REPORT NO. DOT-TSC-OST-72-5

THE NOISE EXPOSURE MODEL MOD-5

VOLUME 2

J. TAUB, T. FOREMAN, B. BROWNFIELD
TRANSPORTATION SYSTEMS CENTER
55 BROADWAY
CAMBRIDGE, MA. 02142

NOVEMBER 1971
TECHNICAL REPORT

transpo 72
U.S. International Transportation Exposition
Dulles International Airport
Washington, D.C.
May 27-June 4, 1972

DEPARTMENT OF TRANSPORTATION • UNITED STATES OF AMERICA

Prepared for
DEPARTMENT OF TRANSPORTATION
OFFICE OF THE SECRETARY
WASHINGTON, D. C. 20590

1. Report No. DOT-TSC-OST-72-5	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle THE NOISE EXPOSURE MODEL MOD 5 VOLUME II		5. Report Date November 1971
		6. Performing Organization Code
7. Author(s) J. Taub, T. Foreman, B. Brownfield		8. Performing Organization Report No.
9. Performing Organization Name and Address Department of Transportation Transportation Systems Center 55 Broadway, Cambridge, MA 02142		10. Work Unit No. R-2519
		11. Contract or Grant No. OS207
12. Sponsoring Agency Name and Address Department of Transportation Office of the Secretary Washington, D.C. 20590		13. Type of Report and Period Covered Technical Report
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This volume is the Programmer's Manual describing the Noise Exposure Model MOD-5 Computer Program.

Volume I is the airport analysis and user manual.

17. Key Words	18. Distribution Statement
•Noise Abatement •Airport Aircraft Noise	Availability is Unlimited. Document may be Released To the National Technical Information Service, Springfield, Virginia 22151, for Sale to the Public.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	175	

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

The Noise Exposure Model MOD-5 simulation computer program is described in this volume, Volume II, of the Noise Exposure Model MOD-5 report. Volume I* of this report contains the Airport Analysis and User's Manual sections.

Volume I and Volume II may be considered as independent reports; however, an understanding of the computer program described here (Volume II) will be difficult to obtain without the physical descriptions and analysis presented in Volume I.

The sample airport used to illustrate Volume I, Sample Airport 1975, is used here as an example. The sample input deck and output is from a computer run simulating the noise exposure around this airport. For a description of Sample Airport 1975 see Volume I. A drawing of the airport is provided in this volume at the close of Appendix F.

*Report No. DOT-TSC-OST-72-5, Vol. I.

THE NEM-5 SIMULATION COMPUTER PROGRAM

PURPOSE

The NEM-5 simulation computer program calculates values of a noise exposure index at observer grid points in the neighborhood of an airport. Additionally, up to three contours of equal noise exposure index values may be plotted.

The selected noise exposure index may be one of the following:

NE: Noise Exposure
NEF: Noise Exposure Forecast
WECPNL: Weighted Equivalent Continuous Perceived Noise Level

LANGUAGE/MACHINE

The NEM-5 program is written in FORTRAN-4, level H. It is being run on the Massachusetts Institute of Technology IBM 360/75 computer.

MODEL/EQUATIONS

The Noise Exposure Model MOD-5 and its submodels are discussed in the Airport Analysis section of Volume I of this report. The NEM-5 computer program compiles the submodels, listed below, to form a numerical simulation of the Noise Exposure Model MOD-5.

- Aircraft Class Model
- Airport Model
- Flight Model
- Aircraft Noise Model
- Noise Exposure Index Model

METHOD

Figure 1 is a flow chart which indicates the major NEM-5

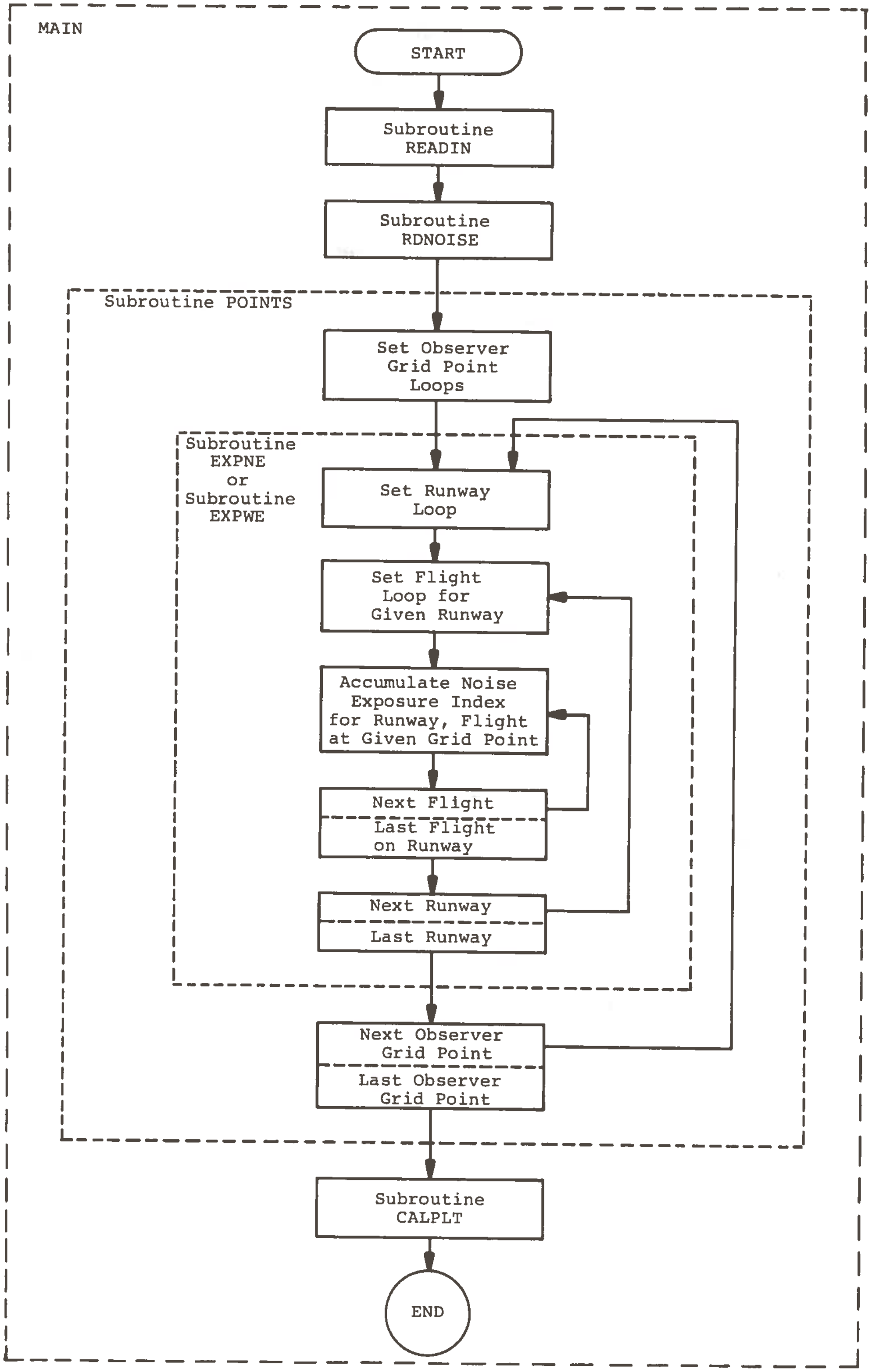


Figure 1. Descriptive Flowchart of the NEM-5 Program

program loops and the subroutines in which they are developed. The MAIN program calls subroutine READIN which reads the information pertaining to runways, ground tracks, flights and the observer grid. If the built-in aircraft noise model is not to be used, MAIN then calls subroutine RDNOISE to read in the user-specified aircraft noise tables. MAIN then calls subroutine POINTS which sets up the x- and y-coordinate loops for the observer grid points and calls subroutine EXPNE if NE or NEF is to be calculated or subroutine EXPWE if WECPNL is to be calculated. Subroutine EXPWE or EXPNE sets up the loops for runways and flights for each runway and accumulates, at the observer grid point specified by subroutine POINTS, the noise exposure index due to all flights and runways. Upon return to MAIN, if contour plots are requested, subroutine CALPLT is called.

The above description and Figure 1 are not intended to be a complete description of the NEM-5 program but are simplified statements of the method of operation. Detailed flowcharts for the routines MAIN, POINTS, EXPNE and EXPWE are contained in Appendix A. Appendix B is a commented program listing.

SUBROUTINES

The NEM-5 program structure, as defined by the subroutines which comprise it, is described by the subroutine call chart in Appendix C.

A complete list of the subroutines and a description of each is contained in Appendix D. Each subroutine is described in terms of its purpose, other subroutines called, input and output.

In both the call charts and the subroutine descriptions, certain subroutines and functions have been left out. These are all system library subroutines and functions and the copyrighted CALCOMP subroutines. The CALCOMP subroutines used in this program are PLOT, PLOTS, LINE, SYMBOL, NUMBER, SCALE, AXIS and FACTOR.

As well, there are five subroutines listed but not described. They are AUTCNT, CRECT, CUBICS, INTPT and STRIKE. These subroutines determine the noise exposure contours from the values of the noise exposure index at the observer grid points. The area contained in each contour is also computed. Appendix F provides sample printed output. Information concerning these subroutines may be obtained from Mr. Walter Messcher of DOT-TSC at 55 Broadway, Cambridge, Massachusetts.

Finally it should be noted that there are three non-functional dummy subroutines. They are READOU, NOMEN and CONTR. These subroutines were intended for later expansion of the NEM-5 program capability.

PROGRAM INPUT

The NEM-5 input is described and discussed in the User's Manual section of Volume I. However, for completeness, a Sample Airport 1975 input listing is contained in Appendix E.

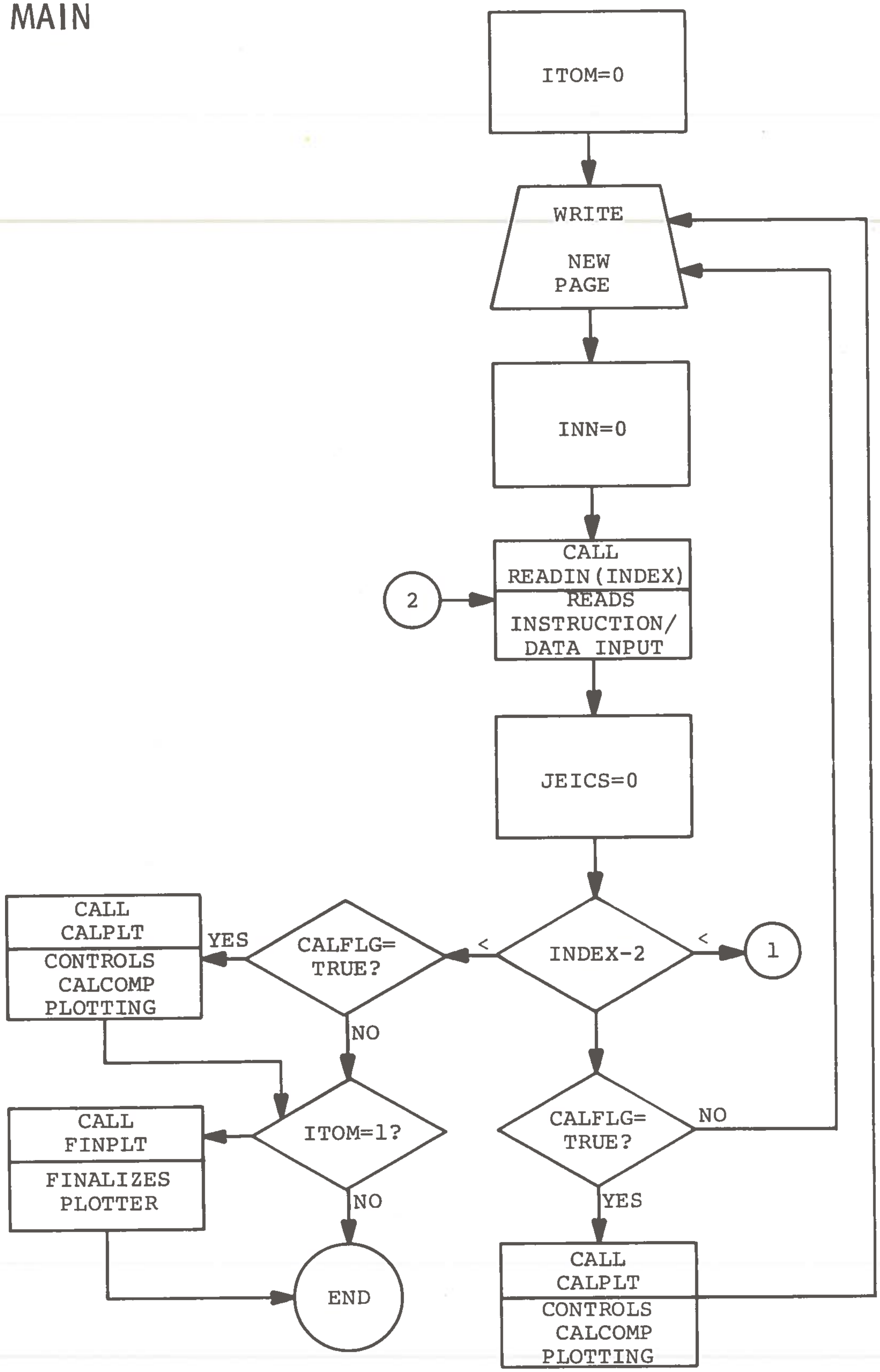
PROGRAM OUTPUT

The NEM-5 output is described and discussed in the Airport Analysis section of Volume I. However, a listing of the printed output and a three contour plot for Sample Airport 1975 is contained in Appendix F.

