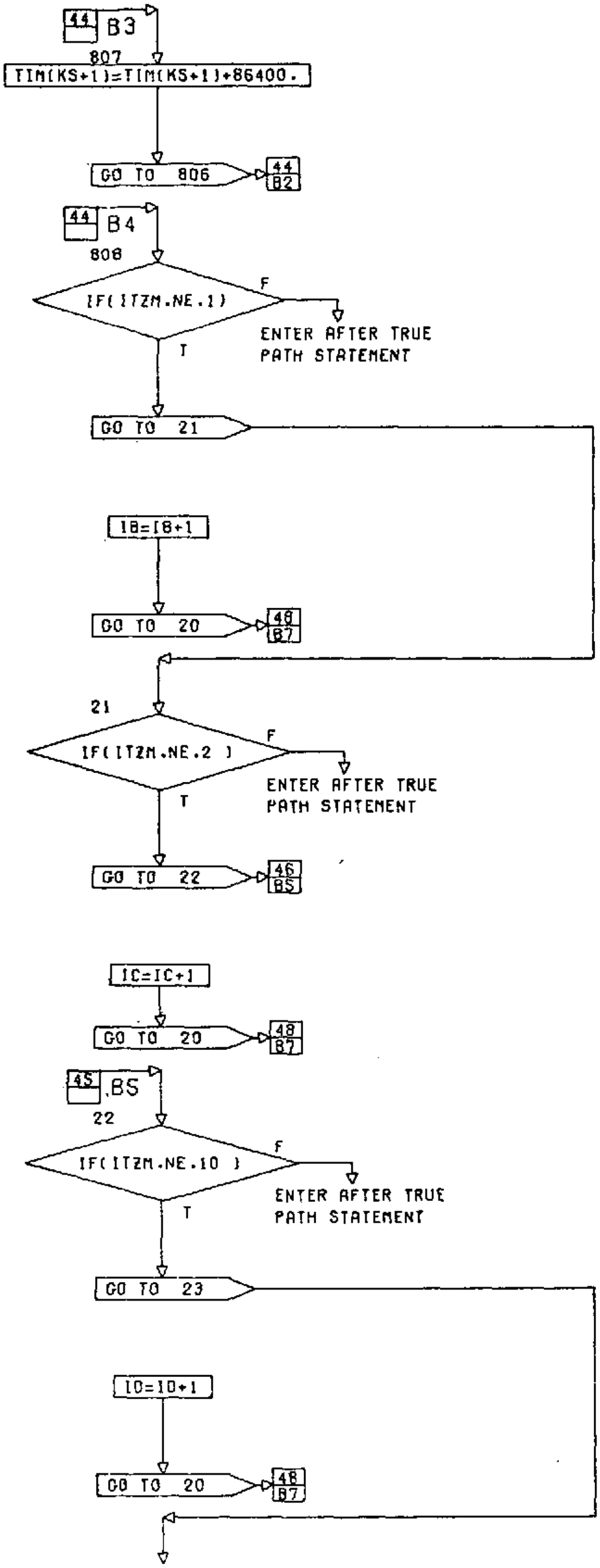
```
102
ANT=46.
103
TIM(K)=((HORO(K)*3600.)+(60.*MINO(K)))+SECO(K)
C     FIT FIVE FINES EACH LINE
IDIF1=EWF2-EWF1
CALL NORMAL(IDIF1)
IDIF2=EWF3-EWF2
CALL NORMAL(IDIF2)
IDIF3=EWF4-EWF3
CALL NORMAL(IDIF3)
IDIF4=EWF5-EWF4
CALL NORMAL(IDIF4)
EEWF(K)=EWF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
IDIF5=((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
C     COUNTER DELAY (TIME) ***********************************************
EEWF(K)=(EEWF(K)-(.05*IDIF5*EWF3))
IDIF1=NSF2-NSF1
```

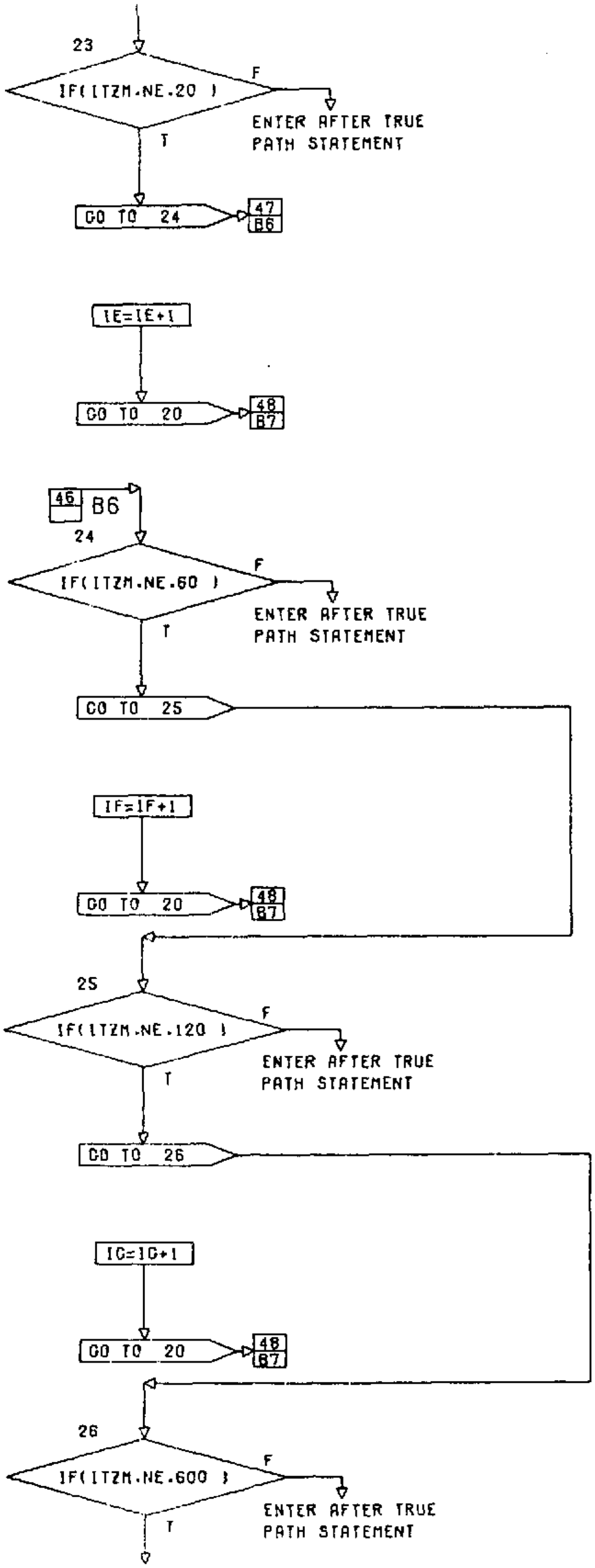


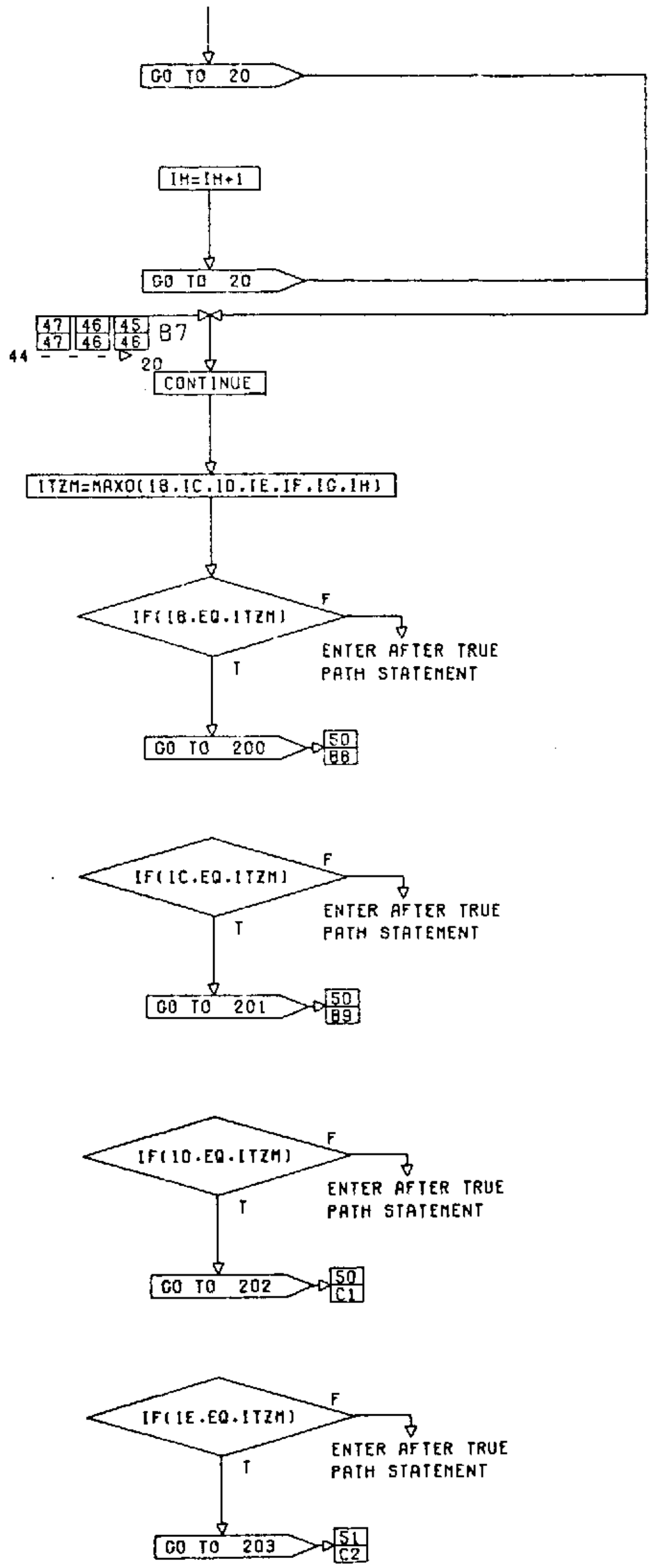
```
CALL NORMAL(IDIF1)
IDIF2=NSF3-NSF2
CALL NORMAL(IDIF2)
IDIF3=NSF4-NSF3
CALL NORMAL(IDIF3)
IDIF4=NSF5-NSF4
CALL NORMAL(IDIF4)
ENSF(K)=NSF3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35.)
IDIF5=(((IDIF1+IDIF2+IDIF3+IDIF4)/4.)
C     FILTER DELAY (TIME)
ENSF(K)=ENSF(K)-(.05*IDIF5*NSF3)
IDIF1=SIG2-SIG1
IDIF2=SIG3-SIG2
IDIF3=SIG4-SIG3
IDIF4=SIG5-SIG4
SIGD(K)=SIG3+(((9.*(IDIF3-IDIF2))-(3.*(IDIF4-IDIF1)))/35)
WRITE(11,151)HORD(K),MIND(K),SECD(K),EEWF(K),EWMD(K),EWCD(K),ENSF(
K),NSMD(K),NSCD(K)
151 FORMAT(X,I2,I2,I2,X,(6(F10.6,X)))
25 - - - ▷ 4 CONTINUE
```

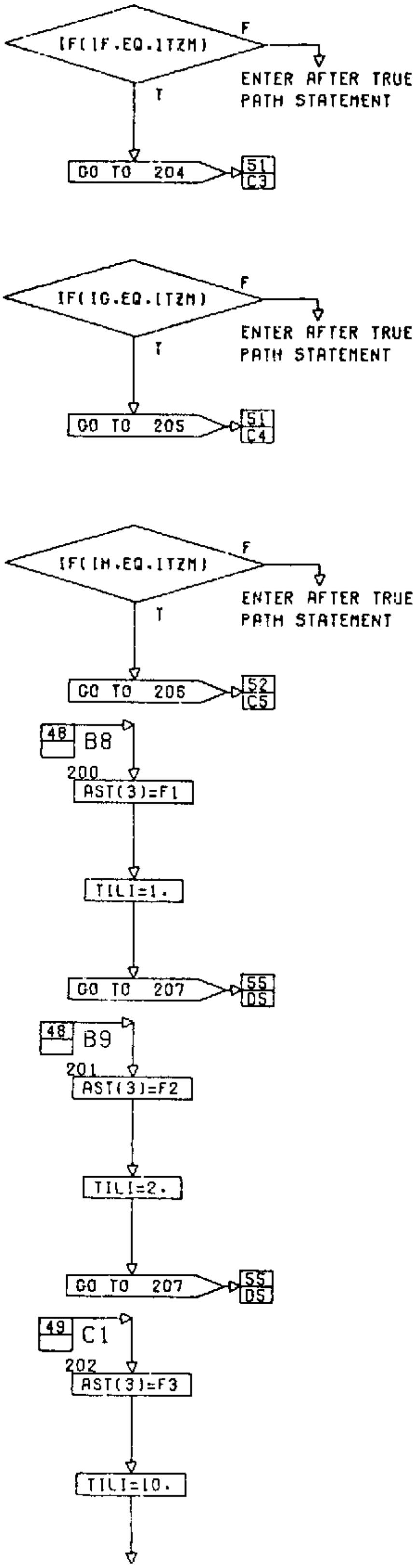

C DATA MSG.COMPLETE.START SMOOTHING ********************************
25
26
B1
B1
92
120
SL=.01
AST(6)=SPA
K=K-1
IF(K.LE.5)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 780
GO TO 782
780
WRITE(6,781)
781
FORMAT(27H LESS THAN 5 LINES OF DATA)
KZ=K-1
GO TO 78
S2
C7
782
JK=K
IF(ANT.EQ.57.)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 90
GO TO 91

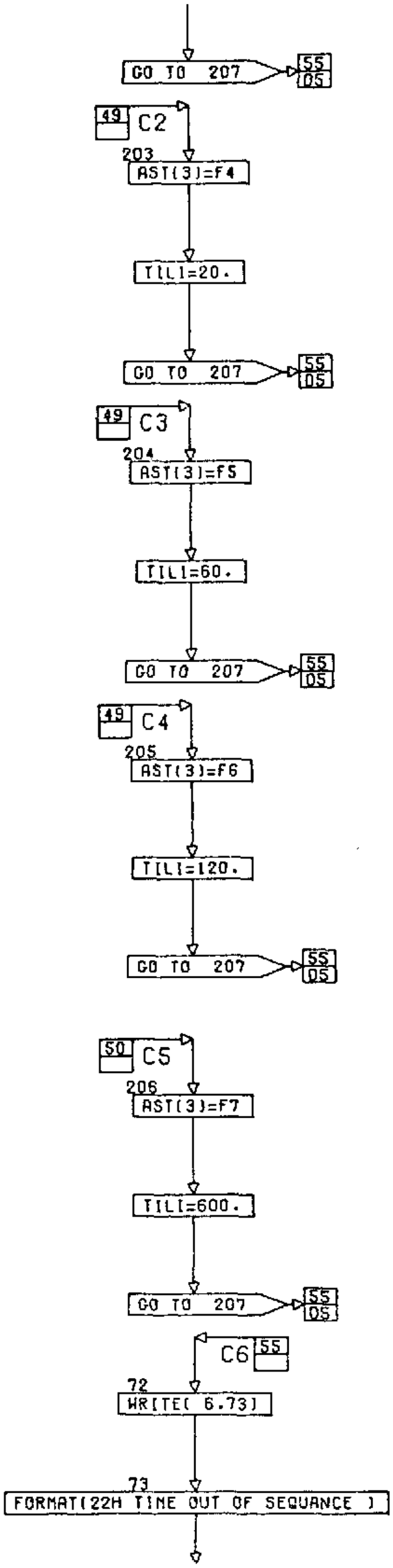

90
CEWF1=CEWF1+EWFPO(L)
CNSF1=CNSF1+NSFPO(L)
AST(2)=POL
GO TO 121
91
CEWF1=CEWF1+EWFEQ(L)
CNSF1=CNSF1+NSFEQ(L)
AST(2)=EQ
121
SM=2.0
WRITE(11.638)
638
FORMAT(32H KC-KS1 USING STATION CONSTANTS)
WRITE(11.1152)CEWF1.EWFPO(L).EWFEQ(L).CNSF1.NSFPO(L).NSFEQ(L)
C CHECK TIME SEQUENCE ***
IB=0
IC=0
ID=0
IE=0
IF=0
IG=0
IH=0
KQ=K-1
DO 20 KS=1.KQ
48
B2 45
806
ITZM=TIM(KS+1)-TIM(KS)
IF(ITZM)807.807.808
−
+
45
84
0
45
83

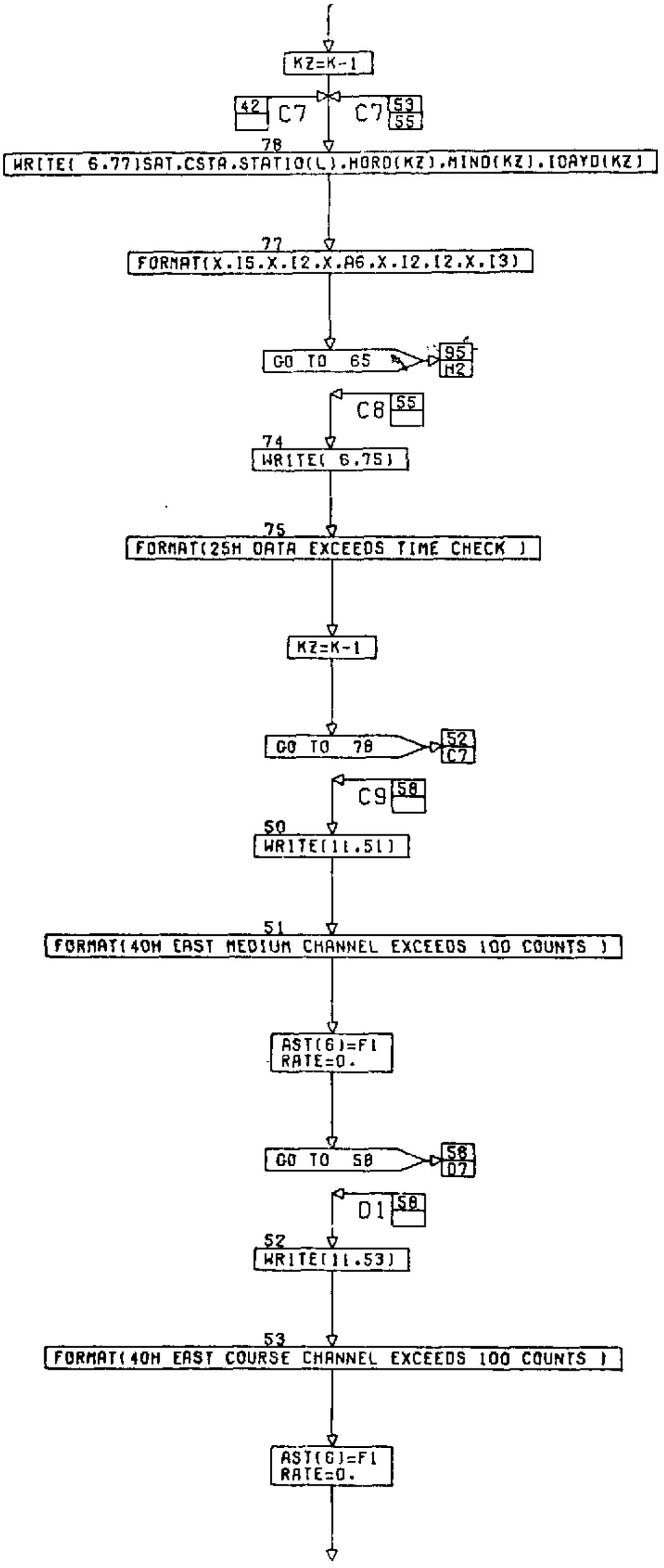

44
B3
807
TIM(KS+1)=TIM(KS+1)+86400.
GO TO 806
44
B2
44
B4
808
IF(ITZM.NE.1)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 21
IB=IB+1
GO TO 20
48
B7
21
IF(ITZM.NE.2)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 22
46
B5
IC=IC+1
GO TO 20
48
B7
45
B5
22
IF(ITZM.NE.10)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 23
ID=ID+1
GO TO 20
48
B7

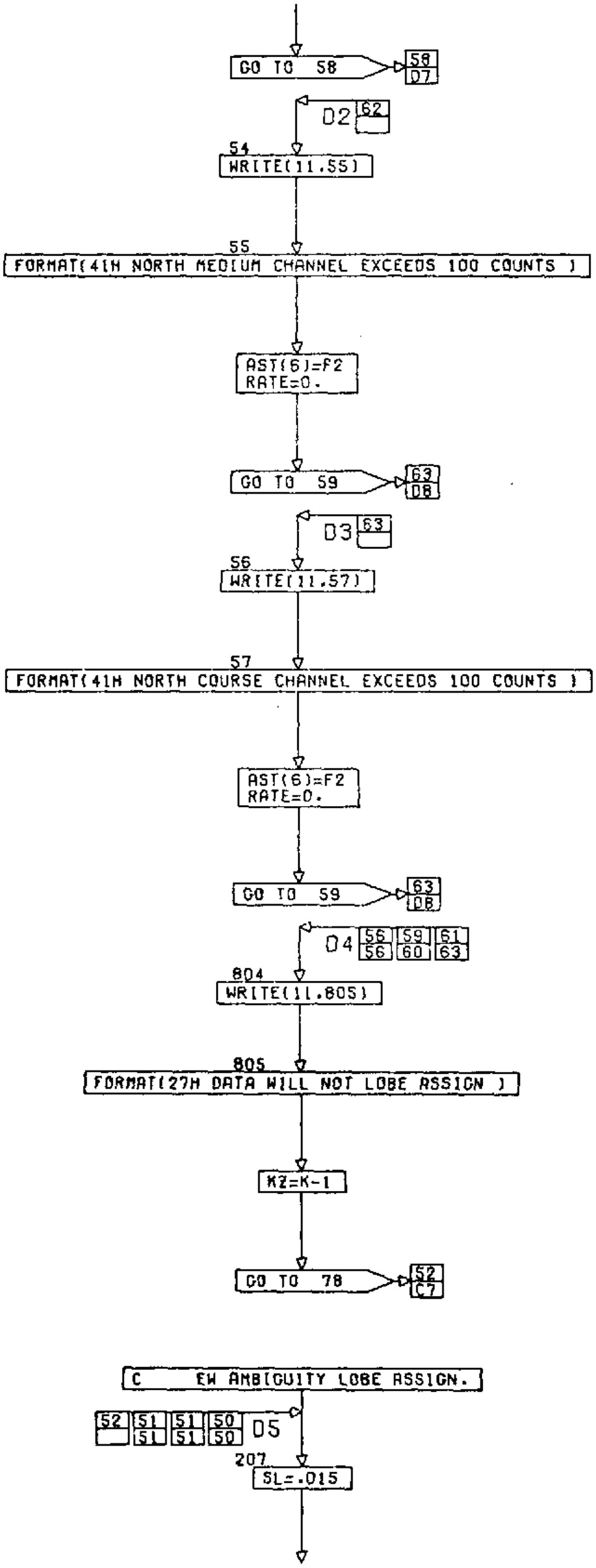

23
IF(ITZM.NE.20)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 24
47
B6
IE=IE+1
GO TO 20
48
B7
46
B6
24
IF(ITZM.NE.60)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 25
IF=IF+1
GO TO 20
48
B7
25
IF(ITZM.NE.120)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 26
IG=IG+1
GO TO 20
48
B7
26
IF(ITZM.NE.600)
F
ENTER AFTER TRUE
PATH STATEMENT
T

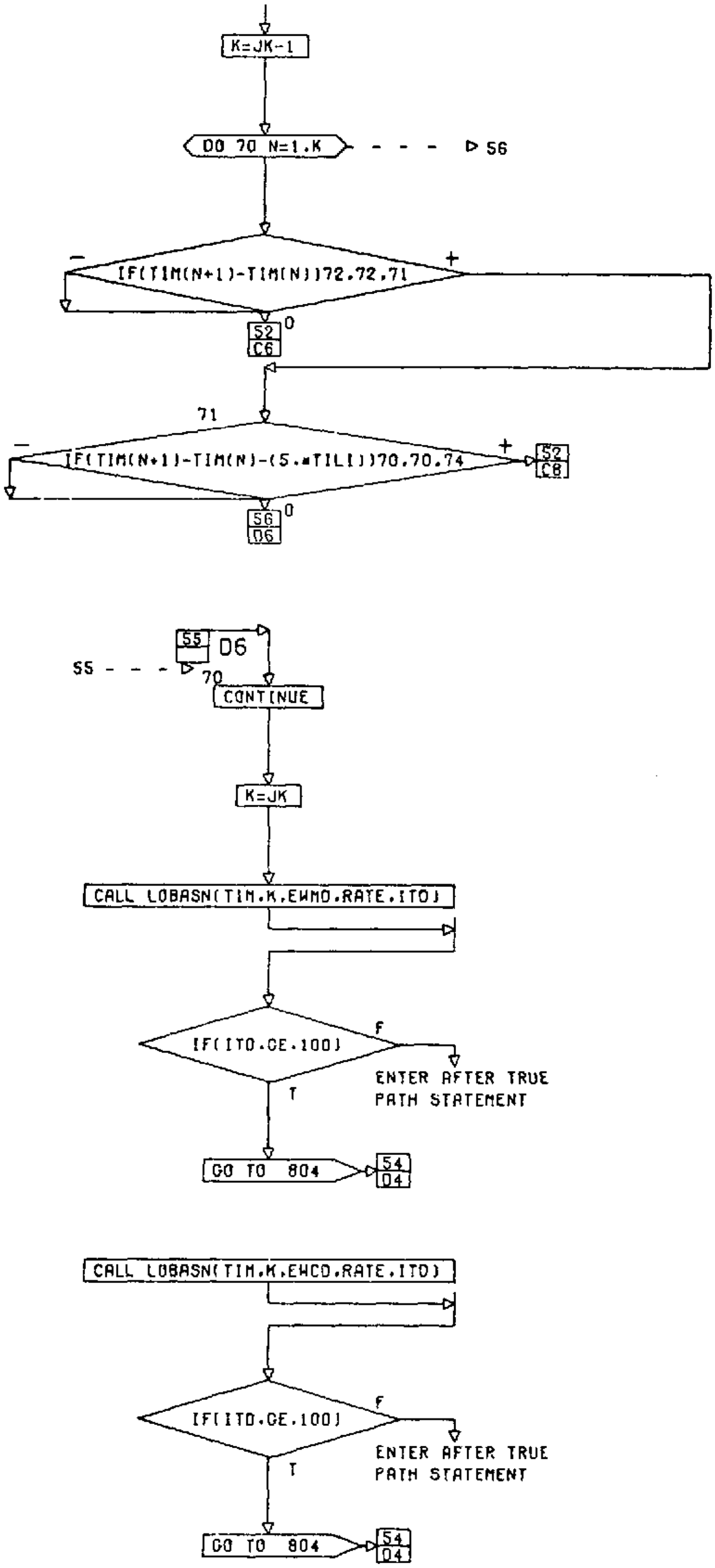

GO TO 20
IH=IH+1
GO TO 20
47
47
46
46
45
46
B7
44
20
CONTINUE
ITZM=MAXO(IB.IC.ID.IE.IF.IG.IH)
IF(IB.EQ.ITZM)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 200
50
B8
IF(IC.EQ.ITZM)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 201
50
B9
IF(ID.EQ.ITZM)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 202
50
C1
IF(IE.EQ.ITZM)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 203
51
C2

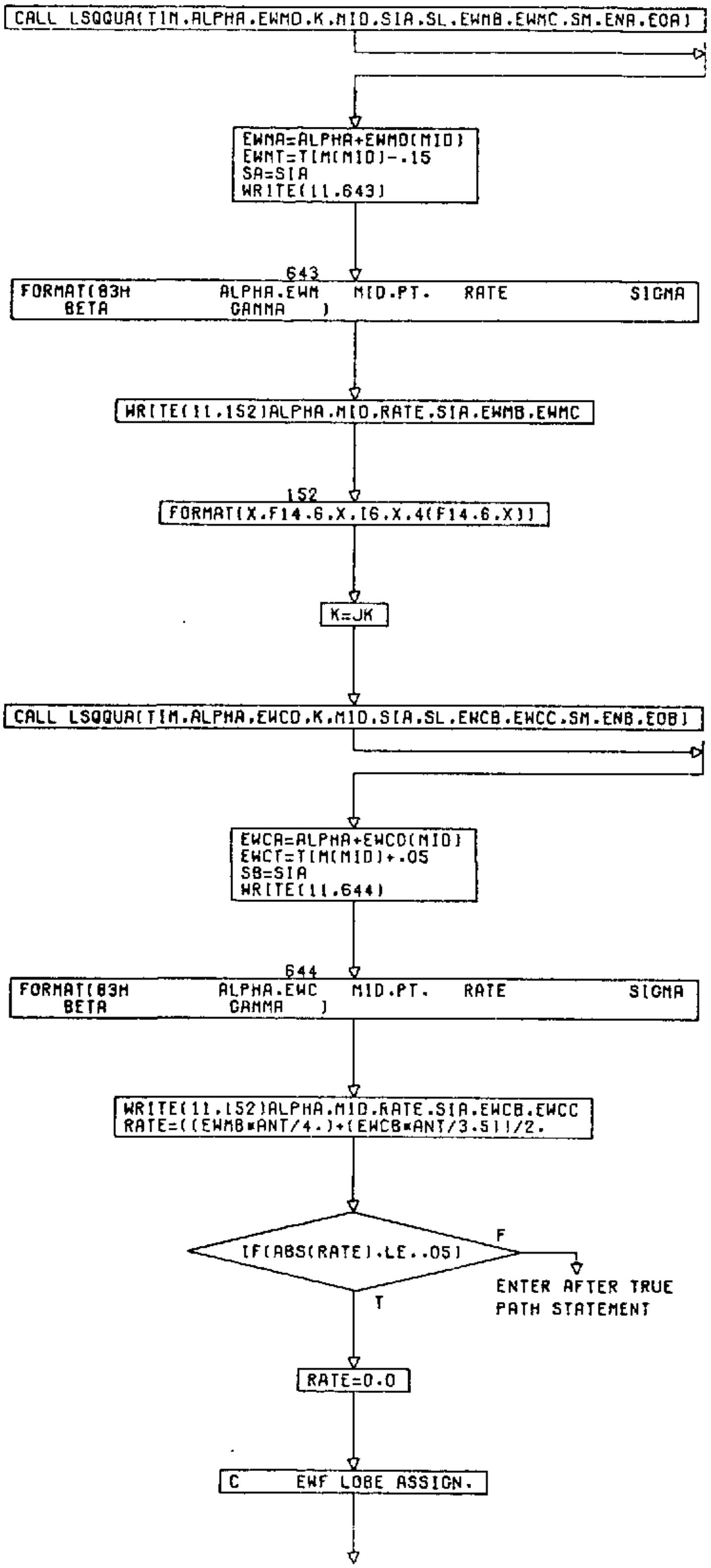

IF(IF.EQ.ITZM)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 204
51
C3
IF(IG.EQ.ITZM)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 205
51
C4
IF(IH.EQ.ITZM)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 206
52
C5
48
B8
200
AST(3)=F1
TILI=1.
GO TO 207
55
D5
48
B9
201
AST(3)=F2
TILI=2.
GO TO 207
55
D5
49
C1
202
AST(3)=F3
TILI=10.

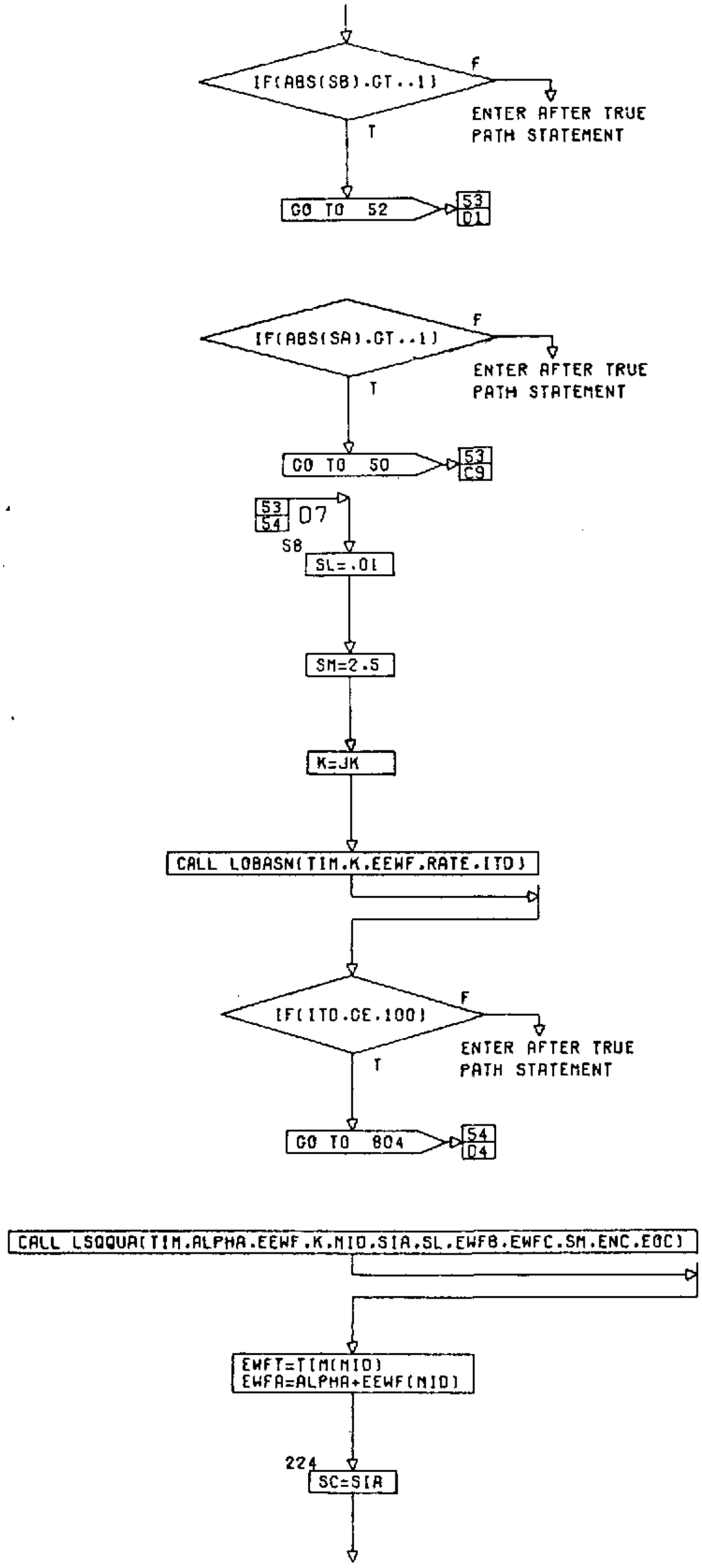

GO TO 207
55
05
49
C2
203
AST(3)=F4
T(L)=20.
GO TO 207
55
05
49
C3
204
AST(3)=F5
T(L)=60.
GO TO 207
55
05
49
C4
205
AST(3)=F6
T(L)=120.
GO TO 207
55
05
50
C5
206
AST(3)=F7
T(L)=600.
GO TO 207
55
05
C6
55
72
WRITE(6.73)
73
FORMAT(22H TIME OUT OF SEQUANCE)

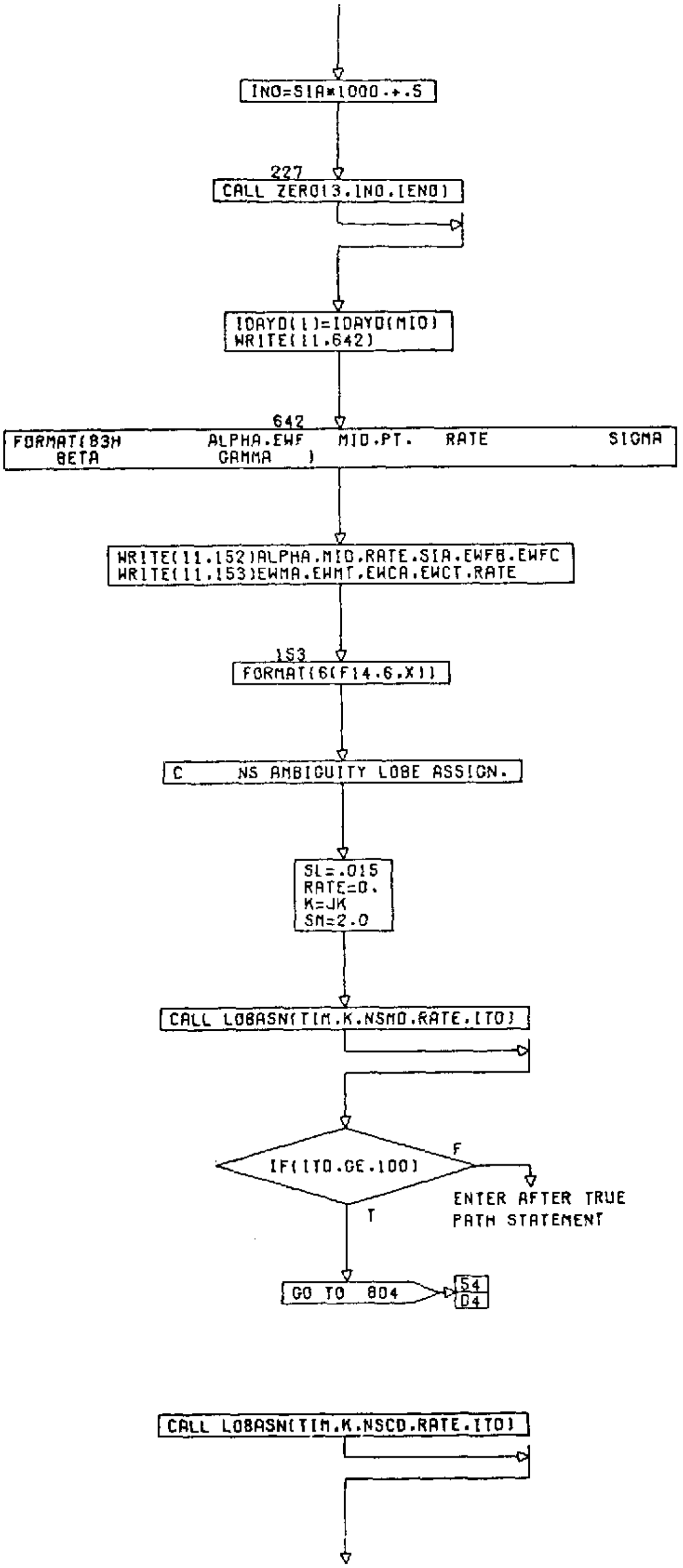

KZ=K-1
42
C7
C7
53
55
78
WRITE(6.77)SAT.CSTA.STATIO(L).HORD(KZ).MIND(KZ).IDAYD(KZ)
77
FORMAT(X.I5.X.I2.X.A6.X.I2.I2.X.I3)
GO TO 65
95
H2
C8
55
74
WRITE(6.75)
75
FORMAT(25H DATA EXCEEDS TIME CHECK)
KZ=K-1
GO TO 78
52
C7
C9
58
50
WRITE(11.51)
51
FORMAT(40H EAST MEDIUM CHANNEL EXCEEDS 100 COUNTS)
AST(6)=F1
RATE=0.
GO TO 58
58
D7
D1
58
52
WRITE(11.53)
53
FORMAT(40H EAST COURSE CHANNEL EXCEEDS 100 COUNTS)
AST(6)=F1
RATE=0.