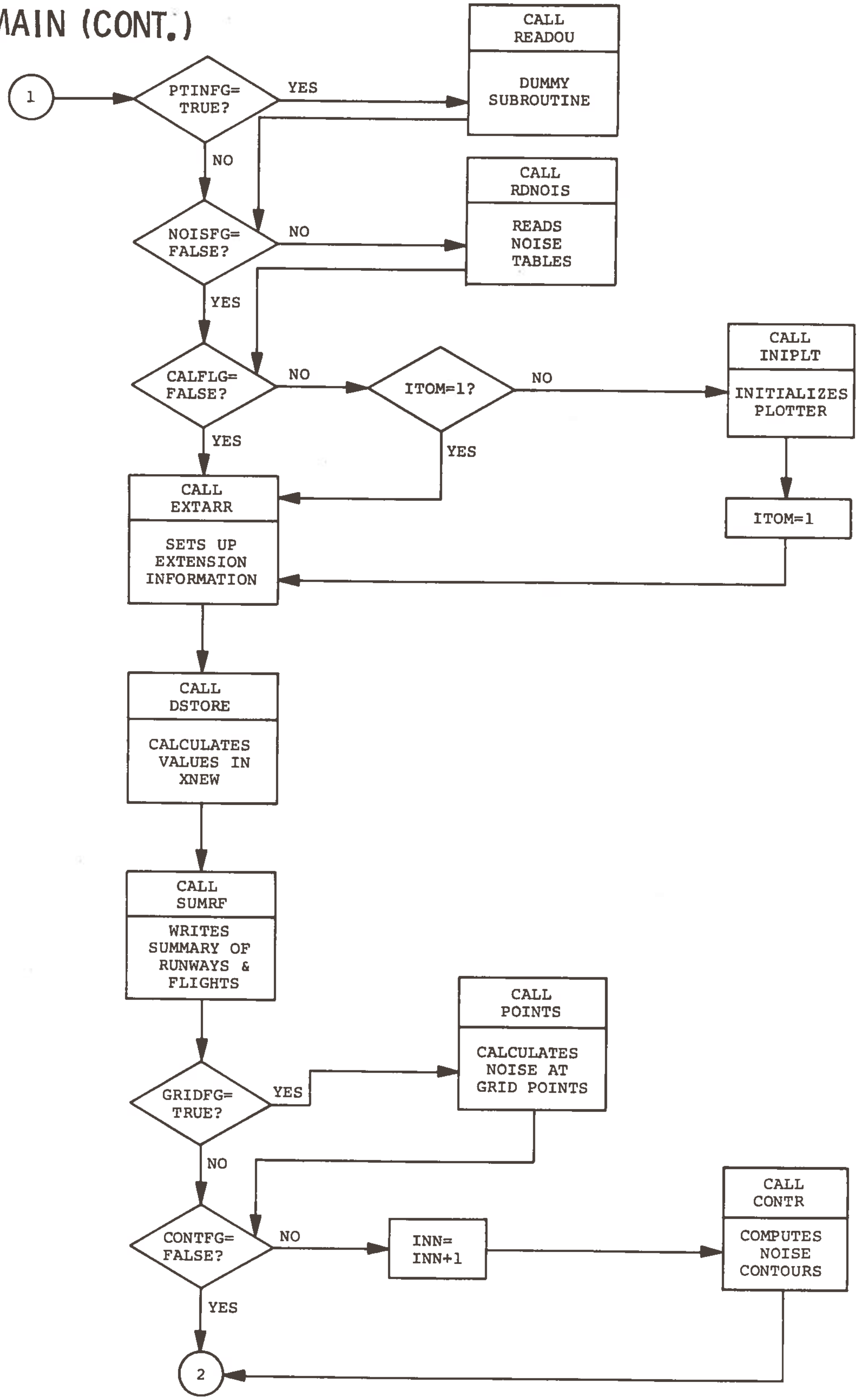
APPENDIX A

FLOWCHARTS

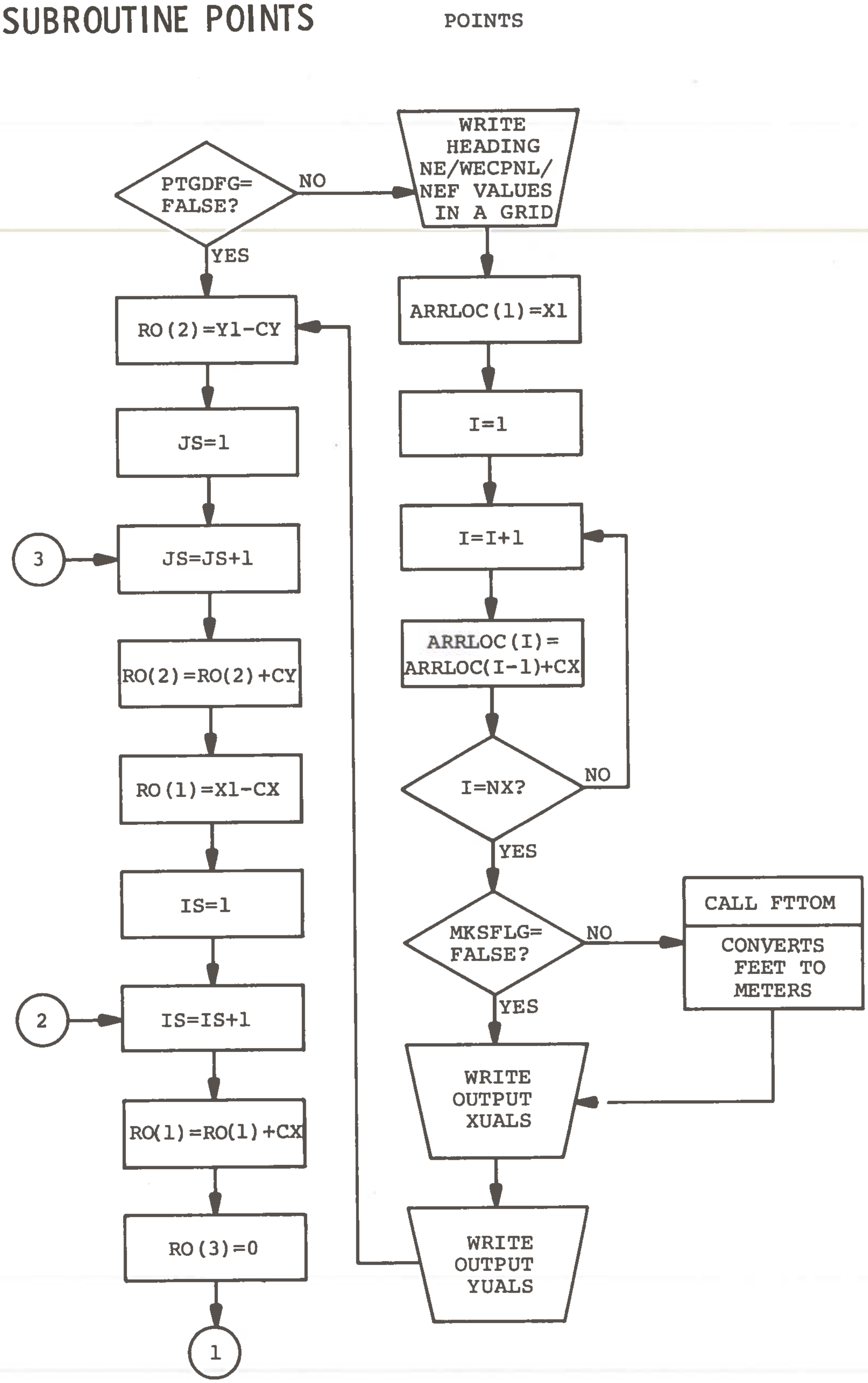
MAIN


ITOM=0
WRITE
NEW
PAGE
INN=0
CALL
READIN(INDEX)
READS
INSTRUCTION/
DATA INPUT
2
JEICS=0
INDEX-2
<
<
1
CALFLG=
TRUE?
YES
NO
CALL
CALPLT
CONTROLS
CALCOMP
PLOTTING
ITOM=1?
NO
CALL
FINPLT
FINALIZES
PLOTTER
END
CALFLG=
TRUE?
NO
YES
CALL
CALPLT
CONTROLS
CALCOMP
PLOTTING

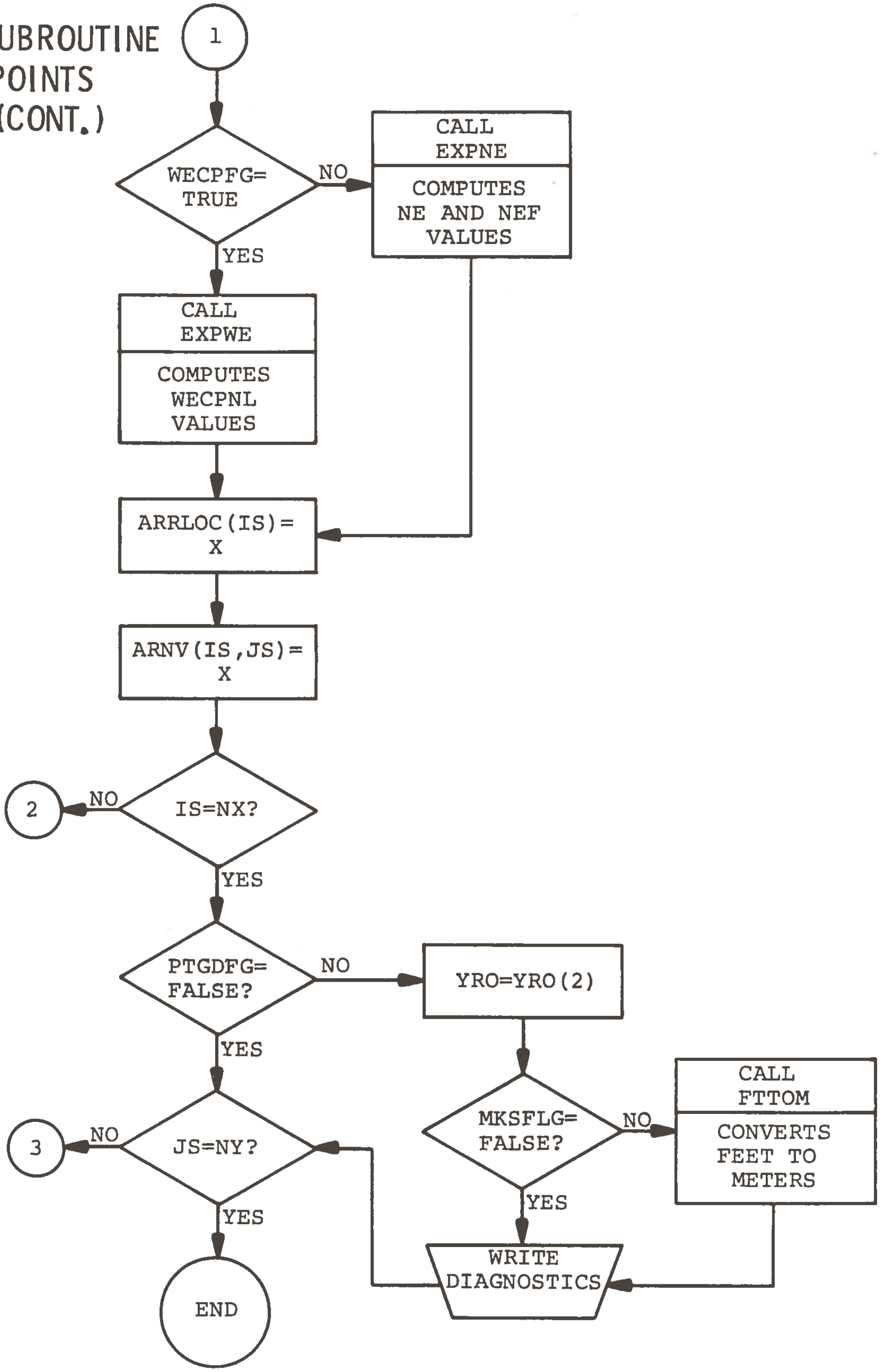
MAIN (CONT.)


1
PTINFG= TRUE?
YES
NO
CALL READOU
DUMMY SUBROUTINE
NOISFG= FALSE?
NO
YES
CALL RDNOIS
READS NOISE TABLES
CALFLG= FALSE?
NO
YES
ITOM=1?
NO
YES
CALL INIPLT
INITIALIZES PLOTTER
ITOM=1
CALL EXTARR
SETS UP EXTENSION INFORMATION
CALL DSTORE
CALCULATES VALUES IN XNEW
CALL SUMRF
WRITES SUMMARY OF RUNWAYS & FLIGHTS
GRIDFG= TRUE?
YES
NO
CALL POINTS
CALCULATES NOISE AT GRID POINTS
CONTFG= FALSE?
NO
YES
INN= INN+1
CALL CONTR
COMPUTES NOISE CONTOURS
2

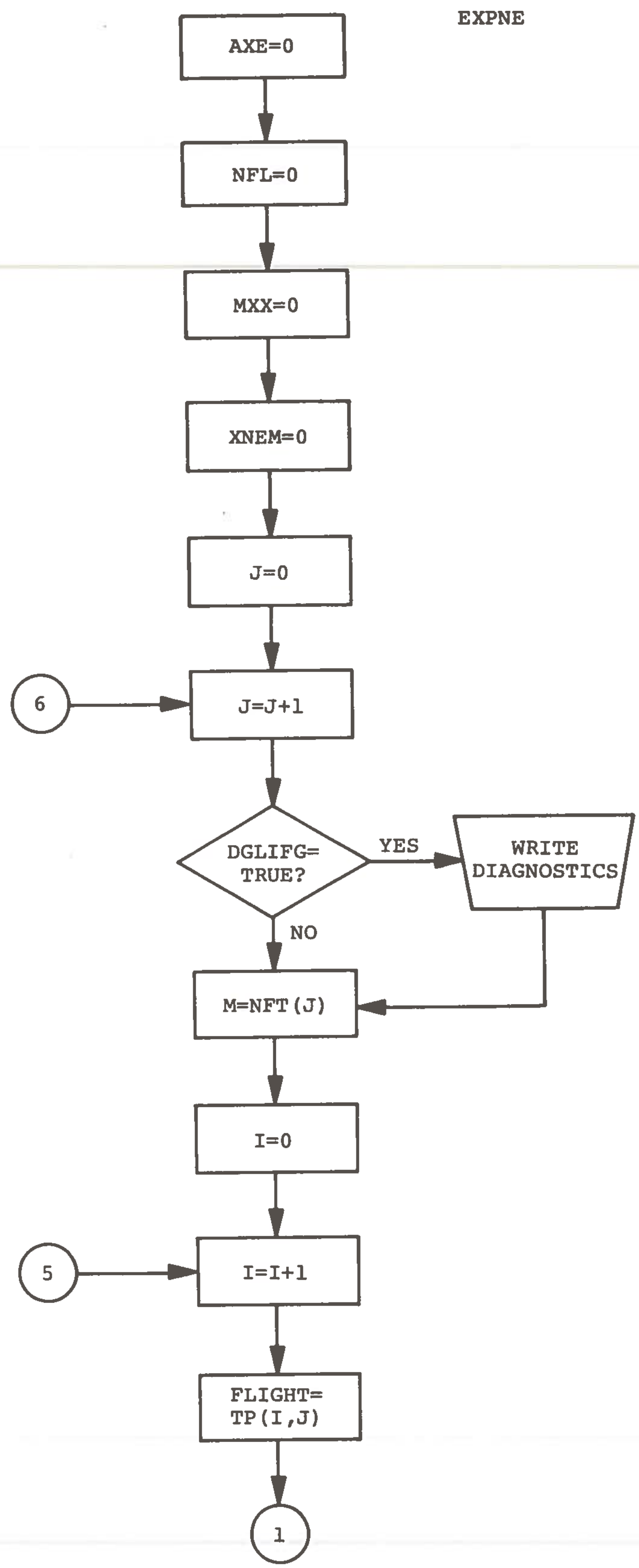
SUBROUTINE POINTS



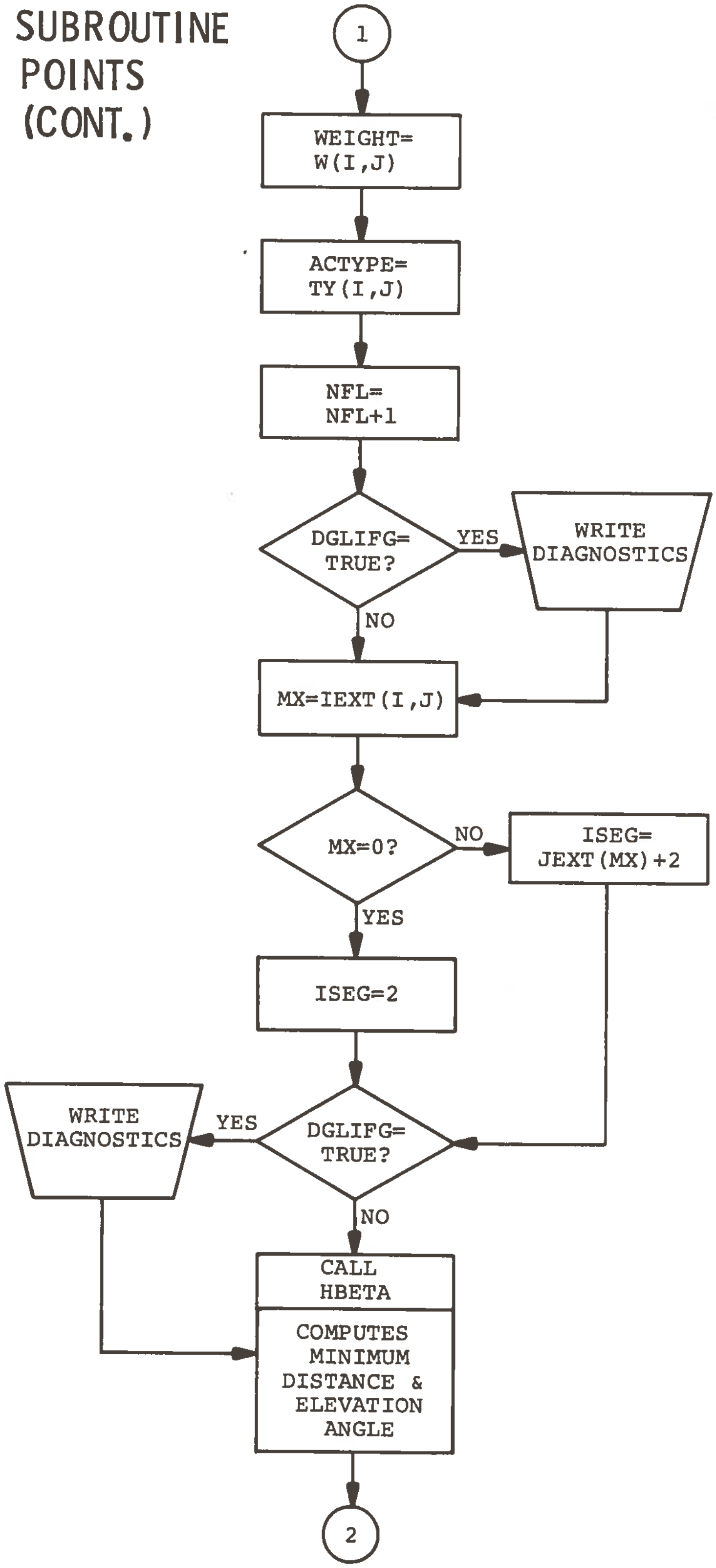
SUBROUTINE POINTS (CONT.)



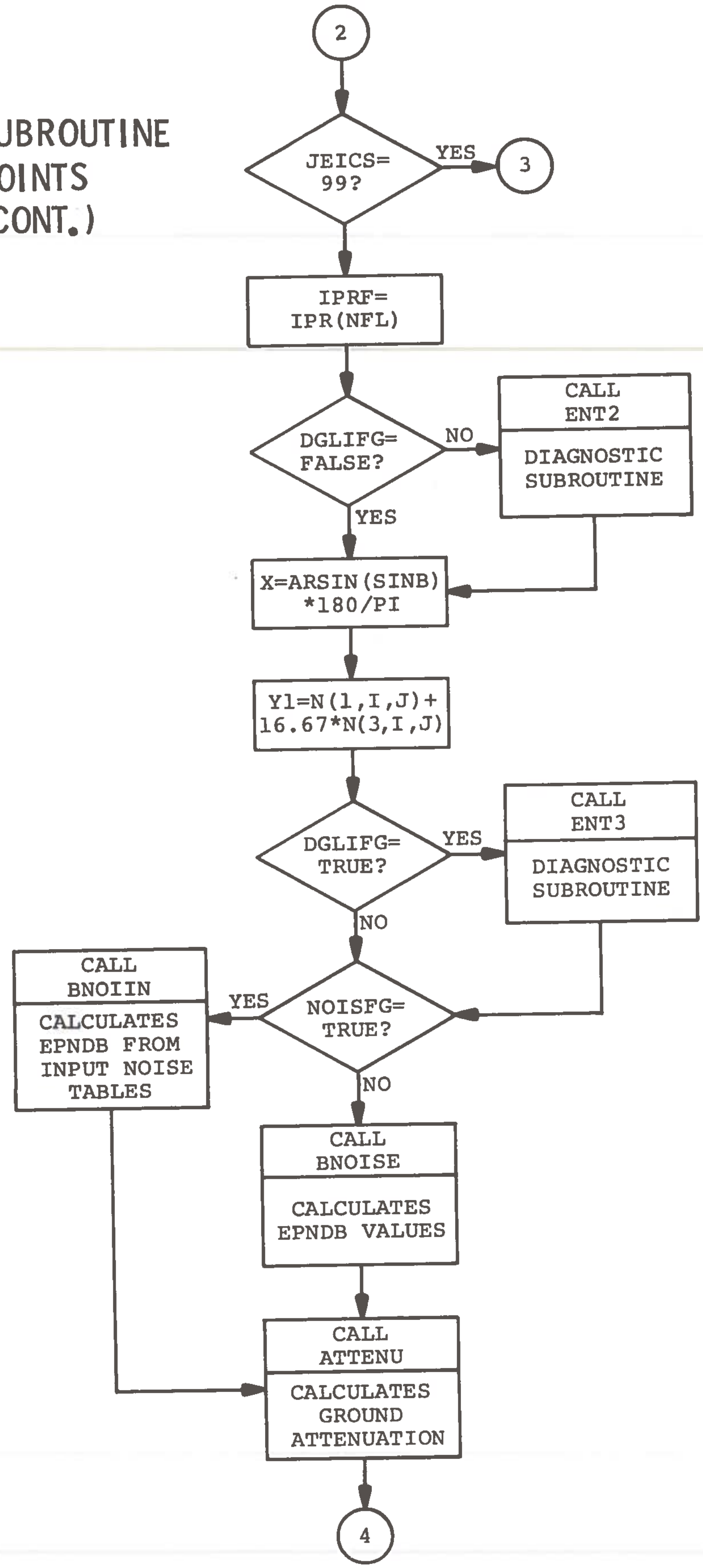
SUBROUTINE
POINTS
(CONT.)



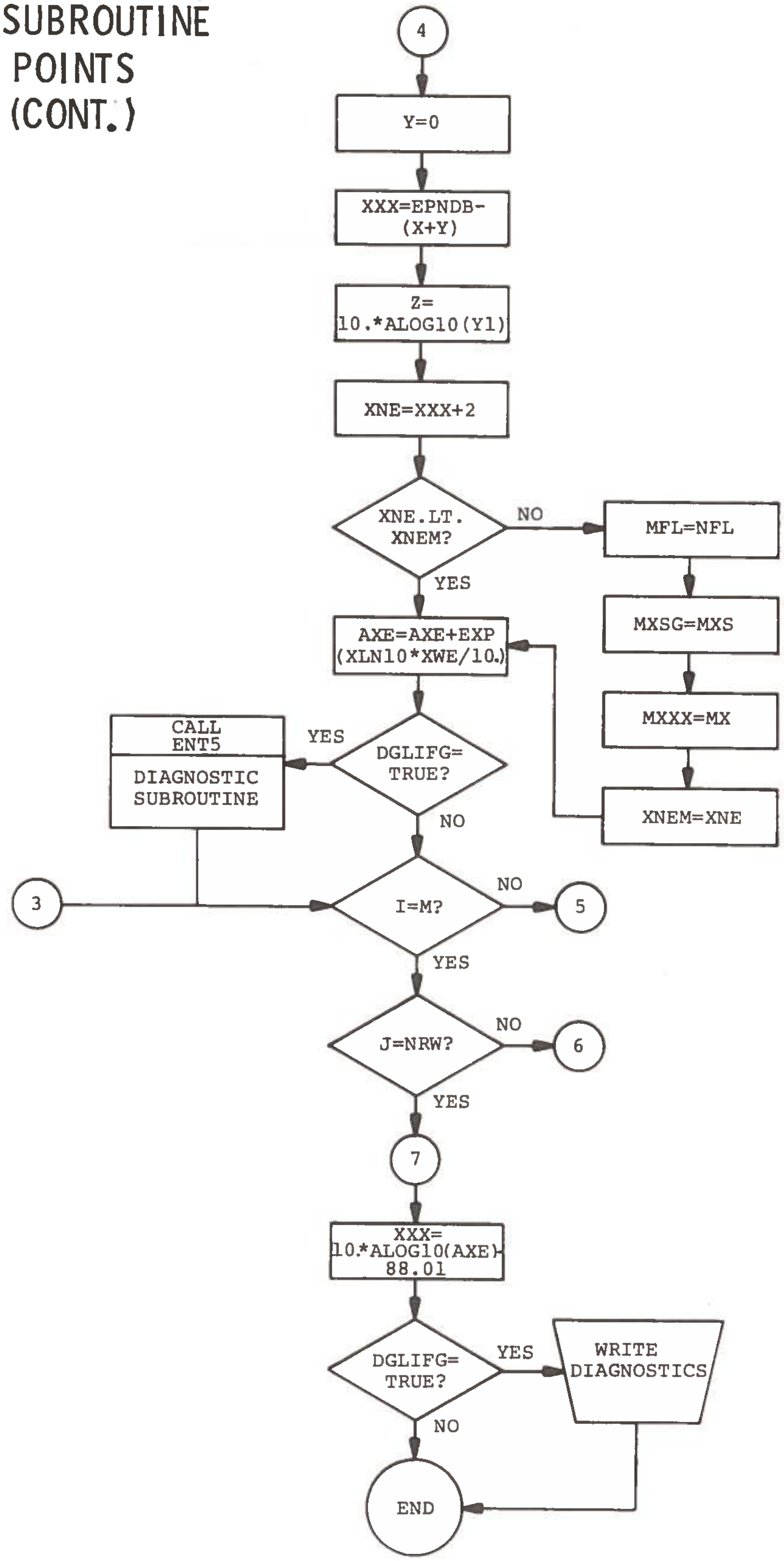
SUBROUTINE POINTS (CONT.)



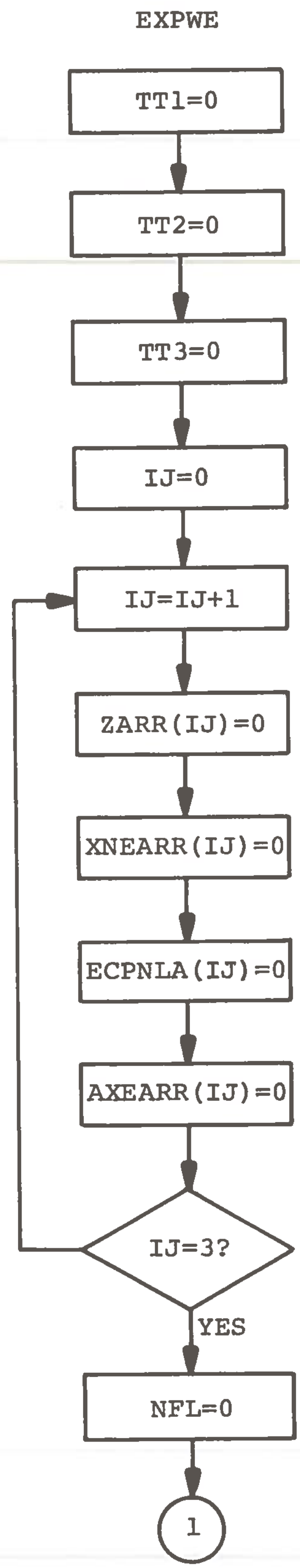
SUBROUTINE POINTS (CONT.)



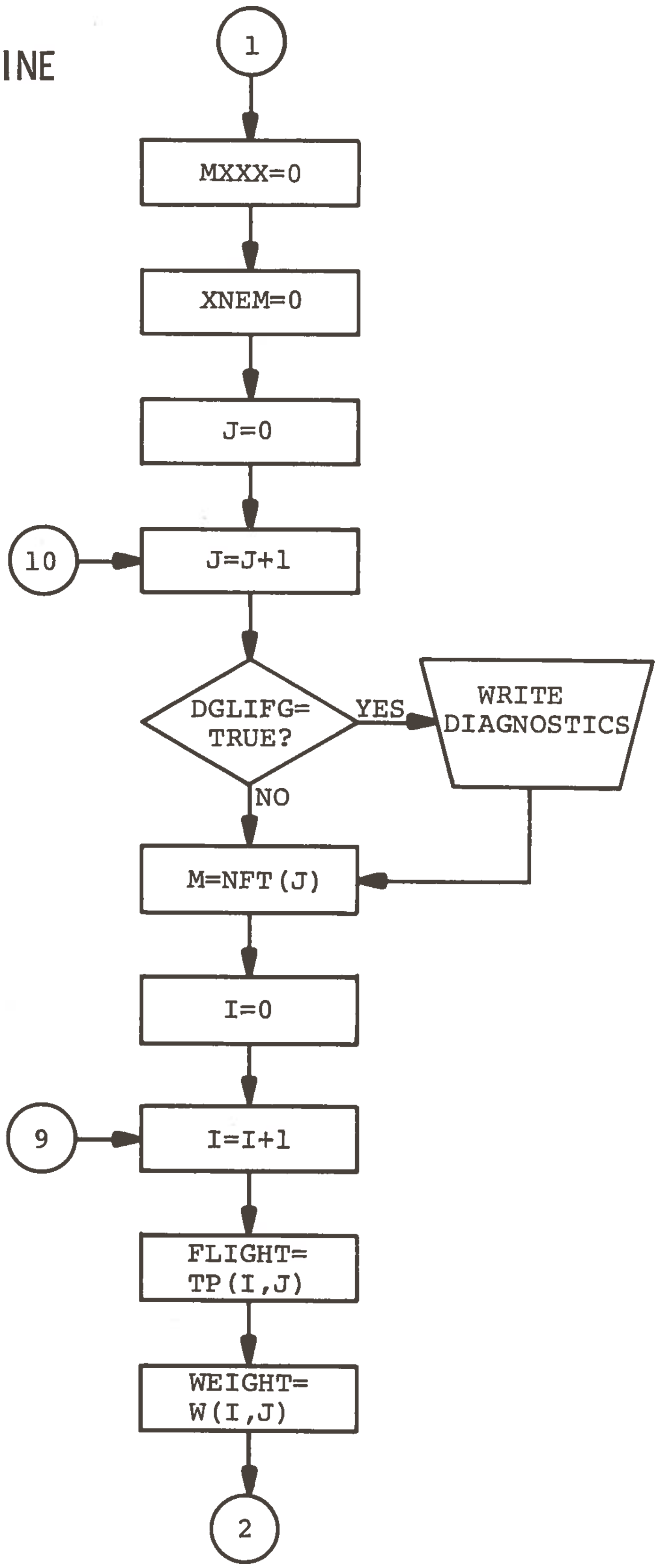
SUBROUTINE POINTS (CONT.)



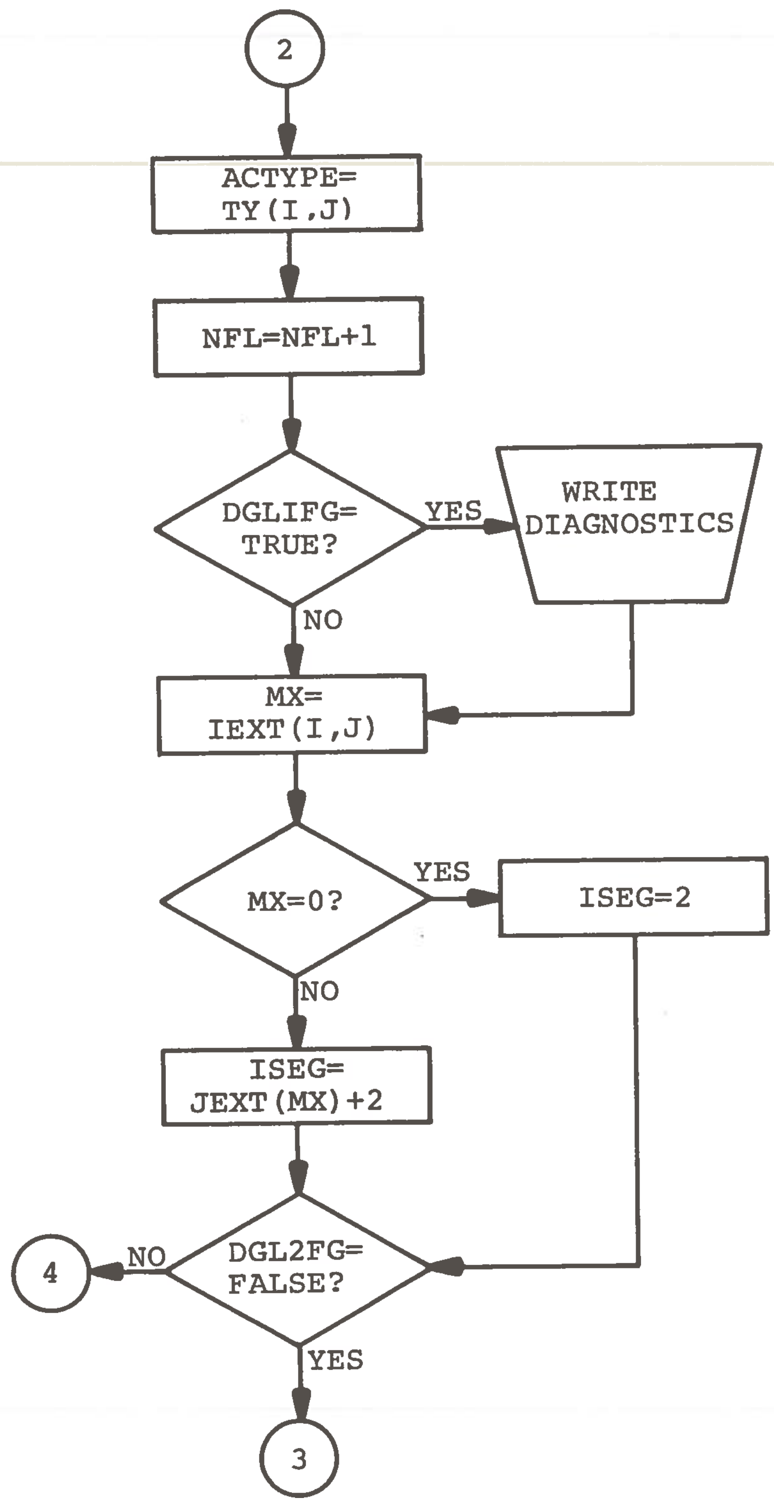
SUBROUTINE
POINTS
(CONT.)