GO TO 58
58
D7
D2
62
54
WRITE(11,55)
55
FORMAT(41H NORTH MEDIUM CHANNEL EXCEEDS 100 COUNTS)
AST(6)=F2
RATE=0.
GO TO 59
63
D8
D3
63
56
WRITE(11,57)
57
FORMAT(41H NORTH COURSE CHANNEL EXCEEDS 100 COUNTS)
AST(6)=F2
RATE=0.
GO TO 59
63
D8
D4
56
56
59
60
61
63
804
WRITE(11,805)
805
FORMAT(27H DATA WILL NOT LOBE ASSIGN)
KZ=K-1
GO TO 78
52
C7
C EW AMBIGUITY LOBE ASSIGN.
52
51
51
51
51
50
50
D5
207
SL=.015

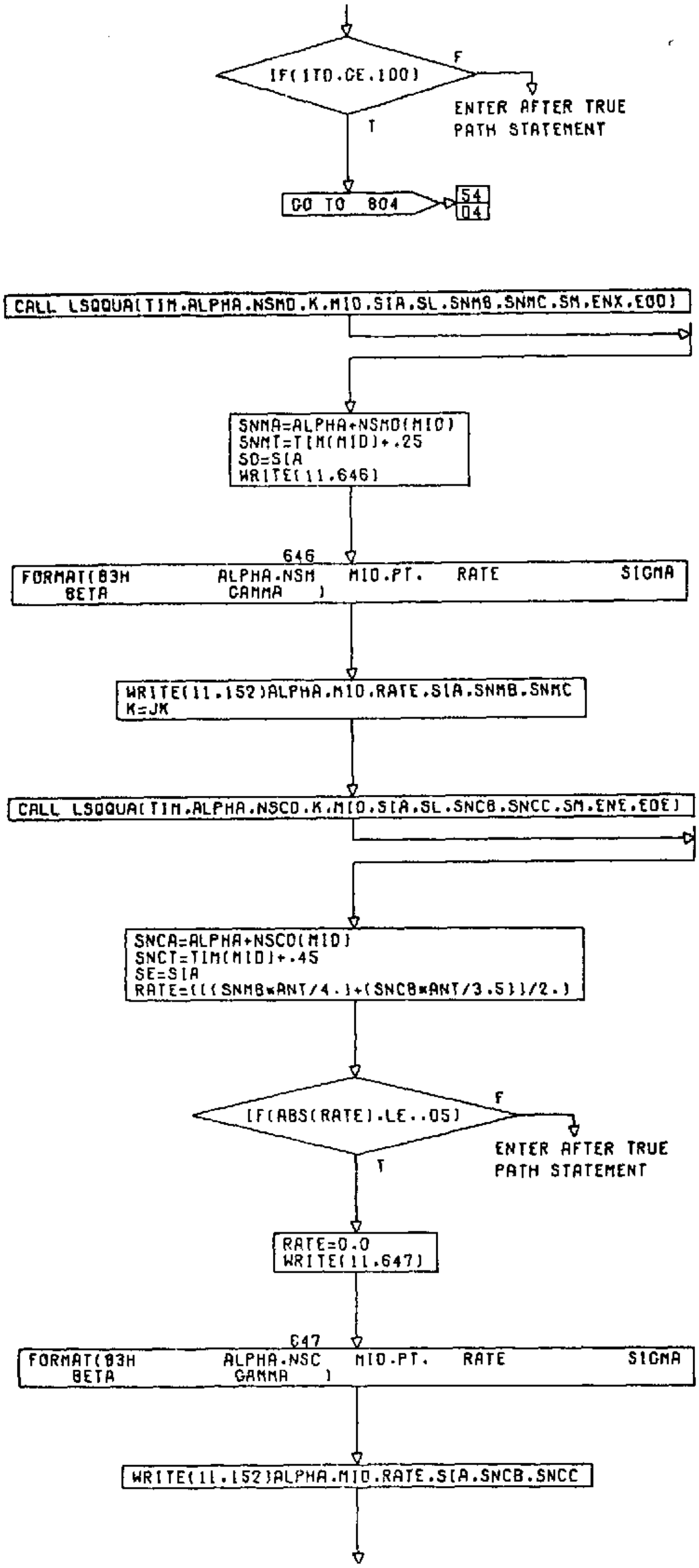

K=JK-1
DO 70 N=1,K
S6
IF(TIM(N+1)-TIM(N))72,72,71
−
+
0
52
C6
71
IF(TIM(N+1)-TIM(N)-(5.*TIL(I))70,70,74
−
+
0
52
C8
56
D6
55
D6
55
70
CONTINUE
K=JK
CALL LOBASN(TIM,K,EWMO,RATE,ITO)
IF(ITO.GE.100)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 804
54
D4
CALL LOBASN(TIM,K,EWCO,RATE,ITO)
IF(ITO.GE.100)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 804
54
D4


CALL LSQQUA(TIM.ALPHA.EWMO.K.MID.SIA.SL.EWMB.EWMC.SM.ENA.EOA)
EWMA=ALPHA+EWMO(MID)
EWMT=TIM(MID)-.15
SA=SIA
WRITE(11.643)
643
FORMAT(83H ALPHA.EWM MID.PT. RATE SIGMA
BETA GAMMA)
WRITE(11.152)ALPHA.MID.RATE.SIA.EWMB.EWMC
152
FORMAT(X.F14.6.X.I6.X.4(F14.6.X))
K=JK
CALL LSQQUA(TIM.ALPHA.EWCO.K.MID.SIA.SL.EWCB.EWCC.SM.ENB.EOB)
EWCA=ALPHA+EWCO(MID)
EWCT=TIM(MID)+.05
SB=SIA
WRITE(11.644)
644
FORMAT(83H ALPHA.EWC MID.PT. RATE SIGMA
BETA GAMMA)
WRITE(11.152)ALPHA.MID.RATE.SIA.EWCB.EWCC
RATE=((EWMB*ANT/4.)+(EWCB*ANT/3.5))/2.
IF(ABS(RATE).LE..05)
F
ENTER AFTER TRUE
PATH STATEMENT
T
RATE=0.0
C EWF LOBE ASSIGN.


IF(ABS(SB).GT..1)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 52
53
D1
IF(ABS(SA).GT..1)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 50
53
C9
53
54
D7
58
SL=.01
SM=2.5
K=JK
CALL LOBASN(TIM.K.EEWF.RATE.ITD)
IF(ITD.GE.100)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 804
54
D4
CALL LSQQUA(TIM.ALPHA.EEWF.K.MID.SIA.SL.EWFB.EWFC.SM.ENC.EOC)
EWFT=TIM(MID)
EWFA=ALPHA+EEWF(MID)
224
SC=SIA



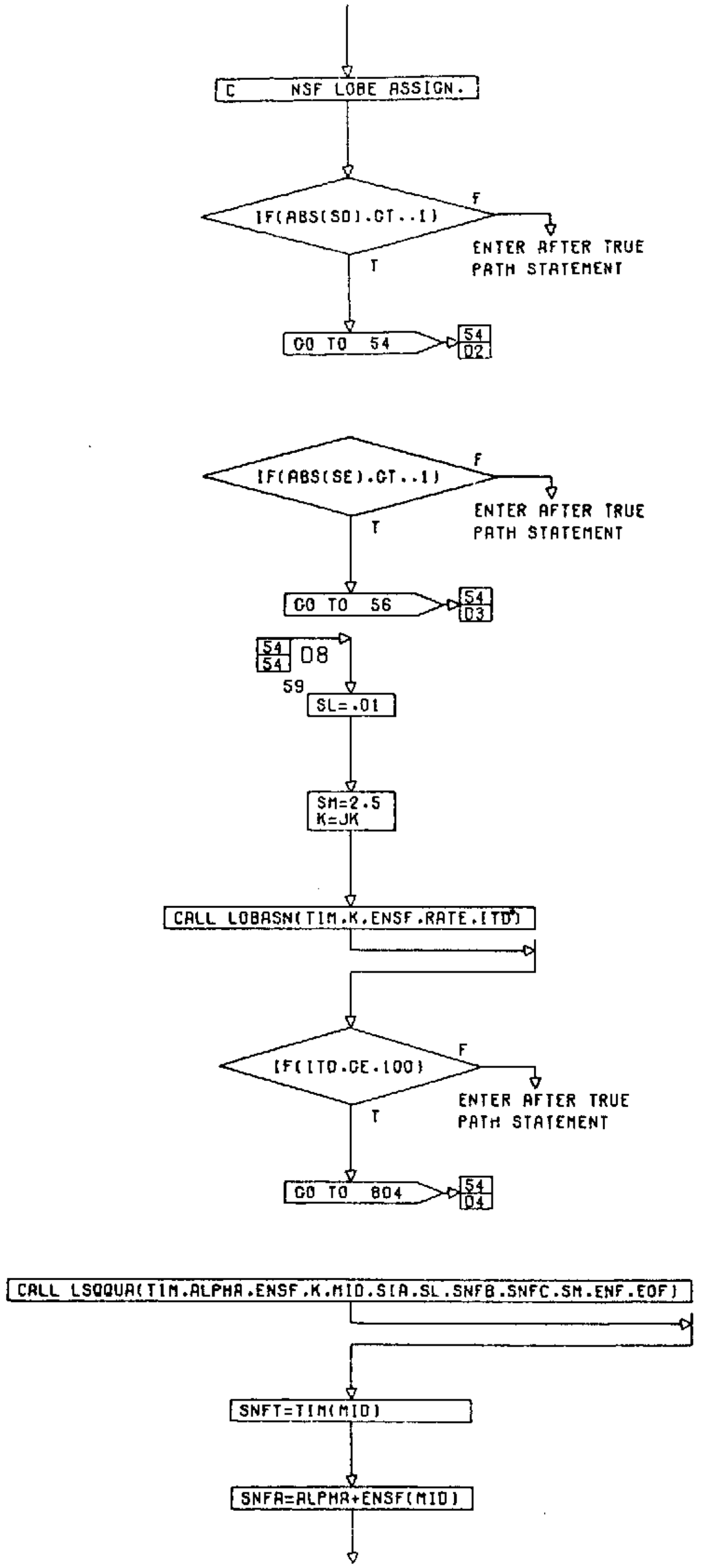
```
      INO=SIA*1000.+.5
227   CALL ZERO(3.INO.IENO)
      IDAYD(1)=IDAYD(MID)
      WRITE(11.642)
642   FORMAT(83H          ALPHA.EWF   MID.PT.   RATE              SIGMA
          BETA            GAMMA    )
      WRITE(11.152)ALPHA.MID.RATE.SIA.EWFB.EWFC
      WRITE(11.153)EWMA.EWMT.EWCA.EWCT.RATE
153   FORMAT(6(F14.6.X))
C       NS AMBIGUITY LOBE ASSIGN.
      SL=.015
      RATE=0.
      K=JK
      SM=2.0
      CALL LOBASN(TIM.K.NSMD.RATE.ITO)
      IF(ITO.GE.100)
      GO TO  804
      CALL LOBASN(TIM.K.NSCD.RATE.ITO)
```

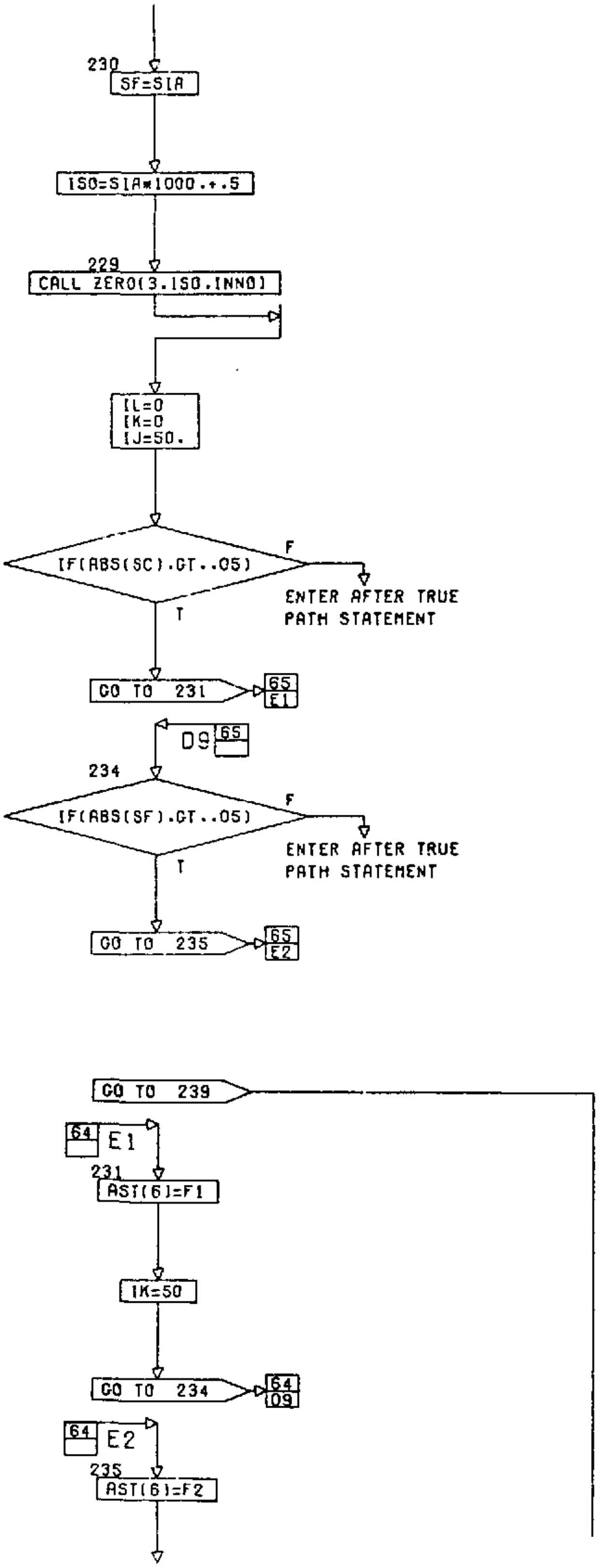


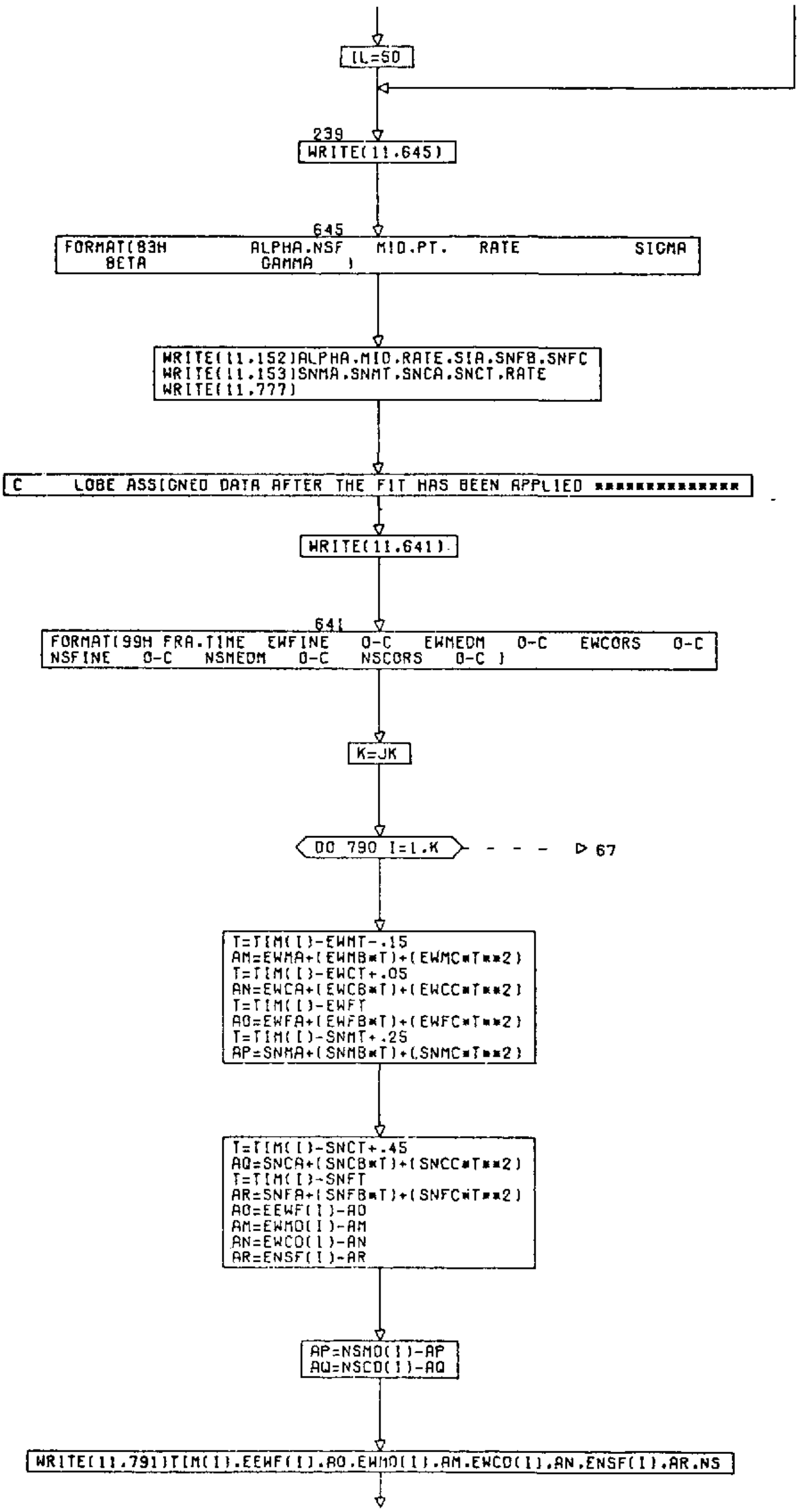
```
IF(ITD.GE.100)
GO TO  804
CALL LSQQUA(TIM.ALPHA.NSMD.K.MID.SIA.SL.SNMB.SNMC.SM.ENX.EDD)
SNMA=ALPHA+NSMD(MID)
SNMT=TIM(MID)+.25
SD=SIA
WRITE(11.646)
646
FORMAT(83H          ALPHA.NSM   MID.PT.   RATE              SIGMA
    BETA               GAMMA    )
WRITE(11.152)ALPHA.MID.RATE.SIA.SNMB.SNMC
K=JK
CALL LSQQUA(TIM.ALPHA.NSCD.K.MID.SIA.SL.SNCB.SNCC.SM.ENE.EDE)
SNCA=ALPHA+NSCD(MID)
SNCT=TIM(MID)+.45
SE=SIA
RATE=(((SNMB*ANT/4.)+(SNCB*ANT/3.5))/2.)
IF(ABS(RATE).LE..05)
RATE=0.0
WRITE(11.647)
647
FORMAT(83H          ALPHA.NSC   MID.PT.   RATE              SIGMA
    BETA               GAMMA    )
WRITE(11.152)ALPHA.MID.RATE.SIA.SNCB.SNCC
```

F
T
ENTER AFTER TRUE PATH STATEMENT
54
04
F
T
ENTER AFTER TRUE PATH STATEMENT

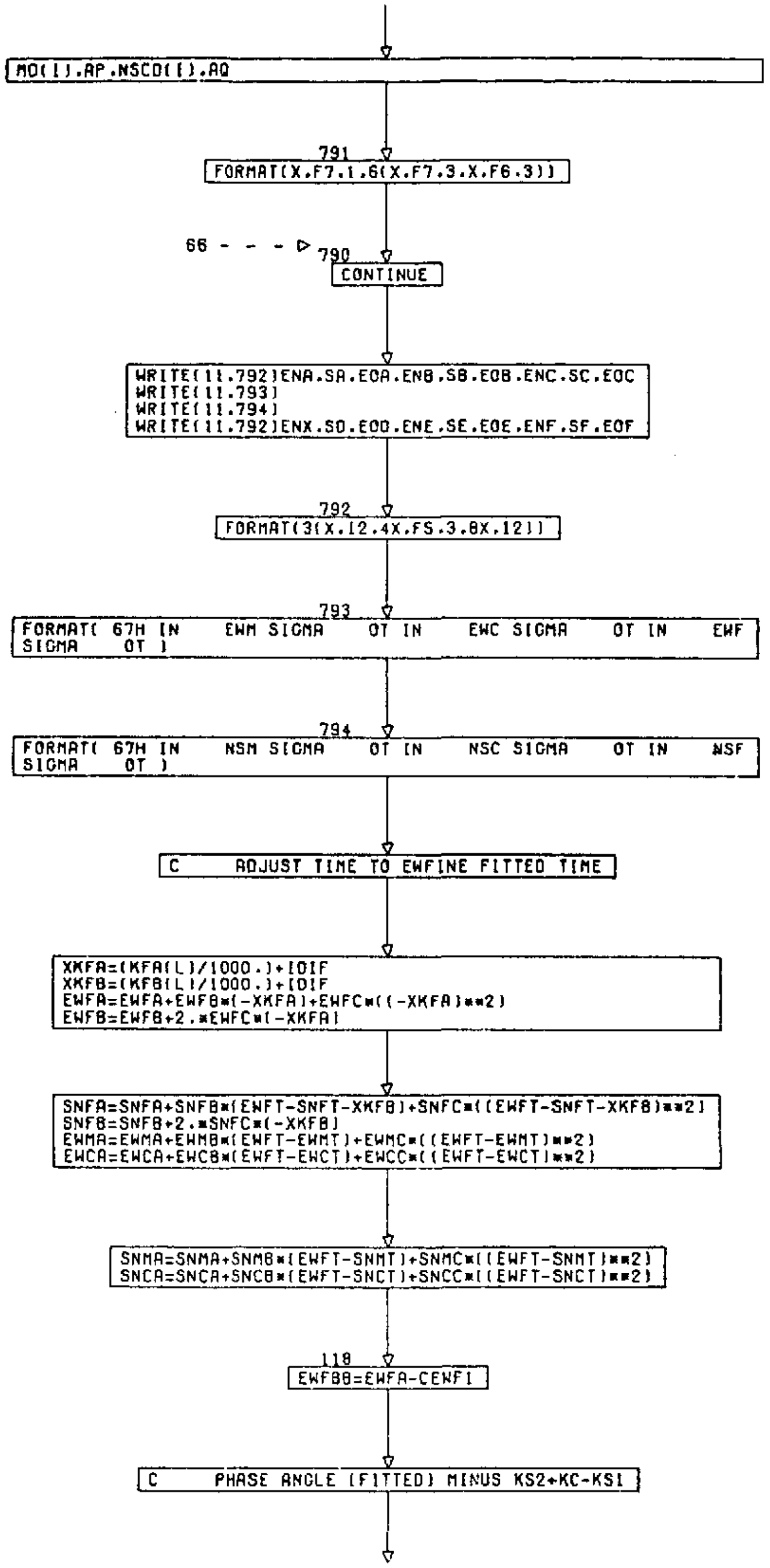
]


C NSF LOBE ASSIGN.
IF(ABS(SD).GT..1)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 54
54
02
IF(ABS(SE).GT..1)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 56
54
03
54
54
08
59
SL=.01
SM=2.5
K=JK
CALL LOBASN(TIM.K.ENSF.RATE.ITD)
IF(ITD.GE.100)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 804
54
04
CALL LSQQUA(TIM.ALPHA.ENSF.K.MID.SIA.SL.SNFB.SNFC.SM.ENF.EOF)
SNFT=TIM(MID)
SNFA=ALPHA+ENSF(MID)

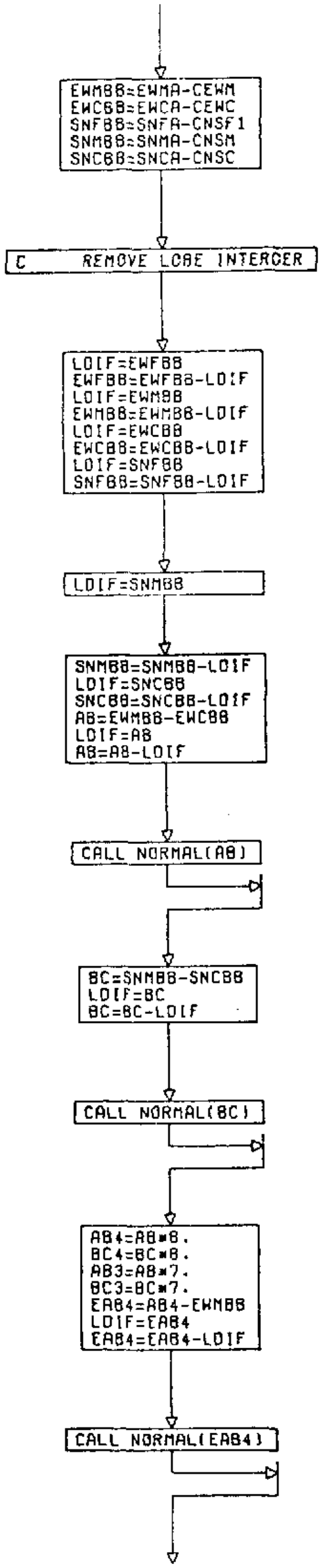

230
SF=SIA
ISO=SIA*1000.+.5
229
CALL ZERO(3,ISO,INNO)
IL=0
IK=0
IJ=50.
IF(ABS(SC).GT..05)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 231
65
E1
D9
65
234
IF(ABS(SF).GT..05)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 235
65
E2
GO TO 239
64
E1
231
AST(6)=F1
IK=50
GO TO 234
64
D9
64
E2
235
AST(6)=F2



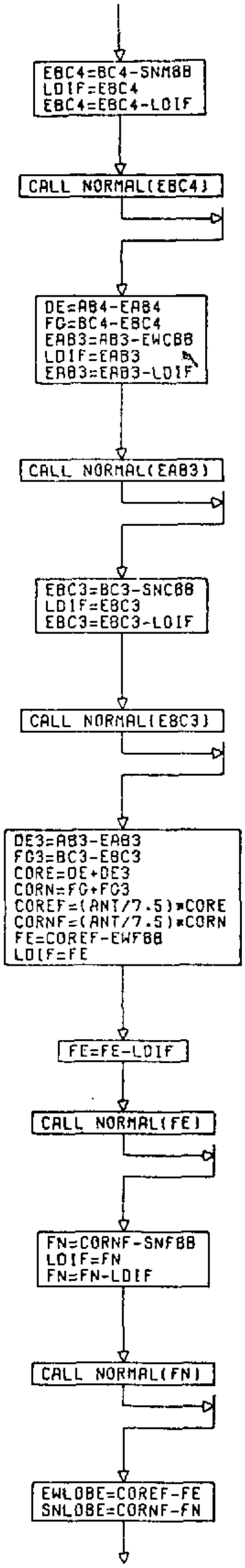
```
IL=50

239 WRITE(11.645)

645 FORMAT(83H          ALPHA.NSF    MID.PT.    RATE              SIGMA
     BETA              GAMMA    )

WRITE(11.152)ALPHA.MID.RATE.SIA.SNFB.SNFC
WRITE(11.153)SNMA.SNMT.SNCA.SNCT.RATE
WRITE(11.777)

C     LOBE ASSIGNED DATA AFTER THE FIT HAS BEEN APPLIED **************

WRITE(11.641)

641 FORMAT(99H FRA.TIME  EWFINE    O-C   EWMEDM    O-C   EWCORS    O-C
NSFINE    O-C   NSMEDM    O-C   NSCORS    O-C )

K=JK

DO 790 I=1.K  - - -  ▷ 67

T=TIM(I)-EWMT-.15
AM=EWMA+(EWMB*T)+(EWMC*T**2)
T=TIM(I)-EWCT+.05
AN=EWCA+(EWCB*T)+(EWCC*T**2)
T=TIM(I)-EWFT
AO=EWFA+(EWFB*T)+(EWFC*T**2)
T=TIM(I)-SNMT+.25
AP=SNMA+(SNMB*T)+(SNMC*T**2)

T=TIM(I)-SNCT+.45
AQ=SNCA+(SNCB*T)+(SNCC*T**2)
T=TIM(I)-SNFT
AR=SNFA+(SNFB*T)+(SNFC*T**2)
AO=EEWF(I)-AO
AM=EWMO(I)-AM
AN=EWCO(I)-AN
AR=ENSF(I)-AR

AP=NSMO(I)-AP
AQ=NSCO(I)-AQ

WRITE(11.791)TIM(I).EEWF(I).AO.EWMO(I).AM.EWCO(I).AN.ENSF(I).AR.NS
```



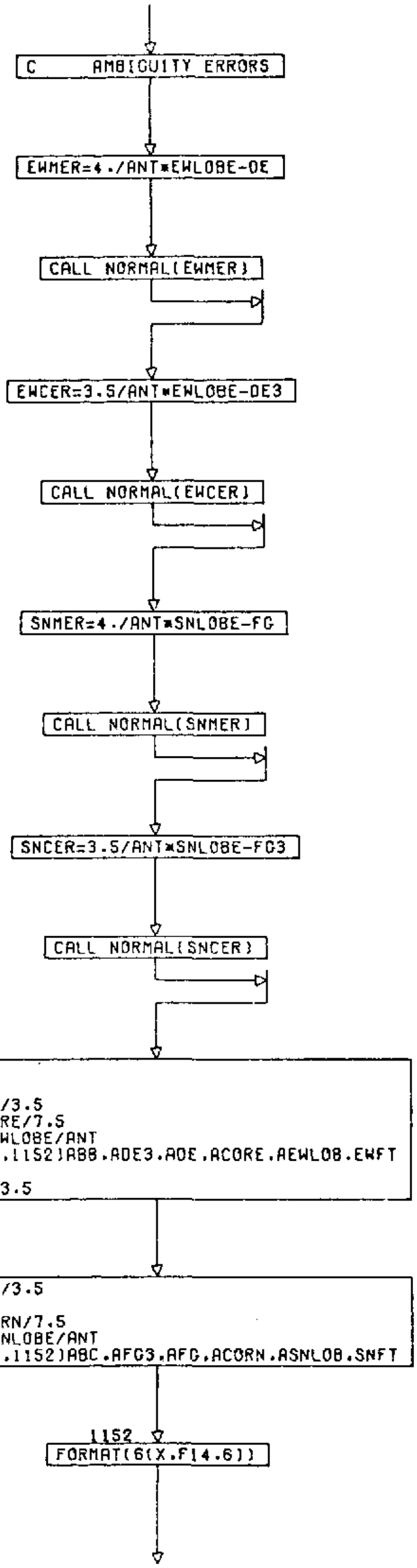
```
MO(I).AP.NSCD(I).AQ

791
FORMAT(X.F7.1.6(X.F7.3.X.F6.3))

66 - - - ▷ 790
CONTINUE

WRITE(11.792)ENA.SA.EOA.ENB.SB.EOB.ENC.SC.EOC
WRITE(11.793)
WRITE(11.794)
WRITE(11.792)ENX.SD.EOD.ENE.SE.EOE.ENF.SF.EOF

792
FORMAT(3(X.I2.4X.F5.3.8X.I2))

793
FORMAT( 67H IN     EWM SIGMA     OT IN     EWC SIGMA     OT IN     EWF
SIGMA     OT )

794
FORMAT( 67H IN     NSM SIGMA     OT IN     NSC SIGMA     OT IN     NSF
SIGMA     OT )

C     ADJUST TIME TO EWFINE FITTED TIME

XKFA=(KFA(L)/1000.)+IDIF
XKFB=(KFB(L)/1000.)+IDIF
EWFA=EWFA+EWFB*(-XKFA)+EWFC*((-XKFA)**2)
EWFB=EWFB+2.*EWFC*(-XKFA)

SNFA=SNFA+SNFB*(EWFT-SNFT-XKFB)+SNFC*((EWFT-SNFT-XKFB)**2)
SNFB=SNFB+2.*SNFC*(-XKFB)
EWMA=EWMA+EWMB*(EWFT-EWMT)+EWMC*((EWFT-EWMT)**2)
EWCA=EWCA+EWCB*(EWFT-EWCT)+EWCC*((EWFT-EWCT)**2)

SNMA=SNMA+SNMB*(EWFT-SNMT)+SNMC*((EWFT-SNMT)**2)
SNCA=SNCA+SNCB*(EWFT-SNCT)+SNCC*((EWFT-SNCT)**2)

118
EWFB8=EWFA-CEWF1

C     PHASE ANGLE (FITTED) MINUS KS2+KC-KS1
```



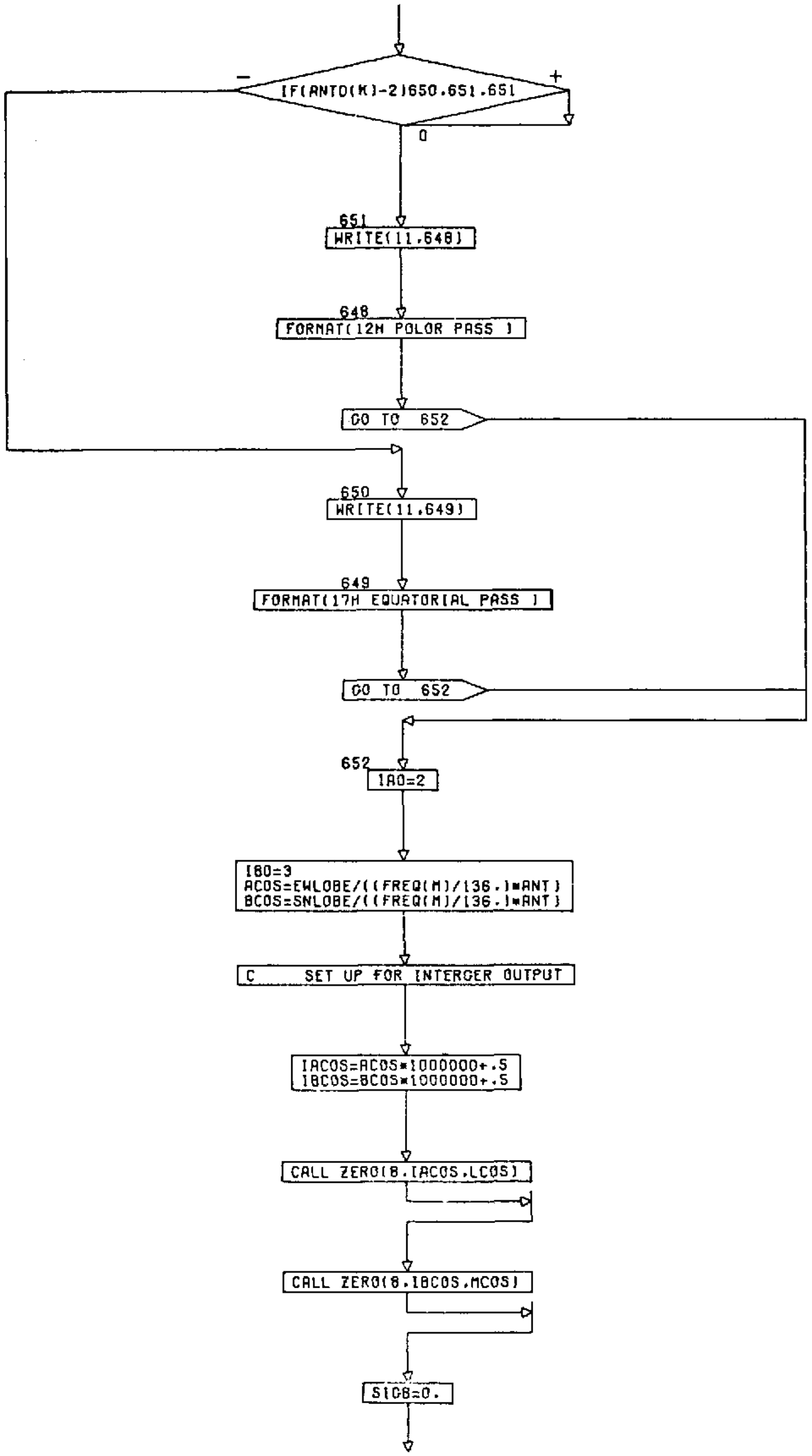
```
EWMBB=EWMA-CEWM
EWCBB=EWCA-CEWC
SNFBB=SNFA-CNSF1
SNMBB=SNMA-CNSM
SNCBB=SNCA-CNSC

C     REMOVE LOBE INTERCER

LDIF=EWFBB
EWFBB=EWFBB-LDIF
LDIF=EWMBB
EWMBB=EWMBB-LDIF
LDIF=EWCBB
EWCBB=EWCBB-LDIF
LDIF=SNFBB
SNFBB=SNFBB-LDIF

LDIF=SNMBB

SNMBB=SNMBB-LDIF
LDIF=SNCBB
SNCBB=SNCBB-LDIF
AB=EWMBB-EWCBB
LDIF=AB
AB=AB-LDIF

CALL NORMAL(AB)

BC=SNMBB-SNCBB
LDIF=BC
BC=BC-LDIF

CALL NORMAL(BC)

AB4=AB*8.
BC4=BC*8.
AB3=AB*7.
BC3=BC*7.
EAB4=AB4-EWMBB
LDIF=EAB4
EAB4=EAB4-LDIF

CALL NORMAL(EAB4)
```



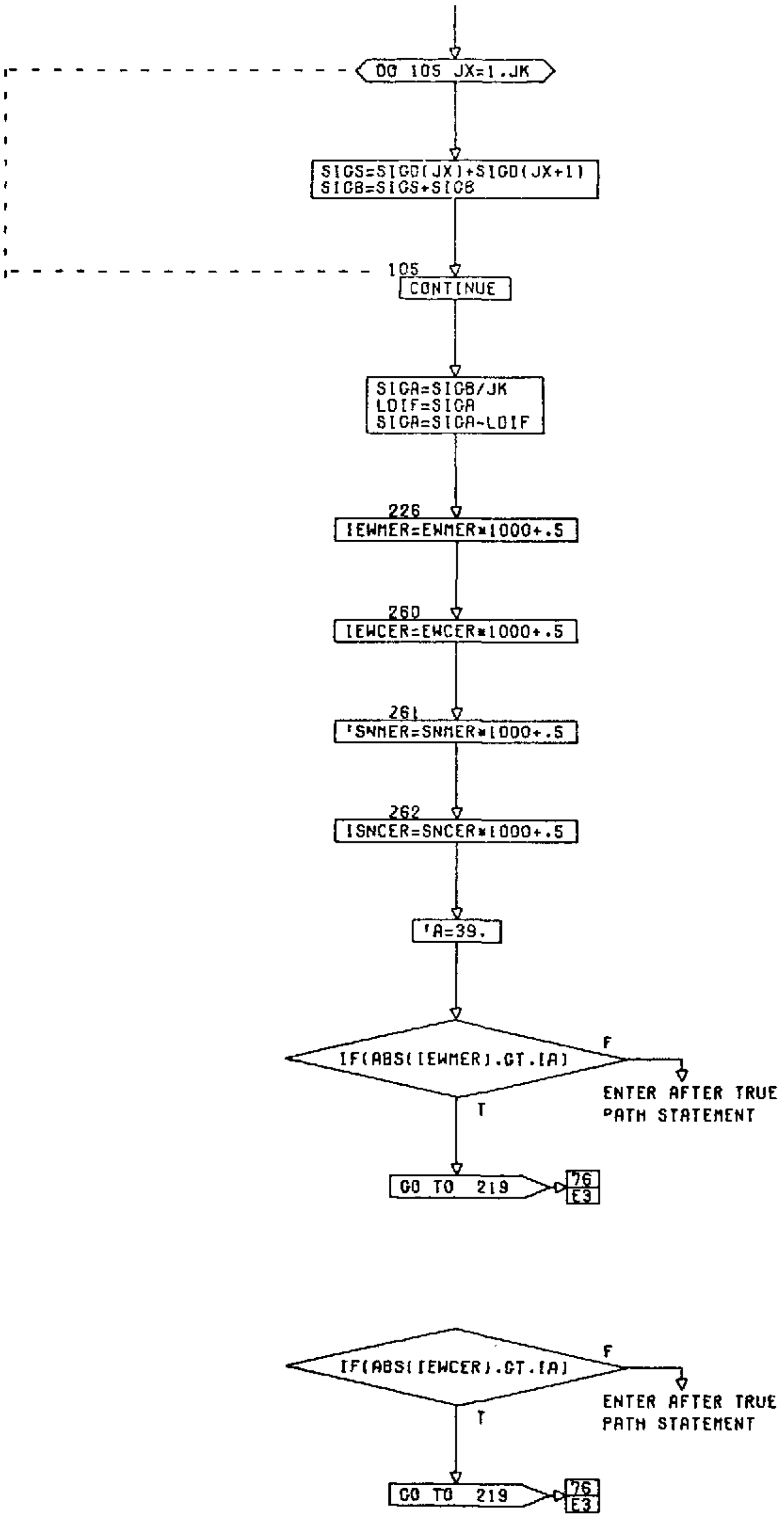
```
EBC4=BC4-SNM8B
LDIF=EBC4
EBC4=EBC4-LDIF

CALL NORMAL(EBC4)

DE=AB4-EAB4
FG=BC4-EBC4
EAB3=AB3-EWCBB
LDIF=EAB3
EAB3=EAB3-LDIF

CALL NORMAL(EAB3)

EBC3=BC3-SNCBB
LDIF=EBC3
EBC3=EBC3-LDIF

CALL NORMAL(EBC3)

DE3=AB3-EAB3
FG3=BC3-EBC3
CORE=DE+DE3
CORN=FG+FG3
COREF=(ANT/7.5)*CORE
CORNF=(ANT/7.5)*CORN
FE=COREF-EWFBB
LDIF=FE

FE=FE-LDIF

CALL NORMAL(FE)

FN=CORNF-SNFBB
LDIF=FN
FN=FN-LDIF

CALL NORMAL(FN)

EWLOBE=COREF-FE
SNLOBE=CORNF-FN
```