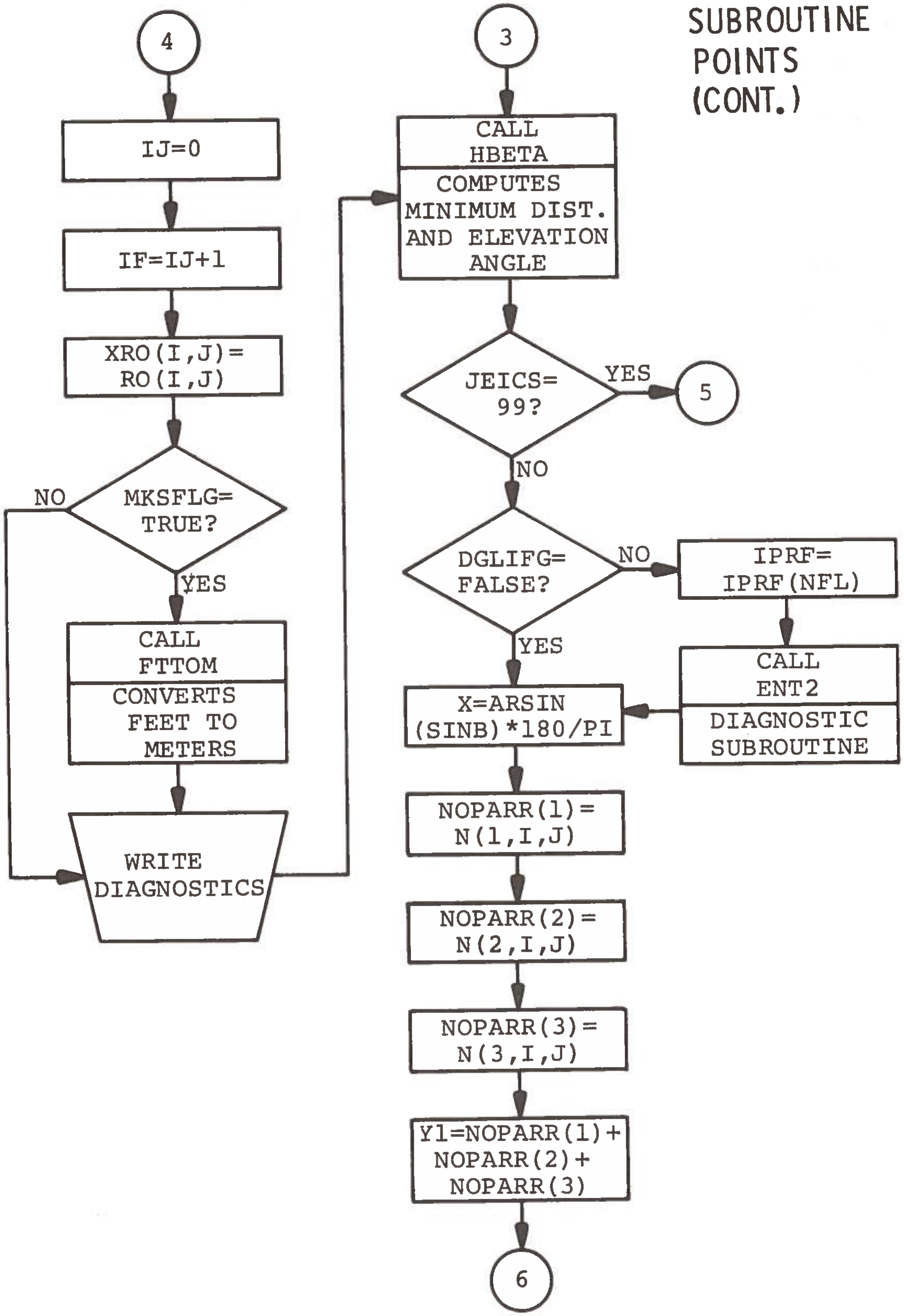
SUBROUTINE
POINTS
(CONT.)



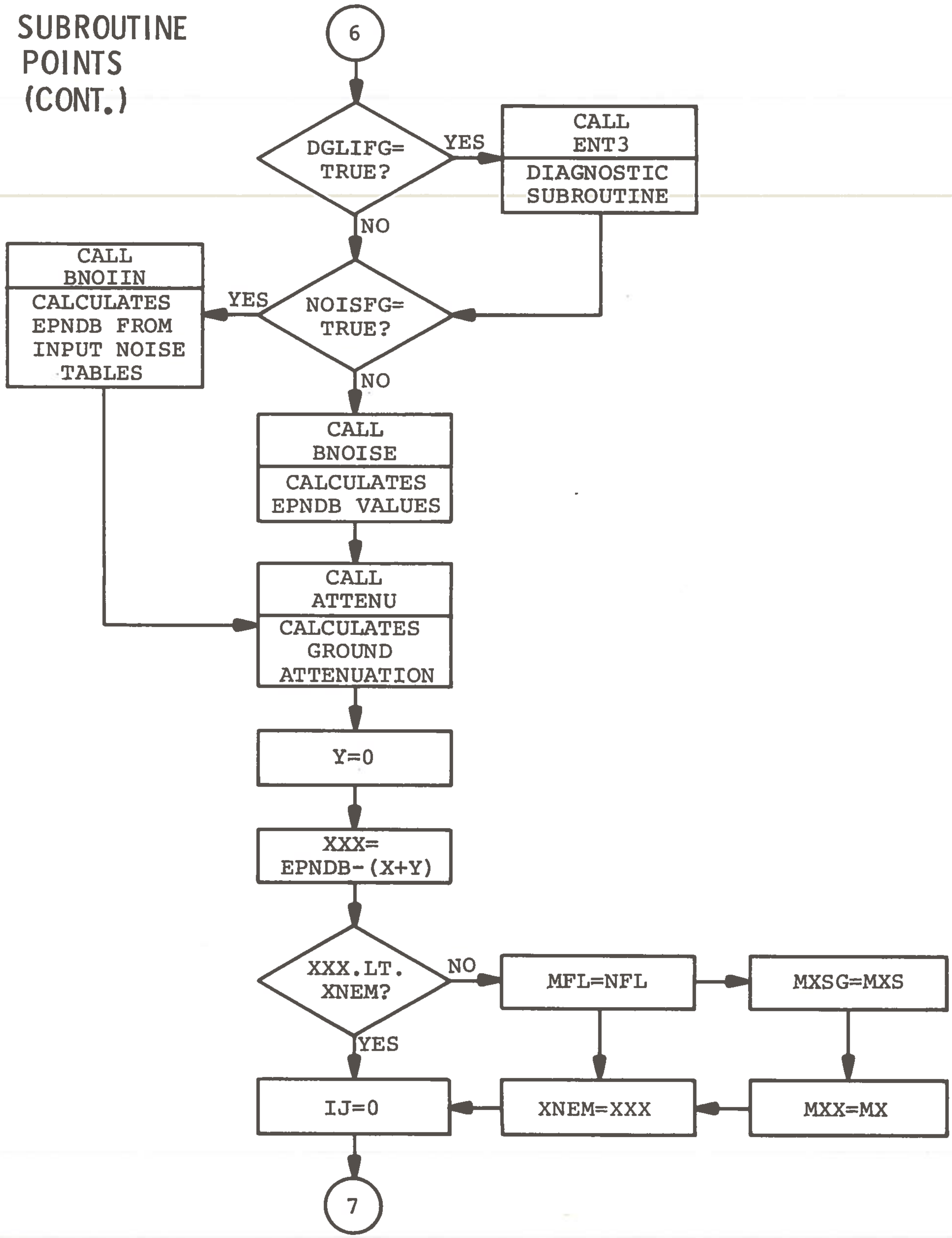
SUBROUTINE
POINTS
(CONT.)



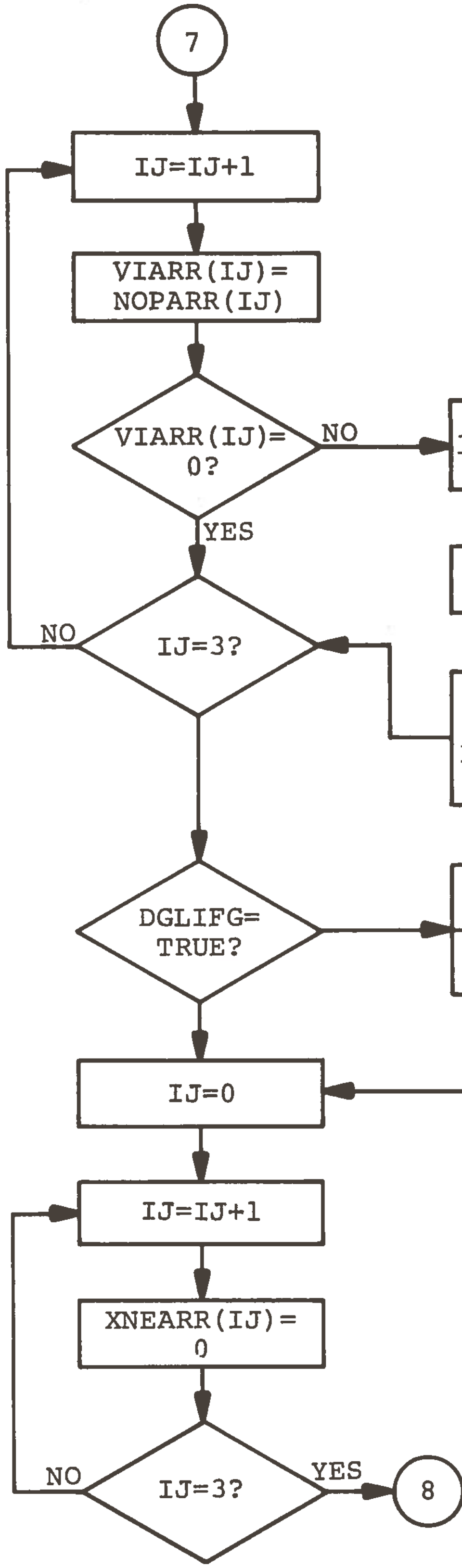
SUBROUTINE POINTS (CONT.)



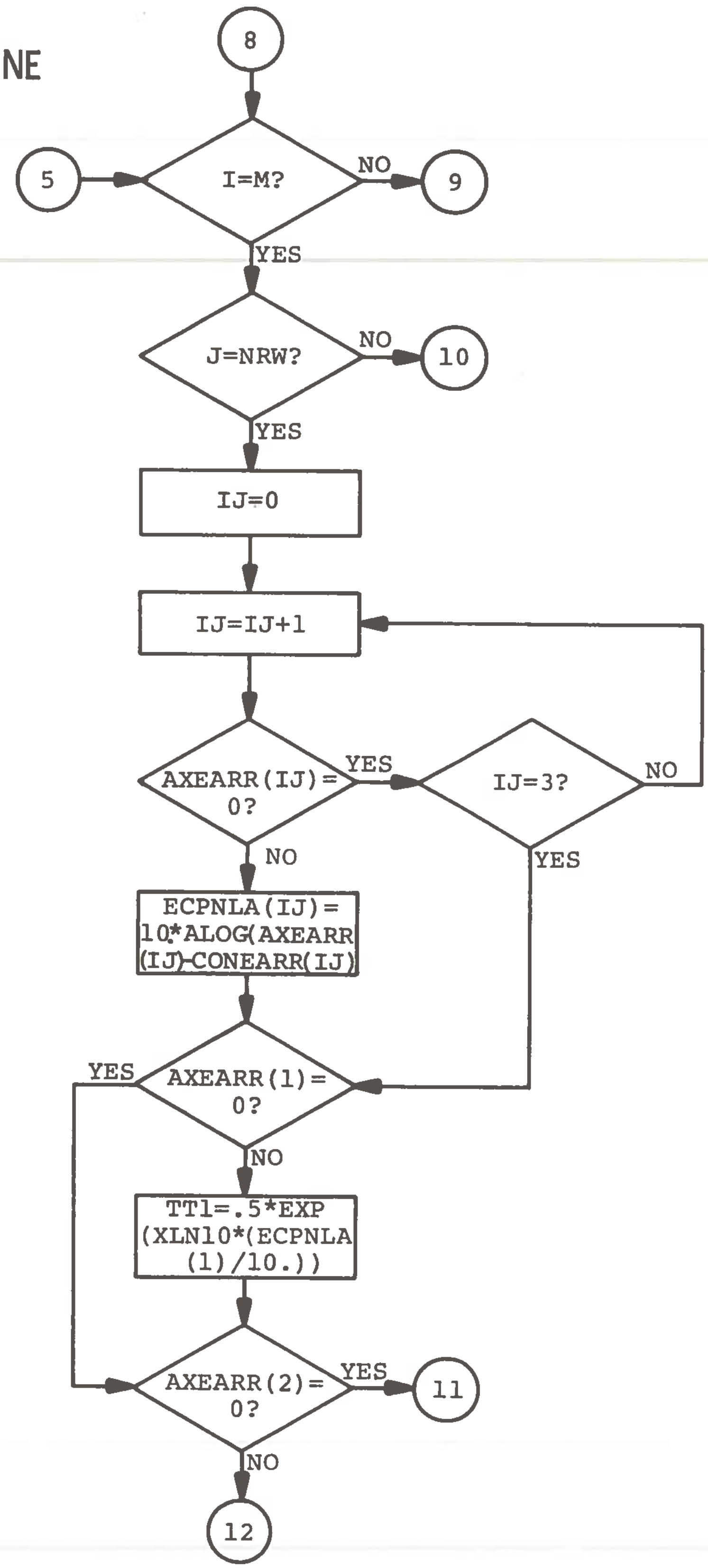
SUBROUTINE
POINTS
(CONT.)



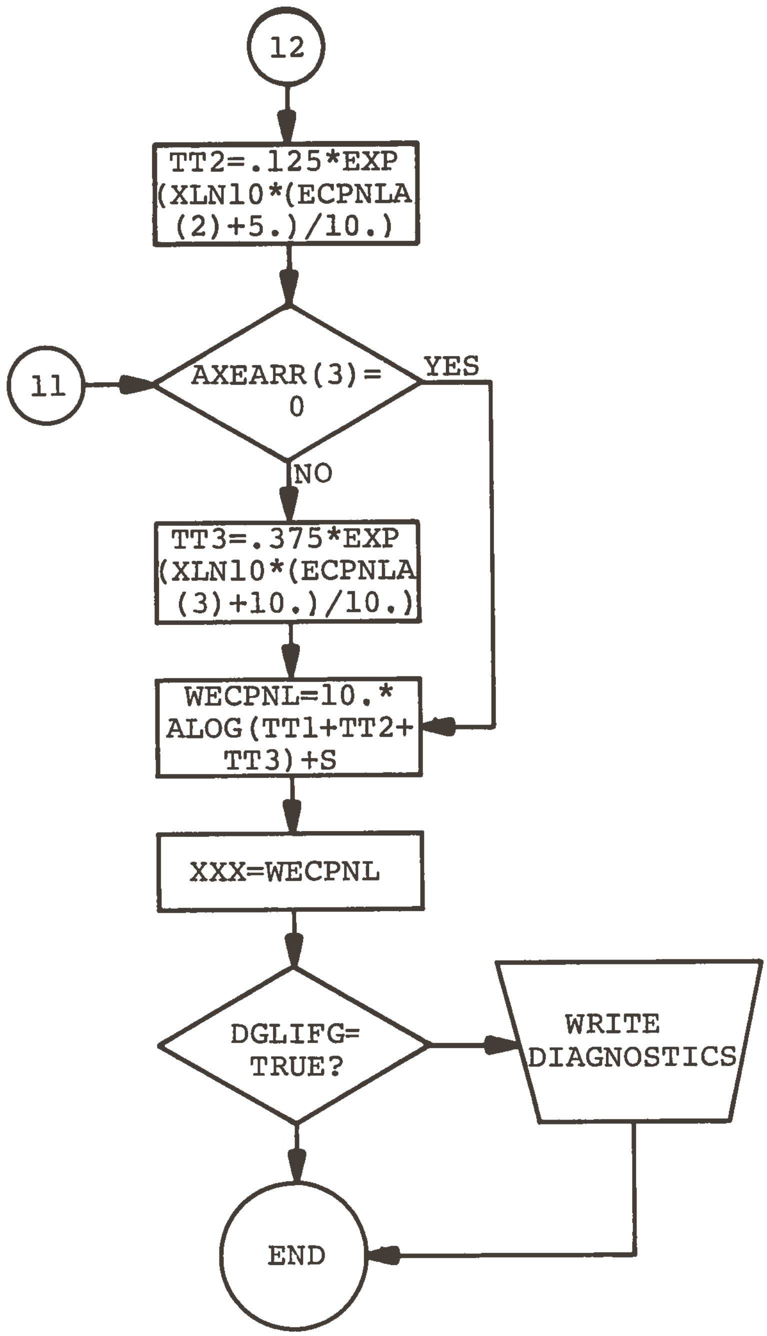
SUBROUTINE
POINTS
(CONT.)



SUBROUTINE POINTS (CONT.)



SUBROUTINE
POINTS
(CONT.)



APPENDIX B

PROGRAM LISTING

```
C NA MOD5 360 VERSION   P. SHAKIR   DOT/TSC/STC
C MAIN PROGRAM-AIRPORT NOISE ABATEMENT PROGRAM-VERSION MOD5
      COMMON/JOE2/NEFFLG,NOISFG
      COMMON/RYNMF/RYNAM(10)
      COMMON/FACTR/AXMAX,AXMIN,AYMAX,AYMIN,DELX,DELY,AINCHX,AINCHY,
     2  SETPLT
      COMMON/ARP/ARNV(20,25)
      COMMON/JOE3/JUMP
      COMMON/JOE1/WECPFG,CANCEL
      COMMON/ERRIN/JFICS
      COMMON/JOE/REFLIN,XCOORD(2),YCOORD(2),TITLE(12),SYMFLG
      COMMON/GDPAR/CX,CY,X1,Y1,NX,NY
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),XLM1(3,10),
     1  XLM2(3,10),STO(10),SL(10)
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/TRNS/PI
      COMMON/NNI/INN
      COMMON/NERAP/EPS1,EPS2,NOITT
      COMMON/OP/TOL,VAL,APPROX,BASEX,XDEL,MX
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/LOGFG3/RWAYFG,GRIDFG,CONTFG
      LOGICAL NEFFLG,NOISFG
      REAL N
      LOGICAL WECPFG,CANCEL
      LOGICAL SYMFLG
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      LOGICAL RWAYFG,GRIDFG,CONTFG
      LOGICAL SETPLT
C INITIALIZATION SECTION
      ITOM=0
    1 WRITE(6,193)
  193 FORMAT(1H1)
      INN=0
   50 CALL READIN(INDEX)
      JFICS=0
C     INDEX          SIGNIFICANCE
C       1            PROCES ENCOUNTERED IN READING INSTRUCTIONS/DATA
C       2            NEWSET READ NOT INCLUDING FIRST CASE IF INCLUDED
C       3            ENDRUN READ
      GO TO(100,200,300),INDEX
  100 IF(PTINFG) CALL READOU
      IF(.NOT. NOISFG) GO TO 110
      CALL RDNOIS
  110 CONTINUE
      IF(.NOT. CALFLG) GO TO 120
      IF(ITOM .EQ. 1) GO TO 120
      CALL INIPLT
      ITOM=1
  120 CONTINUE
```

```
      CALL EXTARR
      CALL DSTORE
C PRINT TABLE OF RUNWAYS AND FLIGHTS
      CALL SUMPE
      IF(GRIDFG) CALL POINTS
      IF(.NOT. CONTFG) GO TO 50
      INN=INN+1
      CALL CONTR
      GO TO 50
C NEWSET READ BUT NOT INCLUDING FIRST CASE IF IT SHOULD APPEAR THERE
  200 IF(CALFLG) CALL CALPLT
      GO TO 1
C ENDRUN READ--THIS SIGNALS END OF INPUT
  300 IF(CALFLG) CALL CALPLT
      IF(ITOM .EQ. 1) CALL FINPLT
      STOP
      END
      SUBROUTINE PDNOIS
C THIS SUBROUTINE READS SETS OF EPNDB VS DISTANCE CURVES INTO MEMORY.
C EACH SET CONSISTS OF ONE OR MORE CURVES CORRESPONDING TO DIFFERENT
C THRUSTS, A SET IS ASSOCIATED WITH A GIVEN AIRCRAFT TYPE NUMBER.
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/NSIN/XXIN(10,40),YYIN(10,40),THRTB(40),TKLDTB(40),TYPIN(40)
      COMMON/INSIN/NSEGIN(40),NATYTB(40),NCVTOT
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      DO 50 I=1,10
      DO 55 J=1,40
      XXIN(I,J)=0.
   55 YYIN(I,J)=0.
   50 CONTINUE
C READ IN NUMBER OF SETS
      READ(5,100) NOSETS
  100 FORMAT(8I10)
      NCURV=0
      DO 500 I=1,NOSETS
C READ IN NUMBER OF CURVES FOR EACH SET AND AIRCRAFT TYPE NUMBER
      READ(5,100) NACTP,NOTHRV
      DO 400 J=1,NOTHRV
      NCURV=NCURV+1
      NATYTB(NCURV)=NACTP
      READ(5,110) THR,TAKLD,TLILOG
  110 FORMAT(8F10.1)
      THRTB(NCURV)=THR
      TKLDTB(NCURV)=TAKLD
      TYPIN(NCURV)=TLILOG
      READ(5,100) NOXY
      READ(5,110) (XXIN(K,NCURV),K=1,NOXY)
      READ(5,110) (YYIN(K,NCURV),K=1,NOXY)
      NSEGIN(NCURV)=NOXY-1
  400 CONTINUE
  500 CONTINUE
      NCVTOT=NCURV
      IF(.NOT. DGL1FG) GO TO 600
```

```
C PRINT THE NOISE TABLES READ IN
      WRITE(6,801)
      WRITE(6,8020)
      DO 800 I=1,NCVTOT
      WRITE(6,802) NATYTB(I),THRTB(I),TKLDTB(I),TYPIN(I),
     1 (XXIN(L,I),L=1,10),(YYIN(L,I),L=1,10)
  802 FORMAT(1X,I6,F10.1,2F6.1,11H  DISTANCE  ,10F8.1/29X,11H      EPN DB,
     110F8.1//)
  800 CONTINUE
  801 FORMAT(120H1 TABULATION OF EPNDB VS DISTANCE CURVES READ BY PROGRA
     1M - THRUST IN PERCENT - TAKEOFF(0.)/LANDING(1.) - ARITH/LOG FOR X)
 8020 FORMAT(120H    TYPE     THRUST   TO/L   A/LG                    1
     12           3          4          5          6          7          8          9          10)
  600 CONTINUE
      RETURN
      END
      SUBROUTINE BNOIIN(TAC,HX ,NFL ,MXS)
C        TAC    AIRCRAFT TYPE NUMBER
C        HX     DISTANCE TO SEGMENT
C        NFL    FLIGHT NUMBER(CUMULATIVE)
C        MXS    SELECTED SEGMENT NUMBER OF FLIGHT PATH
C THIS SUBROUTINE CALCULATES EPNOB FROM INPUT NOISE VALUES
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/NSIN/XXIN(10,40),YYIN(10,40),THRTB(40),TKLDTB(40),TYPIN(40)
      COMMON/INSIN/NSEGIN(40),NATYTB(40),NCVTOT
      COMMON/BB/RO(3),FLIGHT,PEN,EPNDB
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      THRU=PTHRI(MXS,NFL)
      NTAC=TAC
      DO 100 I=1,NCVTOT
      IF(NTAC .EQ. NATYTB(I)) GO TO 200
  100 CONTINUE
      WRITE(6,110) NTAC
  110 FORMAT(21H AIRCRAFT TYPE NUMBER,I10,2X,39HNOT FOUND IN NOISE TABLE
     1S READ AS INPUT)
      WRITE(6,112)
  112 FORMAT(61H PROCESSING CONTINUES WITH ZERO NOISE ASSUMED FOR THIS F
     1LIGHT)
      XQX=0.
      GO TO 400
  200 IF(THRU .EQ. THRTB(I)) GO TO 250
      I=I+1
      IF(NTAC .EQ. NATYTB(I)) GO TO 200
      WRITE(6,210) THRU,NTAC
  210 FORMAT(16H THRUST VALUE OF,F10.4,2X,23H NOT FOUND FOR A/C TYPE,I5)
      WRITE(6,112)
      XQX=0.
      GO TO 400
  250 NCV=I
      CALL GNEMIN(NCV,HX,1,XQX)
  400 EPNDB=XQX
```

```
      IF(.NOT. DGL3FG) GO TO 600
      WRITE(6,450)
  450 FORMAT(12H FROM BNOIIN)
      WRITE(6,460) THRU,TAC,HX,XQX,EPNDB
  460 FORMAT(22H THRU,TAC,HX,XQX,EPNDB/1X,5F12.2)
      WRITE(6,465) NTAC,NCV
  465 FORMAT(10H NTAC,NCV /2I10)
  600 CONTINUE
      RETURN
      END
      SUBROUTINE GNFNIN(M,X,MVAR,XXX)
C THIS SUBROUTINE SELECTS THE CORRECT NOISE CURVE AND INTERPOLATES OR
C EXTRAPOLATES A Y VALUE FOR A GIVEN X VALUE. THE X VALUES MAY BE EITHER
C LINEAR OR LOGARITHMIC.
      COMMON/NSIN/XXIN(10,40),YYIN(10,40),THRTB(40),TKLDTB(40),TYPIN(40)
      COMMON/INSIN/NSEGIN(40),NATYTB(40),NCVTOT
      DIMENSION VAR(3)
      DATA VAR/4H  HX,4H PEN,4H THR/
      IF(X .NE. 0.) GO TO 4
      WRITE(6,2) VAR(MVAR)
    2 FORMAT(24H GNFNIN WAS CALLED WITH ,A4,2H=0)
      CALL EXIT
    4 N=NSEGIN(M)
      IL=1
      IR=N
      IM=II(IL,IR)
    5 XL=XXIN(IL,M)
      XM=XXIN(IM,M)
      XR=XXIN(IR+1,M)
   10 IF(X.GT.XM)GO TO 40
      IR=IM-1
      IM=II(IL,IR)
      IF(IL .EQ. IR) GO TO 100
      GO TO 5
   40 IL=IM
      IM=II(IL,IR)
      IF(IL .NE. IR) GO TO 5
  100 CONTINUE
      X1=XXIN(IM,M)
      X2=XXIN(IM+1,M)
      Y1=YYIN(IM,M)
      Y2=YYIN(IM+1,M)
      IF(TYPIN(M) .EQ. 0.) GO TO 150
      B=X2/X1
      A=(Y2-Y1)/ALOG10(B)
      B=Y1-A*ALOG10(X1)
      XXX=A*ALOG10(X)+B
      GO TO 200
  150 A=(Y2-Y1)/(X2-X1)
      B=Y1-A*X1
      XXX=A*X+B
  200 CONTINUE
      RETURN
      END
      BLOCK DATA
```

```
      COMMON/JOE3/JUMP
      COMMON/NEPAP/EPS1,EPS2,NOITT
      COMMON/DENOP/CONARR(3),S,NOPARR(3)
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IFT(400),MFL,MXX,MXSG
      COMMON/BK/DIAG,YES,MAYBE
      COMMON/ZXX/IPREL,IPP(150)
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/BB/RO(3),FLIGHT,PEN,EPNDB
      COMMON/CC/LAMBD1(3),LAMBD2(3),TAU1,TAU2,WEIGHT,ACTYPE,
     1DIST,SLOPE,UG(3),SINB,XDIST,GLUMP
      COMMON/NN/N(3,20,10),NRW,NET(10)
      COMMON/WTPEV/WECTR,PEGWT(2,2,11),NACT
      COMMON/TOOPRF/DECTR,TOPRF(2,8),NTOP
      COMMON/TRNS/PI
      COMMON/LN/VLN
      COMMON/NCOORD/NC
      COMMON/QR/TOL,VAL,APPROX,BASEX,XDEL,MX
      DIMENSION VT(3),AT(3),RC(3),RHO(3),B(3),C(3),S(3)
      DIMENSION UX(3),VX(3),GMX(3),XNORM(3)
      REAL MAYBE
      REAL LAMBD1,LAMBD2,NE,NEY1,NEY2
      REAL N,NT(3)
      INTEGER PREVEL,PREVSG
      DATA JUMP/0/
      DATA NOITT,EPS1,EPS2/5,.01,.01/
      DATA NACT/13/,WECTR/1000./,PEGWT/1.075,400.,4.825,800.,
     10.,200.,4.,400.,0.6,200.,5.,350.,1.3,200.,5.,307.5,.35,
     2100.,4.,170.,0.,60.,2.75,120.,-1.,40.25,.7,57.,-1.,24.,
     31.5,42.5,-1.,16.,1.5,27.,-1.,8.75,.9,15.,-.375,400.,
     41.75,670./,NTOP/8/,DECTR/1000./,TOPRF/.333,-832.5,
     5.209,-752.4,.169,-929.5,.133,-904.4,
     6.105,-966.,.0875,-927.5,.0895,-223.75,.0812,-324.8/,
     7PI/3.1415927/,VLN/1.0E30/
      DATA NC/3/,YES/1HY/,MAYBE/1HM/
      DATA CONARR/36.35,30.33,35.11/
      DATA S/0./
      END
      SUBROUTINE KGTOLB(A,IDIM)
      DIMENSION A(1)
      DO 1 I=1,IDIM
    1 A(I)=A(I)*2.204618750
      RETURN
      END
      SUBROUTINE LBTOKG(A,IDIM)
      DIMENSION A(1)
      DO 1 I=1,IDIM
    1 A(I)=A(I)*.4535924277
      RETURN
      END
      SUBROUTINE MTOFT(A,IDIM)
      DIMENSION A(1)
      DO 1 I=1,IDIM
```

```
    1 A(I)=A(I)*3.280833333
      RETURN
      END
      SUBROUTINE FTTOM(A,IDIM)
      DIMENSION A(1)
      DO 1 I=1,IDIM
    1 A(I)=A(I)*.3048006096
      RETURN
      END
      SUBROUTINE POINTS
C SUBROUTINE POINTS FINDS THE NE AT THE POINTS OF A GRID.              00053600
C THE GRID MUST BE COMPOSED OF A FINITE NUMBER OF                       00053650
C EQUALLY SPACED ABSCISSAE AND ORDINATES.THE SPACING                    00053700
C NEED NOT BE THE SAME IN EACH DIRECTION.                                  05375
C CX AND CY ARE THE NUMBER OF ABSCISSAE AND ORDINATES.                  00053800
C THEY MAY BE POSITIVE OR NEGATIVE BUT SHOULD NOT BE ZERO.              00053850
C X1 AND Y1 ARE THE STARTING VALUES FOR X AND Y.SUBSEQUENT              00053900
C VALUES ARE FORMED BY INCREMENTING BY CX AND CY.                       00053950
C NX AND NY ARE THE TOTAL NUMBER OF X AND Y VALUES                      00054000
C TO BE CONSIDERED.                                                        54050
      COMMON/ARP/ARNV(20,25)
      COMMON/JOE1/WECPFG,CANCEL
      COMMON/GDPAR/CX,CY,X1,Y1,NX,NY
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/BB/R0(3),FLIGHT,PEN,EPNDB                                     54100
      DIMENSION ARRLOC(20)
      LOGICAL WECPFG,CANCEL
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      IF(.NOT. PTGDFG) GO TO 60
      WRITE(6,10)
      ARRLOC(1)=X1
      DO 40 I=2,NX
   40 ARRLOC(I)=ARRLOC(I-1)+CX
      IF(.NOT. MKSFLG) GO TO 18
      CALL FTTOM(ARRLOC,NX)
   18 CONTINUE
      WRITE(6,20) (ARRLOC(I),I=1,NX)
      WRITE(6,21)
   60 CONTINUE
      R0(2)=Y1-CY
      DO 100 JS=1,NY
      R0(2)=R0(2)+CY
      R0(1)=X1-CX
      DO 200 IS=1,NX
      R0(1)=R0(1)+CX
      R0(3)=0.
      IF(WECPFG) GO TO 800
      CALL FXPNF(DU,X)
      GO TO 810
  800 CALL FXPWF(DU,X)