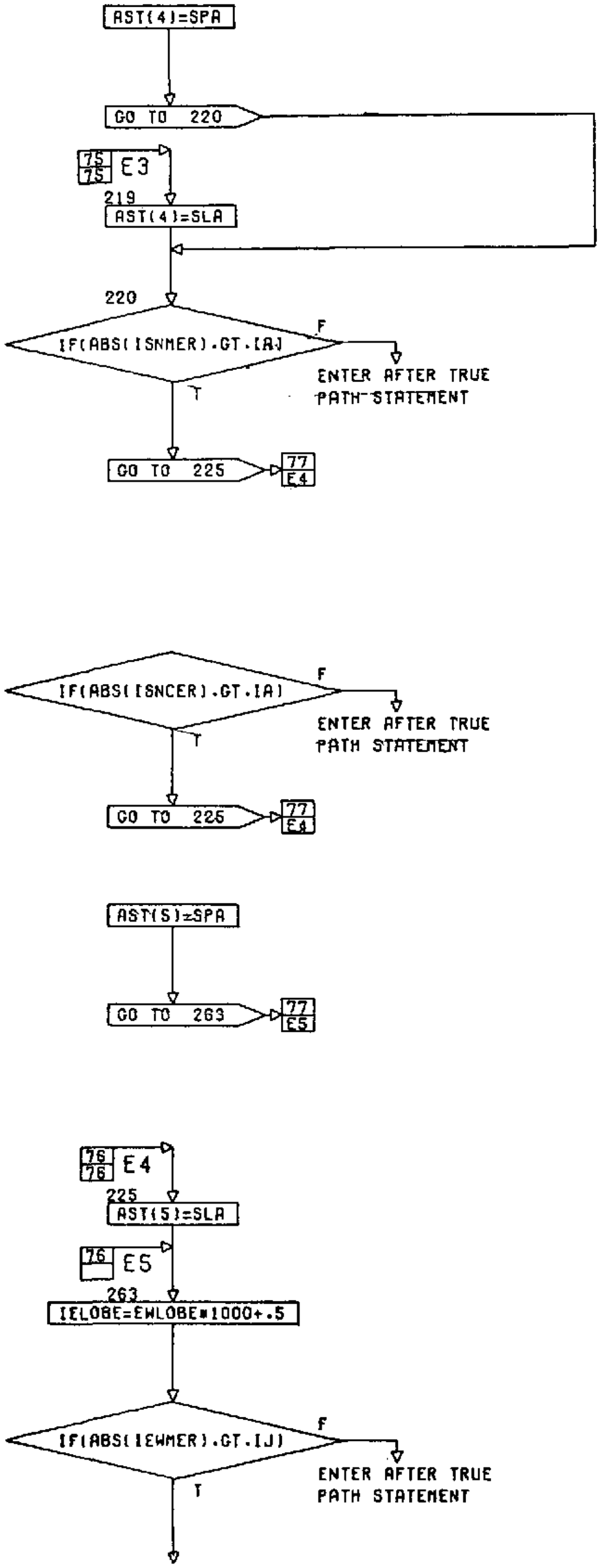


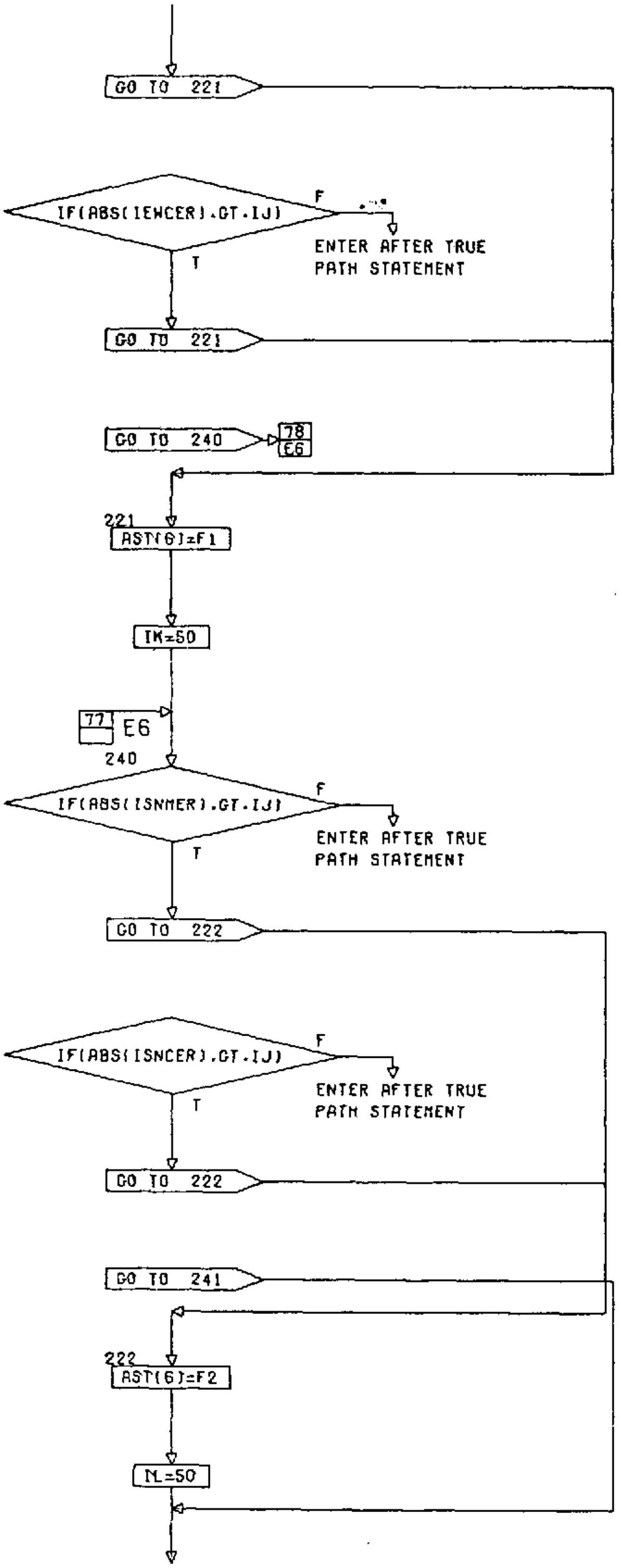
```
C      AMBIGUITY ERRORS
EWMER=4./ANT*EWLOBE-OE
CALL NORMAL(EWMER)
EWCER=3.5/ANT*EWLOBE-OE3
CALL NORMAL(EWCER)
SNMER=4./ANT*SNLOBE-FG
CALL NORMAL(SNMER)
SNCER=3.5/ANT*SNLOBE-FG3
CALL NORMAL(SNCER)
ABB=AB*2
ADE=OE/4
ADE3=DE3/3.5
ACORE=CORE/7.5
AEWLOB=EWLOBE/ANT
WRITE(11,1152)ABB,ADE3,ADE,ACORE,AEWLOB,EWFT
ABC=BC*2
AFG=FG3/3.5
AFG3=FG3/3.5
AFG=FG/4
ACORN=CORN/7.5
ASNLOB=SNLOBE/ANT
WRITE(11,1152)ABC,AFG3,AFG,ACORN,ASNLOB,SNFT
1152 FORMAT(6(X,F14.6))
```

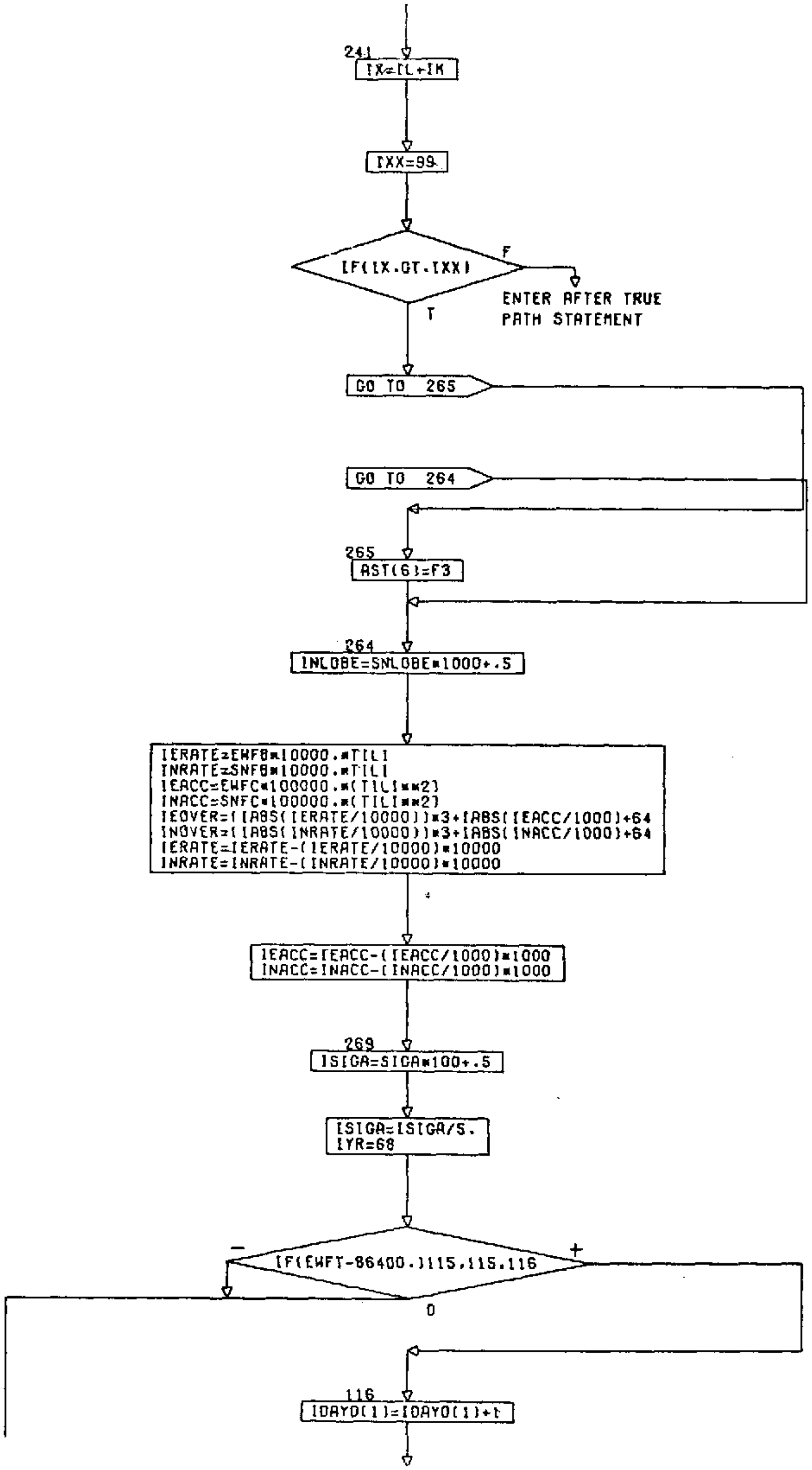


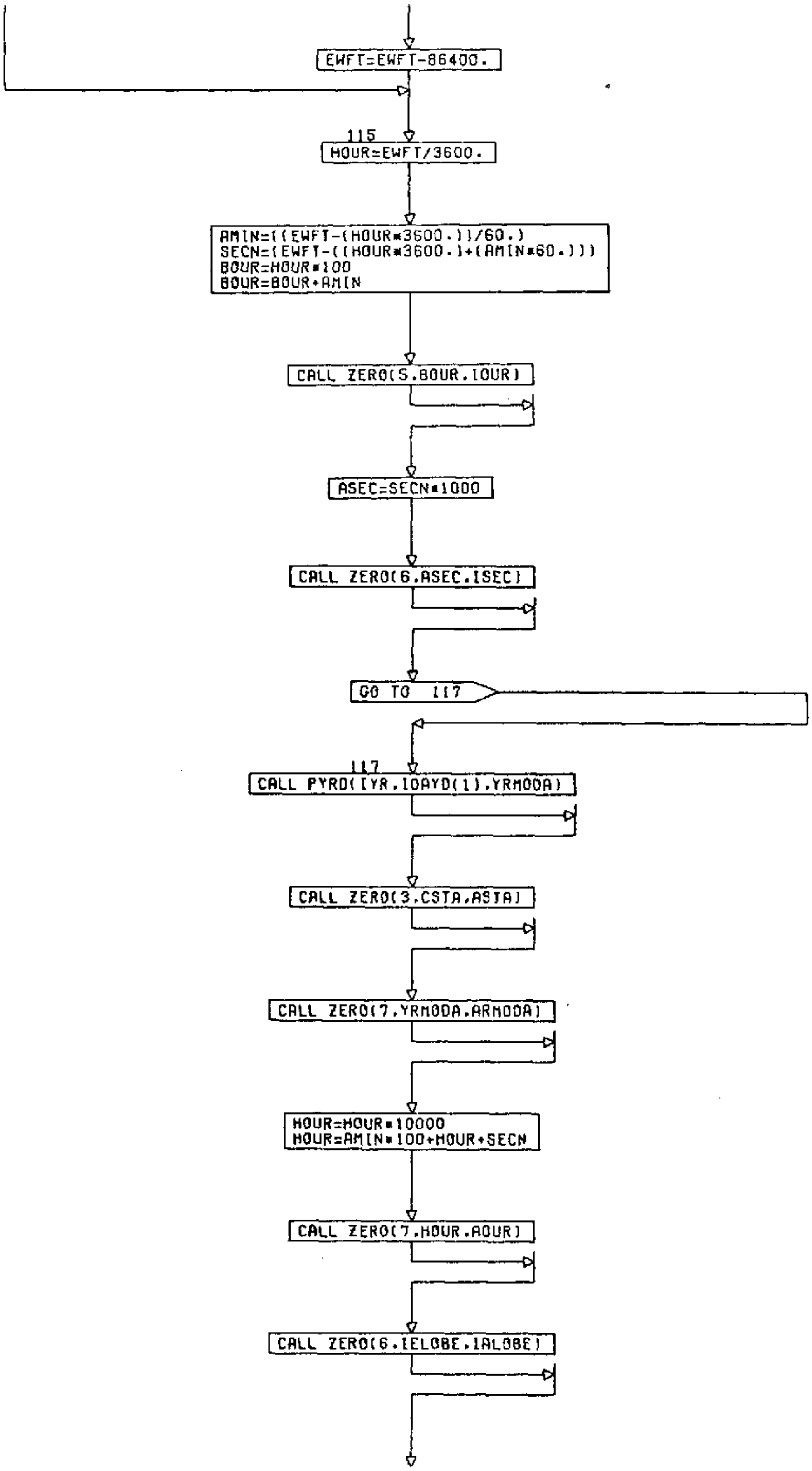
```
      IF(ANTD(K)-2)650,651,651
651   WRITE(11,648)
648   FORMAT(12H POLOR PASS )
      GO TO 652
650   WRITE(11,649)
649   FORMAT(17H EQUATORIAL PASS )
      GO TO 652
652   IAO=2
      IBO=3
      ACOS=EWLOBE/((FREQ(M)/136.)*ANT)
      BCOS=SNLOBE/((FREQ(M)/136.)*ANT)
C     SET UP FOR INTERGER OUTPUT
      IACOS=ACOS*1000000+.5
      IBCOS=BCOS*1000000+.5
      CALL ZERO(8,IACOS,LCOS)
      CALL ZERO(8,IBCOS,MCOS)
      SIGB=0.
```


DO 105 JX=1.JK
SIGS=SIGD(JX)+SIGD(JX+1)
SIGB=SIGS+SIGB
105
CONTINUE
SIGA=SIGB/JK
LDIF=SIGA
SIGA=SIGA-LDIF
226
IEWMER=EWMER*1000+.5
260
IEWCER=EWCER*1000+.5
261
ISNMER=SNMER*1000+.5
262
ISNCER=SNCER*1000+.5
IA=39.
IF(ABS(IEWMER).GT.IA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 219
76
E3
IF(ABS(IEWCER).GT.IA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 219
76
E3

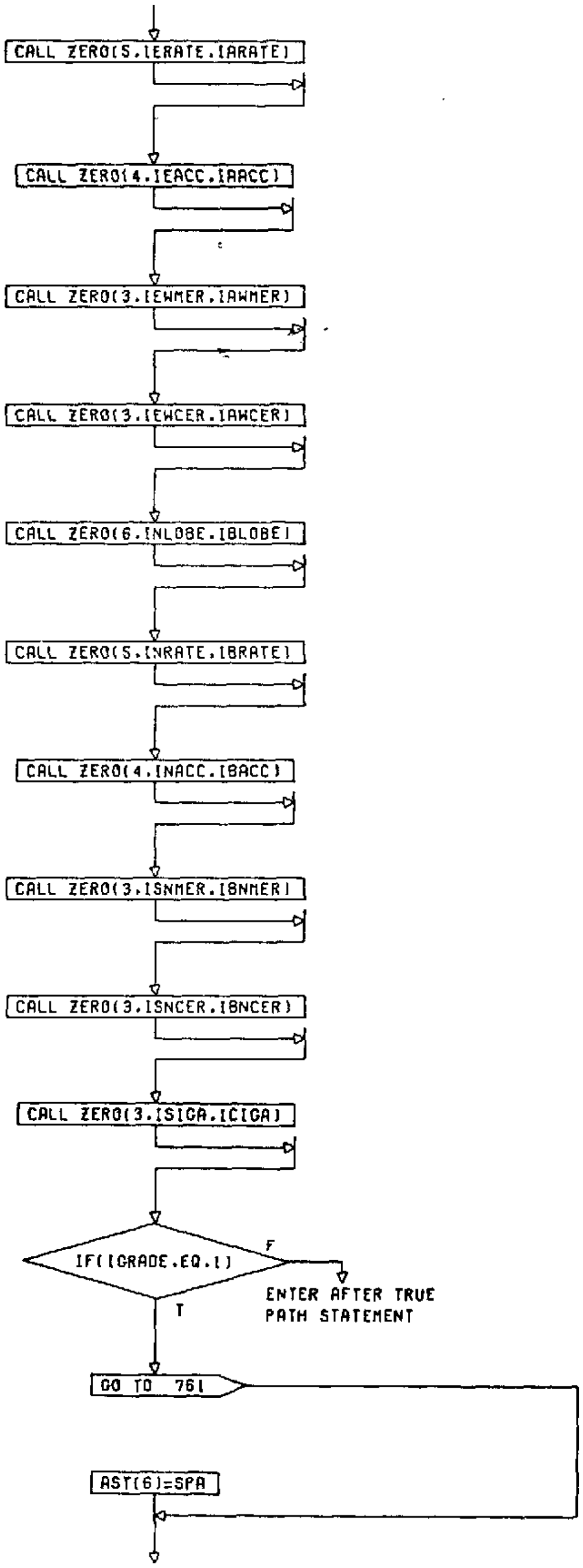

AST(4)=SPA
GO TO 220
75
75
E3
219
AST(4)=SLA
220
IF(ABS(ISNMER).GT.IA)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 225
77
E4
IF(ABS(ISNCER).GT.IA)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 225
77
E4
AST(5)=SPA
GO TO 263
77
E5
76
76
E4
225
AST(5)=SLA
76
E5
263
IELOBE=EWLOBE*1000+.5
IF(ABS(IEWMER).GT.IJ)
F
ENTER AFTER TRUE PATH STATEMENT
T

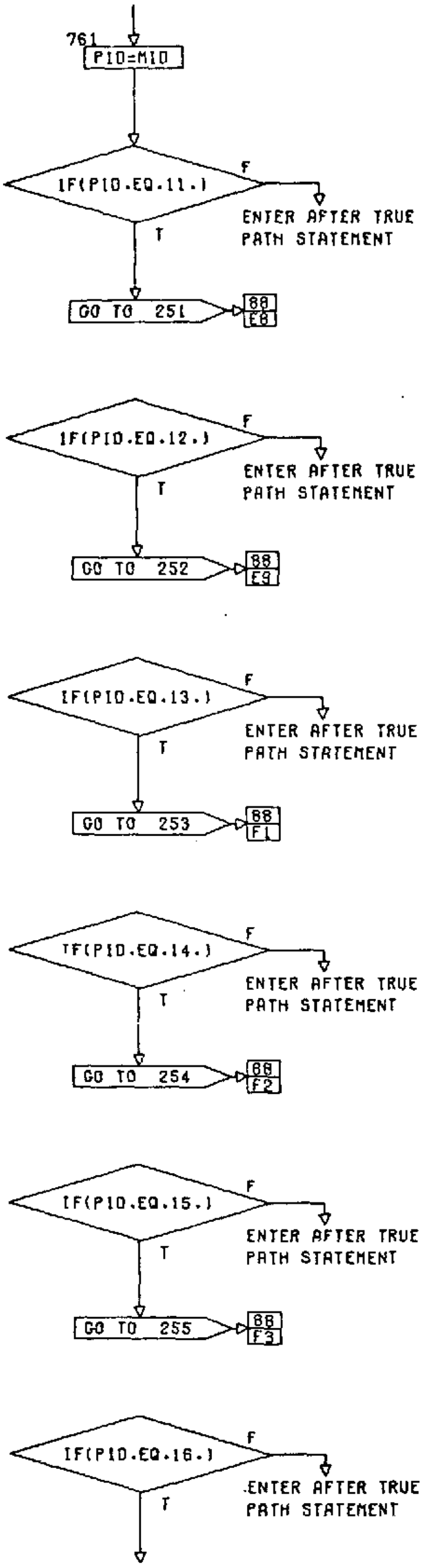

GO TO 221
IF(ABS(IEHCER).GT.IJ)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 221
GO TO 240
78
E6
221
AST(6)=F1
IK=50
77
E6
240
IF(ABS(ISNMER).GT.IJ)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 222
IF(ABS(ISNCER).GT.IJ)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 222
GO TO 241
222
AST(6)=F2
IL=50

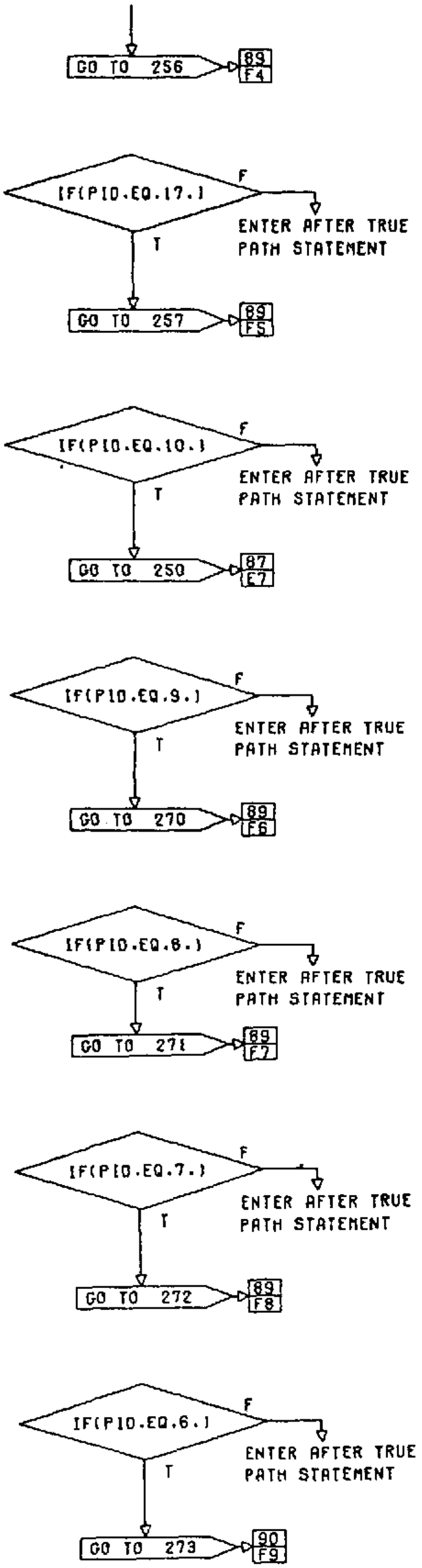

241
IX=IL+IK
IXX=99
IF(IX.GT.IXX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 265
GO TO 264
265
AST(6)=F3
264
INLOBE=SNLOBE*1000+.5
IERATE=EWFB*10000.*T(L)
INRATE=SNFB*10000.*T(L)
IEACC=EWFC*100000.*(T(L)**2)
INACC=SNFC*100000.*(T(L)**2)
IEOVER=(IABS(IERATE/10000))*3+IABS(IEACC/1000)+64
INOVER=(IABS(INRATE/10000))*3+IABS(INACC/1000)+64
IERATE=IERATE-(IERATE/10000)*10000
INRATE=INRATE-(INRATE/10000)*10000
IEACC=IEACC-(IEACC/1000)*1000
INACC=INACC-(INACC/1000)*1000
269
ISIGA=SIGA*100+.5
ISIGA=ISIGA/5.
IYR=68
−
+
IF(EWFT-86400.)115,115,116
0
116
IDAYO(1)=IDAYO(1)+1

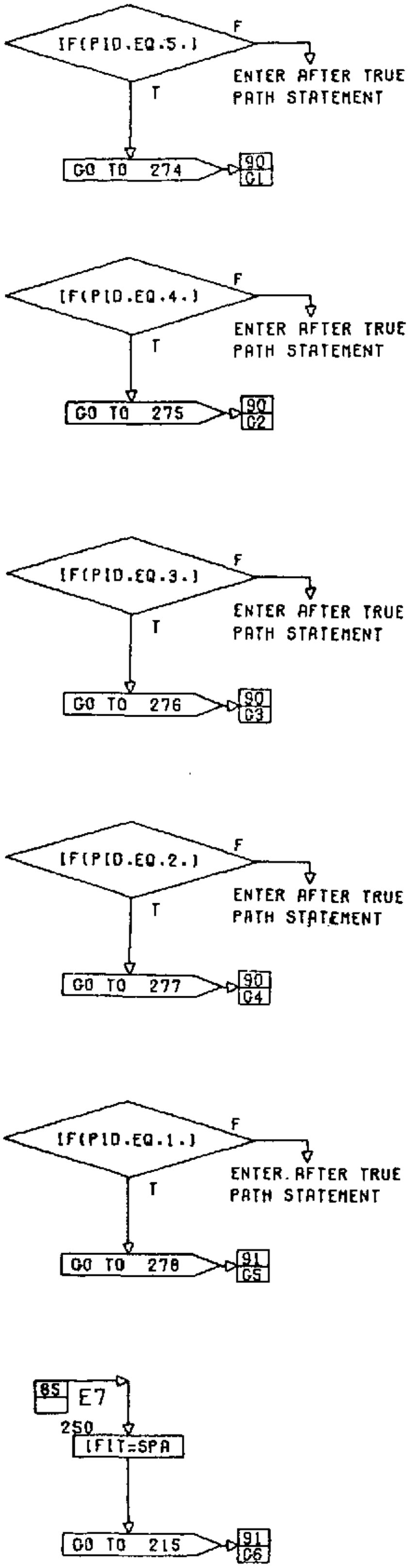


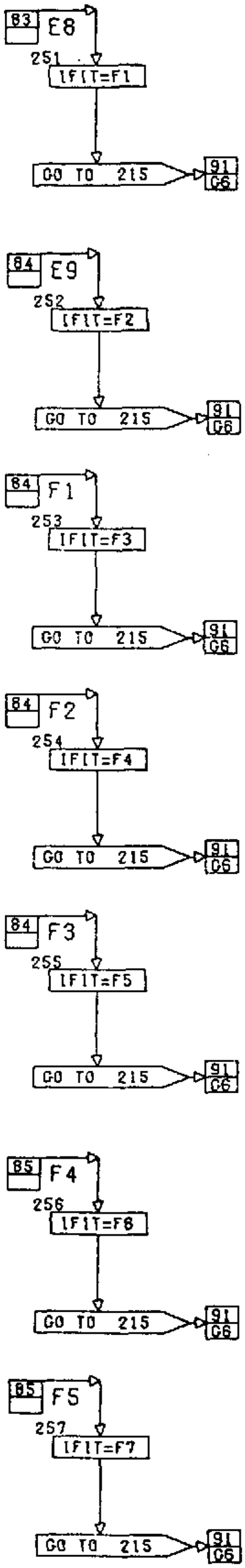
```
EWFT=EWFT-86400.

115
HOUR=EWFT/3600.

AMIN=((EWFT-(HOUR*3600.))/60.)
SECN=(EWFT-((HOUR*3600.)+(AMIN*60.)))
BOUR=HOUR*100
BOUR=BOUR+AMIN

CALL ZERO(5.BOUR.IOUR)

ASEC=SECN*1000

CALL ZERO(6.ASEC.ISEC)

GO TO 117

117
CALL PYRO(IYR.IDAYD(1).YRMODA)

CALL ZERO(3.CSTA.ASTA)

CALL ZERO(7.YRMODA.ARMODA)

HOUR=HOUR*10000
HOUR=AMIN*100+HOUR+SECN

CALL ZERO(7.HOUR.AOUR)

CALL ZERO(6.IELOBE.IALOBE)
```

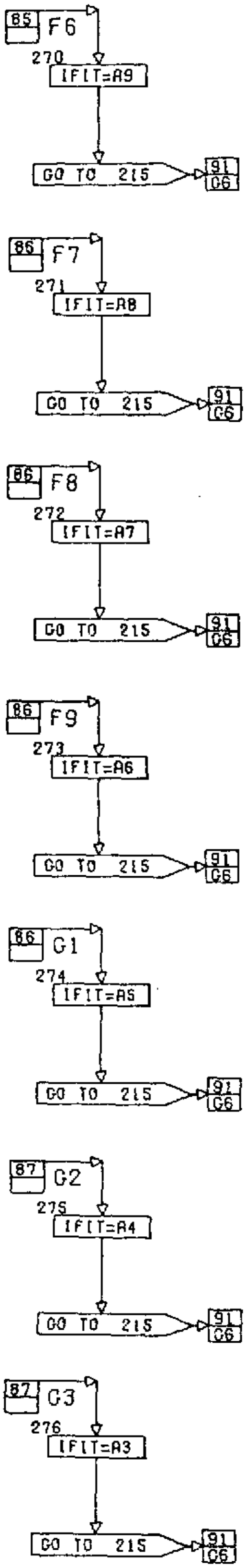

CALL ZERO(5.IERATE.IARATE)
CALL ZERO(4.IEACC.IAACC)
CALL ZERO(3.IEWMER.IAWMER)
CALL ZERO(3.IEWCER.IAWCER)
CALL ZERO(6.INLOBE.IBLOBE)
CALL ZERO(5.INRATE.IBRATE)
CALL ZERO(4.INACC.IBACC)
CALL ZERO(3.ISNMER.IBNMER)
CALL ZERO(3.ISNCER.IBNCER)
CALL ZERO(3.ISIGA.ICIGA)
IF(IGRADE.EQ.1)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 761
AST(6)=SPA

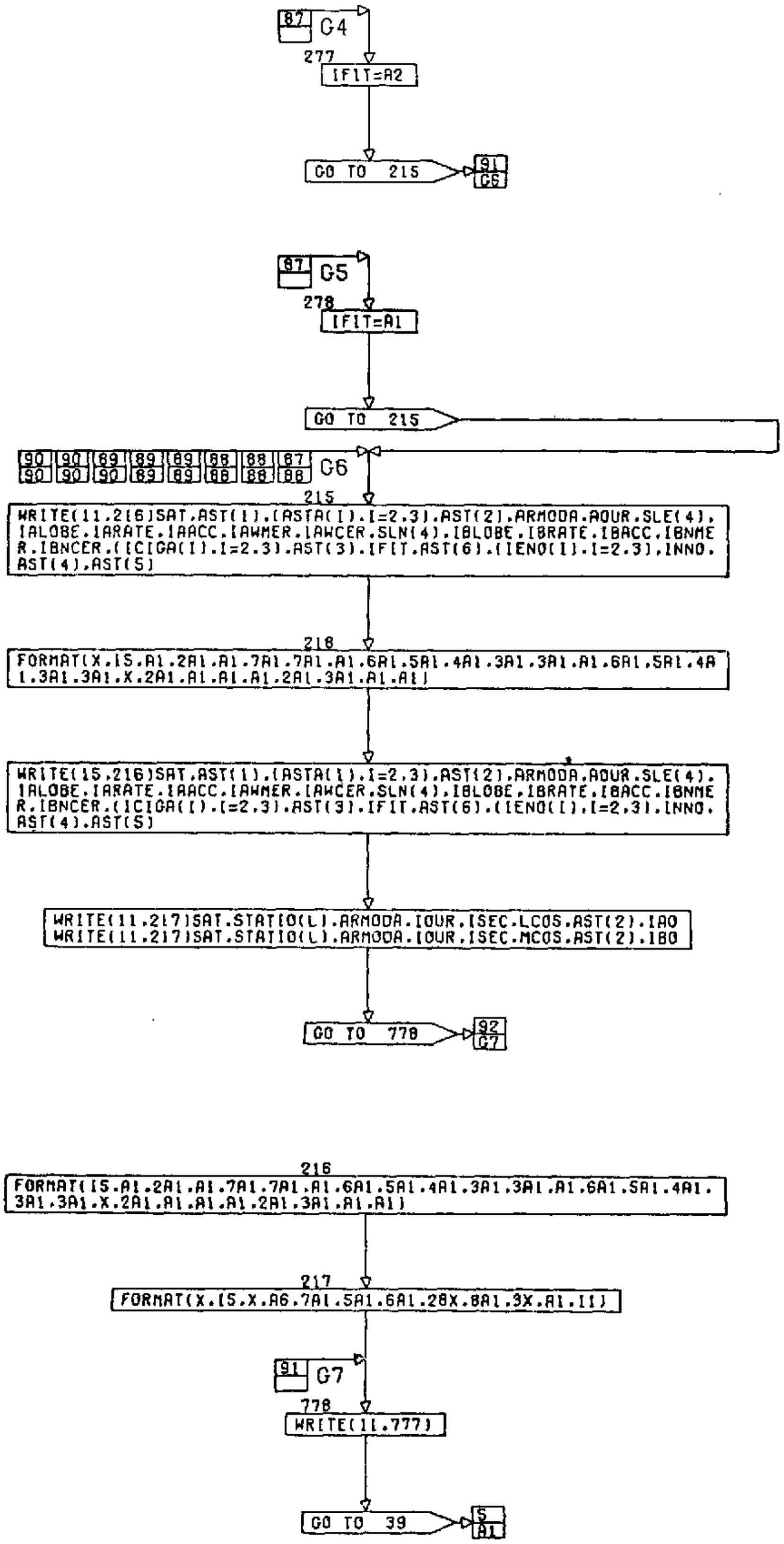

761
PID=MID
IF(PID.EQ.11.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 251
88
E8
IF(PID.EQ.12.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 252
88
E9
IF(PID.EQ.13.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 253
88
F1
TF(PID.EQ.14.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 254
88
F2
IF(PID.EQ.15.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 255
88
F3
IF(PID.EQ.16.)
F
ENTER AFTER TRUE PATH STATEMENT
T


GO TO 256
89
F4
IF(P10.EQ.17.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 257
89
F5
IF(P10.EQ.10.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 250
87
E7
IF(P10.EQ.9.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 270
89
F6
IF(P10.EQ.8.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 271
89
F7
IF(P10.EQ.7.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 272
89
F8
IF(P10.EQ.6.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 273
90
F9

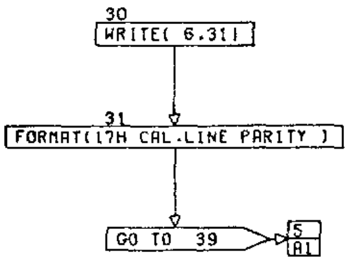

IF(PID.EQ.5.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 274
90
C1
IF(PID.EQ.4.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 275
90
C2
IF(PID.EQ.3.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 276
90
C3
IF(PID.EQ.2.)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 277
90
C4
IF(PID.EQ.1.)
F
ENTER. AFTER TRUE PATH STATEMENT
T
GO TO 278
91
C5
85
E7
250
IFIT=SPA
GO TO 215
91
C6

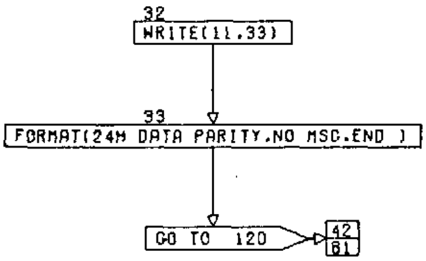

83
E8
251
IF(T=F1
GO TO 215
91
G6
84
E9
252
IF(T=F2
GO TO 215
91
G6
84
F1
253
IF(T=F3
GO TO 215
91
G6
84
F2
254
IF(T=F4
GO TO 215
91
G6
84
F3
255
IF(T=F5
GO TO 215
91
G6
85
F4
256
IF(T=F6
GO TO 215
91
G6
85
F5
257
IF(T=F7
GO TO 215
91
G6

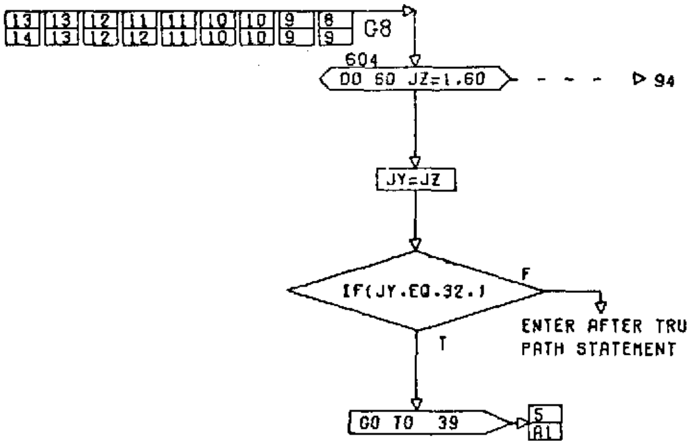

85
F6
270
IFIT=A9
GO TO 215
91
G6
86
F7
271
IFIT=A8
GO TO 215
91
G6
86
F8
272
IFIT=A7
GO TO 215
91
G6
86
F9
273
IFIT=A6
GO TO 215
91
G6
86
G1
274
IFIT=A5
GO TO 215
91
G6
87
G2
275
IFIT=A4
GO TO 215
91
G6
87
G3
276
IFIT=A3
GO TO 215
91
G6

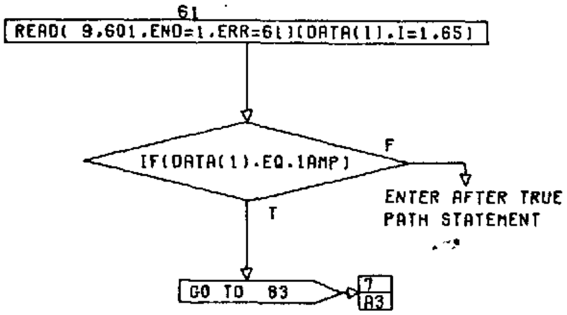


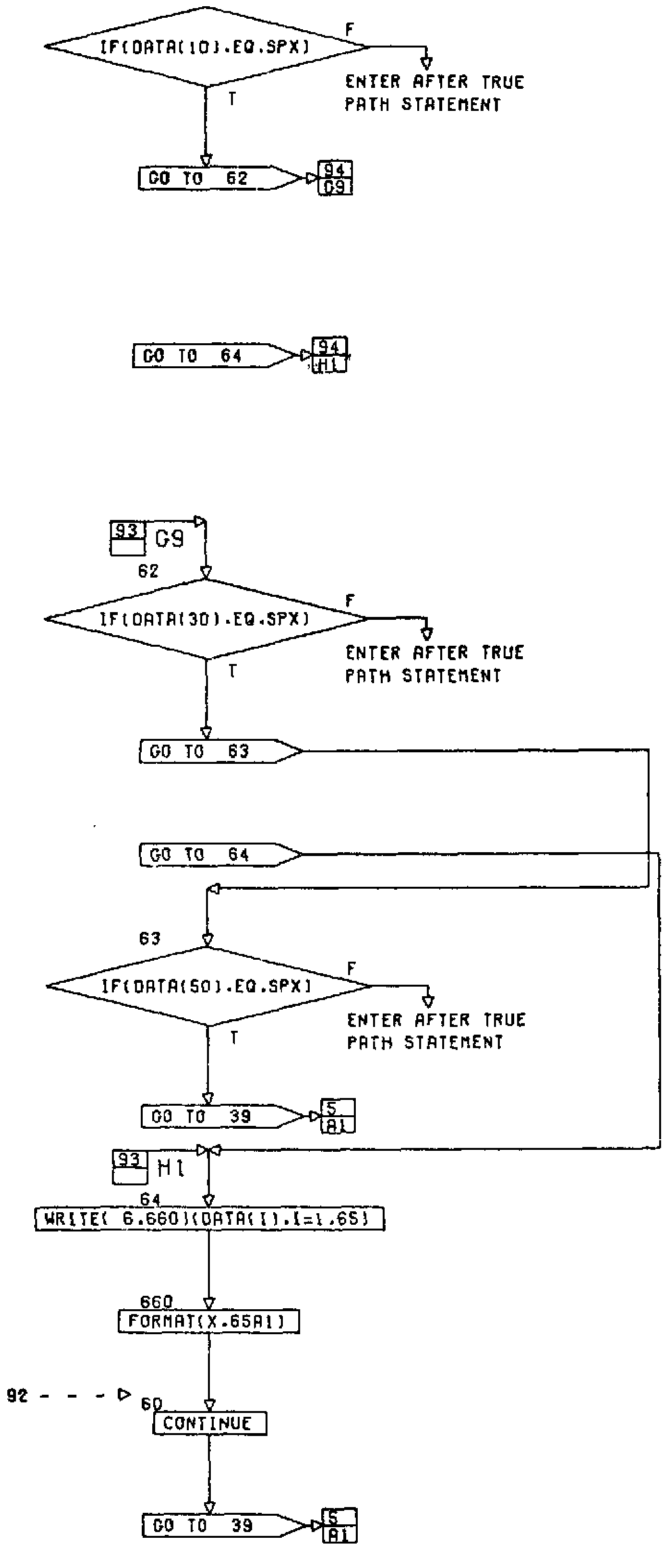
```
277  IFIT=A2
     GO TO  215

278  IFIT=A1
     GO TO  215

215  WRITE(11.2{6)SAT.AST(I).(ASTA(I).I=2.3).AST(2).ARMODA.AOUR.SLE(4).
     IALOBE.IARATE.IAACC.IAWMER.IAWCER.SLN(4).IBLOBE.IBRATE.IBACC.IBNME
     R.IBNCER.((CIGA(I).I=2.3).AST(3).IFIT.AST(6).(IENO(I).I=2.3).INNO.
     AST(4).AST(5)

218  FORMAT(X.I5.A1.2A1.A1.7A1.7A1.A1.6A1.5A1.4A1.3A1.3A1.A1.6A1.5A1.4A
     1.3A1.3A1.X.2A1.A1.A1.A1.2A1.3A1.A1.A1)

     WRITE(15.216)SAT.AST(1).(ASTA(I).I=2.3).AST(2).ARMODA.AOUR.SLE(4).
     IALOBE.IARATE.IAACC.IAWMER.IAWCER.SLN(4).IBLOBE.IBRATE.IBACC.IBNME
     R.IBNCER.((CIGA(I).I=2.3).AST(3).IFIT.AST(6).(IENO(I).I=2.3).INNO.
     AST(4).AST(5)

     WRITE(11.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.AST(2).IAO
     WRITE(11.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.AST(2).IBO

     GO TO  778

216  FORMAT(I5.A1.2A1.A1.7A1.7A1.A1.6A1.5A1.4A1.3A1.3A1.A1.6A1.5A1.4A1.
     3A1.3A1.X.2A1.A1.A1.A1.2A1.3A1.A1.A1)

217  FORMAT(X.I5.X.A6.7A1.5A1.6A1.28X.8A1.3X.A1.I1)

778  WRITE(11.777)

     GO TO  39
```

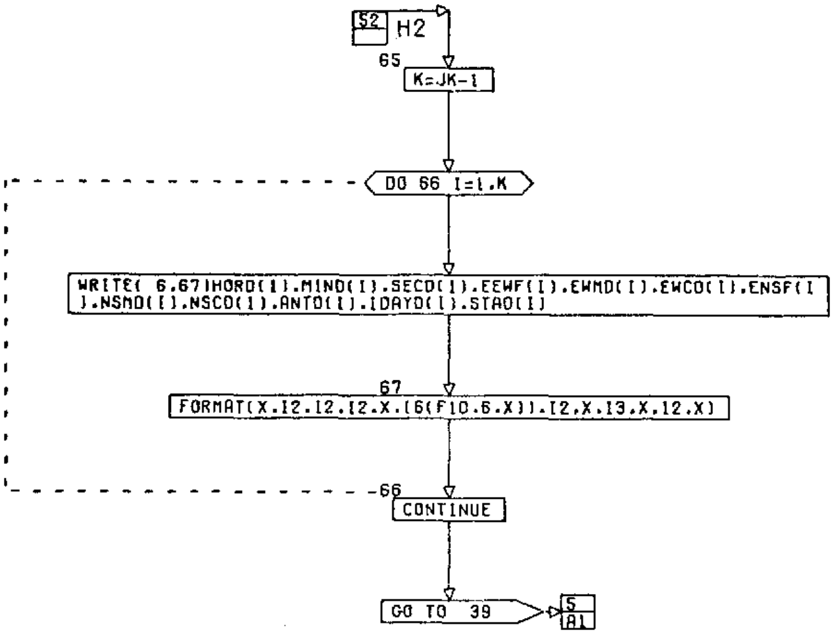

30
WRITE(6.31)
31
FORMAT(17H CAL.LINE PARITY)
GO TO 39
5
A1

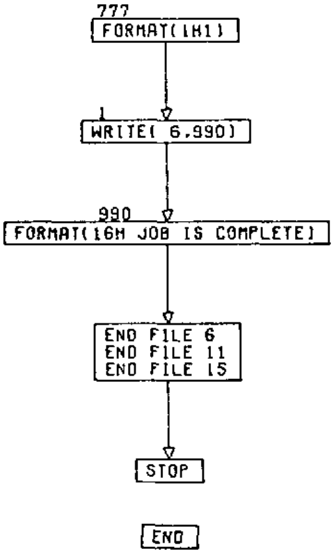

32
WRITE(11.33)
33
FORMAT(24H DATA PARITY.NO MSC.END)
GO TO 120
42
B1

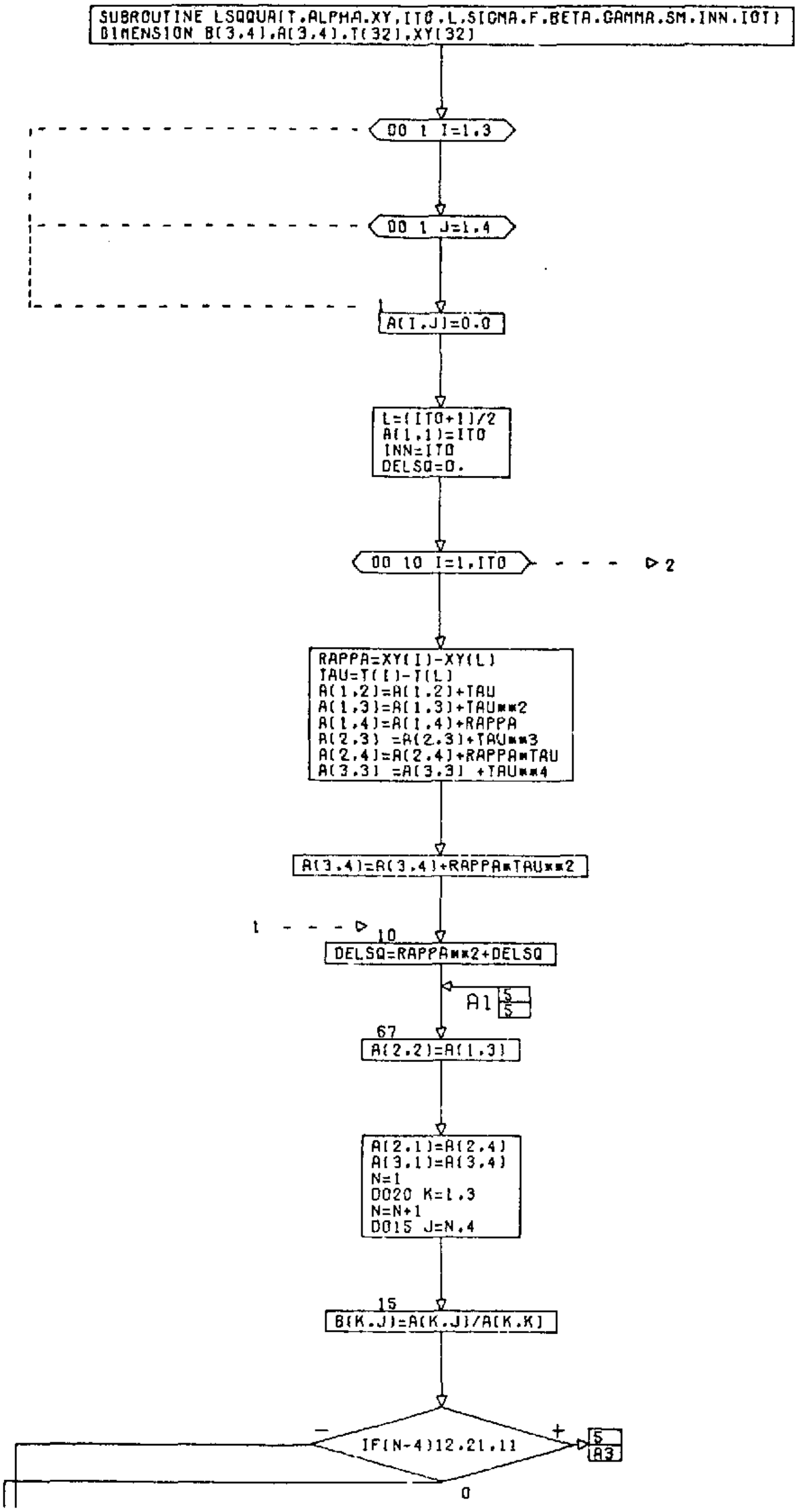

13
14
13
13
12
12
11
12
11
11
10
10
10
10
9
9
8
9
C8
604
DO 60 JZ=1.60
94
JY=JZ
IF(JY.EQ.32.)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 39
5
A1


61
READ(9,601.END=1.ERR=61)(DATA(I).I=1.65)
IF(DATA(1).EQ.1AMP)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 83
7
A3

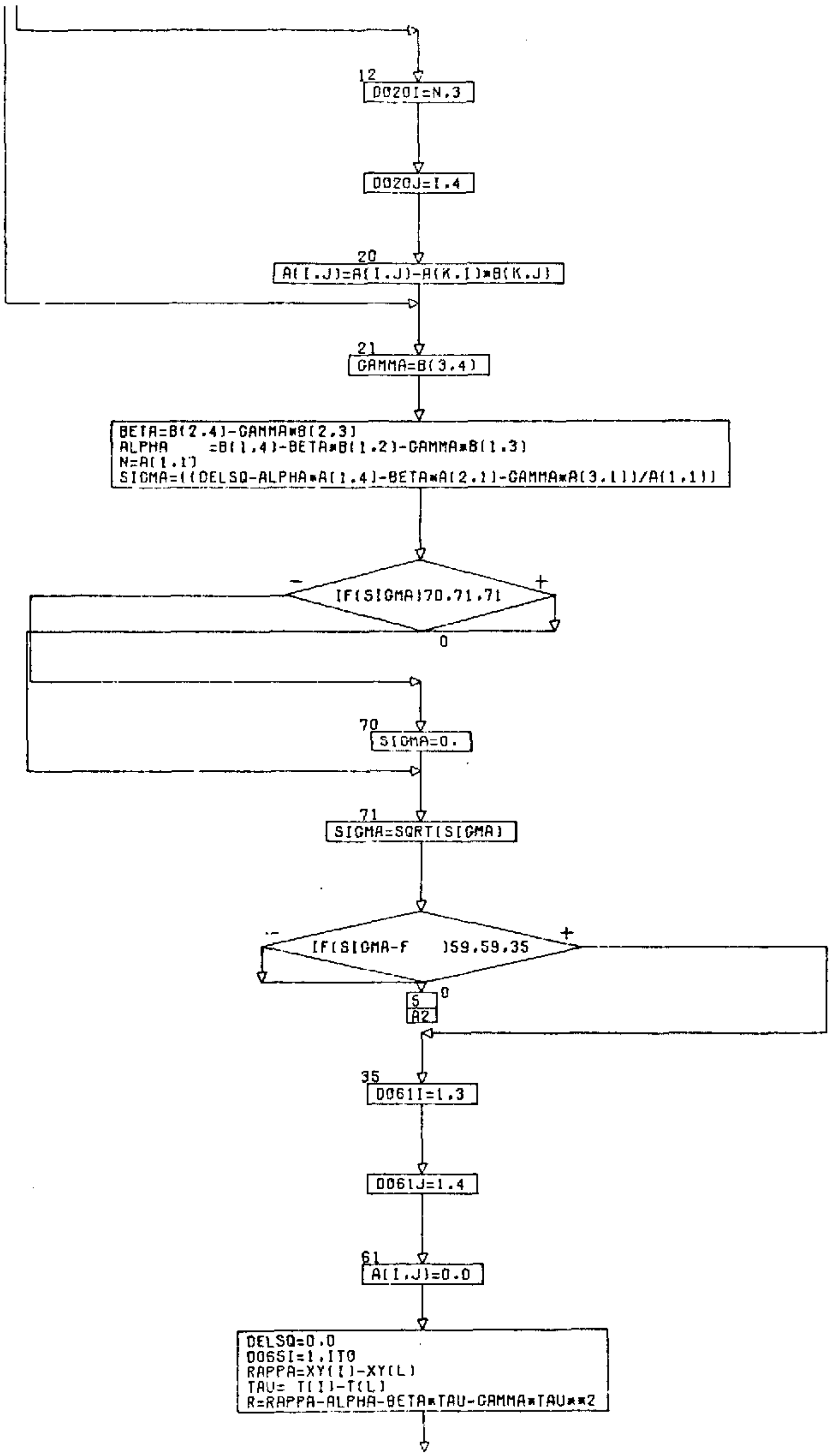

IF(DATA(10).EQ.SPX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 62
94
G9
GO TO 64
94
H1
93
G9
62
IF(DATA(30).EQ.SPX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 63
GO TO 64
63
IF(DATA(50).EQ.SPX)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 39
5
A1
93
H1
64
WRITE(6.660)(DATA(I).I=1.65)
660
FORMAT(X.65A1)
92
60
CONTINUE
GO TO 39
5
A1

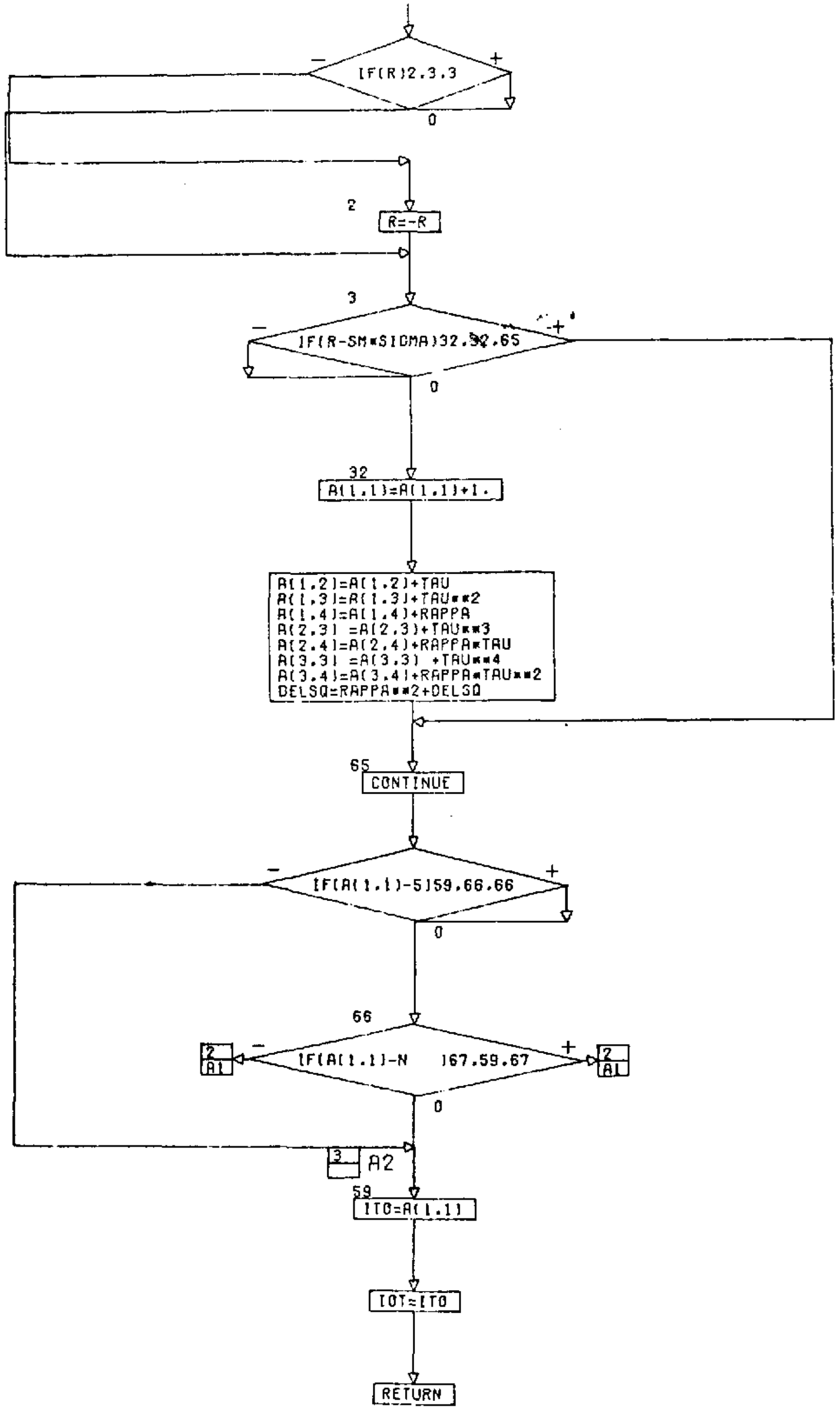

52
H2
65
K=JK-1
DO 66 I=1.K
WRITE(6.67)HORD(I).MIND(I).SECD(I).EEWF(I).EWMD(I).EWCD(I).ENSF(I
).NSMD(I).NSCD(I).ANTD(I).IDAYD(I).STAD(I)
67
FORMAT(X.I2.I2.I2.X.(6(F10.6.X)).I2.X.I3.X.I2.X)
66
CONTINUE
GO TO 39
5
A1

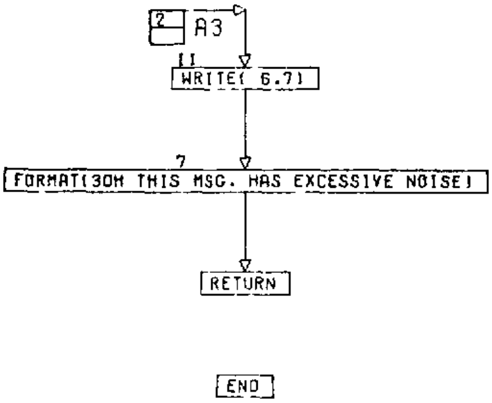

777
FORMAT(1H1)
1
WRITE(6.990)
990
FORMAT(16H JOB IS COMPLETE)
END FILE 6
END FILE 11
END FILE 15
STOP
END



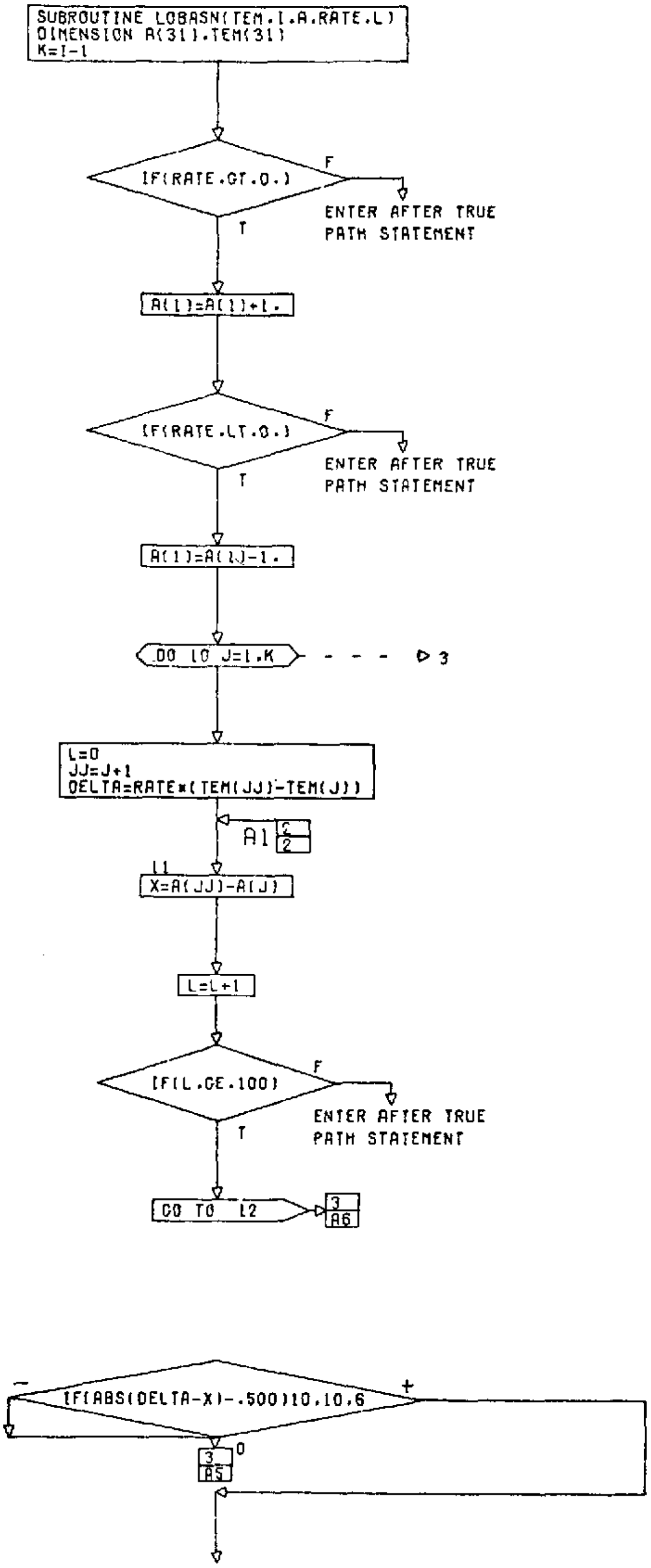
```
SUBROUTINE LSQQUA(T.ALPHA.XY,ITO.L,SIGMA.F.BETA.GAMMA.SM.INN.IOT)
DIMENSION B(3.4).A(3.4).T(32).XY(32)

DO 1 I=1.3

DO 1 J=1.4

A(I.J)=0.0

L=(ITO+1)/2
A(1.1)=ITO
INN=ITO
DELSQ=0.

DO 10 I=1.ITO

RAPPA=XY(I)-XY(L)
TAU=T(I)-T(L)
A(1.2)=A(1.2)+TAU
A(1.3)=A(1.3)+TAU**2
A(1.4)=A(1.4)+RAPPA
A(2.3) =A(2.3)+TAU**3
A(2.4)=A(2.4)+RAPPA*TAU
A(3.3) =A(3.3) +TAU**4

A(3.4)=A(3.4)+RAPPA*TAU**2

10
DELSQ=RAPPA**2+DELSQ

67
A(2.2)=A(1.3)

A(2.1)=A(2.4)
A(3.1)=A(3.4)
N=1
DO20 K=1.3
N=N+1
DO15 J=N.4

15
B(K.J)=A(K.J)/A(K.K)

IF(N-4)12.21.11
```

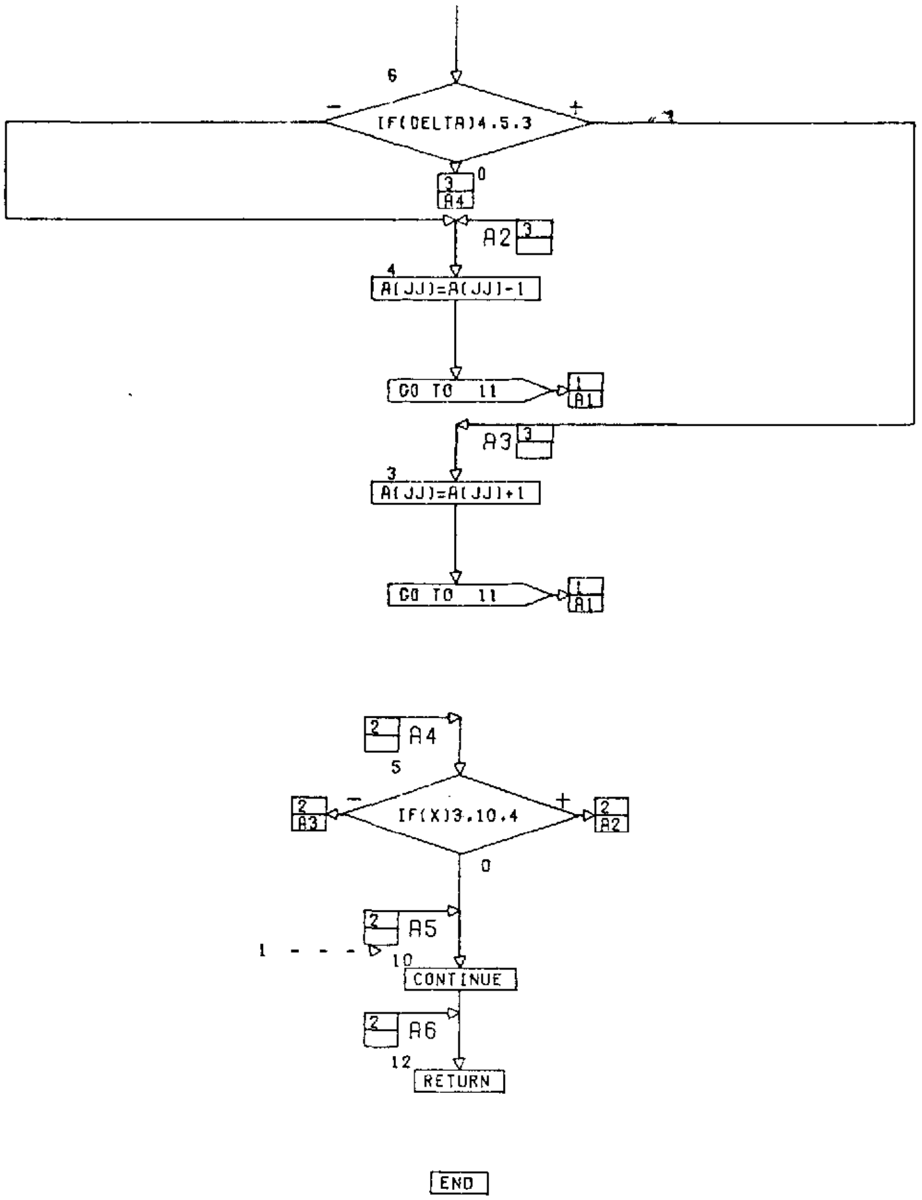

12
DO20I=N.3
DO20J=I.4
20
A(I.J)=A(I.J)-A(K.I)*B(K.J)
21
GAMMA=B(3.4)
BETA=B(2.4)-GAMMA*B(2.3)
ALPHA =B(1.4)-BETA*B(1.2)-GAMMA*B(1.3)
N=A(1.1)
SIGMA=((DELSQ-ALPHA*A(1.4)-BETA*A(2.1)-GAMMA*A(3.1))/A(1.1))
-
IF(SIGMA)70.71.71
+
0
70
SIGMA=0.
71
SIGMA=SQRT(SIGMA)
-
IF(SIGMA-F)59.59.35
+
0
5
A2
35
DO61I=1.3
DO61J=1.4
61
A(I.J)=0.0
DELSQ=0.0
DO65I=1.ITO
RAPPA=XY(I)-XY(L)
TAU= T(I)-T(L)
R=RAPPA-ALPHA-BETA*TAU-GAMMA*TAU**2

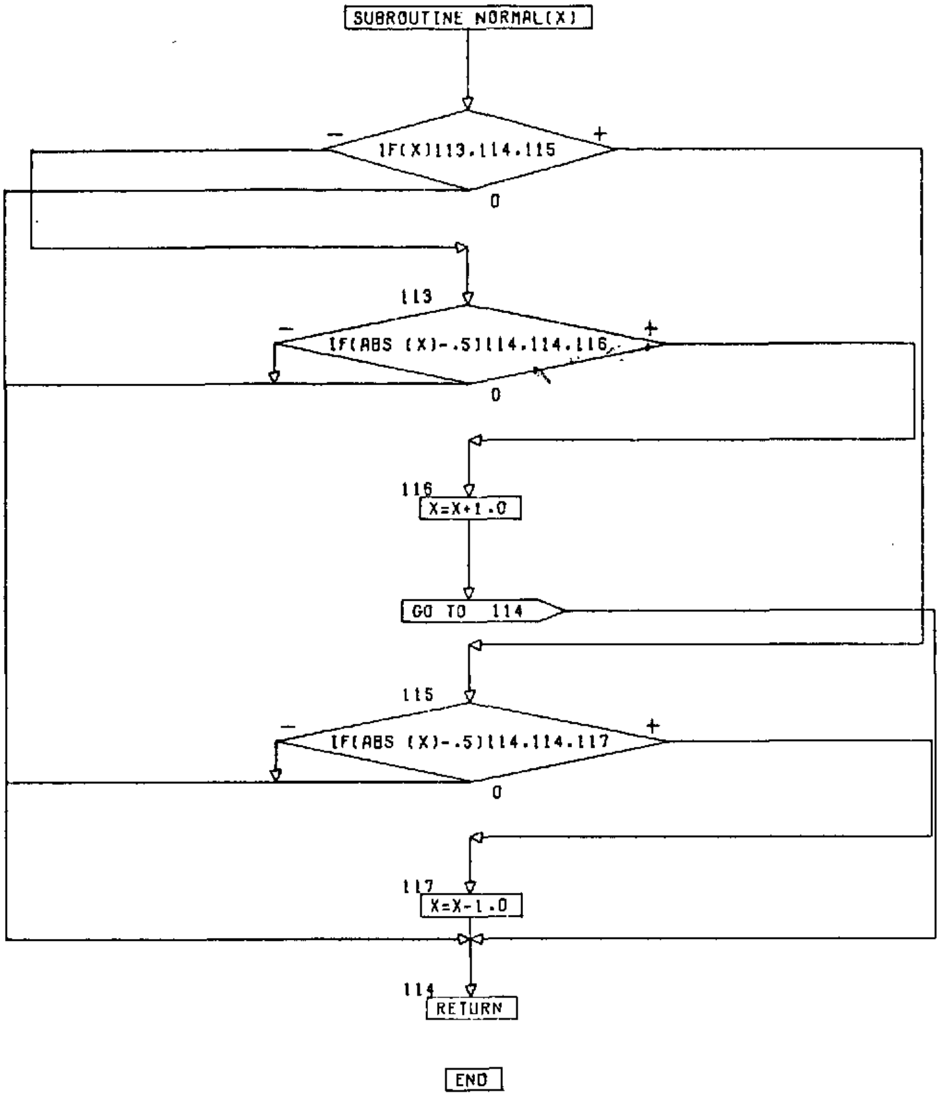

IF(R)2.3.3
−
+
0
2
R=-R
3
IF(R-SM*SIGMA)32.32.65
−
+
0
32
A(1.1)=A(1.1)+1.
A(1.2)=A(1.2)+TAU
A(1.3)=A(1.3)+TAU**2
A(1.4)=A(1.4)+RAPPA
A(2.3) =A(2.3)+TAU**3
A(2.4)=A(2.4)+RAPPA*TAU
A(3.3) =A(3.3) +TAU**4
A(3.4)=A(3.4)+RAPPA*TAU**2
DELSQ=RAPPA**2+DELSQ
65
CONTINUE
IF(A(1.1)-5)59.66.66
−
+
0
66
IF(A(1.1)-N)67.59.67
−
+
0
2
A1
2
A1
3
A2
59
ITO=A(1.1)
IOT=ITO
RETURN

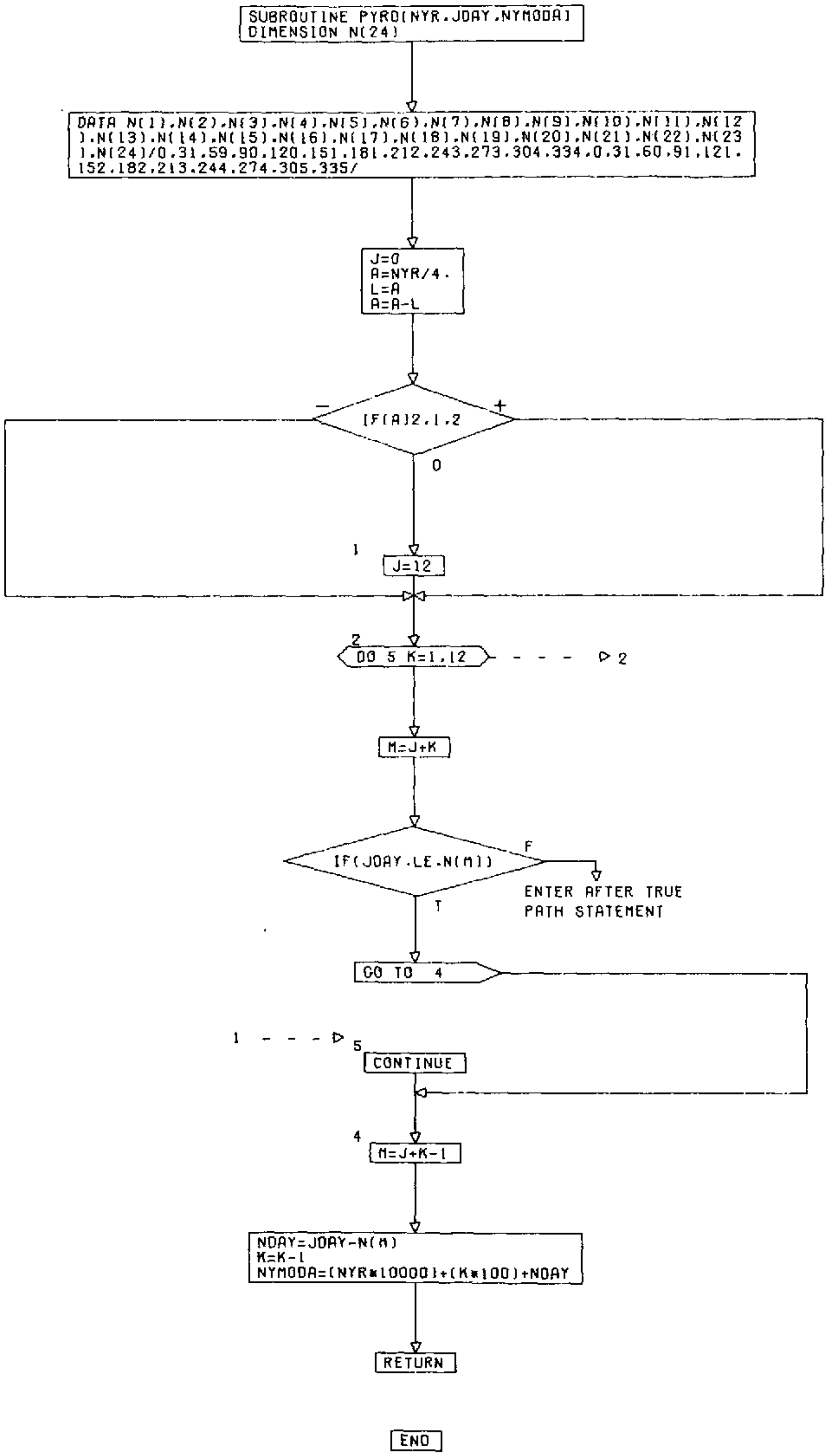


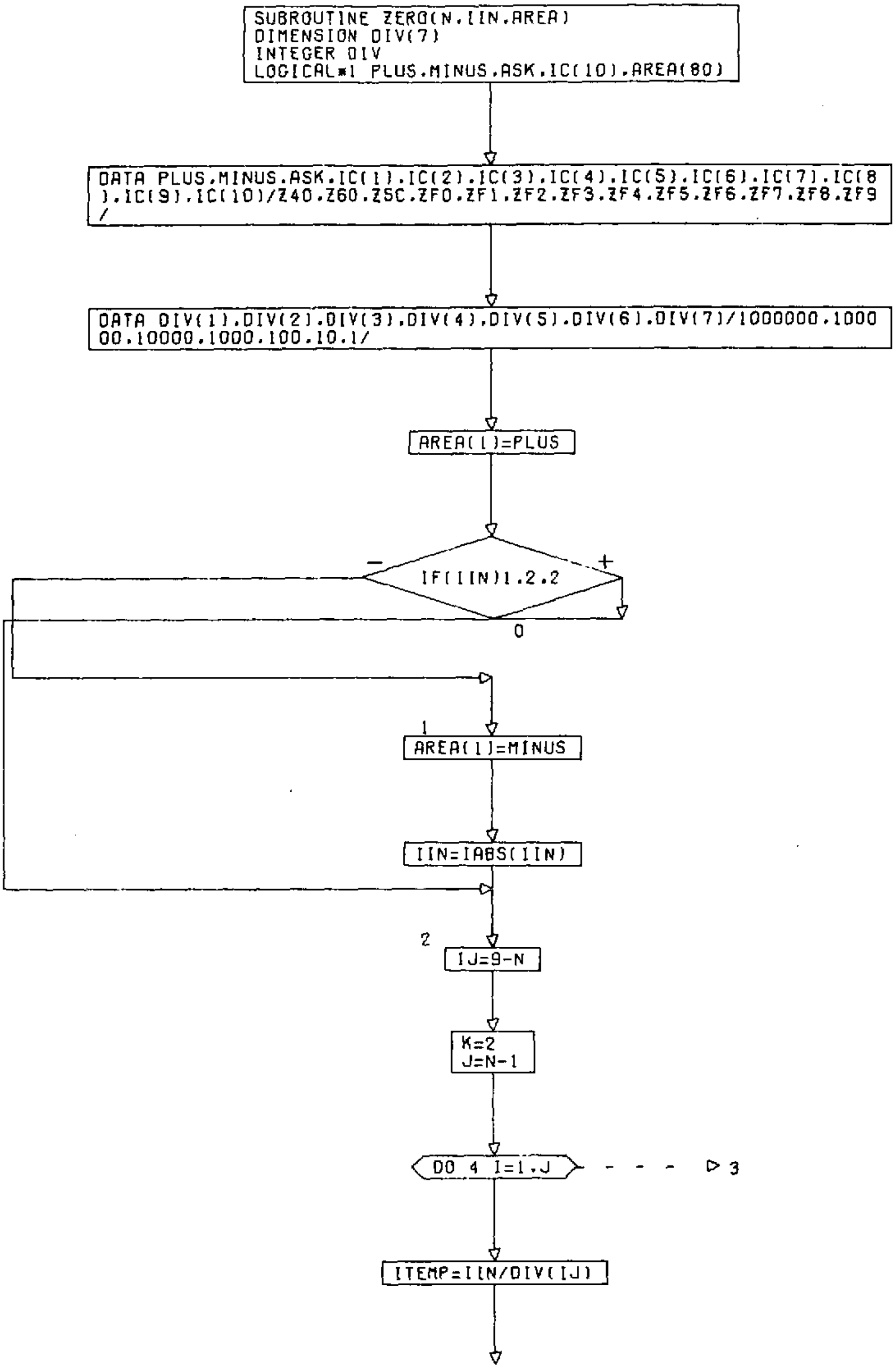
```
1
WRITE( 6.7)
7
FORMAT(30H THIS MSG. HAS EXCESSIVE NOISE)
RETURN
END
```