  810 CONTINUE
```

```
      ARRLOC(IS)=X
      ARNV(IS,JS)=X
  200 CONTINUE
      IF(.NOT. PTGDFG) GO TO 100
      YR0=R0(2)
      IF(.NOT. MKSFLG) GO TO 65
      CALL FTTOM(YR0,1)
   65 WRITE(6,30) YR0,(ARRLOC(IS),IS=1,NX)
      WRITE (7,35) (ARRLOC(IS),IS=1,NX)
   35 FORMAT(16F5.1/16F5.1)
  100 CONTINUE
   10 FORMAT(37H1***NF/WECPNL/NEF VALUES IN A GRID***//)
   20 FORMAT(7H X VALS,1X,16F7.0/8X,4F7.0)
   21 FORMAT(/8H   Y VALS)
   30 FORMAT(/1X,F7.0,16F7.1/8X,4F7.1)
      RETURN                                                             55000
      END                                                                55050
      SUBROUTINE GENFN(M,X,MVAR,XXX)
      DIMENSION TYPE(20),NSEGS(20),VAR(10)
      DIMENSION XX(10,20),YY(10,20)
      DATA TYPE/9*1.,4*0.,2*1.,5*1./
      DATA XX/
     1200.,400.,700.,4000.,10000.,16000.,4*0.,200.,600.,1000.,           00037100
     22000.,4000.,6000.,10000.,16000.,2*0.,200.,1000.,16000.,            00037150
     37*0.,200.,400.,1000.,5000.,7000.,10000.,4*0.,200.,400.,1000.,      00037200
     42000.,4000.,10000.,4*0.,200.,400.,1000.,2000.,4000.,5*0.,          00037250
     5200.,3000.,16000.,7*0.,200.,3000.,16000.,7*0.,200.,2000.,          00037300
     616000.,7*0.,20.,40.,60.,80.,100.,5*0.,20.,40.,60.,80.,             00037350
     7100.,5*0.,20.,40.,60.,80.,100.,5*0.,20.,40.,60.,80.,100.,          00037400
     85*0.,200.,400.,1000.,2000.,4000.,20000.,4*0.,200.,400.,1000.,      00037450
     92000.,4000.,20000.,4*0.,200.,500.,1000.,4000.,20000.,              00037500
     15*0.,200.,500.,1000.,4000.,16000.,35*0./
      DATA YY/129.38,
     1124.38,119.06,100.,91.25,87.5,4*0.,123.13,113.44,108.13,           00037650
     298.75,86.88,80.31,74.06,70.,2*0.,121.89,109.38,83.13,7*0.,         00037700
     3115.5,109.38,99.38,77.5,73.75,70.,4*0.,117.5,112.5,105.63,         00037750
     499.38,92.5,81.88,4*0.,107.5,102.5,95.,                              0037800
     588.13,78.8,5*0.,136.88,118.75,105.63,7*0.,126.25,107.5,            00037850
     695.,7*0.,115.63,95.,79.38,7*0.,108.75,116.25,121.13,               00037900
     7123.63,124.,5*0.,106.88,111.25,                                       37950
     8114.7,117.7,120.,5*0.,104.75,108.5,111.75,114.88,117.,             00038000
     95*0.,102.50,106.5,110.13,112.5,114.,5*0.,1.88,2.5,6.88,            00038050
     111.88,14.38,15.,4*0.,1.88,2.5,5.63,8.13,9.69,10.,4*0.,             00038100
     2104.,100.,95.,83.75,70.,5*0.,100.,93.,87.5,73.8,                   00038150
     360.,35*0./
      DATA NSEGS/5,7,2,5,5,4,3*2,4*4,2*5,2*4,3*0/
      DATA VAR/4H  HX,4H PEN,4H THR,4H    ,4H    ,4H    ,4H    ,
     14H    ,4H    ,4H    /
    2 FORMAT(22H GENFN WAS CALLED WITH  ,A4,2H=0)                        00038400
      IF(X.NE.0.)GO TO 4                                                    38350
C OR EXTRAPOLATION.TO USE THIS FN PICK TEN OR LESS POINTS                00038500
C FROM A GRAPH IN SUCH A WAY THAT THE GRAPH IS FAIRLY LINEAR             00038550
C IN THE INTERVALS THE POINTS DEFINE,E.G.                                  038600
C                                                                          38650
C                                                                          38700
```

```
C
C   Y
C   *
C   *
C   *       +
C   *    +     +
C   *  +          +
C   * +               +
C   *                      +
C   *                             +
C   ************************************** X
C
C THE GRAPH MAY BE LINEAR OR LOGARITHMIC IN THE X DIRECTION
C BUT IS ASSUMED TO BE LINEAR IN THE Y DIRECTION THE ARRAY
C TYPE(FUNCTION NO.) SHOULD BE 1(ONE) TO INDICATE LOG OR 0 TO
C INDICATE LINEAR.20 FUNCTIONS ARE ALLOWED.
C THE POINT COORDINATES OF THE TEN POINTS ARE STORED IN THE
C ARRAYS XX(FN NO.,POINT NO.) AND YY(FN NO.,POINT NO.).NSEGS(FN NO.)
C SHOULD CONTAIN ONE LESS THAN THE NUMBER OF POINTS CHOSEN.
C GENFN ITERATES ON THE QUESTION  DOES X LIE IN THE RIGHT OR LEFT
C HALF OF THE INTERVALS THAT HAVE NOT YET BEEN EXCLUDED
C UNTIL ONLY ONE INTERVAL REMAINS.
      WRITE(6,2) VAR(MVAR)
      CALL EXIT
    4 N=NSEGS(M)
      IL=1
      IR=N
C FIND THE MIDDLE INTERVAL
      IM=II(IL,IR)
    5 XL=XX(IL,M)
      XM=XX(IM,M)
      XR=XX(IR+1,M)
C DOES X LIE IN THE RIGHT HALF
   10 IF(X.GT.XM)GO TO 40
C REDEFINE THE RIGHTMOST INTERVAL AND FIND THE NEW MIDDLE
      IR=IM-1
      IM=II(IL,IR)
C HAS THE CORRECT INTERVAL BEEN LOCATED
      IF(IL.EQ.IR)GO TO 100
      GO TO 5
C REDEFINE THE LEFTMOST INTERVAL TO BE THE MIDDLE
C AND FIND THE NEW MIDDLE
   40 IL=IM
      IM=II(IL,IR)
C IF THE CORRECT INTERVAL HAS NOT BEEN LOCATED TRY AGAIN
      IF(IL.NE.IR)GO TO 5
  100 CONTINUE
C IT IS KNOWN THAT XX(IM,M) LE X LE XX(IM+1,M)
C IT IS POSSIBLE TO INTERPOLATE BETWEEN THESE TWO
C VALUES.NOTICE THAT IF X IS,SAY,GREATER THAN ANY OF THE
C MEMBERS OF XX GENFN WILL AUTOMATICALLY EXTRAPOLATE
C SINCE THE RIGHTMOST INTERVAL WILL BE CHOSEN FOR THE
C  INTERPOLATION .
      X1=XX(IM,M)
      X2=XX(IM+1,M)
```

```
      Y1=YY(IM,M)
      Y2=YY(IM+1,M)
      IF(TYPE(M).EQ.0.)GO TO 150
      B=X2/X1
      A=(Y2-Y1)/ALOG10(B)
      B=Y1-A*ALOG10(X1)
      XXX=A*ALOG10(X)+B
      GO TO 200
  150 A=(Y2-Y1)/(X2-X1)
      B=Y1-A*X1
      XXX=A*X+B
  200 CONTINUE
      RETURN
      END
      FUNCTION II(N1,N2)
      II=FLOAT(N2-N1+1)/2.
      II=II+N1
      RETURN
      END
      SUBROUTINE VUNIT(C,A)
      COMMON/NCOORD/NC
      DIMENSION C(1),A(1)
C THIS SUBROUTINE FORMS A UNIT VECTOR U FROM ANY VECTOR C
      CALL VMAG(A,VM)
      IF(VM.EQ.0.)VM=1.
      DO 10 I=1,NC
      C(I)=A(I)/VM
   10 CONTINUE
      RETURN
      END
      FUNCTION VDOT(A,B)
      COMMON/NCOORD/NC
      DIMENSION A(1),B(1)
C THIS SUBROUTINE PERFORMS THE VECTOR DOT PRODUCT A.B
      VDOT=0.
      DO 10 I=1,NC
   10 VDOT=VDOT+A(I)*B(I)
      RETURN
      END
      SUBROUTINE VCROS(C,A,B)
      COMMON/NCOORD/NC
      DIMENSION A(1),B(1),C(1)
C THIS SUBROUTINE PERFORMS THE VECTOR CROSS PRODUCT,I.E.
C C=AXB. C MUST BE DIFFERENT FROM A OR B.
      C(1)=A(2)*B(3)-A(3)*B(2)
      C(2)=A(3)*B(1)-A(1)*B(3)
      C(3)=A(1)*B(2)-A(2)*B(1)
      RETURN
      END
      SUBROUTINE VADD(C,A,B)
      COMMON/NCOORD/NC
      DIMENSION A(1),B(1),C(1)
C THIS SUBROUTINE PERFORMS VECTOR ADDITON AS C=A+B
      DO 10 I=1,NC
   10 C(I)=A(I)+B(I)
```

```
      SL(J)=0.
      N(1,I,J)=0.
      N(2,I,J)=0.
  608 N(3,I,J)=0.
      GO TO 10
  132 CONTINUE
  133 CONTINUE
  134 CONTINUE
  135 CONTINUE
  136 CONTINUE
      JUMP=0
      INDEX=2
      RETURN
  140 CONTINUE
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      IST=IST+2
      INDEX=3
      RETURN
  154 CONTINUE
  155 CONTINUE
      IF(.NOT. DGL3FG) GO TO 188
      WRITE(6,6000)
11000 FORMAT(4E16.8)
      WRITE(6,11000) ZETI(1,1),ZETI(2,1),ZETI(3,1),ZETI(4,1)
      WRITE(6,11000) ZETI(1,2),ZETI(2,2),ZETI(3,2),ZETI(4,2)
      WRITE(6,11000) ZETI(1,3),ZETI(2,3),ZETI(3,3),ZETI(4,3)
      WRITE(6,11000) THETAI(1,1),THETAI(1,2),THETAI(1,3),THETAI(1,4)
      WRITE(6,11000) THETAI(2,1),THETAI(2,2),THETAI(2,3),THETAI(2,4)
      WRITE(6,11000) THETAI(3,1),THETAI(3,2),THETAI(3,3),THETAI(3,4)
      WRITE(6,11000) THETAI(4,1),THETAI(4,2),THETAI(4,3),THETAI(4,4)
      WRITE(6,11000) PTHRI(1,1),PTHRI(1,2),PTHRI(1,3),PTHRI(1,4)
      WRITE(6,11000) PTHRI(2,1),PTHRI(2,2),PTHRI(2,3),PTHRI(2,4)
      WRITE(6,11000) PTHRI(3,1),PTHRI(3,2),PTHRI(3,3),PTHRI(3,4)
      WRITE(6,11000) PTHRI(4,1),PTHRI(4,2),PTHRI(4,3),PTHRI(4,4)
      WRITE(6,11000) RADI(1,1),RADI(2,1),RADI(3,1),RADI(4,1)
      WRITE(6,11000) RADI(1,2),RADI(2,2),RADI(3,2),RADI(4,2)
      WRITE(6,11000) RADI(1,3),RADI(2,3),RADI(3,3),RADI(4,3)
      WRITE(6,11000) CX,CY,X1,Y1
      WRITE(6,11000) VAL,XDFL,BASEX,APPROX
      WRITE(6,18000) MKSFLG,HESGFG,PTALFG,PTTBFG,PTINFG,DGL0FG,DGL1FG,
     2DGL2FG,DGL3FG,CALFLG,SSIFLG,LSIFLG,LINEFG,PTOUFG,PTGDFG,PTCOFG,
     3CONTFG,SYMFLG
18000 FORMAT(19L4)
      WRITE(6,12000) NUMRUN,NUMFLI,NEXTS,N(1,1,1),N(2,1,1),N(3,1,1)
12000 FORMAT(3I5,3F10.5)
      WRITE(6,11000) XLM1(1,1),XLM2(1,1),XLM1(2,1),XLM2(2,2),XLM1(3,1),
     2XLM1(4,1),XLM2(4,1),STO(1),SL(1),TP(1,1),TY(1,1)
      WRITE(6,6000)
  188 CONTINUE
      IF(IPHIL.NE.500) RETURN
      IF(.NOT.CANCEL) RETURN
      STOP
      END
```

```
      IF(ALPHA.EQ.EXTNSN(1)) GO TO 134
      IF(ALPHA.EQ.GRID(1)) GO TO 95
      IF(ALPHA.EQ.CONTOR(1)) GO TO 110
500   NEXTS=0
      NUMRUN=0
      IF(.NOT. MKSFLG) GO TO 6053
      NUMFLI=0
      NFLGHT=0
      DGL1FG=.FALSE.
      DGL2FG=.FALSE.
      DGL3FG=.FALSE.
      CONVRT=.FALSE.
      CANCEL=.FALSE.
      WFCPFG=.FALSE.
      NEFFLG=.FALSE.
      SETPLT=.FALSE.
      LSIFLG=.FALSE.
      SSIFLG=.TRUE.
      MKSFLG=.TRUE.
      HESGFG=.FALSE.
      PTALFG=.TRUE.
      PTTBFG=.TRUE.
      PTINFG=.TRUE.
      DGLOFG=.TRUE.
      PTOUFG=.TRUE.
      PTGDFG=.TRUE.
      PTCOFG=.TRUE.
      CONTFG=.FALSE.
      NOISFG=.FALSE.
      JUMP=1
      SYMFLG=.FALSE.
      LINEFG=.FALSE.
600   CONTINUE
      DO 610 J=1,150
      DO 610 I=1,10
      RADI(I,J)=0.
      ZETI(I,J)=0.
      THETAI(I,J)=0.
      PTHRI(I,J)=0.
      NSG(J)=0
      CLIMBA(J)=0.
610   NET(I)=0
      DO 608 I=1,20
      DO 608 J=1,10
      TP(I,J)=0.
      PT(I,J)=0.
      W(I,J)=0.
      TY(I,J)=0.
      XLM1(1,J)=0.
      XLM1(2,J)=0.
      XLM1(3,J)=0.
      XLM2(1,J)=0.
      XLM2(2,J)=0.
      XLM2(3,J)=0.
      STO(J)=0.
```

```
C
 1111 CONTINUE
      STORE(IST+1)=DUMMY1
      IF(JUMP2.LT.9) GO TO 11111
      STORE(IST+ 2)=DUMMY2
      STORE(IST+3)=-1.
       IST=IST+4
      GO TO 111
11111 STORE(IST+2)=-1.
      IST=IST+3
      GO TO 111
  130 CONTINUE
C     PROCESS THIS CASE
      INDEX=1
      IF(CONVRT) GO TO 6053
      CALL MTOFT(RADI,1500)
      CALL MTOFT(STO,NUMRUN)
      CALL MTOFT(SL  ,NUMRUN)
      CALL MTOFT(XLM1,30)
      CALL MTOFT(XLM2,30)
      CALL KGTOLB(W,200)
      CONVRT=.TRUE.
 6053 CONTINUE
      NRW=NUMRUN
      JUMP=1
      ISTART=1
      WRITE(6,6000)
 6000 FORMAT(1H1)
 4000 JOE=0
      IEND=ISTART+3
      DO 150 ILOOP=ISTART,IEND
      JOE=JOE+1
      IF(STORE(ILOOP).EQ.-1.) GO TO 4005
  150 CONTINUE
 4005 K=ISTART+JOE-2
      WRITE(6,1003)(STORE(ILOOP),ILOOP=ISTART,K)
      IF(STORE(ISTART) .EQ. HEADER) WRITE(6,5050) (TITLE(I),I=1,12)