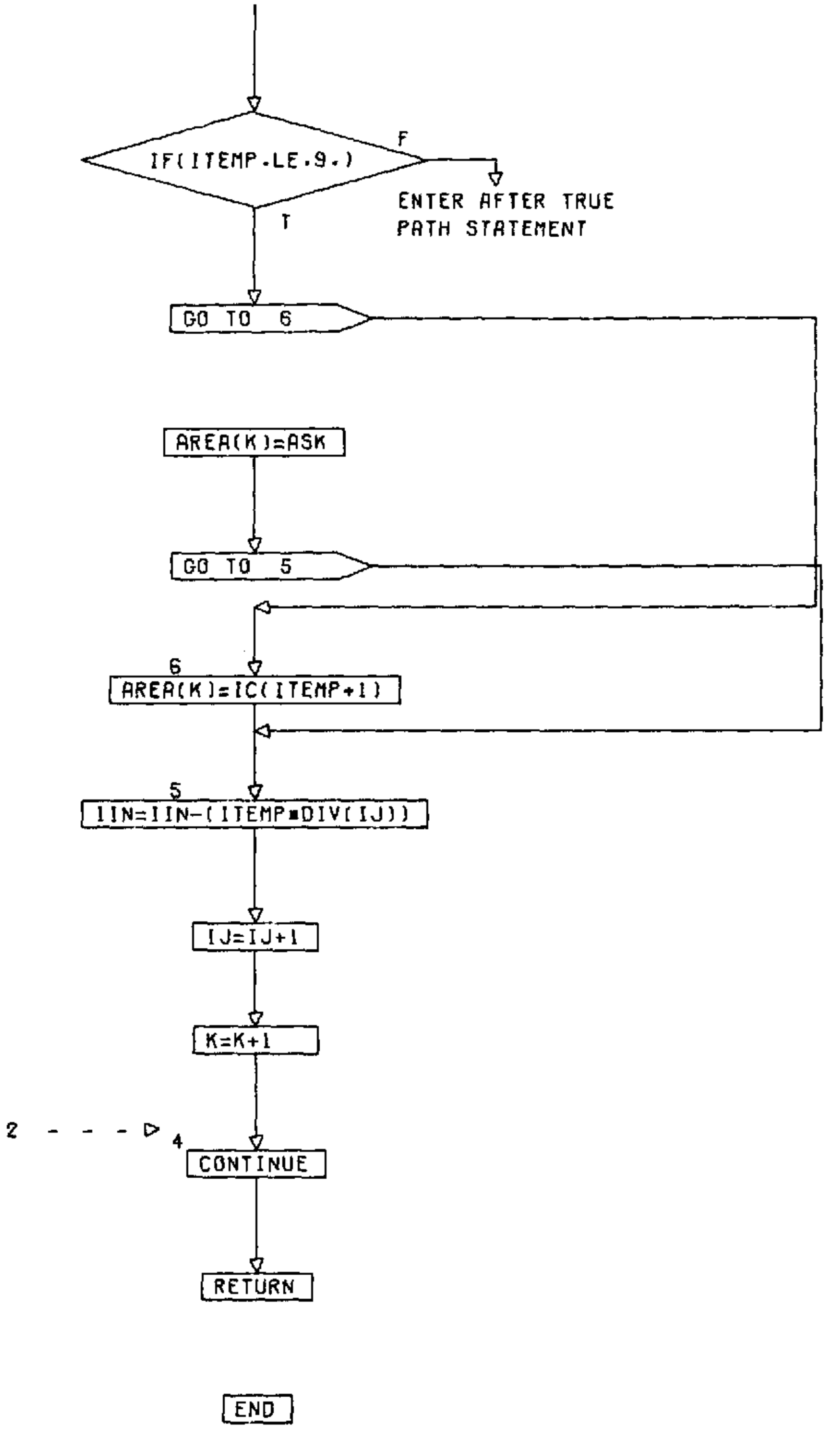

SUBROUTINE LOBASN(TEM.I.A.RATE.L)
DIMENSION A(31).TEM(31)
K=I-1
IF(RATE.GT.0.)
F
T
ENTER AFTER TRUE
PATH STATEMENT
A(1)=A(1)+1.
IF(RATE.LT.0.)
F
T
ENTER AFTER TRUE
PATH STATEMENT
A(1)=A(1)-1.
DO 10 J=1.K
3
L=0
JJ=J+1
DELTA=RATE*(TEM(JJ)-TEM(J))
A1
2
2
11
X=A(JJ)-A(J)
L=L+1
IF(L.GE.100)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 12
3
A6
IF(ABS(DELTA-X)-.500)10,10,6
−
+
0
3
A5

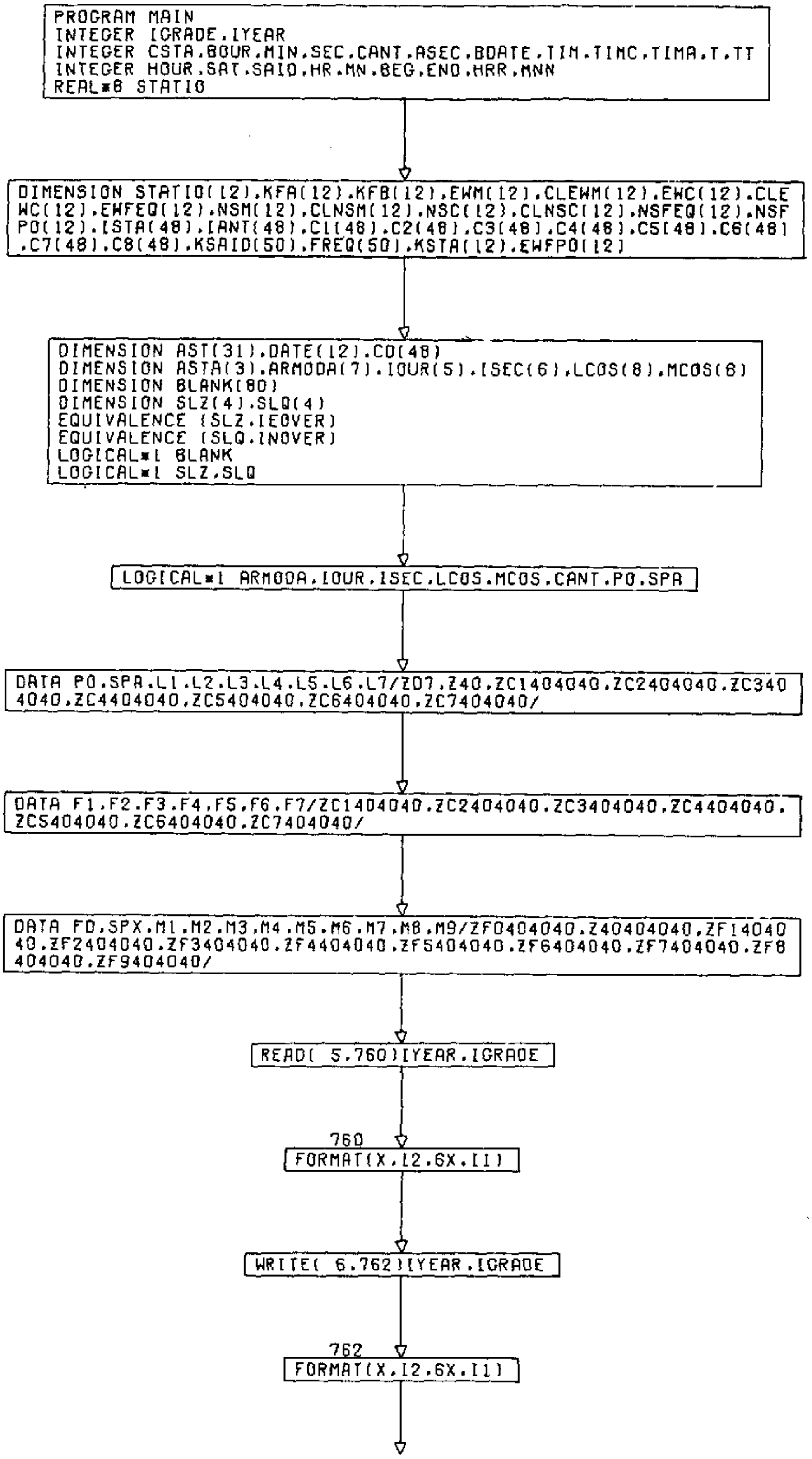

6
IF(DELTA)4.5.3
−
+
0
3
A4
A2
3
4
A(JJ)=A(JJ)-1
GO TO 11
1
A1
A3
3
3
A(JJ)=A(JJ)+1
GO TO 11
1
A1
2
A4
5
IF(X)3.10.4
−
+
2
A3
2
A2
0
2
A5
1
10
CONTINUE
2
A6
12
RETURN
END


SUBROUTINE NORMAL(X)
−
+
IF(X)113,114,115
0
113
−
+
IF(ABS (X)-.5)114,114,116
0
116
X=X+1.0
GO TO 114
115
−
+
IF(ABS (X)-.5)114,114,117
0
117
X=X-1.0
114
RETURN
END


SUBROUTINE PYRO(NYR.JDAY.NYMODA)
DIMENSION N(24)
DATA N(1).N(2).N(3).N(4).N(5).N(6).N(7).N(8).N(9).N(10).N(11).N(12
).N(13).N(14).N(15).N(16).N(17).N(18).N(19).N(20).N(21).N(22).N(23
).N(24)/0.31.59.90.120.151.181.212.243.273.304.334.0.31.60,91.121.
152,182,213.244.274.305.335/
J=0
A=NYR/4.
L=A
A=A-L
−
+
IF(A)2.1.2
0
1
J=12
2
DO 5 K=1.12
2
M=J+K
IF(JDAY.LE.N(M))
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 4
1
5
CONTINUE
4
M=J+K-1
NDAY=JDAY-N(M)
K=K-1
NYMODA=(NYR*10000)+(K*100)+NDAY
RETURN
END


SUBROUTINE ZERO(N.IIN.AREA)
DIMENSION DIV(7)
INTEGER DIV
LOGICAL*1 PLUS.MINUS.ASK.IC(10).AREA(80)
DATA PLUS.MINUS.ASK.IC(1).IC(2).IC(3).IC(4).IC(5).IC(6).IC(7).IC(8
).IC(9).IC(10)/Z40.Z60.Z5C.ZF0.ZF1.ZF2.ZF3.ZF4.ZF5.ZF6.ZF7.ZF8.ZF9
/
DATA DIV(1).DIV(2).DIV(3).DIV(4).DIV(5).DIV(6).DIV(7)/1000000.1000
00.10000.1000.100.10.1/
AREA(1)=PLUS
IF(IIN)1.2.2
-
+
0
1
AREA(1)=MINUS
IIN=IABS(IIN)
2
IJ=9-N
K=2
J=N-1
DO 4 I=1.J
3
ITEMP=IIN/DIV(IJ)


IF(ITEMP.LE.9.)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 6
AREA(K)=ASK
GO TO 5
6
AREA(K)=IC(ITEMP+1)
5
IIN=IIN-(ITEMP*DIV(IJ))
IJ=IJ+1
K=K+1
2
4
CONTINUE
RETURN
END



```
PROGRAM MAIN
INTEGER IGRADE.IYEAR
INTEGER CSTA.BOUR.MIN.SEC.CANT.ASEC.BDATE.TIM.TIMC.TIMA.T.TT
INTEGER HOUR.SAT.SAIO.HR.MN.BEG.END.HRR.MNN
REAL*8 STATIO
```

```
DIMENSION STATIO(12).KFA(12).KFB(12).EWM(12).CLEWM(12).EWC(12).CLE
WC(12).EWFEQ(12).NSM(12).CLNSM(12).NSC(12).CLNSC(12).NSFEQ(12).NSF
PO(12).ISTA(48).IANT(48).C1(48).C2(48).C3(48).C4(48).C5(48).C6(48)
.C7(48).C8(48).KSAID(50).FREQ(50).KSTA(12).EWFPO(12)
```

```
DIMENSION AST(31).DATE(12).CD(48)
DIMENSION ASTA(3).ARMODA(7).IOUR(5).ISEC(6).LCOS(8).MCOS(8)
DIMENSION BLANK(80)
DIMENSION SLZ(4).SLQ(4)
EQUIVALENCE (SLZ.IEOVER)
EQUIVALENCE (SLQ.INOVER)
LOGICAL*1 BLANK
LOGICAL*1 SLZ.SLQ
```

```
LOGICAL*1 ARMODA.IOUR.ISEC.LCOS.MCOS.CANT.PO.SPA
```

```
DATA PO.SPA.L1.L2.L3.L4.L5.L6.L7/Z07.Z40.ZC1404040.ZC2404040.ZC340
4040.ZC4404040.ZC5404040.ZC6404040.ZC7404040/
```

```
DATA F1.F2.F3.F4.F5.F6.F7/ZC1404040.ZC2404040.ZC3404040.ZC4404040.
ZC5404040.ZC6404040.ZC7404040/
```

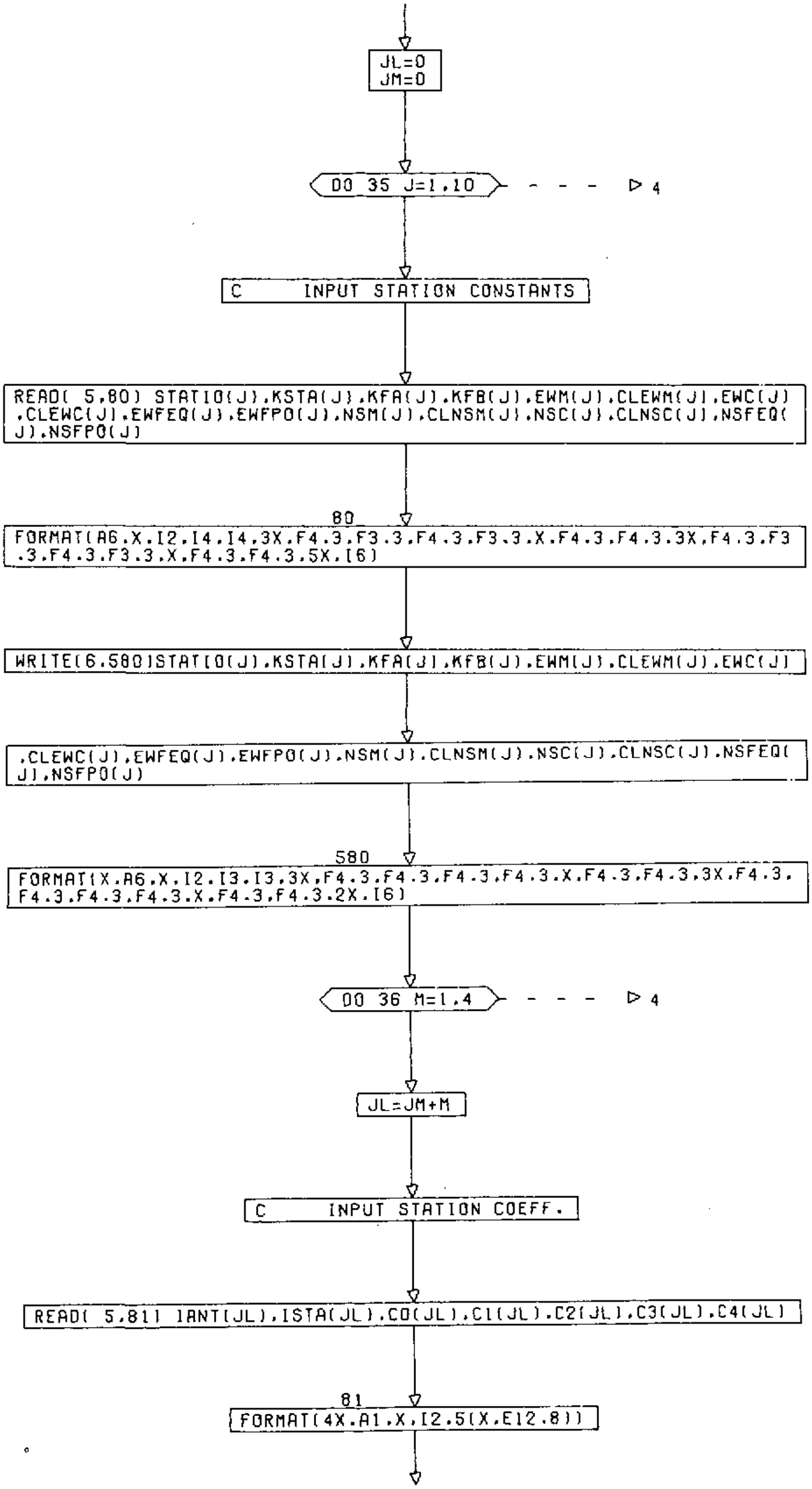
```
DATA F0.SPX.M1.M2.M3.M4.M5.M6.M7.M8.M9/ZF0404040.Z40404040.ZF14040
40.ZF2404040.ZF3404040.ZF4404040.ZF5404040.ZF6404040.ZF7404040.ZF8
404040.ZF9404040/
```

```
READ( 5.760)IYEAR.IGRADE
```

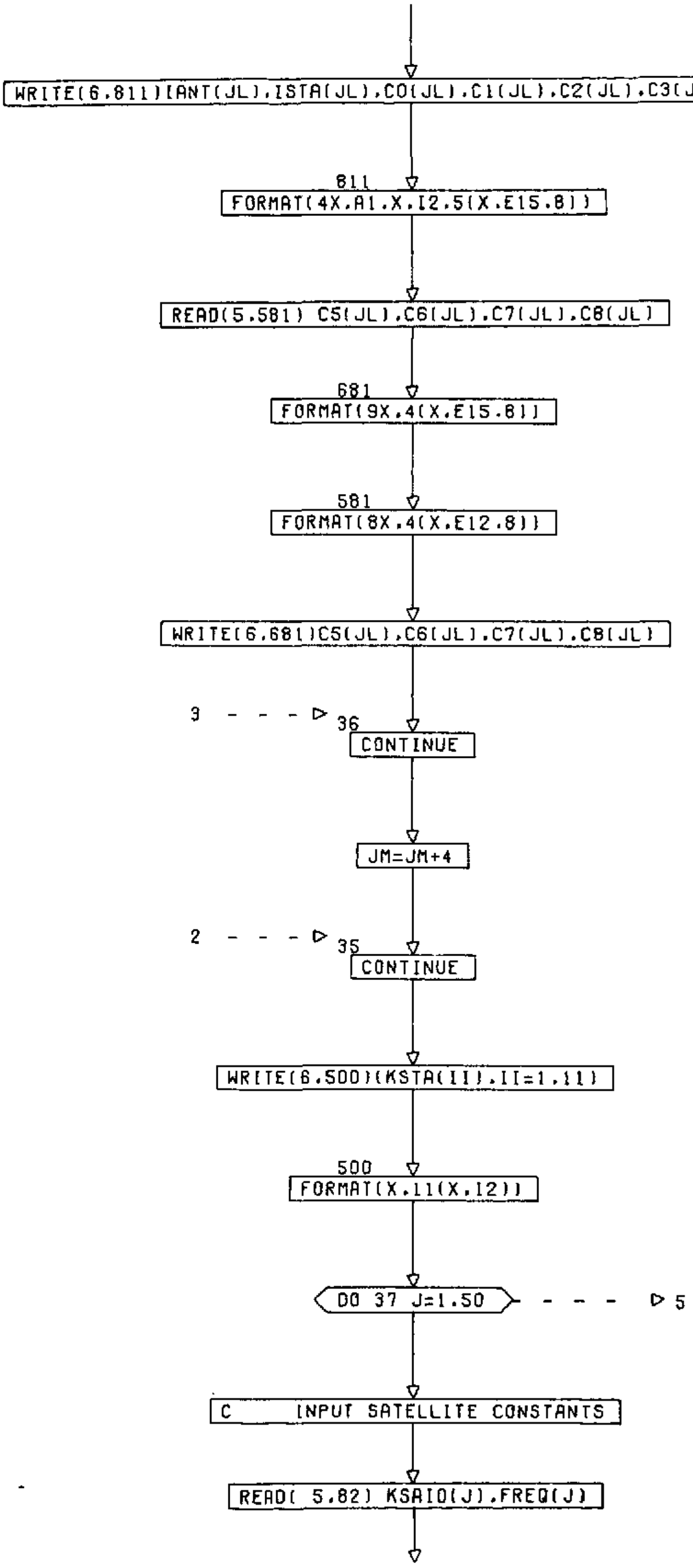
```
760
FORMAT(X.I2.6X.I1)
```

```
WRITE( 6.762)IYEAR.IGRADE
```

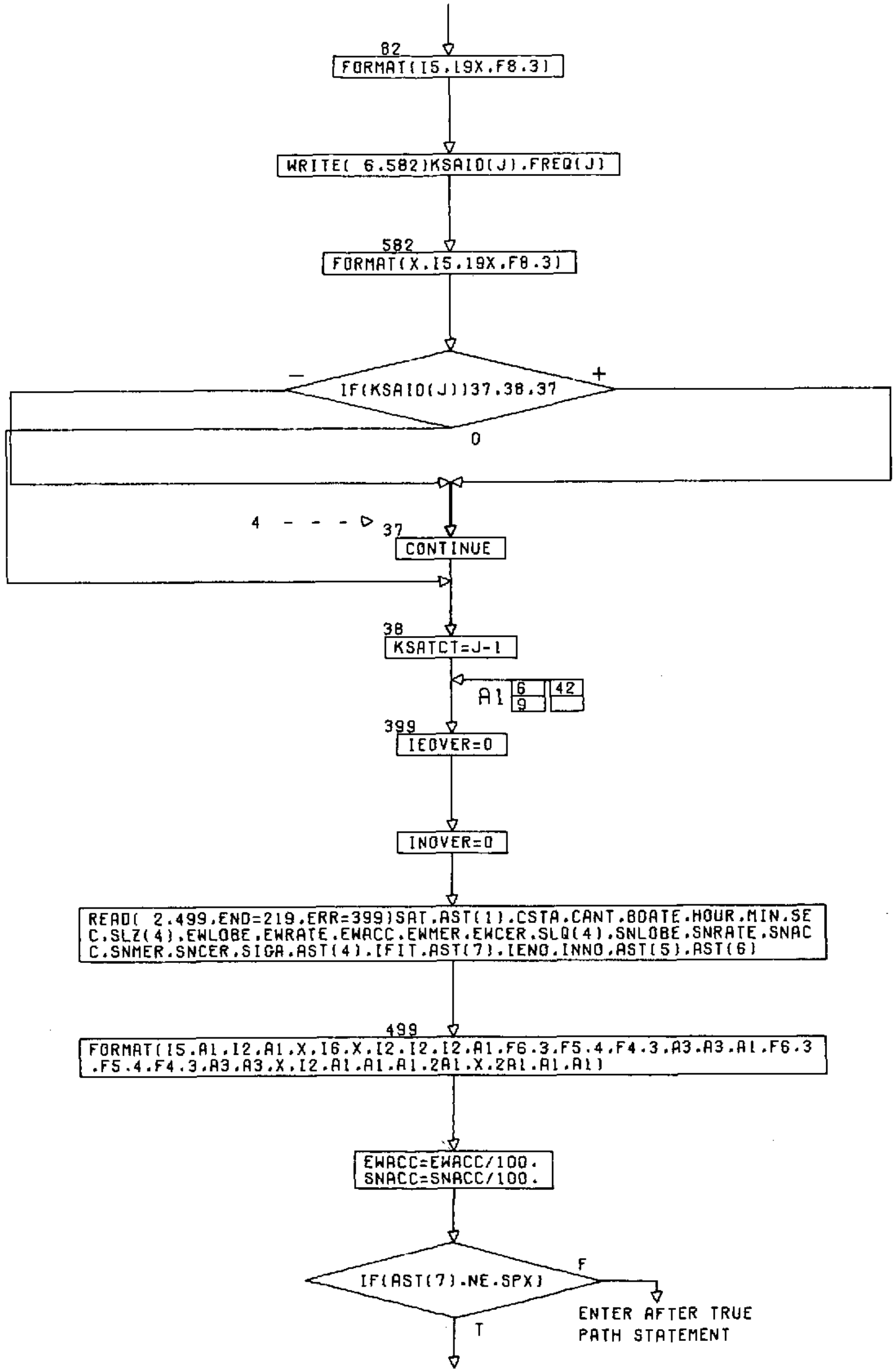
```
762
FORMAT(X.I2.6X.I1)
```

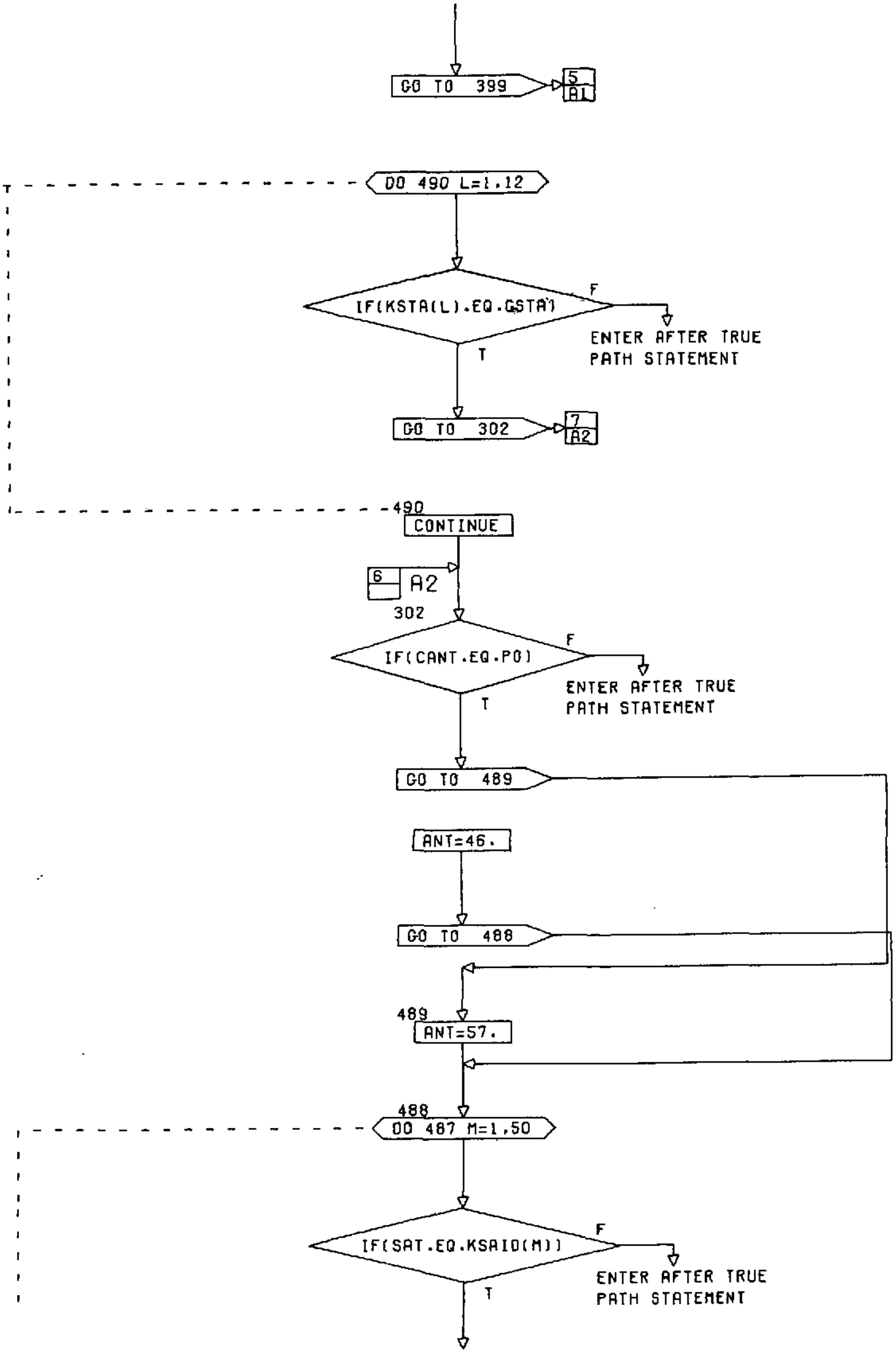


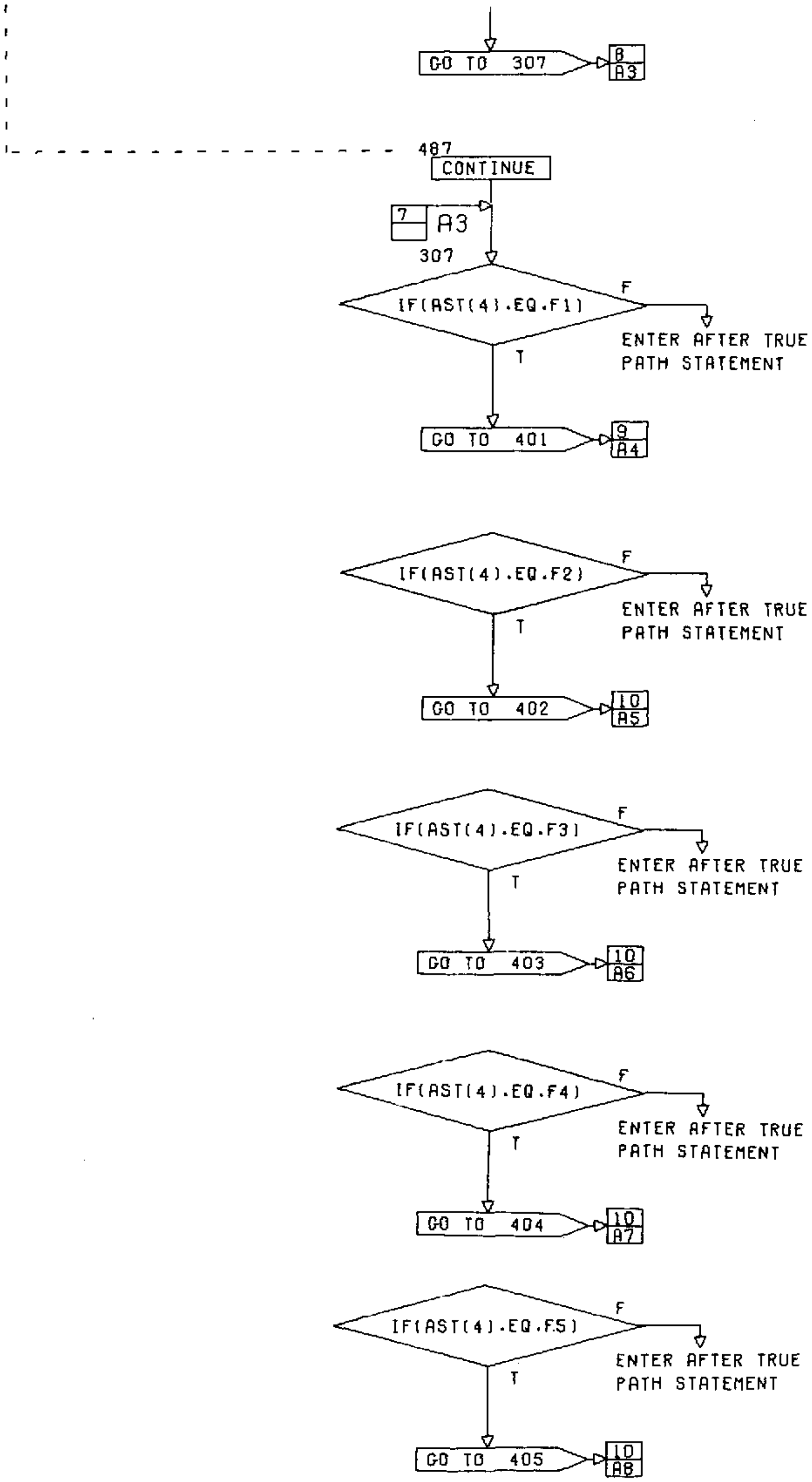
```
JL=0
JM=0

DO 35 J=1,10

C      INPUT STATION CONSTANTS

READ( 5,80) STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)
,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
J),NSFPO(J)

80
FORMAT(A6,X,I2,I4,I4,3X,F4.3,F3.3,F4.3,F3.3,X,F4.3,F4.3,3X,F4.3,F3
.3,F4.3,F3.3,X,F4.3,F4.3,5X,I6)

WRITE(6,580)STATIO(J),KSTA(J),KFA(J),KFB(J),EWM(J),CLEWM(J),EWC(J)

,CLEWC(J),EWFEQ(J),EWFPO(J),NSM(J),CLNSM(J),NSC(J),CLNSC(J),NSFEQ(
J),NSFPO(J)

580
FORMAT(X,A6,X,I2,I3,I3,3X,F4.3,F4.3,F4.3,F4.3,X,F4.3,F4.3,3X,F4.3,
F4.3,F4.3,F4.3,X,F4.3,F4.3,2X,I6)

DO 36 M=1,4

JL=JM+M

C      INPUT STATION COEFF.

READ( 5,81) IANT(JL),ISTA(JL),C0(JL),C1(JL),C2(JL),C3(JL),C4(JL)

81
FORMAT(4X,A1,X,I2,5(X,E12.8))
```

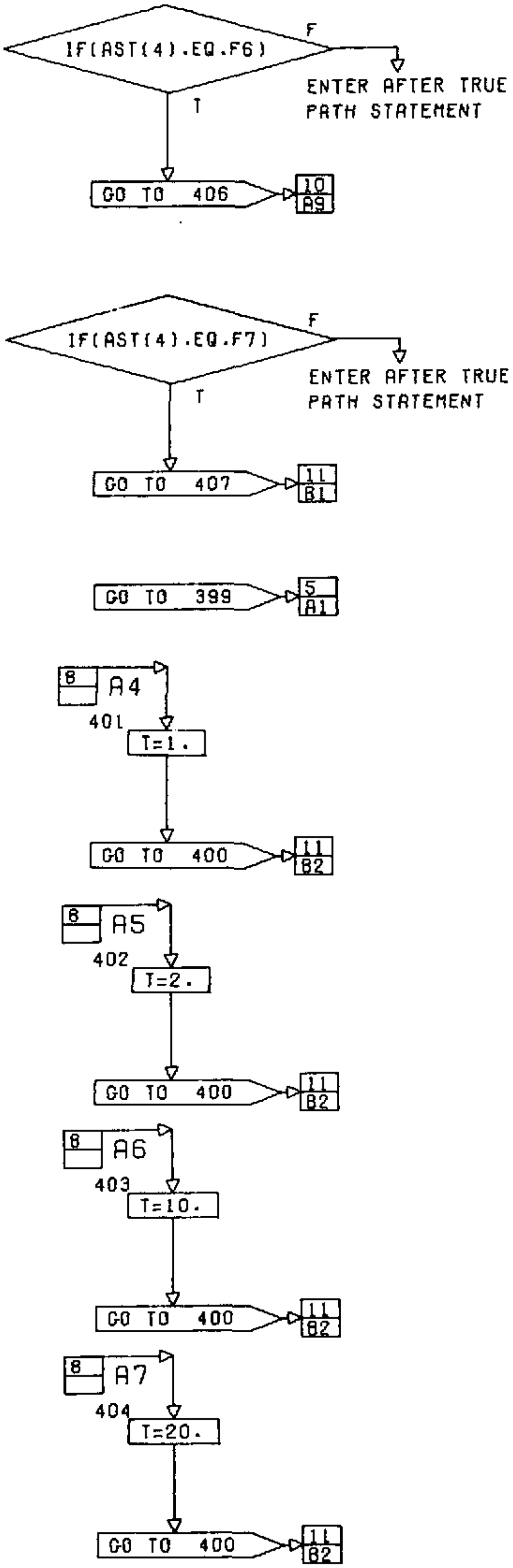


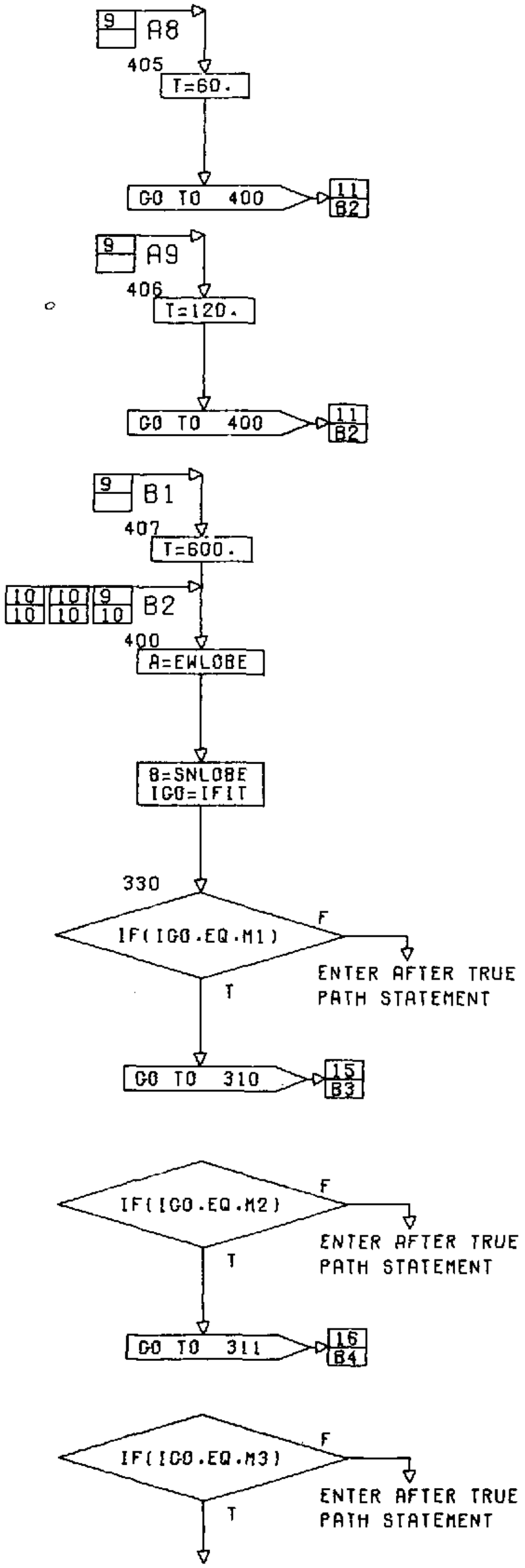
```
WRITE(6.811)IANT(JL),ISTA(JL),CO(JL).C1(JL).C2(JL).C3(JL).C4(JL)
811 FORMAT(4X.A1.X.I2.5(X.E15.8))
READ(5.581) C5(JL).C6(JL).C7(JL).C8(JL)
681 FORMAT(9X.4(X.E15.8))
581 FORMAT(8X.4(X.E12.8))
WRITE(6.681)C5(JL).C6(JL).C7(JL).C8(JL)
3 - - - ▷ 36 CONTINUE
JM=JM+4
2 - - - ▷ 35 CONTINUE
WRITE(6.500)(KSTA(II).II=1.11)
500 FORMAT(X.11(X.I2))
DO 37 J=1.50 - - - ▷ 5
C       INPUT SATELLITE CONSTANTS
READ( 5.82) KSAIO(J).FREQ(J)
```