      ISTART=K+2
      IF(ISTART.GE.IST) GO TO 155
 4010 FORMAT(A6,2F16.8)
      GO TO 4000
  131 CONTINUE
      ASSIGN 131 TO ERROR
      GROUP=8
C     THIS IS A REPEAT CASE WITH CERTAIN CHANGES
      DO 1311 J=1,NFLGHT
      IF(CLIMBA(J).NE.0.) GO TO 1311
      THETAI(2,J)=0.
 1311 CONTINUE
      READ(5,1002) ALPHA
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      IST=IST+2
      IF(ALPHA.EQ.FLIGHT(1)) GO TO 132
      IF(ALPHA.EQ.RUNWAY(1)) GO TO 133
```

```
      CONTEG=.TRUE.
      IF(J.EQ.54) IST=IST+2
C
  111 READ(5,1002) ALPHA,DUMMY1,DUMMY2
      WRITE(6,1003) ALPHA,DUMMY1,DUMMY2
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      ASSIGN 111 TO ERROR
      GROUP=7
      DO 112 J=54,63
      JUMP2=J
      IF(ALPHA.EQ.PSEUDO(J)) JUMP2=J-53
      IF(JUMP2.NE.J) GO TO (110,113,114,115,116,117,118,119,120,121,122,
     1123,124),JUMP2
  112 CONTINUE
      IST=IST+2
      GO TO 2000
  113 VAL=DUMMY1
      GO TO 1111
  114 XDEL=DUMMY1
      IF(.NOT. MKSFLG) GO TO 1111
      CALL MTOFT(XDEL,1)
      GO TO 1111
  115 BASEX=DUMMY1
      IF(.NOT. MKSFLG) GO TO 1111
      CALL MTOFT(BASEX,1)
      GO TO 1111
  116 MX=IFIX(DUMMY1)
      GO TO 1111
  117 TOL=DUMMY1
      GO TO 1111
  118 APPROX=DUMMY1
      IF(.NOT. MKSFLG) GO TO 1111
      CALL MTOFT(APPROX,1)
      GO TO 1111
  119 SYMFLG=.TRUE.
      GO TO 1111
  120 XCOORD(1)=DUMMY1
      XCOORD(2)=DUMMY2
      LINEFG=.TRUE.
      GO TO 1111
  121 YCOORD(1)=DUMMY1
      YCOORD(2)=DUMMY2
      LINEFG=.TRUE.
      GO TO 1111
  122 REFLIN=1.
C     IMPLIES REFLECTION IS ABOUT X AXIS
      LINEFG=.TRUE.
      GO TO 1111
  123 REFLIN=2.
      LINEFG=.TRUE.
C     IMPLIES REFLECTION IS ABOUT Y AXIS
      GO TO 1111
  124 CONTINUE
      GO TO 1111
```

```
      GO TO 866
   93 CONTINUE
      GO TO 866
  866 CONTINUE
      STORE(IST+1)=DUMMY1
      STORE(IST+2)=-1.
      IST=IST+3
      GO TO 86
C     WE ARE NOW PROCESSING THE GRID INFORMATION
   95 CONTINUE
      ASSIGN 96 TO ERROR
      GROUP=6
      GRIDFG=.TRUE.
   96 READ(5,1002) ALPHA,DUMMY1
      WRITE(6,1003) ALPHA,DUMMY1
      STORE(IST)=ALPHA
      DO 97 J=45,53
      JUMP2=J
      IF(ALPHA.EQ.PSEUDO(J)) JUMP2=J-44
      IF(JUMP2.NE.J) GO TO ( 95,99,100,101,102,103,104,105,106),JUMP2
   97 CONTINUE
      STORE(IST+1)=-1.
      IST=IST+2
      GO TO 2000
   99 CX=DUMMY1
      IF(.NOT. MKSFLG) GO TO 966
      CALL MTOFT(CX,1)
      GO TO 966
  100 CY=DUMMY1
      IF(.NOT. MKSFLG) GO TO 966
      CALL MTOFT(CY,1)
      GO TO 966
  101 X1=DUMMY1
      IF(.NOT. MKSFLG) GO TO 966
      CALL MTOFT(X1,1)
      GO TO 966
  102 Y1=DUMMY1
      IF(.NOT. MKSFLG) GO TO 966
      CALL MTOFT(Y1,1)
      GO TO 966
  103 NX=IFIX(DUMMY1)
      GO TO 966
  104 NY=IFIX(DUMMY1)
      GO TO 966
  105 CONTINUE
      GO TO 966
  106 CONTINUE
      GO TO 966
  966 CONTINUE
      STORE(IST+1)=DUMMY1
      STORE(IST+2)=-1.
      IST=IST+3
      GO TO 96
C     WE ARE NOW PROCESSING THE CONTOURS
  110 CONTINUE
```

```
      STORE(IST+2)=-1.
      IST=IST+3
      GO TO 66
   85 CONTINUE
C     WE ARE NOW IN THE EXTENSION SECTION
      NEXTS=NEXTS+1
      ASSIGN 86 TO ERROR
      GROUP=5
C   THE FOLLOWING CARD HAS BEEN TEMPORARILY COMMENTED
C     IF(J.NE.39) GO TO 86
      IF(ICOFG .EQ. 0) GO TO 86
      IF(RADI(NEXTS+1,NFLGHT).NE.0.) ZETI(NEXTS+1,NFLGHT)=ZETI(NEXTS+1,
     2NFLGHT)/180.*3.1415927
      LOC=NEXTS+1
      IF(RADI(NEXTS+1,NFLGHT).EQ.0..AND.MKSFLG) CALL MTOFT(ZETI(LOC,NFLG
     2HT),1)
      IST=IST+2
   86 ICOFG=1
      READ(5,1002) ALPHA,DUMMY1
      WRITE(6,1003) ALPHA,DUMMY1
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      DO 87 J=39,44
      JUMP2=J
      IF(ALPHA.EQ.PSEUDO(J)) JUMP2=J-38
      IF(JUMP2.NE.J) GO TO (85,88,89,90,91,92,93),JUMP2
   87 CONTINUE
      IST=IST+2
      IF(RADI(NEXTS+2,NFLGHT).NE.0.) ZETI(NEXTS+2,NFLGHT)=ZETI(NEXTS+2,
     2NFLGHT)/180.*3.1415927
      LOC=NEXTS+2
      IF(RADI(NEXTS+2,NFLGHT).EQ.0..AND.MKSFLG) CALL MTOFT(ZETI(LOC,NFLG
     2HT),1)
15000 FORMAT(61H NUMBER OF EXTENSIONS PROVIDED DOES NOT AGREE WITH NO. S
     2TATED)
      IF(NEXTS.NE.NOFXTS) WRITE(6,15000)
      IF(NEXTS .NE.NOFXTS) CANCEL=.TRUE.
      GO TO 2000
C     EXTENT   GROUND PROJECTION DISTANCE OF LINEAR SEGMENT OR ANGULAR EXTENT
C     IN DEGREES IF HELICAL SEGMENT
   88 ZETI(NEXTS+2,NFLGHT)=DUMMY1
       GO TO 866
   89 THETAI(NEXTS+2,NFLGHT)=DUMMY1/180.*3.1415927
C     THIS WAS READ IN AS DEGREES AND IS NOW IN RADIANS
       GO TO 866
   90 PTHRI(NEXTS+2,NFLGHT)=DUMMY1
C     PERCENTAGE THRUST OVER EACH SEGMENT
       GO TO 866
   91 RADI(NEXTS+2,NFLGHT)=DUMMY1
C     IF RADI =  0  THIS IS A LINEAR SEGMENT  OTHERWISE EXTENT IS IN DEGREES
C THE FOLLOWING CARD HAS BEEN TEMPORARILY COMMENTED
C     IF(RADI.NE.0.) HESGFG=.TRUE.
C
       GO TO 866
   92 CONTINUE
```

```
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 666
   74 N(1,NUMFLI,NUMRUN)=DUMMY1
      ICAT(GROUP)=ICAT(GROUP)+1
C     NUMBER OF DAY OPERATIONS FOR EACH FLIGHT AND EACH RUNWAY
      GO TO 666
   75 IF(WECPEG) GO TO 755
      WRITE(6,10000)
10000 FORMAT(41H EVENING OPERATIONS CURRENTLY NOT ALLOWED)
      WRITE(6,10001)
10001 FORMAT(59H THESE FLIGHTS WILL BE IGNORED ONLY THE DAY OPS WILL BE
     2RUN)
      GO TO 666
C     WHEN AVAILABLE CHANGE THIS SECTION FOR EVENING FLIGHTS
C     MAKE  75 A CONTINUE
C     N(2,NUMFLI,NUMRUN)=DUMMY1
  755 N(2,NUMFLI,NUMRUN)=DUMMY1
      GO TO 666
   76 IF(WECPEG) GO TO 766
      IF(NEEFLG) GO TO 766
      WRITE(6,10002)
10002 FORMAT(32H NIGHT OPS CURRENTLY NOT ALLOWED)
      WRITE(6,10001)
      GO TO 666
C
C     CHANGES WHEN AVAILABLE
C
C     N(3,NUMFLI,NUMRUN)=DUMMY1
  766 N(3,NUMFLI,NUMRUN)=DUMMY1
      GO TO 666
   77 NOEXTS=IFIX(DUMMY1)
      NSG(NFLGHT)=NOEXTS+2
      GO TO 666
   78 CONTINUE
      CLIMBA(NFLGHT)=DUMMY1/180.*3.1415927
      THETAI(2,NFLGHT)=CLIMBA(NFLGHT)
      GO TO 666
   79 CONTINUE
      ZETI(2,NFLGHT)=DUMMY1
      IF(.NOT. MKSFLG) GO TO 666
      CALL MTOFT(ZETI(2,NFLGHT),1)
      GO TO 666
   80 CONTINUE
      ZETI(1,NFLGHT)=DUMMY1
      IF(.NOT. MKSFLG) GO TO 666
      CALL MTOFT(ZETI(1,NFLGHT),1)
      GO TO 666
C
   83 CONTINUE
      WRITE(6,777) THETAI(2,NFLGHT),ZETI(1,NFLGHT)
  777 FORMAT(25H CLIMBA OR PRSEG1 MISSING,2(2X,F15.5))
      CANCEL=.TRUE.
      GO TO 2000
  666 CONTINUE
      STORE(IST+1)=DUMMY1
```

```
      IST=IST+4
      GO TO 51
 5111 STORE(IST+1)=DUMMY1
      STORE(IST+2)=-1.
      IST=IST+3
      GO TO ERROR,(10,51,66,86,96,111,131)
C     EXPANSION
C LOOP TO SETUP FLIGHT PARAMETERS
   65 NUMFLI=NUMFLI+1
      NFLGHT=NFLGHT+1
      NEXTS=0
      ASSIGN 66 TO ERROR
      GROUP=4
   66 READ(5,1002) ALPHA,DUMMY1
      WRITE(6,1003) ALPHA,DUMMY1
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      DO 67 J=26,38
      JUMP2=J
      IF(ALPHA.EQ.PSEUDO(J)) JUMP2=J-25
      IF(JUMP2.NE.J) GO TO (68,69,70,71,72,73,74,75,76,77,78,79,80)
     2,JUMP2
   67 CONTINUE
      IST=IST+2
16000 FORMAT(55H NO. OF FLIGHTS PROVIDED DOESNT AGREE WITH NUMBERSTATED)
      IF(NUMFLI.NE.NOFLTS) WRITE(6,16000)
      IF(NUMFLI.NE.NOFLTS)CANCEL=.TRUE.
C THE FOLLOWING CARD HAS BEEN TEMPORARILY COMMENTED
C     IF(CLIMBA.NE.0..AND.CHECK.EQ.0.) GO TO 83
C     IF(ZETI(1,NFLGHT).NE.0..AND.CHECK.EQ.0.) GO TO 83
      GO TO 2000
   68 NUMFLI=NUMFLI+1
       NFLGHT=NFLGHT+1
      IST=IST+2
      GO TO 66
   69 TP(NUMFLI,NUMRUN)=1.
      ICAT(GROUP)=ICAT(GROUP)+1
      IST=IST+2
      GO TO 66
   70 TP(NUMFLI,NUMRUN)=0.
      ICAT(GROUP)=ICAT(GROUP)+1
C     THIS IS A TAKEOFF
      IST=IST+2
      GO TO 66
   71 W(NUMFLI,NUMRUN)=DUMMY1
C     IF NECESSARY CONVERT BETWEEN FEETLB AND MKS SYSTEMS
      GO TO 666
   72 PT(NUMFLI,NUMRUN)=DUMMY1
      PTHRI(1,NFLGHT)=PT(NUMFLI,NUMRUN)
      PTHRI(2,NFLGHT)=PT(NUMFLI,NUMRUN)
      ICAT(GROUP)=ICAT(GROUP)+1
C  PERCENTAGE THRUST OVER SEGMENTS 1 AND 2
      GO TO 666
   73 TY(NUMFLI,NUMRUN)=DUMMY1
C     TYPE OF PLANE
```

```
   50 NUMRUN=NUMRUN+1
      NUMFLT=0
      ASSIGN 51 TO ERROR
      GROUP=3
   51 READ(5,1002) ALPHA,DUMMY1,DUMMY2
      STORE(IST+1)=-1.
 1002 FORMAT(A6,2X,F15.5,2X,F15.5)
      WRITE(6,1003) ALPHA,DUMMY1,DUMMY2
      STORE(IST)=ALPHA
 1001 FORMAT(A6,F10.2,6X,F10.2)
      DO 52 J=17,25
      JUMP2=J
      IF(ALPHA.EQ.PSEUDO(J)) JUMP2=J-16
      IF(JUMP2.NE.J) GO TO (53,54,55,56,57,58,59,60,61),JUMP2
   52 CONTINUE
      STORE(IST+1)=-1.
      IST=IST+2
C     NO MATCH FOR A VARIABLE WITHIN THE RUNWAY SET CHECK TO SEE IF IT IS A NEW
C     IDENTIFIER OTHER THAN A RUNWAY IDENTIFIER
      GO TO 2000
   53 NUMRUN=NUMRUN+1
       IST=IST+2
      GO TO 511
C SETTING OF ENDPOINTS OF RUNWAYS
   54 XLM1(1,NUMRUN)=DUMMY1
      XLM2(1,NUMRUN)=DUMMY2
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 511
   55 XLM1(2,NUMRUN)=DUMMY1
      XLM2(2,NUMRUN)=DUMMY2
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 511
   56 XLM1(3,NUMRUN)=DUMMY1
      XLM2(3,NUMRUN)=DUMMY2
      GO TO 511
   57 STO(NUMRUN)=DUMMY1
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 5111
   58 SL(NUMRUN)=DUMMY1
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 5111
   59 CONTINUE
      NOFLTS=IFIX(DUMMY1)
      NFT(NUMRUN)=NOFLTS
      ICAT(GROUP)=ICAT(GROUP)+1
      GO TO 5111
   60 CONTINUE
C     EXPANSION
      GO TO 511
   61 CONTINUE
      GO TO 511
  511 CONTINUE
      STORE(IST+1)=DUMMY1
      STORE(IST+2)=DUMMY2
      STORE(IST+3)=-1.
```

```
 2030 IF(ALPHA.EQ.CONTOR(1)) GO TO 110
 2040 IF(ALPHA.EQ.RUNNFO(1)) GO TO 130
 2050 IF(ALPHA.EQ.RUNNFO(2)) GO TO 131
 2060 IF(ALPHA.EQ.RUNNFO(3)) GO TO 140
 2070 IF(ALPHA.EQ.RUNNFO(4)) GO TO 136
 2080 IF(ALPHA.EQ.RUNNFO(5)) GO TO 154
 2090 IF(ALPHA.EQ.RUNNFO(6)) GO TO 155
      IF(ALPHA.EQ.RUNWAY(1)) GO TO  50
C     IF WE GET HERE THERE IS AN ERROR IN THE INITIAL IDENTIFIER
 1003 FORMAT(1X,A6,2X,F15.5,2X,F15.5)
      IPHIL=500
       WRITE(6,5000) ALPHA,GROUP
      CANCEL=.TRUE.
 5000 FORMAT(24H FATAL ERROR VARIABLE = ,A6,14HCHECK CATEGORY,I3)
      GO TO ERROR,(10,51,66,86,96,111,131)
   21 MKSFLG=.FALSE.
      GO TO 10
   22 MKSFLG=.TRUE.
      GO TO 10
   23 PTINFG=.TRUE.
      GO TO 10
   24 PTTBFG=.TRUE.
      GO TO 10
   25 PTOUFG=.TRUE.
   27 PTCOFG=.TRUE.
      IF(ALPHA.NE.CONTRL(8)) PTALFG=.FALSE.
      GO TO 10
   26 NOISFG=.TRUE.
      GO TO 10
   28 PTALFG=.TRUE.