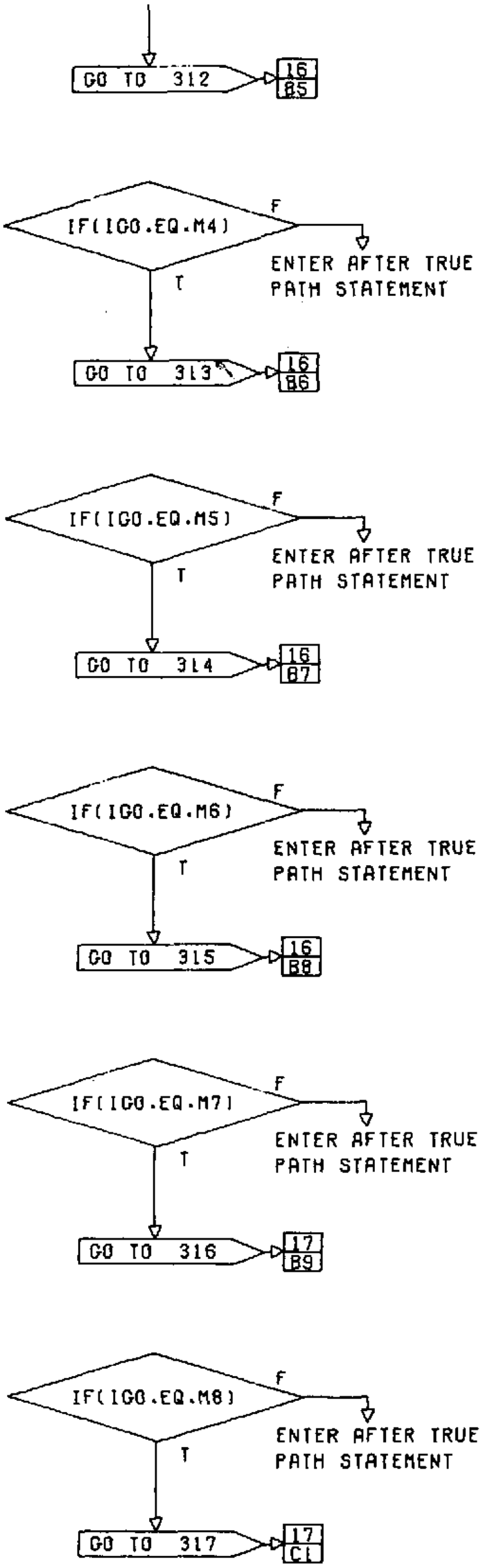

82
FORMAT(I5,19X,F8.3)
WRITE(6.582)KSAID(J).FREQ(J)
582
FORMAT(X.I5.19X.F8.3)
IF(KSAID(J))37.38.37
−
+
0
4
37
CONTINUE
38
KSATCT=J-1
A1
6
9
42
399
IEOVER=0
INOVER=0
READ(2.499.END=219.ERR=399)SAT.AST(1).CSTA.CANT.BDATE.HOUR.MIN.SEC.SLZ(4).EWLOBE.EWRATE.EWACC.EWMER.EWCER.SLQ(4).SNLOBE.SNRATE.SNACC.SNMER.SNCER.SIGA.AST(4).IFIT.AST(7).IENO.INNO,AST(5).AST(6)
499
FORMAT(I5.A1.I2.A1.X.I6.X.I2.I2.I2.A1.F6.3.F5.4.F4.3.A3.A3.A1.F6.3.F5.4.F4.3.A3.A3.X.I2.A1.A1.A1.2A1.X.2A1.A1.A1)
EWACC=EWACC/100.
SNACC=SNACC/100.
IF(AST(7).NE.SPX)
F
T
ENTER AFTER TRUE
PATH STATEMENT

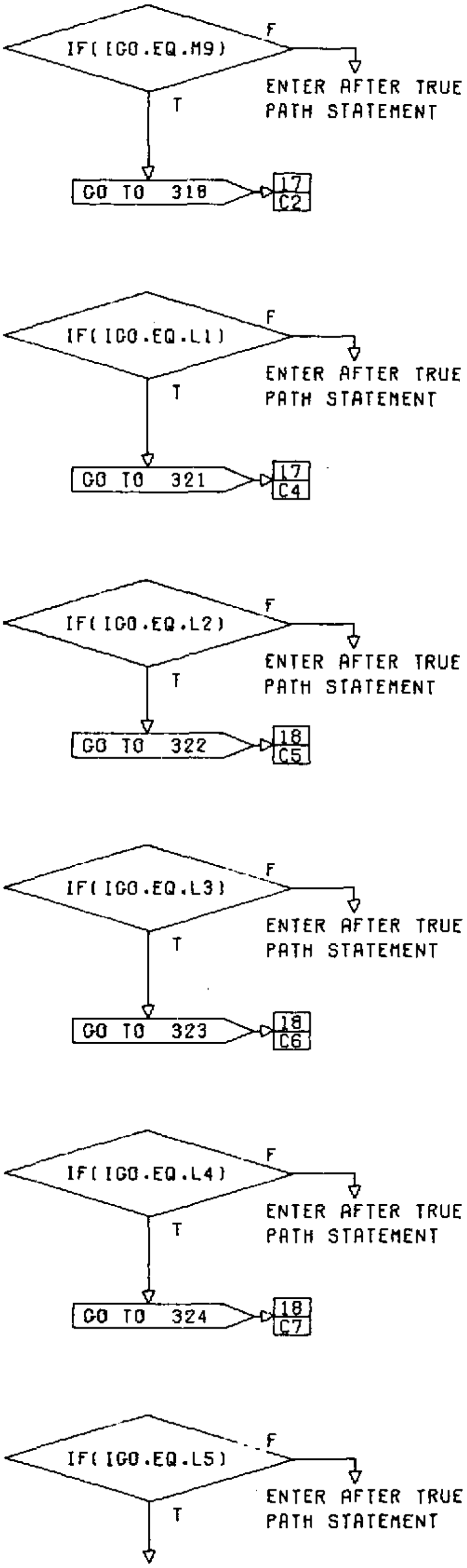

GO TO 399
5
A1
DO 490 L=1.12
T
IF(KSTA(L).EQ.GSTA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 302
7
A2
490
CONTINUE
6
A2
302
IF(CANT.EQ.PO)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 489
ANT=46.
GO TO 488
489
ANT=57.
488
DO 487 M=1,50
IF(SAT.EQ.KSAID(M))
F
ENTER AFTER TRUE
PATH STATEMENT
T

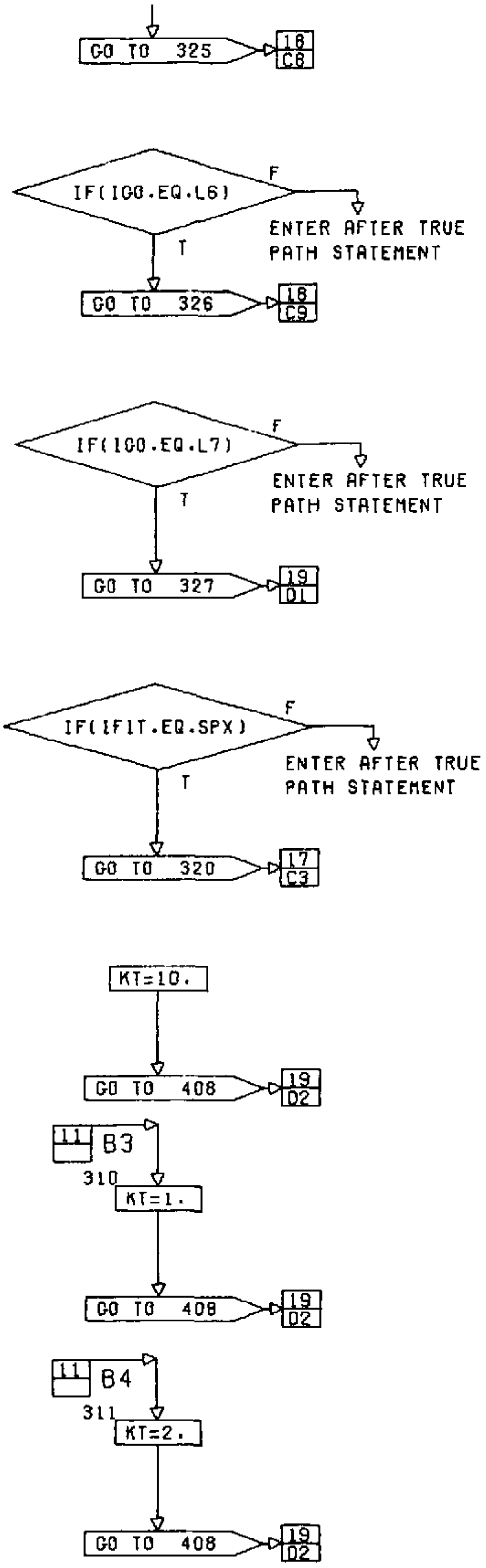

GO TO 307
8
A3
487
CONTINUE
7
A3
307
IF(AST(4).EQ.F1)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 401
9
A4
IF(AST(4).EQ.F2)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 402
10
A5
IF(AST(4).EQ.F3)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 403
10
A6
IF(AST(4).EQ.F4)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 404
10
A7
IF(AST(4).EQ.F5)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 405
10
A8

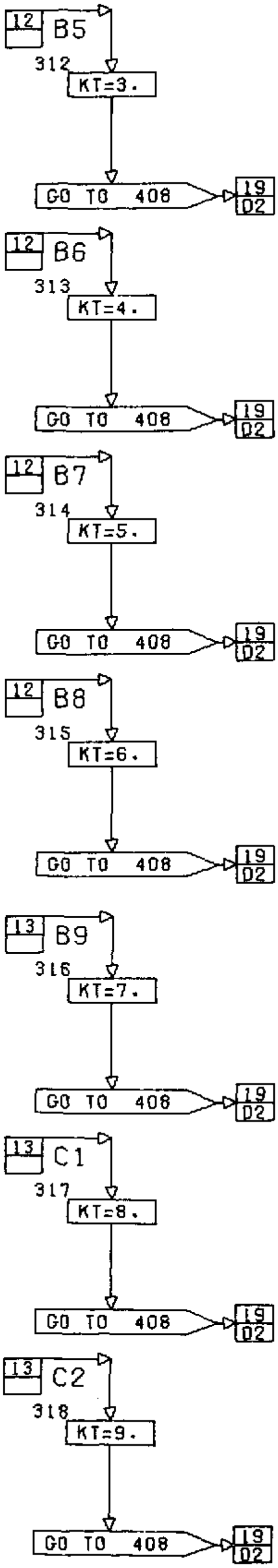

IF(AST(4).EQ.F6)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 406
10
A9
IF(AST(4).EQ.F7)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 407
11
B1
GO TO 399
5
A1
8
A4
401
T=1.
GO TO 400
11
B2
8
A5
402
T=2.
GO TO 400
11
B2
8
A6
403
T=10.
GO TO 400
11
B2
8
A7
404
T=20.
GO TO 400
11
B2

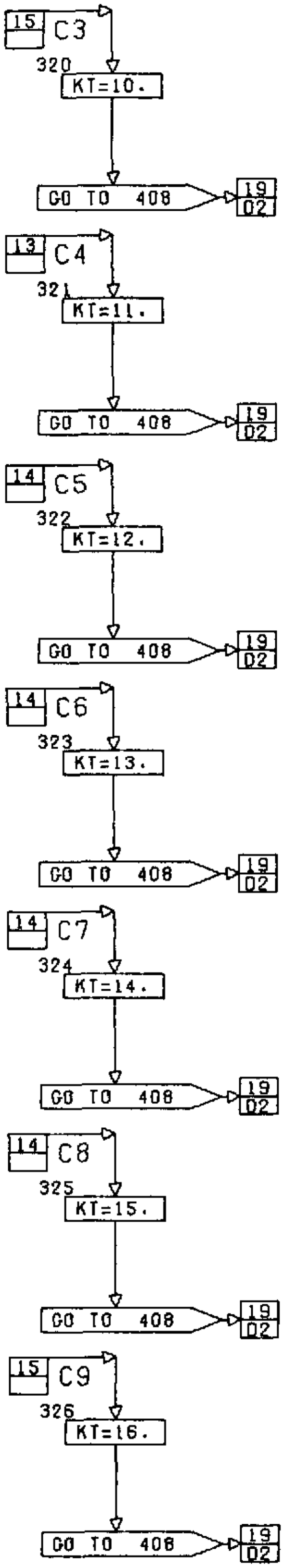

9
A8
405
T=60.
GO TO 400
11
B2
9
A9
406
T=120.
GO TO 400
11
B2
9
B1
407
T=600.
10
10
10
10
9
10
B2
400
A=EWLOBE
B=SNLOBE
IGO=IFIT
330
IF(IGO.EQ.M1)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 310
15
B3
IF(IGO.EQ.M2)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 311
16
B4
IF(IGO.EQ.M3)
F
ENTER AFTER TRUE
PATH STATEMENT
T

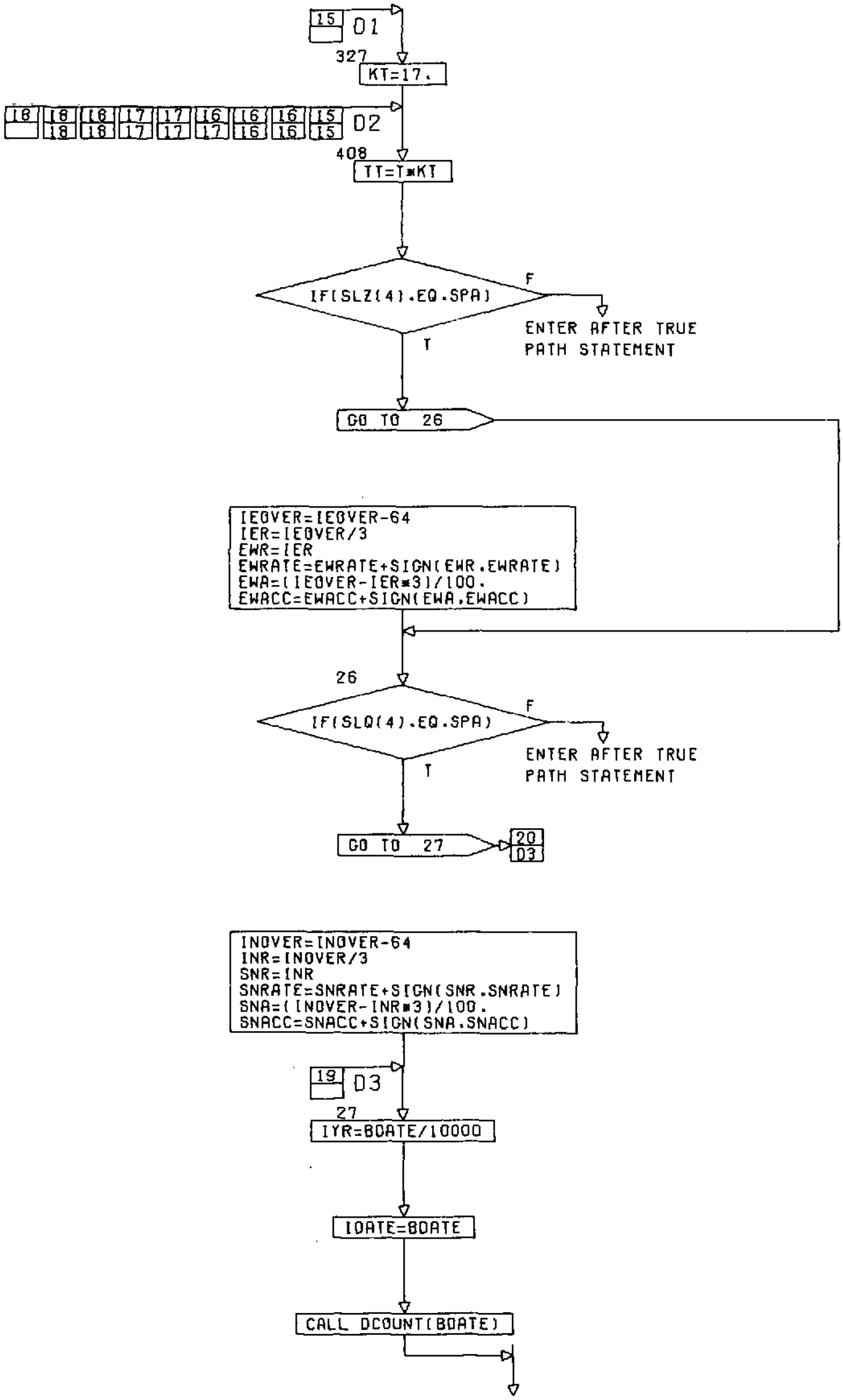

GO TO 312
16
B5
IF(IGO.EQ.M4)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 313
16
B6
IF(IGO.EQ.M5)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 314
16
B7
IF(IGO.EQ.M6)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 315
16
B8
IF(IGO.EQ.M7)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 316
17
B9
IF(IGO.EQ.M8)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 317
17
C1

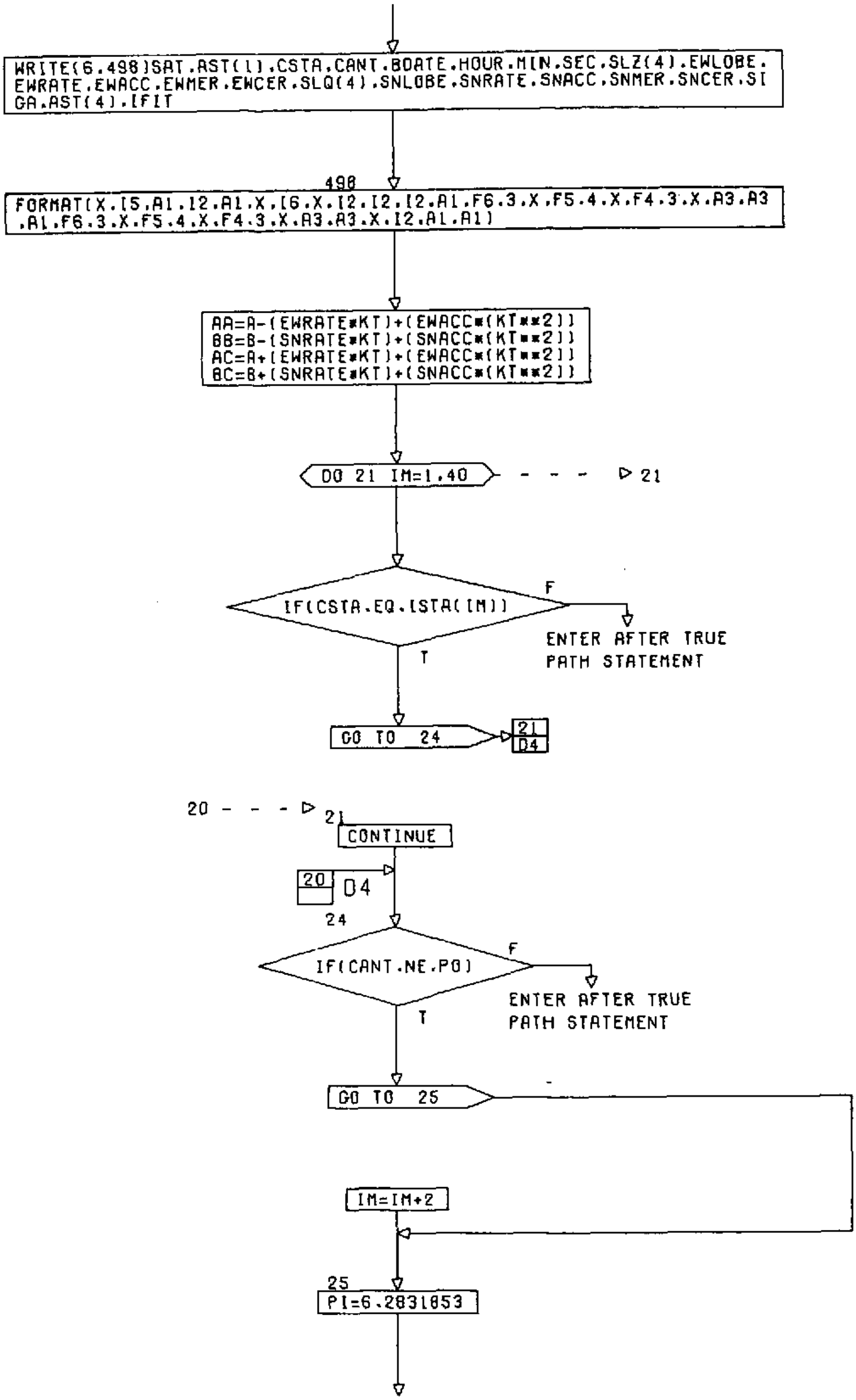

IF(IGO.EQ.M9)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 318
17
C2
IF(IGO.EQ.L1)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 321
17
C4
IF(IGO.EQ.L2)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 322
18
C5
IF(IGO.EQ.L3)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 323
18
C6
IF(IGO.EQ.L4)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 324
18
C7
IF(IGO.EQ.L5)
F
ENTER AFTER TRUE
PATH STATEMENT
T


GO TO 325
18
C8
IF(IGO.EQ.L6)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 326
18
C9
IF(IGO.EQ.L7)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 327
19
D1
IF(IFIT.EQ.SPX)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 320
17
C3
KT=10.
GO TO 408
19
D2
11
B3
310
KT=1.
GO TO 408
19
D2
11
B4
311
KT=2.
GO TO 408
19
D2

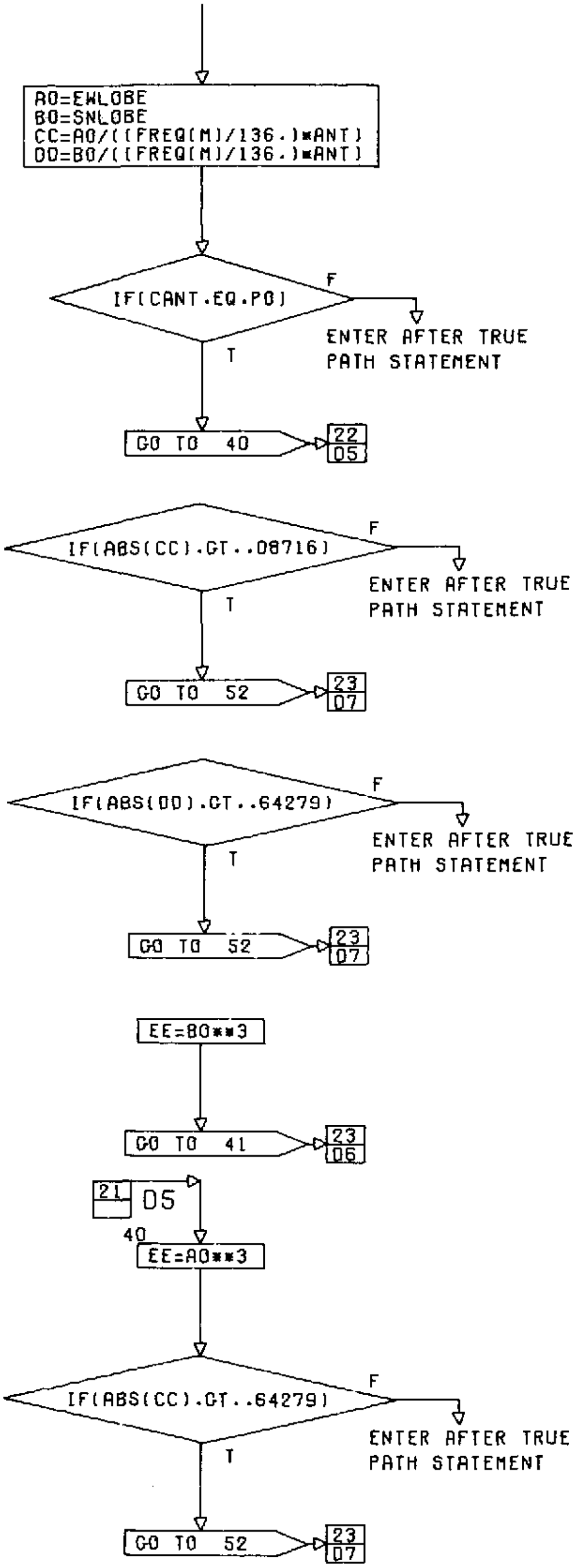

12
B5
312
KT=3.
GO TO 408
19
D2
12
B6
313
KT=4.
GO TO 408
19
D2
12
B7
314
KT=5.
GO TO 408
19
D2
12
B8
315
KT=6.
GO TO 408
19
D2
13
B9
316
KT=7.
GO TO 408
19
D2
13
C1
317
KT=8.
GO TO 408
19
D2
13
C2
318
KT=9.
GO TO 408
19
D2

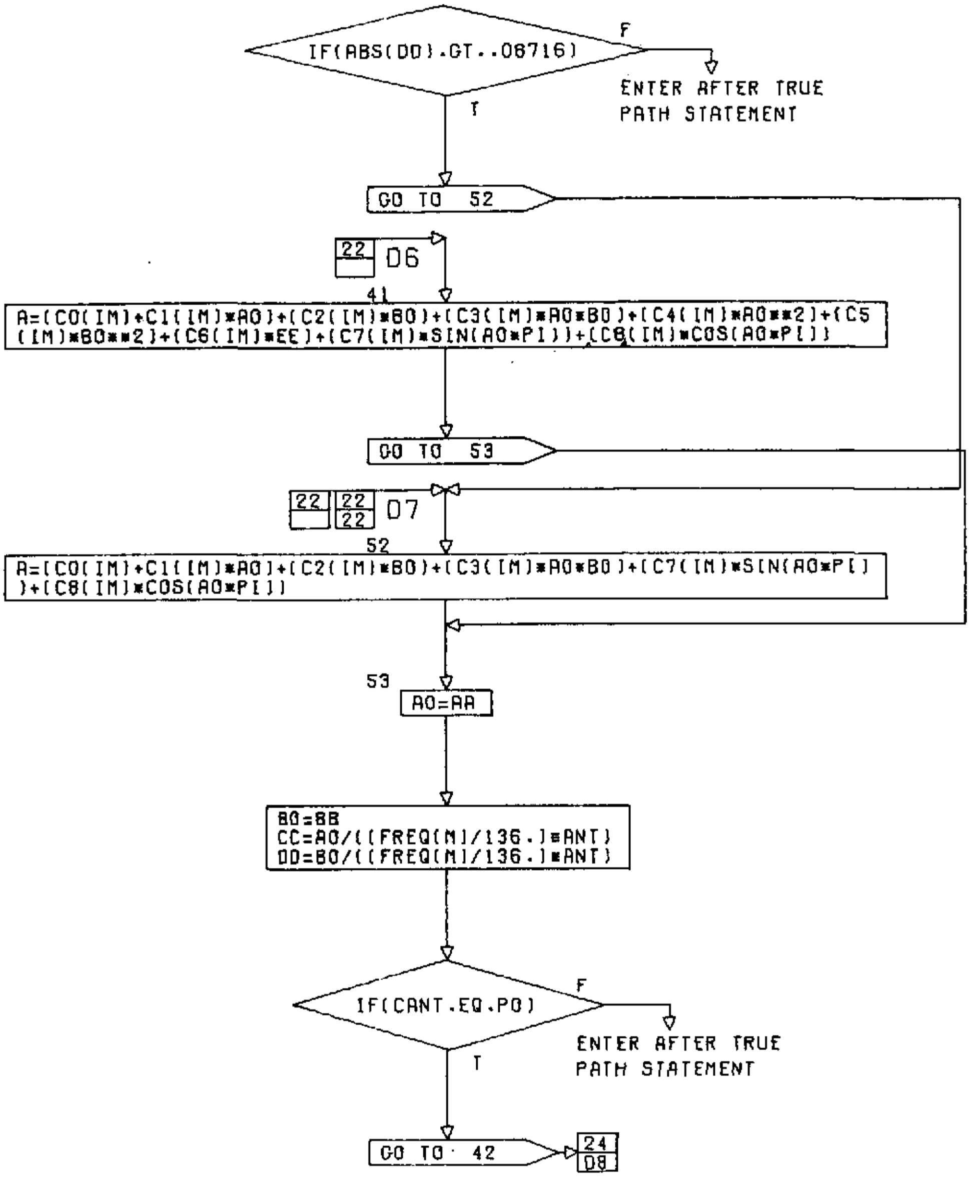

15
C3
320
KT=10.
GO TO 408
19
02
13
C4
321
KT=11.
GO TO 408
19
02
14
C5
322
KT=12.
GO TO 408
19
02
14
C6
323
KT=13.
GO TO 408
19
02
14
C7
324
KT=14.
GO TO 408
19
02
14
C8
325
KT=15.
GO TO 408
19
02
15
C9
326
KT=16.
GO TO 408
19
02

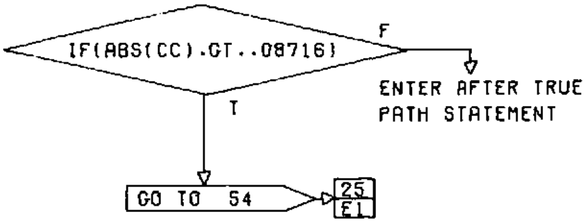

15
D1
327
KT=17.
18 18 18 17 17 16 16 16 15
18 18 17 17 17 16 16 15
D2
408
TT=T*KT
IF(SLZ(4).EQ.SPA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 26
IEOVER=IEOVER-64
IER=IEOVER/3
EWR=IER
EWRATE=EWRATE+SIGN(EWR.EWRATE)
EWA=(IEOVER-IER*3)/100.
EWACC=EWACC+SIGN(EWA.EWACC)
26
IF(SLQ(4).EQ.SPA)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 27
20
D3
INOVER=INOVER-64
INR=INOVER/3
SNR=INR
SNRATE=SNRATE+SIGN(SNR.SNRATE)
SNA=(INOVER-INR*3)/100.
SNACC=SNACC+SIGN(SNA.SNACC)
19
D3
27
IYR=BDATE/10000
IDATE=BDATE
CALL DCOUNT(BDATE)



```
WRITE(6.498)SAT.AST(l).CSTA.CANT.BOATE.HOUR.MIN.SEC.SLZ(4).EWLOBE.
EWRATE.EWACC.EWMER.EWCER.SLQ(4).SNLOBE.SNRATE.SNACC.SNMER.SNCER.SI
GA.AST(4).IFIT

498
FORMAT(X.I5.A1.I2.A1.X.I6.X.I2.I2.I2.A1.F6.3.X.F5.4.X.F4.3.X.A3.A3
.A1.F6.3.X.F5.4.X.F4.3.X.A3.A3.X.I2.A1.A1)

AA=A-(EWRATE*KT)+(EWACC*(KT**2))
BB=B-(SNRATE*KT)+(SNACC*(KT**2))
AC=A+(EWRATE*KT)+(EWACC*(KT**2))
BC=B+(SNRATE*KT)+(SNACC*(KT**2))

DO 21 IM=1.40

IF(CSTA.EQ.ISTA(IM))

GO TO 24

21
CONTINUE

24
IF(CANT.NE.PO)

GO TO 25

IM=IM+2

25
PI=6.2831853
```

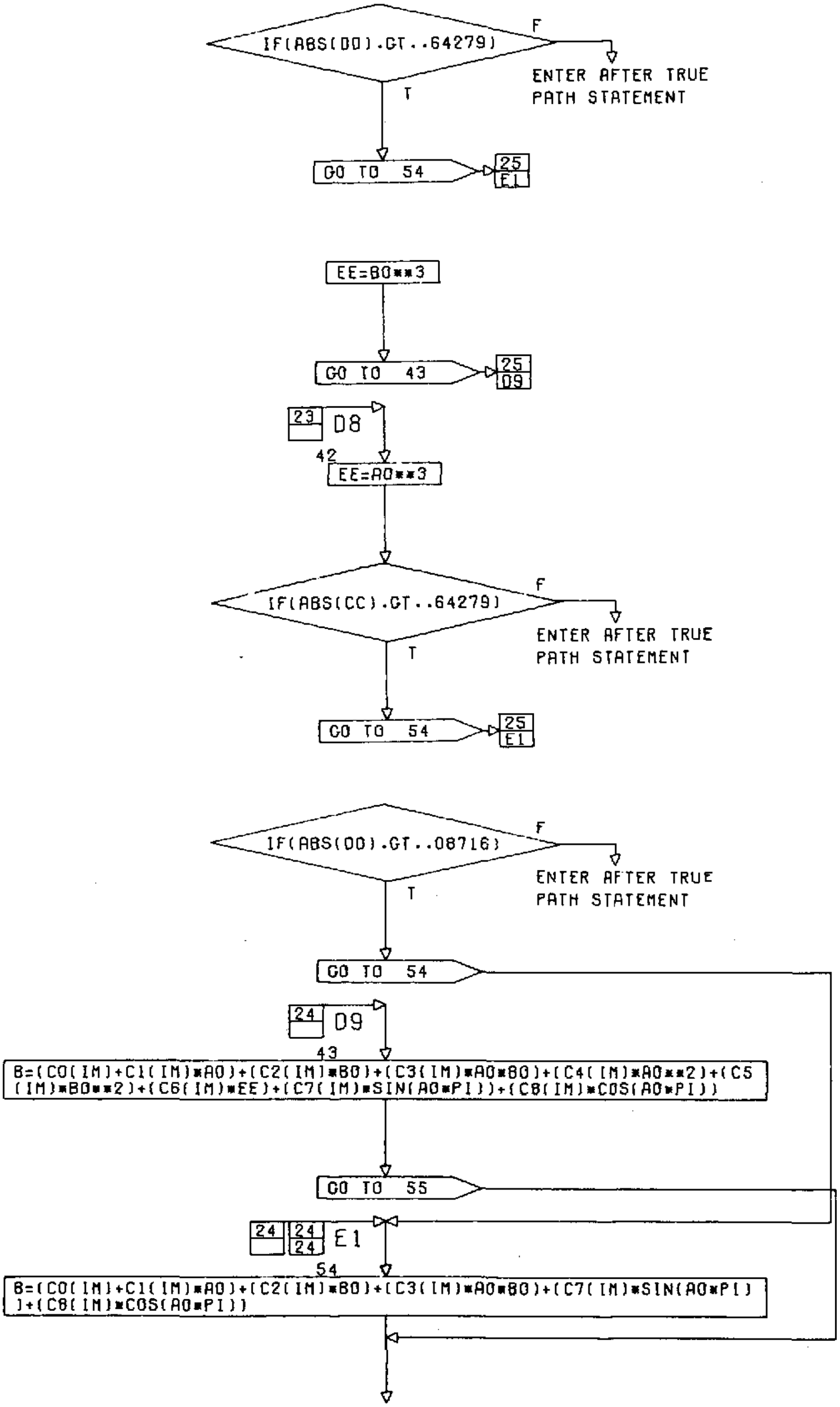

A0=EWLOBE
B0=SNLOBE
CC=A0/((FREQ(M)/136.)*ANT)
DD=B0/((FREQ(M)/136.)*ANT)
IF(CANT.EQ.P0)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 40
22
05
IF(ABS(CC).GT..08716)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 52
23
07
IF(ABS(DD).GT..64279)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 52
23
07
EE=B0**3
GO TO 41
23
06
21
05
40
EE=A0**3
IF(ABS(CC).GT..64279)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 52
23
07

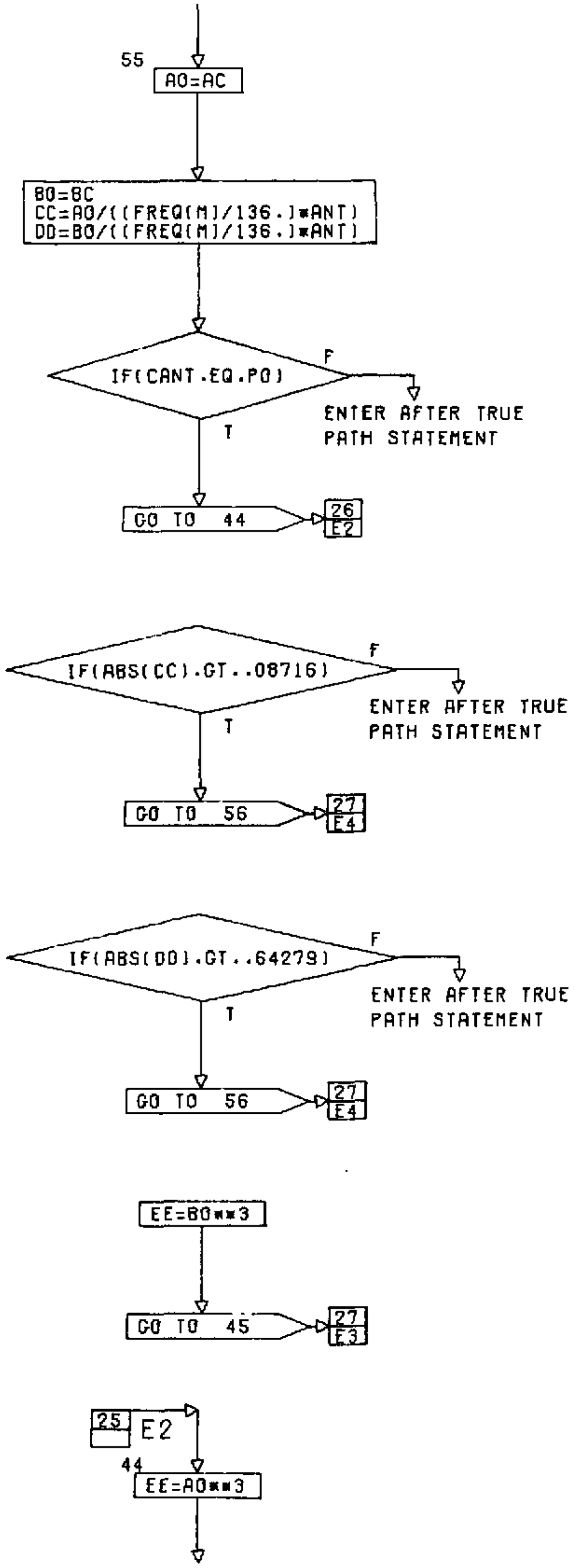

IF(ABS(DO).GT..08716)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 52
22
D6
41
A=(CO(IM)+C1((M)*AO)+(C2((M)*BO)+(C3((M)*AO*BO)+(C4((M)*AO**2)+(C5
((M)*BO**2)+(C6((M)*EE)+(C7((M)*SIN(AO*PI))+(C8((M)*COS(AO*PI))
GO TO 53
22
22
22
D7
52
A=(CO(IM)+C1((M)*AO)+(C2((M)*BO)+(C3((M)*AO*BO)+(C7((M)*SIN(AO*PI)
)+(C8(IM)*COS(AO*PI))
53
AO=AA
BO=BB
CC=AO/((FREQ(M)/136.)*ANT)
DD=BO/((FREQ(M)/136.)*ANT)
IF(CANT.EQ.PO)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 42
24
D8

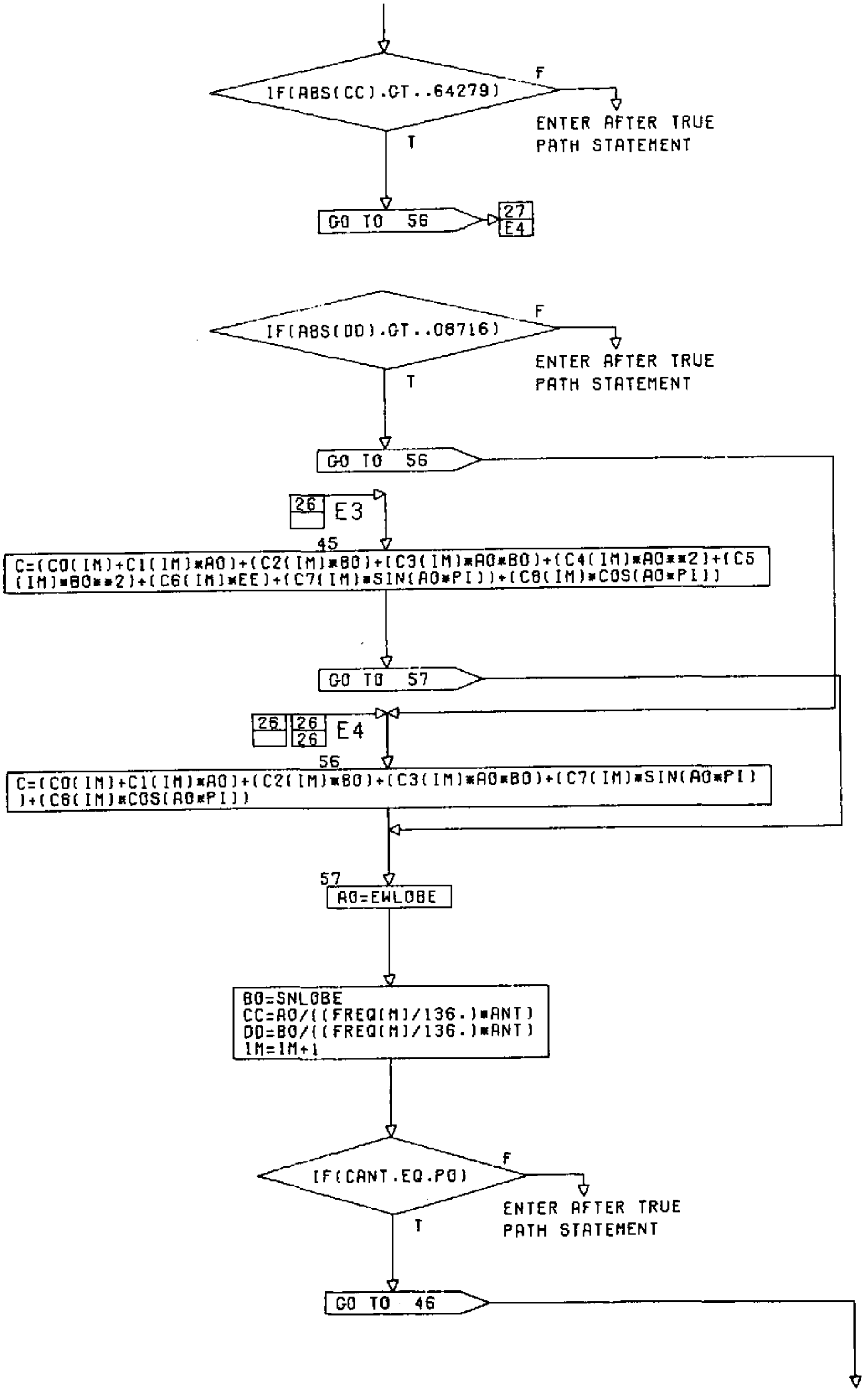

IF(ABS(CC).GT..08716)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 54
25
E1

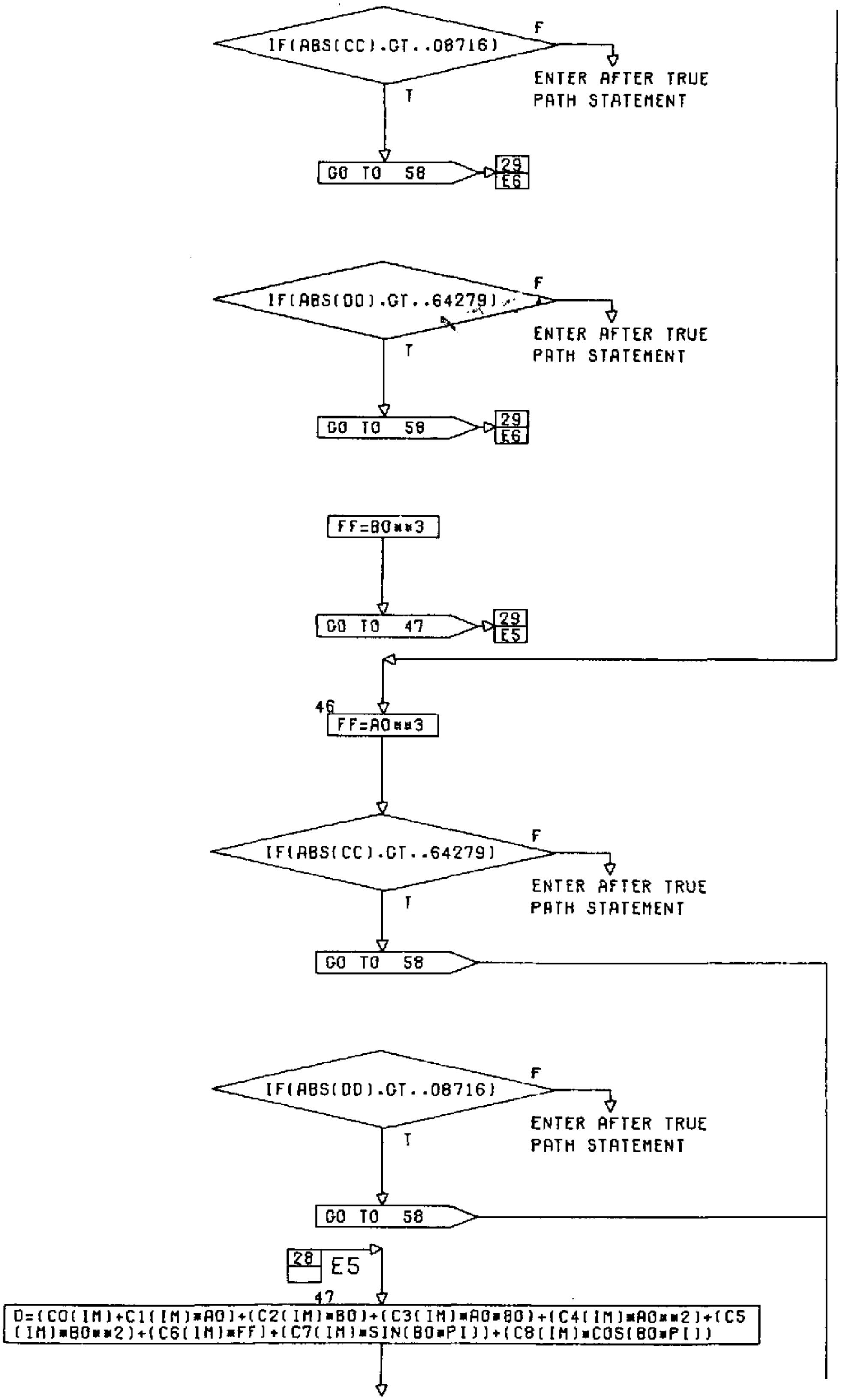
C4

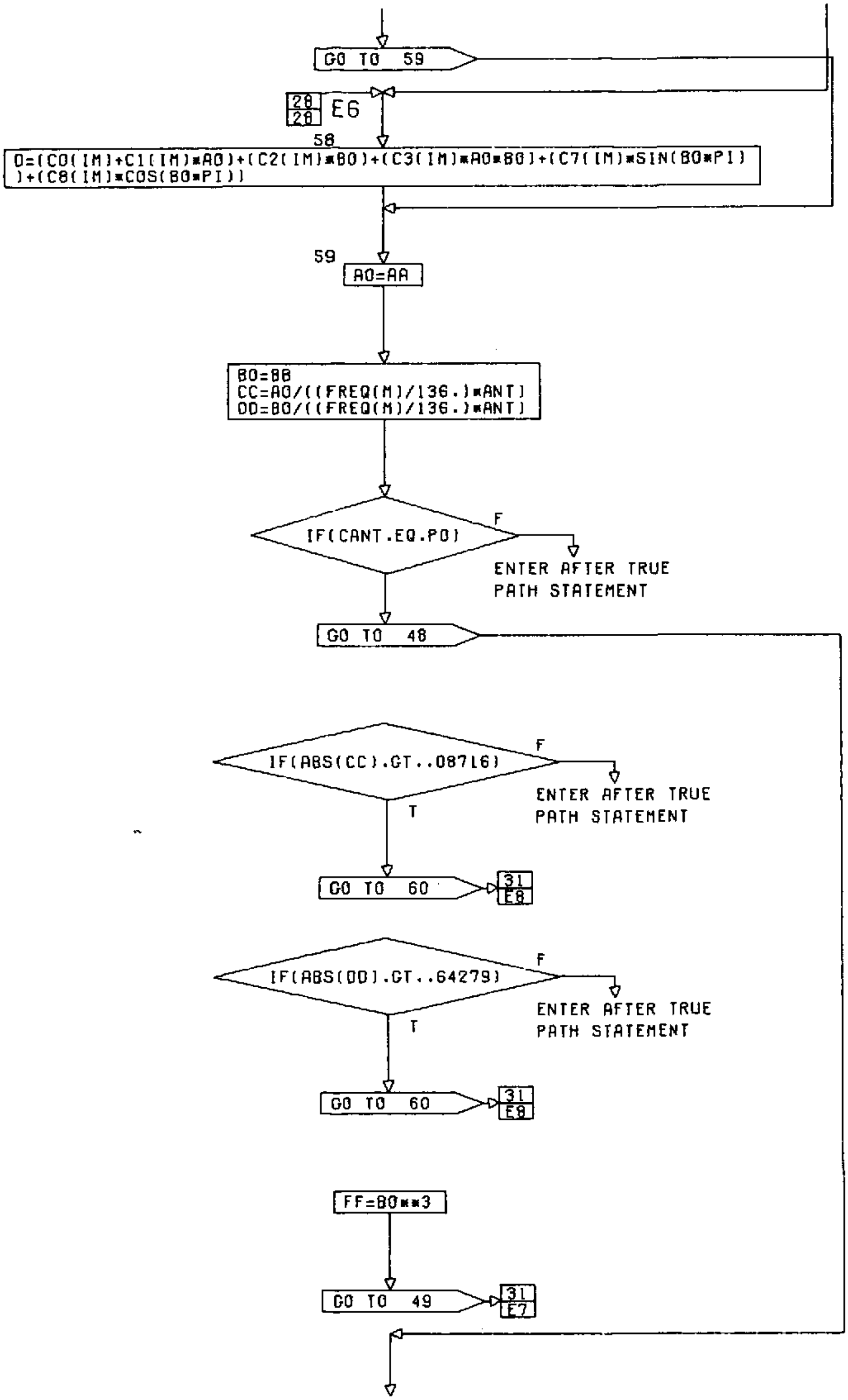
C4

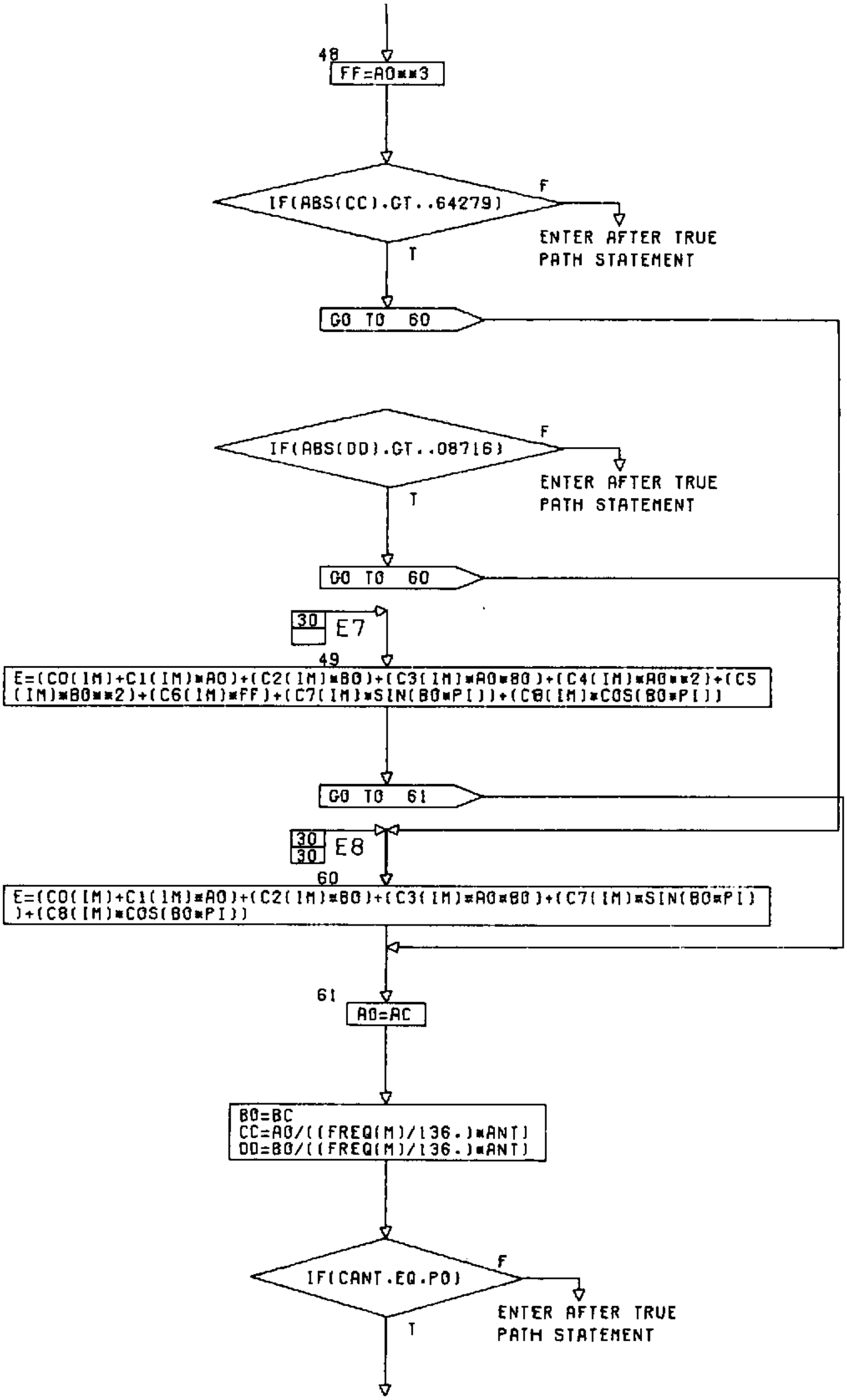

IF(ABS(DD).GT..64279)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 54
25
E1
EE=B0**3
GO TO 43
25
D9
23
D8
42
EE=A0**3
IF(ABS(CC).GT..64279)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 54
25
E1
IF(ABS(DD).GT..08716)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 54
24
D9
43
B=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C4(IM)*A0**2)+(C5
(IM)*B0**2)+(C6(IM)*EE)+(C7(IM)*SIN(A0*PI))+(C8(IM)*COS(A0*PI))
GO TO 55
24
24
24
E1
54
B=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C7(IM)*SIN(A0*PI)
)+(C8(IM)*COS(A0*PI))

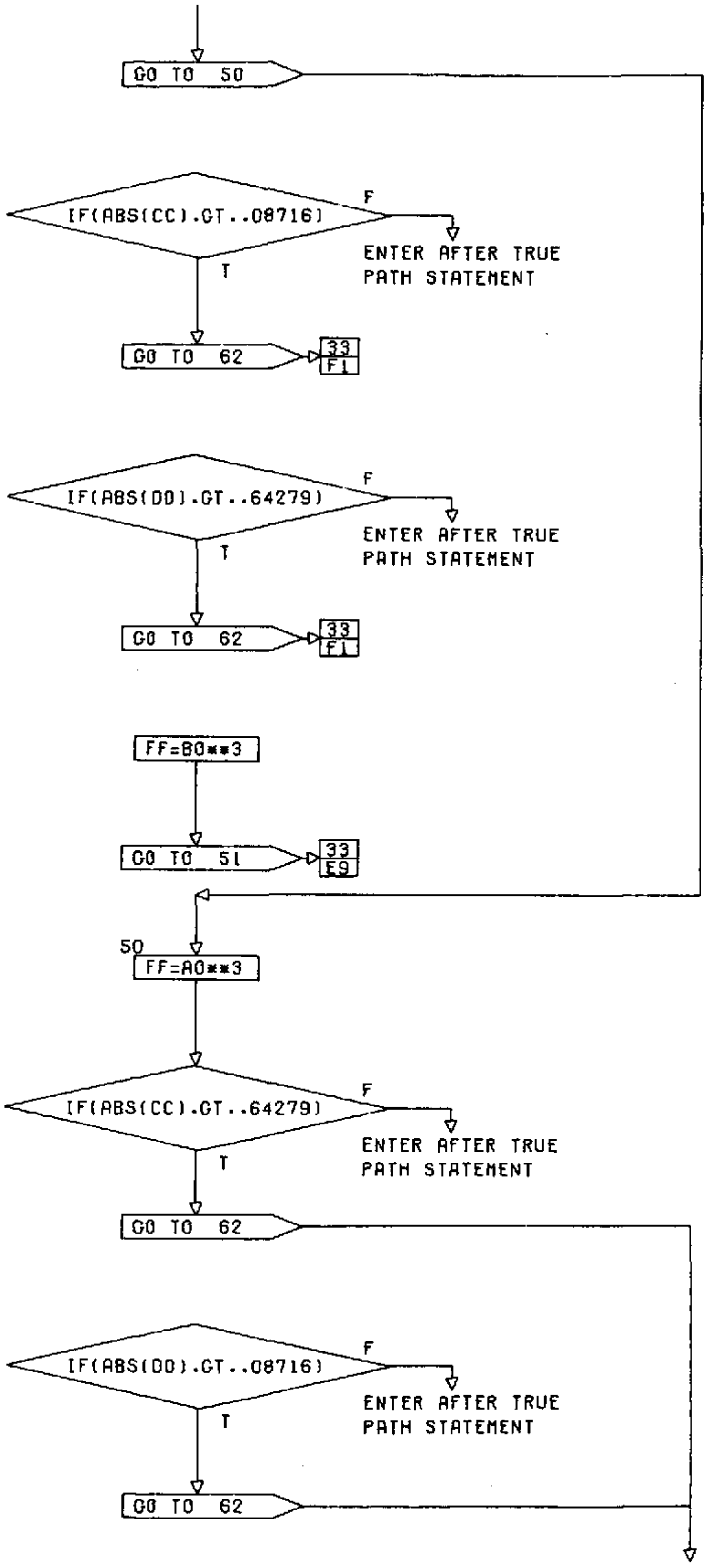


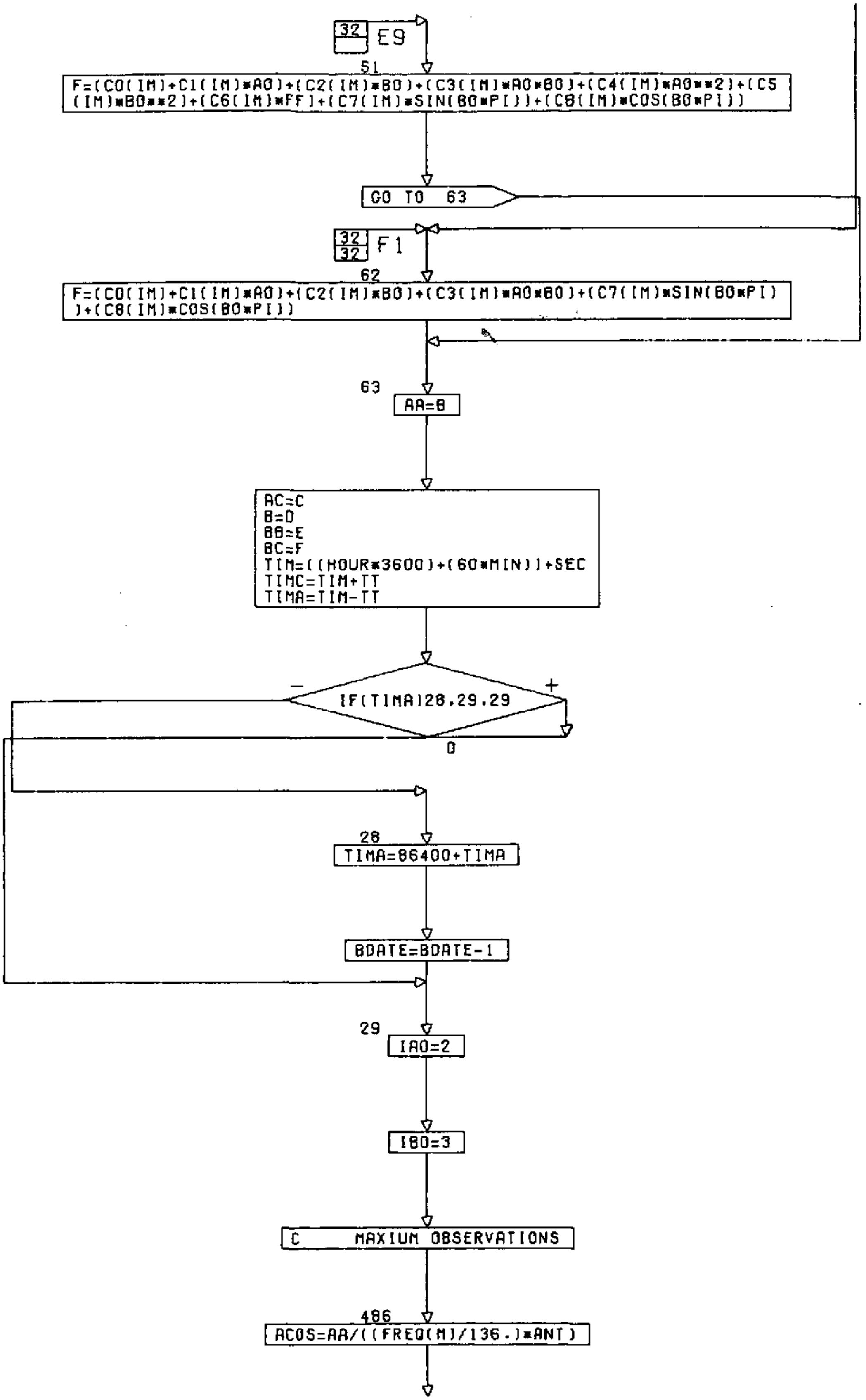

IF(ABS(CC).GT..64279)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 56
27
E4
IF(ABS(DD).GT..08716)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 56
26
E3
45
C=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C4(IM)*A0**2)+(C5(IM)*B0**2)+(C6(IM)*EE)+(C7(IM)*SIN(A0*PI))+(C8(IM)*COS(A0*PI))
GO TO 57
26
26
26
E4
56
C=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C7(IM)*SIN(A0*PI))+(C8(IM)*COS(A0*PI))
57
A0=EWLOBE
B0=SNLOBE
CC=A0/((FREQ(M)/136.)*ANT)
DD=B0/((FREQ(M)/136.)*ANT)
IM=IM+1
IF(CANT.EQ.P0)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 46

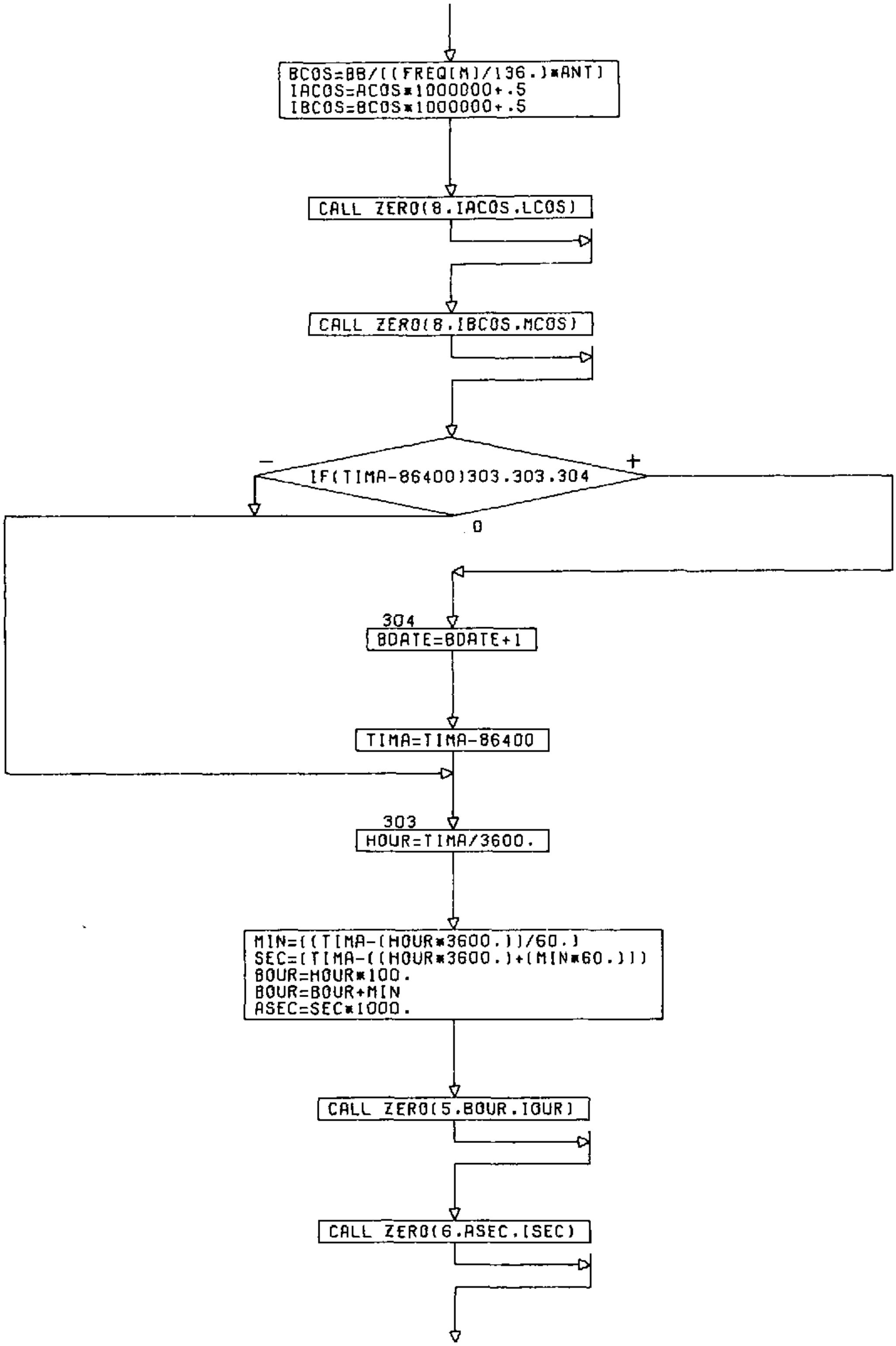

IF(ABS(CC).GT..08716)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 58
29
E6
IF(ABS(DD).GT..64279)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 58
29
E6
FF=BO**3
GO TO 47
29
E5
46
FF=AO**3
IF(ABS(CC).GT..64279)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 58
IF(ABS(DD).GT..08716)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 58
28
E5
47
D=(CO(IM)+C1(IM)*AO)+(C2(IM)*BO)+(C3(IM)*AO*BO)+(C4(IM)*AO**2)+(C5
(IM)*BO**2)+(C6(IM)*FF)+(C7(IM)*SIN(BO*PI))+(C8(IM)*COS(BO*PI))

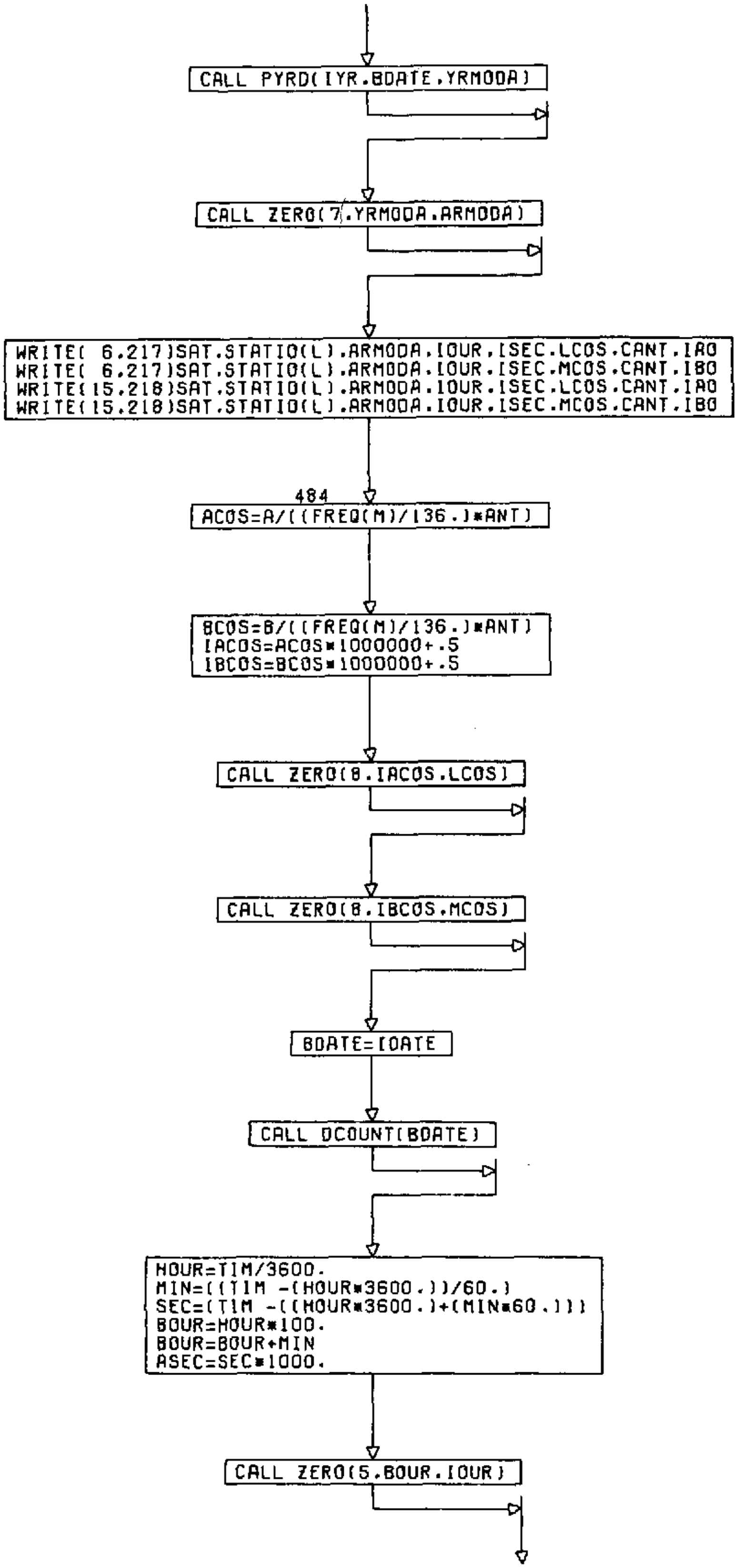

GO TO 59
28
28
E6
58
D=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C7(IM)*SIN(B0*PI)
)+(C8(IM)*COS(B0*PI))
59
A0=AA
B0=BB
CC=A0/((FREQ(M)/136.)*ANT)
DD=B0/((FREQ(M)/136.)*ANT)
IF(CANT.EQ.P0)
F
ENTER AFTER TRUE
PATH STATEMENT
GO TO 48
IF(ABS(CC).GT..08716)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 60
31
E8
IF(ABS(DD).GT..64279)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 60
31
E8
FF=B0**3
GO TO 49
31
E7

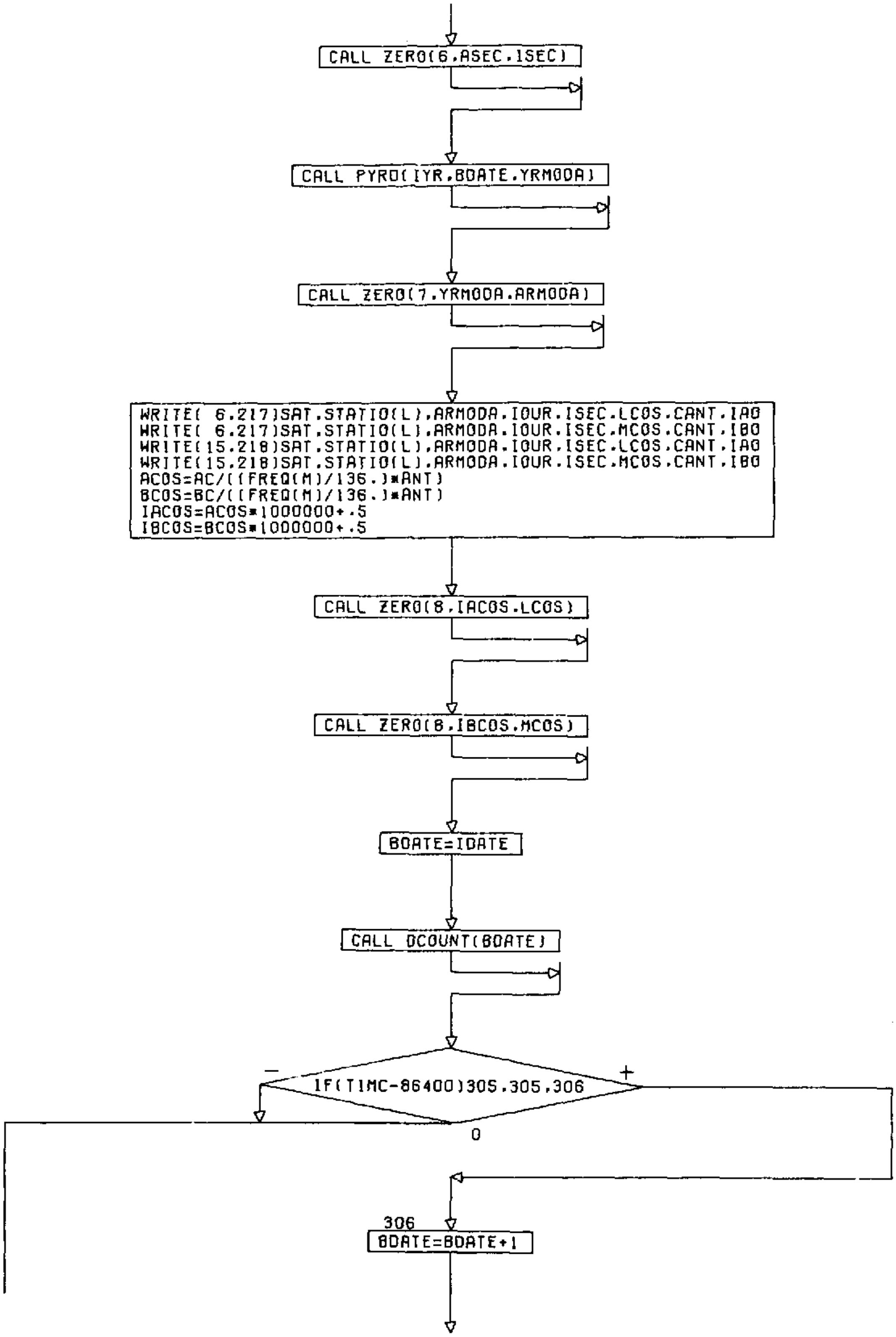

48
FF=A0**3
IF(ABS(CC).GT..64279)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 60
IF(ABS(DD).GT..08716)
F
ENTER AFTER TRUE PATH STATEMENT
T
GO TO 60
30
E7
49
E=(CO(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C4(IM)*A0**2)+(C5
(IM)*B0**2)+(C6(IM)*FF)+(C7(IM)*SIN(B0*PI))+(C8(IM)*COS(B0*PI))
GO TO 61
30
30
E8
60
E=(CO(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C7(IM)*SIN(B0*PI)
)+(C8(IM)*COS(B0*PI))
61
A0=AC
B0=BC
CC=A0/((FREQ(M)/136.)*ANT)
DD=B0/((FREQ(M)/136.)*ANT)
IF(CANT.EQ.P0)
F
ENTER AFTER TRUE PATH STATEMENT
T


GO TO 50
IF(ABS(CC).GT..08716)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 62
33
F1
IF(ABS(DD).GT..64279)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 62
33
F1
FF=B0**3
GO TO 51
33
E9
50
FF=A0**3
IF(ABS(CC).GT..64279)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 62
IF(ABS(DD).GT..08716)
F
T
ENTER AFTER TRUE PATH STATEMENT
GO TO 62

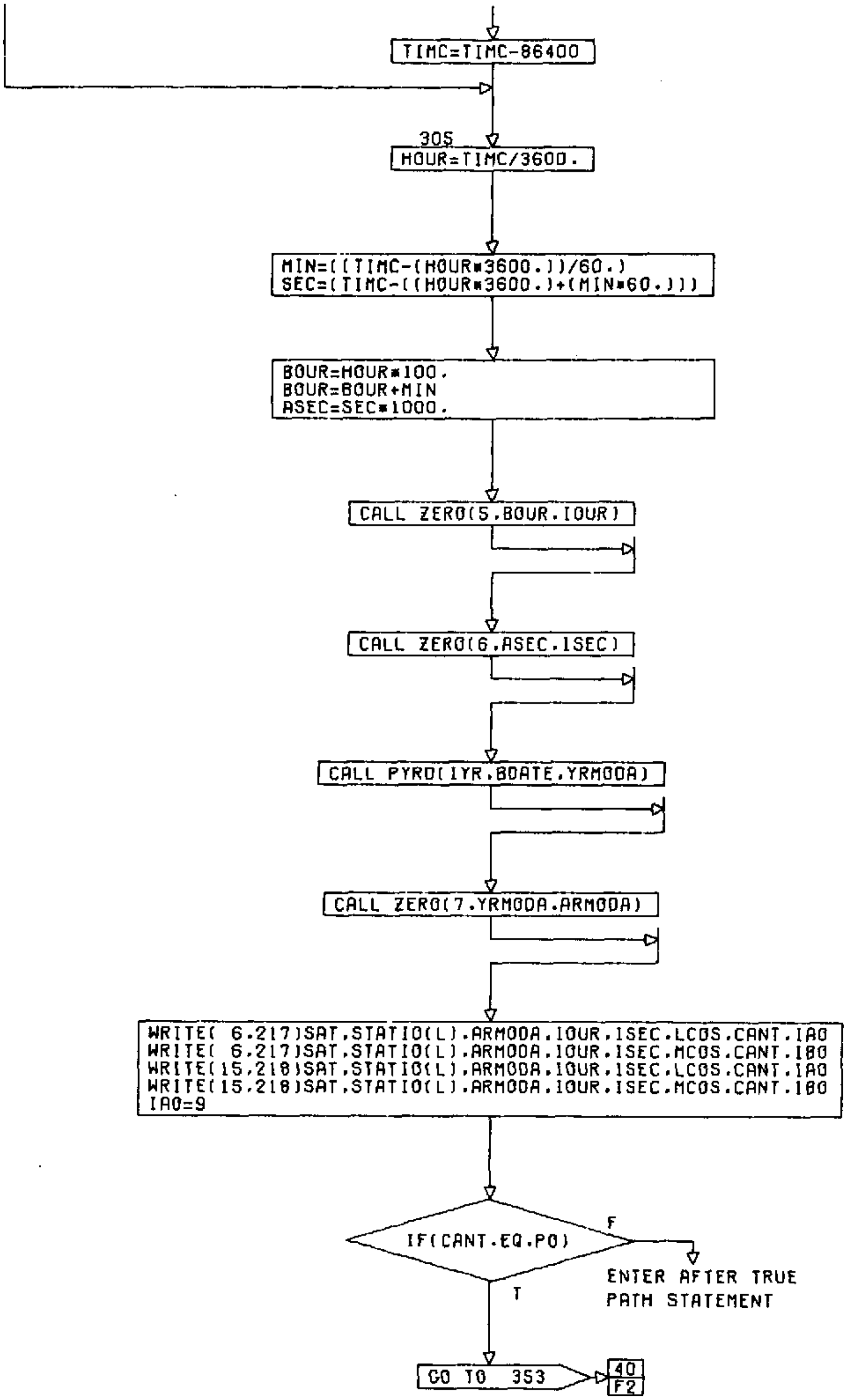

32
E9
51
F=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C4(IM)*A0**2)+(C5
(IM)*B0**2)+(C6(IM)*FF)+(C7(IM)*SIN(B0*PI))+(C8(IM)*COS(B0*PI))
GO TO 63
32
32
F1
62
F=(C0(IM)+C1(IM)*A0)+(C2(IM)*B0)+(C3(IM)*A0*B0)+(C7(IM)*SIN(B0*PI)
)+(C8(IM)*COS(B0*PI))
63
AA=B
AC=C
B=D
BB=E
BC=F
TIM=((HOUR*3600)+(60*MIN))+SEC
TIMC=TIM+TT
TIMA=TIM-TT
−
IF(TIMA)28,29,29
+
0
28
TIMA=86400+TIMA
BDATE=BDATE-1
29
IA0=2
IB0=3
C MAXIUM OBSERVATIONS
486
ACOS=AA/((FREQ(M)/136.)*ANT)

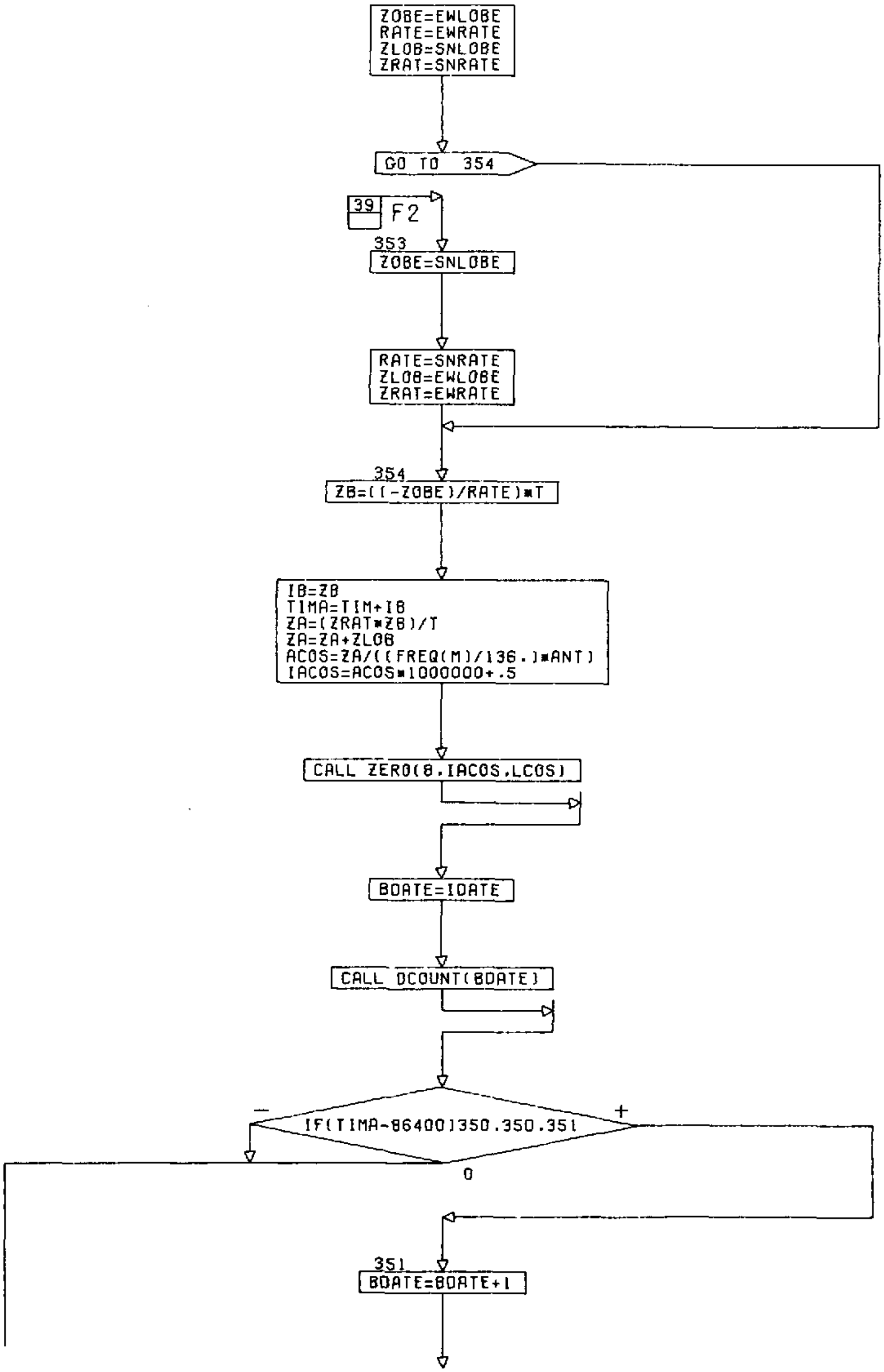

BCOS=BB/((FREQ(M)/136.)*ANT)
IACOS=ACOS*1000000+.5
IBCOS=BCOS*1000000+.5
CALL ZERO(8.IACOS.LCOS)
CALL ZERO(8.IBCOS.MCOS)
−
IF(TIMA-86400)303.303.304
+
0
304
BDATE=BDATE+1
TIMA=TIMA-86400
303
HOUR=TIMA/3600.
MIN=((TIMA-(HOUR*3600.))/60.)
SEC=(TIMA-((HOUR*3600.)+(MIN*60.)))
BOUR=HOUR*100.
BOUR=BOUR+MIN
ASEC=SEC*1000.
CALL ZERO(5.BOUR.IOUR)
CALL ZERO(6.ASEC.ISEC)

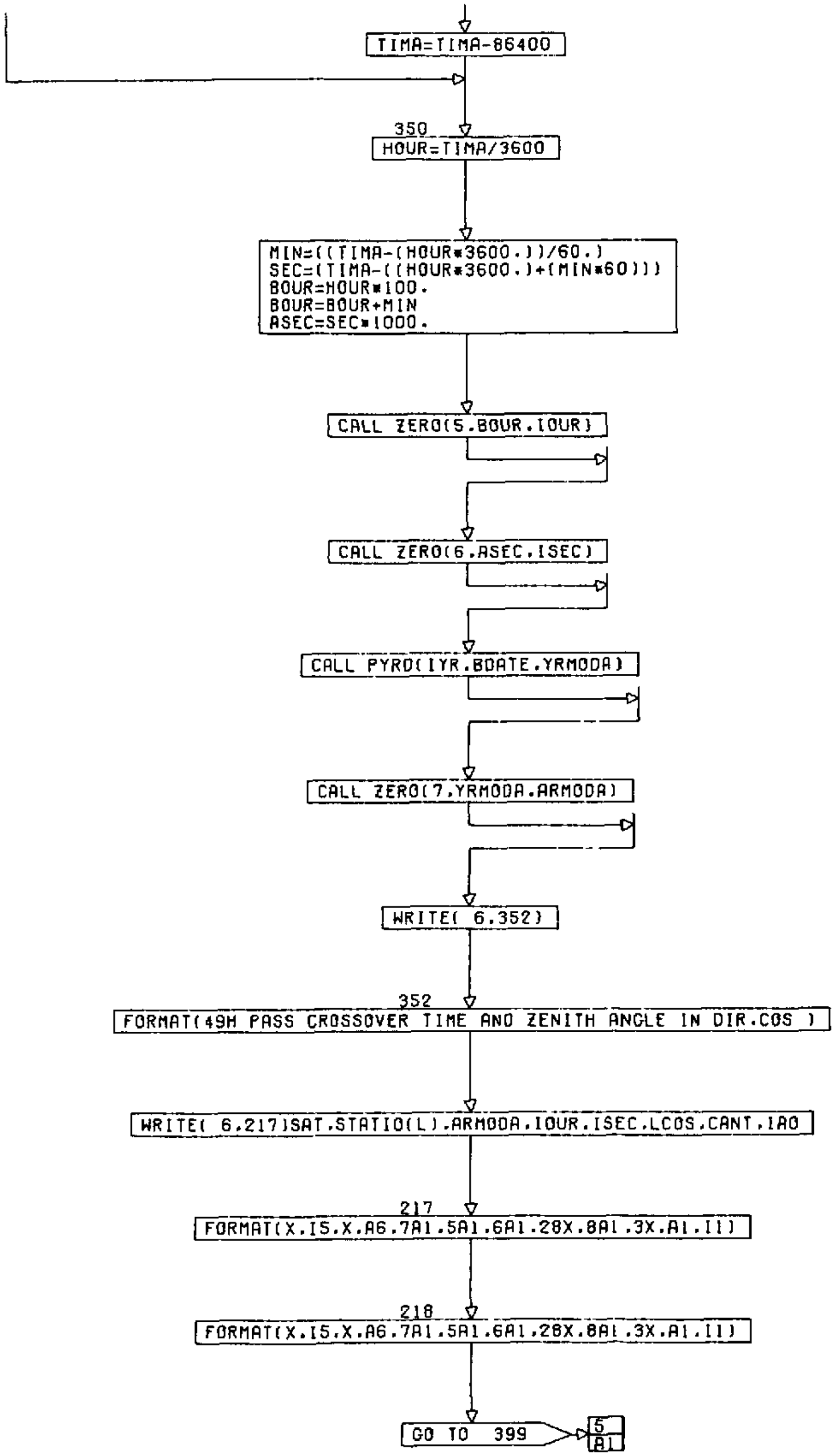

CALL PYRD(IYR.BDATE.YRMODA)
CALL ZERO(7.YRMODA.ARMODA)
WRITE(6.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.CANT.IAO
WRITE(6.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.CANT.IBO
WRITE(15.218)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.CANT.IAO
WRITE(15.218)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.CANT.IBO
484
ACOS=A/((FREQ(M)/136.)*ANT)
BCOS=B/((FREQ(M)/136.)*ANT)
IACOS=ACOS*1000000+.5
IBCOS=BCOS*1000000+.5
CALL ZERO(8.IACOS.LCOS)
CALL ZERO(8.IBCOS.MCOS)
BDATE=IDATE
CALL DCOUNT(BDATE)
HOUR=TIM/3600.
MIN=((TIM -(HOUR*3600.))/60.)
SEC=(TIM -((HOUR*3600.)+(MIN*60.)))
BOUR=HOUR*100.
BOUR=BOUR+MIN
ASEC=SEC*1000.
CALL ZERO(5.BOUR.IOUR)

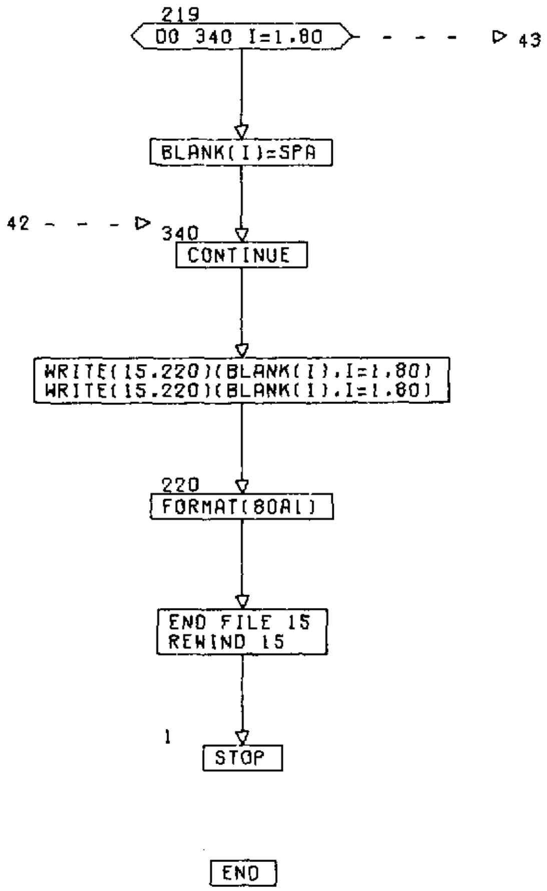


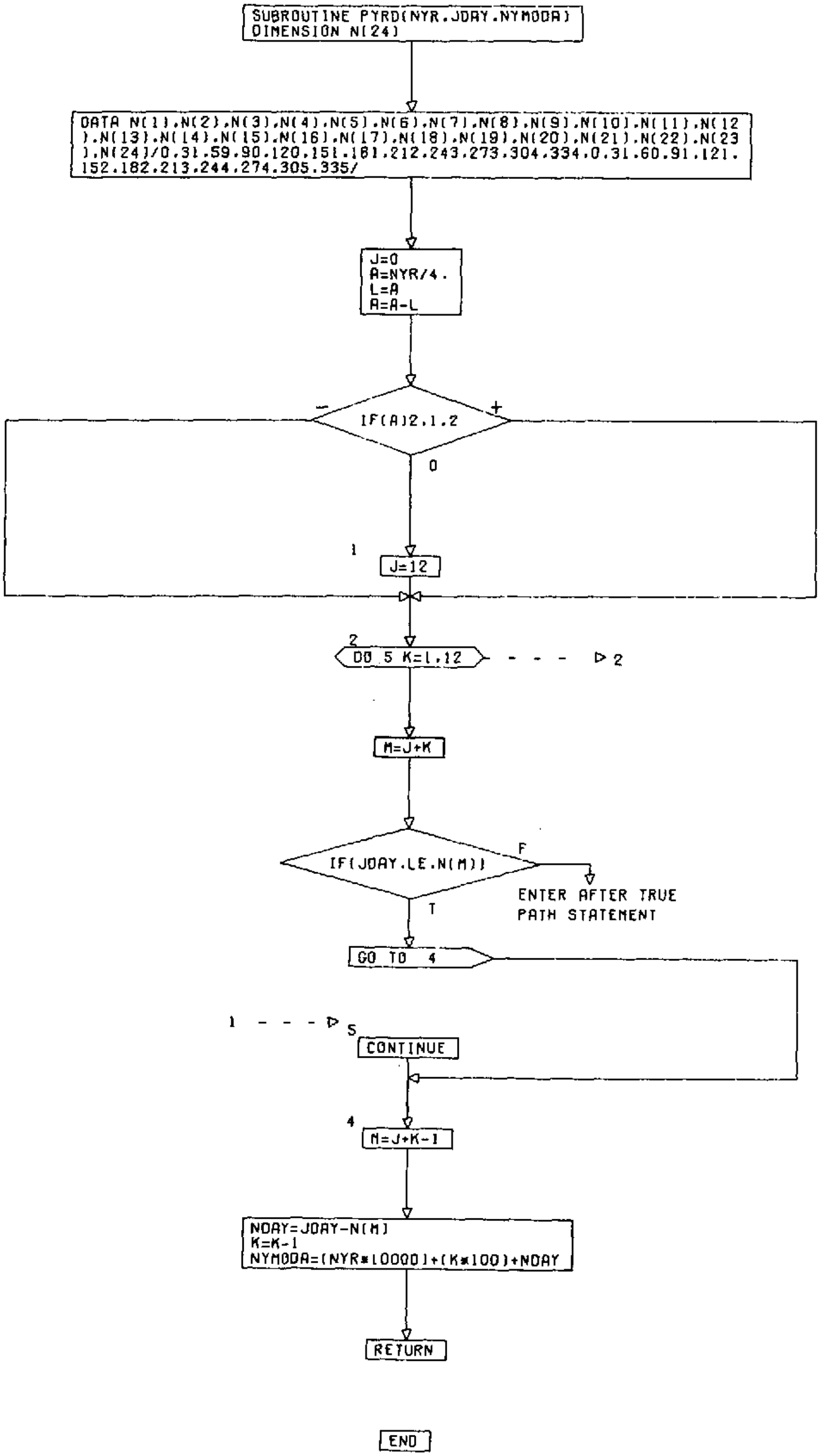
```
CALL ZERO(6.ASEC.ISEC)
CALL PYRO(IYR.BDATE.YRMODA)
CALL ZERO(7.YRMODA.ARMODA)
WRITE( 6.217)SAT.STATIO(L),ARMODA.IOUR.ISEC.LCOS.CANT.IAO
WRITE( 6.217)SAT.STATIO(L),ARMODA.IOUR.ISEC.MCOS.CANT.IBO
WRITE(15.218)SAT.STATIO(L),ARMODA.IOUR.ISEC.LCOS.CANT.IAO
WRITE(15.218)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.CANT.IBO
ACOS=AC/((FREQ(M)/136.)*ANT)
BCOS=BC/((FREQ(M)/136.)*ANT)
IACOS=ACOS*1000000+.5
IBCOS=BCOS*1000000+.5
CALL ZERO(8.IACOS.LCOS)
CALL ZERO(8.IBCOS.MCOS)
BDATE=IDATE
CALL DCOUNT(BDATE)
IF(TIMC-86400)305.305.306
306 BDATE=BDATE+1
```

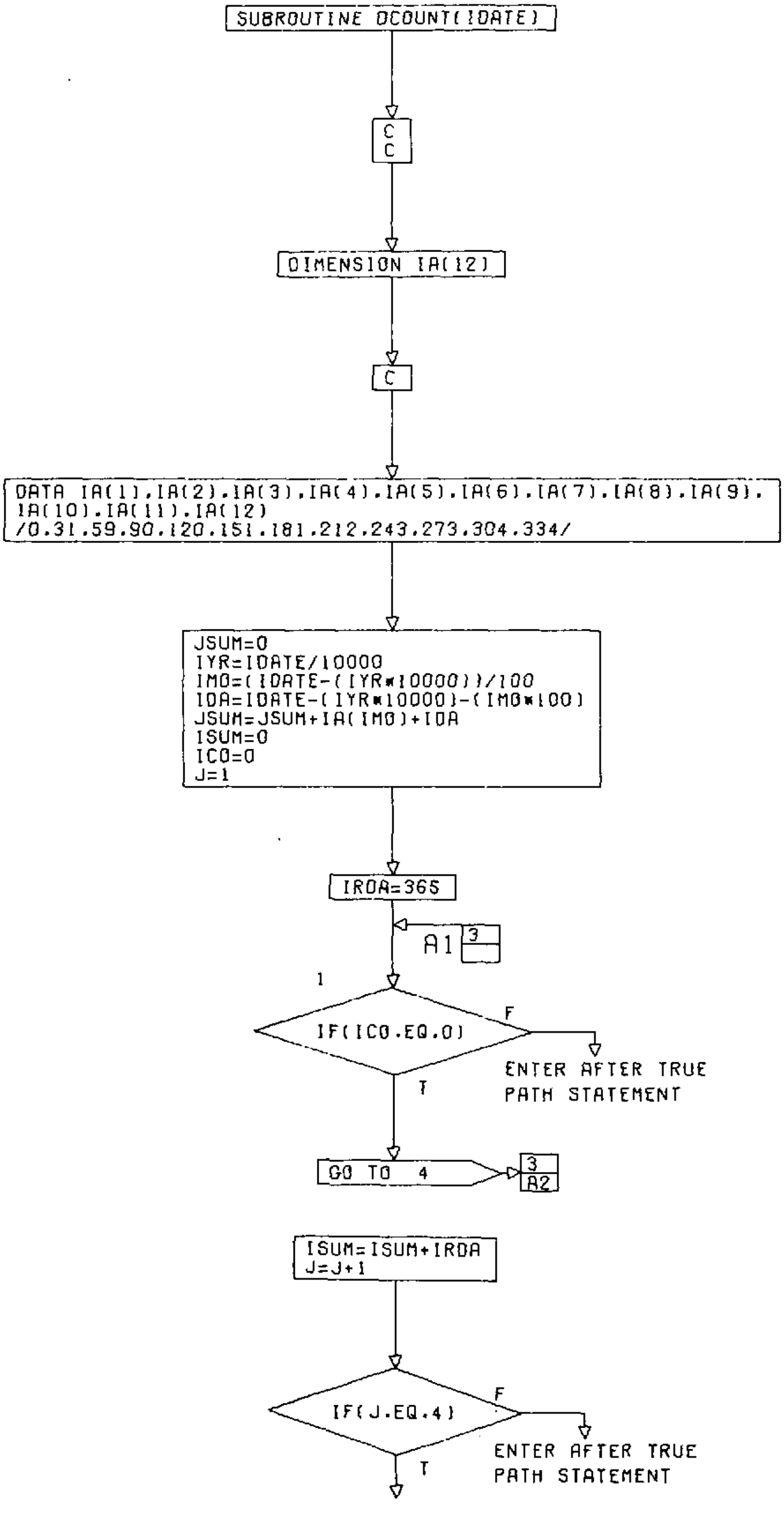

TIMC=TIMC-86400
305
HOUR=TIMC/3600.
MIN=((TIMC-(HOUR*3600.))/60.)
SEC=(TIMC-((HOUR*3600.)+(MIN*60.)))
BOUR=HOUR*100.
BOUR=BOUR+MIN
ASEC=SEC*1000.
CALL ZERO(5.BOUR.IOUR)
CALL ZERO(6.ASEC.ISEC)
CALL PYRO(IYR.BDATE.YRMODA)
CALL ZERO(7.YRMODA.ARMODA)
WRITE(6.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.CANT.IA0
WRITE(6.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.CANT.IB0
WRITE(15.218)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.CANT.IA0
WRITE(15.218)SAT.STATIO(L).ARMODA.IOUR.ISEC.MCOS.CANT.IB0
IA0=9
IF(CANT.EQ.P0)
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 353
40
F2


ZOBE=EWLOBE
RATE=EWRATE
ZLOB=SNLOBE
ZRAT=SNRATE
GO TO 354
39
F2
353
ZOBE=SNLOBE
RATE=SNRATE
ZLOB=EWLOBE
ZRAT=EWRATE
354
ZB=((-ZOBE)/RATE)*T
IB=ZB
TIMA=TIM+IB
ZA=(ZRAT*ZB)/T
ZA=ZA+ZLOB
ACOS=ZA/((FREQ(M)/136.)*ANT)
IACOS=ACOS*1000000+.5
CALL ZERO(8,IACOS,LCOS)
BDATE=IDATE
CALL DCOUNT(BDATE)
−
IF(TIMA-86400)350,350,351
+
0
351
BDATE=BDATE+1

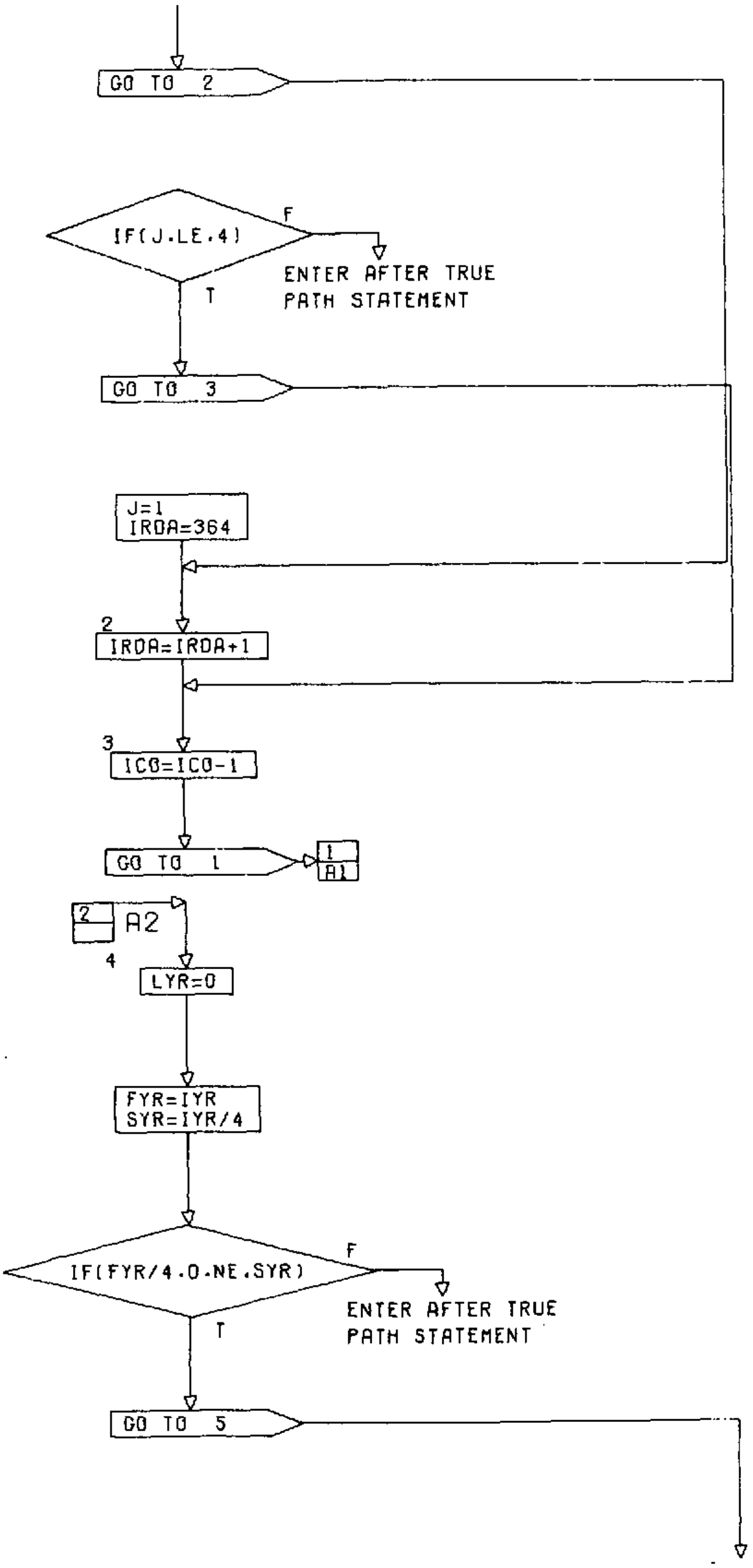

TIMA=TIMA-86400
350
HOUR=TIMA/3600
MIN=((TIMA-(HOUR*3600.))/60.)
SEC=(TIMA-((HOUR*3600.)+(MIN*60)))
BOUR=HOUR*100.
BOUR=BOUR+MIN
ASEC=SEC*1000.
CALL ZERO(5.BOUR.IOUR)
CALL ZERO(6.ASEC.ISEC)
CALL PYRD(IYR.BDATE.YRMODA)
CALL ZERO(7.YRMODA.ARMODA)
WRITE(6.352)
352
FORMAT(49H PASS CROSSOVER TIME AND ZENITH ANGLE IN DIR.COS)
WRITE(6.217)SAT.STATIO(L).ARMODA.IOUR.ISEC.LCOS.CANT.IAO
217
FORMAT(X.I5.X.A6.7A1.5A1.6A1.28X.8A1.3X.A1.I1)
218
FORMAT(X.I5.X.A6.7A1.5A1.6A1.28X.8A1.3X.A1.I1)
GO TO 399
5
A1

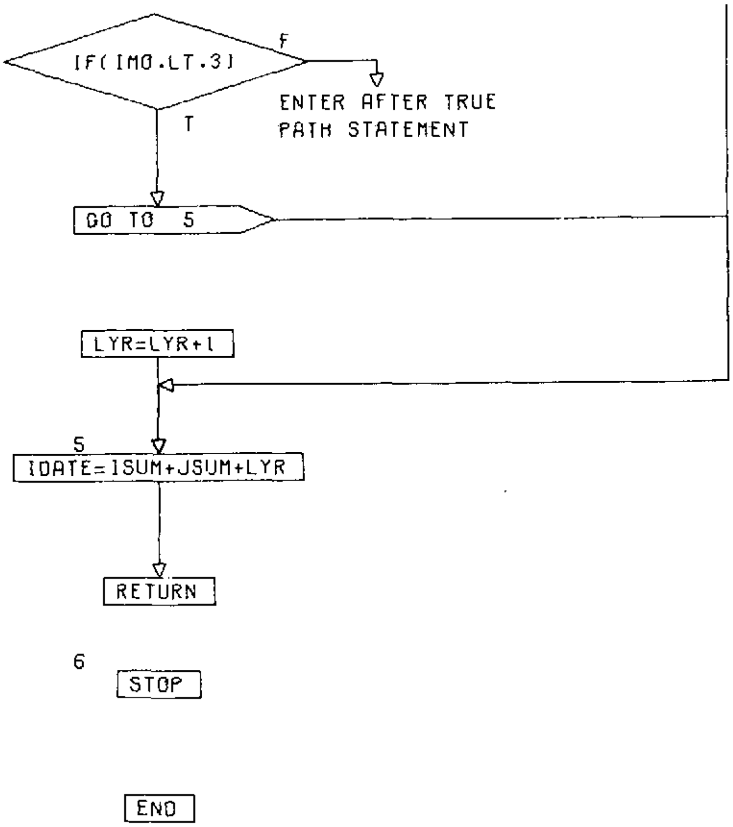

219
DO 340 I=1,80
43
BLANK(I)=SPA
42
340
CONTINUE
WRITE(15.220)(BLANK(I),I=1,80)
WRITE(15.220)(BLANK(I),I=1,80)
220
FORMAT(80A1)
END FILE 15
REWIND 15
1
STOP
END

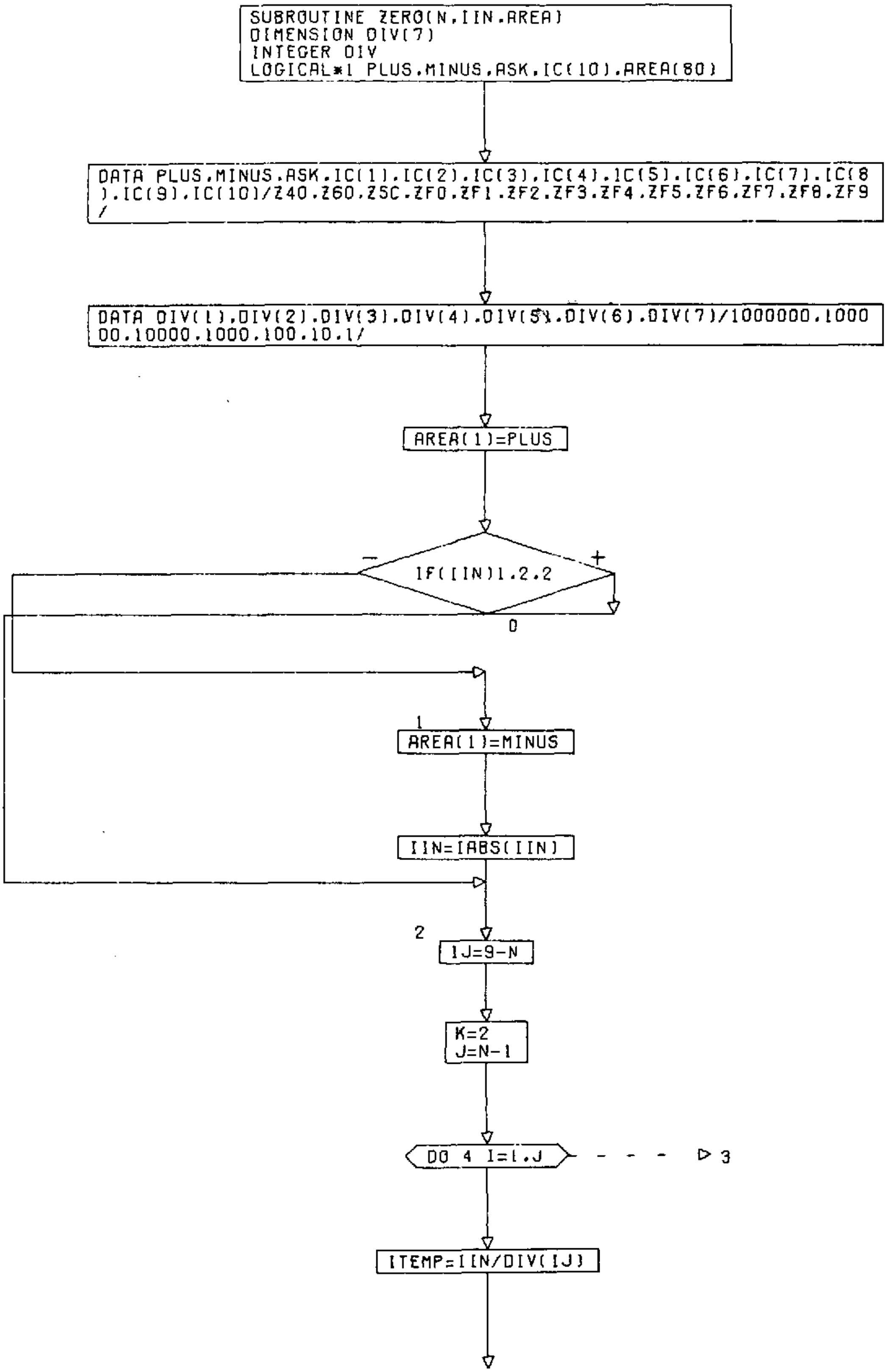

SUBROUTINE PYRD(NYR.JDAY.NYMODA)
DIMENSION N(24)
DATA N(1).N(2).N(3).N(4).N(5).N(6).N(7).N(8).N(9).N(10).N(11).N(12
).N(13).N(14).N(15).N(16).N(17).N(18).N(19).N(20).N(21).N(22).N(23
).N(24)/0.31.59.90.120.151.181.212.243.273.304.334.0.31.60.91.121.
152.182.213.244.274.305.335/
J=0
A=NYR/4.
L=A
A=A-L
IF(A)2.1.2
−
+
0
1
J=12
2
DO 5 K=1.12
▷ 2
M=J+K
IF(JDAY.LE.N(M))
F
ENTER AFTER TRUE
PATH STATEMENT
T
GO TO 4
1 ▷ 5
CONTINUE
4
M=J+K-1
NDAY=JDAY-N(M)
K=K-1
NYMODA=(NYR*10000)+(K*100)+NDAY
RETURN
END



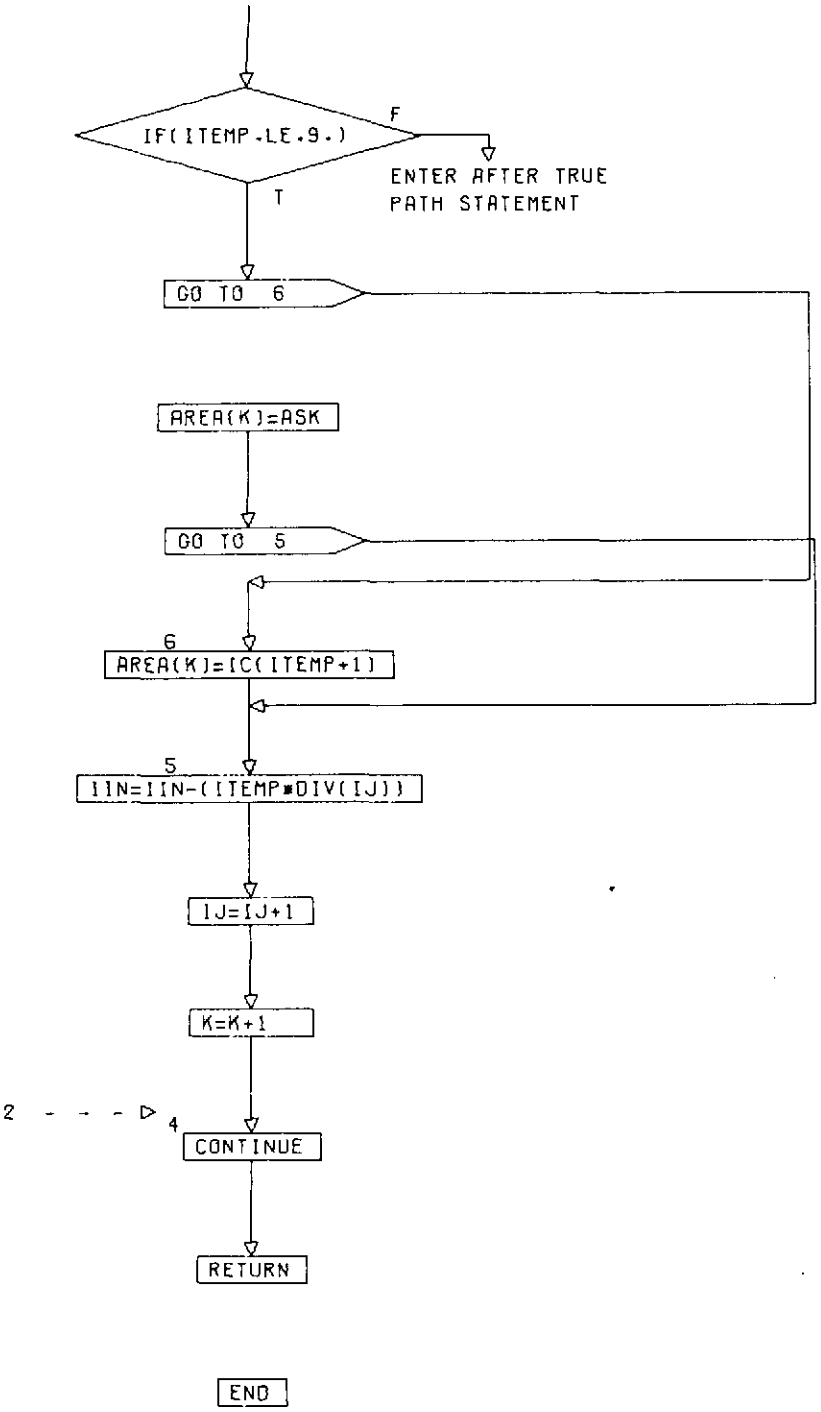
```
SUBROUTINE DCOUNT(IDATE)
C
C
DIMENSION IA(12)
C
DATA IA(1).IA(2).IA(3).IA(4).IA(5).IA(6).IA(7).IA(8).IA(9).
IA(10).IA(11).IA(12)
/0.31.59.90.120.151.181.212.243.273.304.334/
JSUM=0
IYR=IDATE/10000
IMO=(IDATE-(IYR*10000))/100
IDA=IDATE-(IYR*10000)-(IMO*100)
JSUM=JSUM+IA(IMO)+IDA
ISUM=0
ICO=0
J=1
IRDA=365
1 IF(ICO.EQ.0)
GO TO 4
ISUM=ISUM+IRDA
J=J+1
IF(J.EQ.4)
```


GO TO 2
IF(J.LE.4)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 3
J=1
IRDA=364
2
IRDA=IRDA+1
3
ICO=ICO-1
GO TO 1
1
A1
2
A2
4
LYR=0
FYR=IYR
SYR=IYR/4
IF(FYR/4.0.NE.SYR)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 5


IF(IMO.LT.3)
F
T
ENTER AFTER TRUE
PATH STATEMENT
GO TO 5
LYR=LYR+1
5
IDATE=ISUM+JSUM+LYR
RETURN
6
STOP
END



```
SUBROUTINE ZERO(N,IIN,AREA)
DIMENSION DIV(7)
INTEGER DIV
LOGICAL*1 PLUS,MINUS,ASK,IC(10),AREA(80)

DATA PLUS,MINUS,ASK,IC(1),IC(2),IC(3),IC(4),IC(5),IC(6),IC(7),IC(8
),IC(9),IC(10)/Z40,Z60,Z5C,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,ZF6,ZF7,ZF8,ZF9
/

DATA DIV(1),DIV(2),DIV(3),DIV(4),DIV(5),DIV(6),DIV(7)/1000000,1000
00,10000,1000,100,10,1/

AREA(1)=PLUS

IF(IIN)1,2,2

1 AREA(1)=MINUS

IIN=IABS(IIN)

2 IJ=9-N

K=2
J=N-1

DO 4 I=1,J

ITEMP=IIN/DIV(IJ)
```



☆ U.S. GOVERNMENT PRINTING OFFICE : 1972—735-967/595