      PTTBFG=.TRUE.
      PTINFG=.TRUE.
C
C     BRANCH TO SET UP THE REST OF THE PRINT FLAGS
      GO TO 25
   34 DGL3FG=.TRUE.
   31 DGL2FG=.TRUE.
   30 DGL1FG=.TRUE.
      DGLCFG=.FALSE.
      GO TO 10
   32 CALFLG=.TRUE.
      SSIFLG=.TRUE.
      GO TO 10
   33 CALFLG=.TRUE.
      LSIFLG=.TRUE.
      GO TO 10
   29 CONTINUE
      NEFFLG=.TRUE.
      GO TO 10
   35 CONTINUE
      WECPFG=.TRUE.
      GO TO 10
   36 CONTINUE
      WECPFG=.FALSE.
      GO TO 10
```

```
 1011 FORMAT(12A4)
      IF(ALPHA.EQ.HEADER)
     1 WRITE(6,5050) (TITLE(I),I=1,12)
 5050 FORMAT(1X,12A4)
      IF(ALPHA.EQ.RUNNEO(4).OR.JUMP.EQ.0) GO TO 500
 1000 FORMAT(A6)
      IF(ALPHA.EQ.HEADER) GO TO 10
      IF(ALPHA.EQ.RUNWAY(1)) GO TO 50
      DO 20 J=1,16
      IF(ALPHA.EQ.PSEUDO(J    )) GO TO (21,22,23,24,25,26,27,28,29,30,31
     2,32,33,34,35,36),J
C     PUT  ERROR MESSAGES IN HERE FOR WHEN THERE ARE NO MATCHES
C     A COUNTER IS KEPT UNTIL THE PROCESS CARD IS ENCOUNTERED IF THE COUNTER
C     IS NOT  0 NE STOP BECAUSE OFFATAL ERRORS
   20 CONTINUE
      IF(ALPHA.NE.PLOTS(1)) GO TO 2000
  200 SETPLT=.TRUE.
      DO 203 I=1,8
      READ(5,1002) ALPHA,DUMMY1
      STORE(IST)=ALPHA
      STORE(IST+1)=DUMMY1
      STORE(IST+2)=-1.
      IST=IST+3
      DO 202 J=1,9
      IF(ALPHA.EQ.PLOTS(J)) GO TO (204,205,206,207,208,209,210,211,212),J
     2J
  202 CONTINUE
      WRITE(6,300)
  300 FORMAT(56H SOME VARIABLE USED WITH SETPLT IS MISSING OR MISSPELLED
     2)
      CANCEL=.TRUE.
      GO TO 2000
  204 SETPLT=.TRUE.
      GO TO 203
  205 DFLX=DUMMY1
      GO TO 203
  206 DFLY=DUMMY1
      GO TO 203
  207 AINCHX=DUMMY1
      GO TO 203
  208 AINCHY=DUMMY1
      GO TO 203
  209 AXMAX=DUMMY1
      GO TO 203
  210 AXMIN=DUMMY1
      GO TO 203
  211 AYMAX=DUMMY1
      GO TO 203
  212 AYMIN=DUMMY1
  203 CONTINUE
      GO TO 10
 2000 IF(ALPHA.EQ.FLIGHT(1)) GO TO 65
 2010 ICOFG=0
      IF(ALPHA .EQ. EXTNSN(1)) GO TO 85
 2020 IF(ALPHA.EQ.GRID(1)) GO TO 95
```

```
      REAL*8 STORE(6000)
      REAL*8 PLOTS(9),HEADER,CONTRL(16),RUNWAY(9),FLIGHT(13),
     1 EXTNSN(6),GRID(9),CONTOR(13),RUNNFO(6)
      DATA PLOTS/6HSETPLT,6HDELTAX,6HDELTAY,6HAINCHX,6HAINCHY,6HAXMAXV,6
     2HAXMINV,6HAYMAXV,6HAYMINV/
      DATA  HEADER/6HTITLEE/
      DATA CONTRL/6HFEETLB,6HMKSSYS,6HPRTINP,6HPRTTAB,6HPRTOUT,6HNOISIN
     2,6HPRTCTR,6HPRTALL,6HNEFCAL,6HDIAGL1,6HDIAGL2,6HCLCMPS,6HCLCMPL,6H   DATA1
     3DIAGL3,6HWECPNL,6HNEFVAL/
      DATA RUNWAY/6HRUNWAY,6HXCOORD,6HYCOORD,6HZCOORD,6HTKGDRL,6HTCHDWN,   DATA2
     26HNOFLTS,6H      ,6H      /
      DATA FLIGHT/6HFLIGHT,6HLANDNG,6HTAKOFF,6HACWGHT,6HTHRUST,6HACTYPE,   DATA3
     26HNDAYOP,6HNEVNOP,6HNNGTOP,6HNOEXTS,6HCLIMBA,6HPRSFG2,6HPRSFG1/      DATA
      DATA EXTNSN/6HEXTNSN,6HEXTENT,6HELEVAT,6HTHRUST,6HRADIUS,6H      /   DATA4
      DATA GRID/6HGRIDCL,6HDELTAX,6HDELTAY,6HFIRSTX,6HFIRSTY,6HNOOFXS,6H
     2NOOFYS,6H      ,6H      /                                            DATA6
      DATA CONTOR/6HCONTOR,6HNOISEV,6HDELTAX,6HFIRSTX,6HNOOFXS,6HTOLNCE,   DATA7
     26HYAPPRX,6HSYMMET,6HXCOORD,6HYCOORD,6HRXAXIS,6HRYAXIS,6H      /      DATA7
      DATA RUNNFO/6HPROCES,6HREPEAT,6HENDRUN,6HNEWSET,6H      ,6H      /   DATA8
      INTEGER ERROR,GROUP
      DIMENSION CLIMBA(150)
      DIMENSION ICHECK(7),ICAT(7)
      EQUIVALENCE (CONTRL(1),PSEUDO(1)),(RUNWAY(1),PSEUDO(17)),(FLIGHT(1  EQUIV1
     2),PSEUDO(26)),(EXTNSN(1),PSEUDO(39)),(GRID(1),PSEUDO(45)),(CONTOR(  EQUIV2
     31),PSEUDO(54)),(RUNNFO(1),PSEUDO(67))
C  THE EQUIVALENCE IS SET UP TO CUT DOWN ON RUN TIME SEARCHING FOR A MATCH  TO
C  EACH OF THE ALPHABETIC INPUT CONTROL VARIABLES THIS WILL ALLOW US TO COMPARE
C  ONLY PARTICULAR SECTIONS OF PSEUDO
C THE FOLLOWING CARD HAS BEEN TEMPORARILY COMMENTED
C     DATA ICHECK/0,0,5,5,4,6,6/
      REAL N
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1 RADI(10,150),NSG(150),NFLGHT
      COMMON/GDPAR/CX,CY,X1,Y1,NX,NY
      COMMON/QR/TOL,VAL,APPROX,BASEX,XDEL,MX
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGLOFG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,     LOGFG2
     2PTGDFG,PTCOFG,PTALFG
      COMMON/LOGFG3/RWAYFG,GRIDFG,CONTFG
      COMMON/JOE/PEFLIN,XCOORD(2),YCOORD(2),TITLE(12),SYMFLG
      ASSIGN 131 TO ERROR
      IF(JUMP.NE.0) READ(5,1000) ALPHA
      IF(JUMP.NE.0) GO TO 2040
C     OVER ALL CONTROL SECTION
      IPHIL=0
      IST=1
      ASSIGN 10 TO ERROR
      GROUP=2
   10 READ(5,1000) ALPHA
      WRITE(6,1003) ALPHA
      STORE(IST)=ALPHA
      STORE(IST+1)=-1.
      IST=IST+2
C     CHECK FOR TITLE
      IF(ALPHA.EQ.HEADER) READ(5,1011) (TITLE(I),I=1,12)
```

```
      WRITE(6,1)
    1 FORMAT(1H1)
      WRITE(6,100)
  100 FORMAT(36X,45(1H*))
      WRITE(6,101)
  101 FORMAT(36X,45H* SUMMARY OF RUNWAYS AND ASSOCIATED FLIGHTS *)
      WRITE(6,102)
  102 FORMAT(36X,45H*   (INCLUDING AIRCRAFT TYPE NUMBERS AND    *)
      WRITE(6,103)
  103 FORMAT(36X,45H*        LANDING/TAKEOFF INFORMATION)        *)
      WRITE(6,100)
      WRITE(6,106)
  106 FORMAT(//118H     FLIGHTS 1     2     3     4     5     6     7     8     9
     1    10    11    12    13    14    15    16    17    18    19    20       TOTAL)
      WRITE(6,107)
  107 FORMAT(8H RUNWAYS)
      DO 500 I=1,NRW
      K=NFT(I)
      DO 400 J=1,K
      XTP(J)=XDG
      IF(TP(J,I) .EQ. 0.) XTP(J)=TKF
  400 NTY(J)=TY(J,I)
      WRITE(6,120) RYNAM(I),(XTP(J),NTY(J),J=1,K)
  120 FORMAT(1H ,A6,1X,20(1X,A1,I3))
      WRITE(6,121) K
  121 FORMAT(1H+,112X,I5/)
      WRITE(6,122)
  122 FORMAT(1H )
  500 CONTINUE
      WRITE(6,125)
  125 FORMAT(113X,5H-----)
      WRITE(6,126) NFLGHT
  126 FORMAT(107X,7HFLIGHTS,I4)
      RETURN
      END
      SUBROUTINE READOU
      RETURN
      END
      SUBROUTINE READIN(INDEX)
      COMMON/JOE3/JUMP
      COMMON/JOE2/NEFFLG,NOISFG
      COMMON/JOE1/WECPFG,CANCEL
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),XLM1(3,10),XLM2(3       AA
     2,10),STO(10),SL(10)                                                     AA
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/FACTR/AXMAX,AXMIN,AYMAX,AYMIN,DFLX,DFLY,AINCHX,AINCHY,
     2SETPLT
      LOGICAL NEFFLG,NOISFG
      LOGICAL SETPLT
      LOGICAL CONVRT
      LOGICAL MKSFLG,PTINFG,PTTRFG,PTOUFG,PTGDFG,PTCOFG,PTALFG,DGL3FG,
     2DGL2FG,DGL1FG,DGL0FG,CALFLG,SSIFLG,LSIFLG,HESGFG,CONTFG,SYMFLG,
     3LINEFG,GRIDFG,RWAYFG
      LOGICAL CANCEL,WECPFG
      REAL*8 ALPHA,PSEUDO(72)
```

```
     1  SETPLT
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL SETPLT
      DIMENSION DATA(1024)
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      CALL PLOTS(DATA,1024)
      IF(SETPLT) GO TO 10
      SIZE=11.
      IF(LSIFLG)SIZE=30.
      CALL FACTOR(SIZE/11.)
      GO TO 20
   10 SIZE=10.
      IF(LSIFLG) SIZE=30.
      CALL FACTOR(SIZE/10.)
   20 CONTINUE
      CALL PLOT(0.,-11.,-3)
      CALL PLOT(0.,.5,-3)
      RETURN
      ENTRY FINPLT
      CALL PLOT(0.,0.,999)
      RETURN
      END
      SUBROUTINE NOMEN(XM,YM,XD,YD)
      RETURN
      END
      SUBROUTINE ATTENU(SB,HX,FTT,XXX)
C THIS FUNCTION GIVES THE GROUND ATTENUATION AS A FUNCTION OF THE        00048500
C DISTANCE TO THE OBSERVER AND THE ELEVATION ANGLE BETA.                 00048550
      XXX=0.
      M=14                                                                  48650
      B=ARSIN(SB)                                                           48700
      IF(B.GT.0.1208)GO TO 100
      T=SQRT(ABS(TAN(13.*B)))
      IF(FTT.EQ.0.)M=15                                                     48850
      CALL GENEN(M,HX,1,YYY)
      TEMP1=EXP(T)
      XXX=YYY/TEMP1
  100 CONTINUE
      RETURN
      END                                                                   48970
      SUBROUTINE SUMPE
C THIS SUBROUTINE WRITES THE SUMMARY TABLE OF RUNWAYS,FLIGHTS,AND
C ASSOCIATED INFORMATION
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1 RADI(10,150),NSG(150),NFLGHT
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),XLM1(3,10),
     1   XLM2(3,10),STO(10),SL(10)
      COMMON/RYNME/RYNAM
      REAL*8 RYNAM(10)
      REAL N
      DIMENSION XTP(20),NTY(20)
      DATA XDG/1HL/,TKF/1HT/
      READ(5,1000) (RYNAM(I),I=1,10)
 1000 FORMAT(10A8)
```

```
      GO TO 1111
1000  CALL SYMBOL(SX1-.2,SY1-.2,.063,RYNAM(I),ANG,3)
1111  CONTINUE
      CALL PLOT(-AINCH-4.,-2.,-3)
      RETURN
      END
      SUBROUTINE AXIS2(X0,Y0,AMAX,AMIN,DELX,AINCH,BCD,NCH,NDEC,DELN)
      DIMENSION BCD(1)
      HT=.12
      CINCH=ABS(AINCH)
      IF(AINCH.LT.0.)GO TO 5
      W2=1.
      W1=0.
      GO TO 10
    5 W1=1.
      W2=0.
   10 CALL PLOT(X0,Y0,3)
      NUM=(AMAX-AMIN)/DELX+1.
      ANC=CINCH/FLOAT(NUM-1)
      DELN=DELX/ANC
      ANUM=AMIN-DELX
      X=0.
      Y=0.
      XM=0.
      DO 40 I=1,NUM
      ANUM=ANUM+DELX
      II=0
   25 IF(ABS(ANUM)/10.**II.LT.1.)GO TO 20
      II=II+1
      GO TO 25
   20 IF(ANUM.LT.0.)II=II+1
      IMORE=NDEC+1
      II=II+IMORE
      CENTER=FLOAT(II)*HT
      IF(W2.EQ.0.)CENTER=CENTER/2.
      OFF=HT/2.
      XC=X-CENTER-.15*W2
      IF(XC.LT.XM)XM=XC
      YC=Y-(HT+.15)*W1-OFF*W2
      CALL PLOT(X0+X,Y0+Y,2)
      CALL PLOT(X0+X+.1*W2,Y0+Y+.1*W1,3)
      CALL PLOT(X0+X-.1*W2,Y0+Y-.1*W1,2)
      CALL NUMBER(X0+XC,Y0+YC,HT,ANUM,0.,NDEC)
      CALL PLOT(X0+X,Y0+Y,3)
      X=X+ANC*W1
      Y=Y+ANC*W2
   40 CONTINUE
      BST=(CINCH-FLOAT(NCH)*HT)/2.
      XXC=(X0+BST)*W1+(X0+XM-OFF)*W2
      YYC=(Y0+YC-HT-OFF)*W1+(Y0+BST)*W2
      CALL SYMBOL(XXC,YYC,.12,BCD,90.*W2,NCH)
      RETURN
      END
      SUBROUTINE INIPLT
      COMMON/FACTR/AXMAX,AXMIN,AYMAX,AYMIN,DELX,DELY,AINCHX,AINCHY,
```

```
      IF(J.EQ.11)SAY11=Y
      IF(J.EQ.15)Y=SAY7
      IF(J.EQ.15)X=X+1.
      IF(J.EQ.19)Y=SAY11
      KKK=(J-6)/4+1
      IF(J.EQ.7.OR.J.EQ.11.OR.J.EQ.15.OR.J.EQ.19)
     X  CALL NUMBER(X+.4,Y+2.*DELY,.15,FLOAT(KKK),0.,-1)
      CALL NUMBER(X,Y,.15,FLIGHT(J,I),0.,NK)
      GO TO 80
C WECPNL OPERATIONS
   67 CALL SYMBOL(X,Y+.13,.1,3HDAY,0.,3)
      CALL NUMBER(X+.4,Y+.13,.1,N(1,KK,JJ),0.,3)
      CALL SYMBOL(X,Y,.1,3HEVN,0.,3)
      CALL NUMBER(X+.4,Y,.1,N(2,KK,JJ),0.,3)
      CALL SYMBOL(X,Y-.13,.1,3HNGT,0.,3)
      CALL NUMBER(X+.4,Y-.13,.1,N(3,KK,JJ),0.,3)
      GO TO 80
C NEF OPERATIONS
  670 CALL SYMBOL(X,Y+.07,.12,3HDAY,0.,3)
      CALL NUMBER(X+.5,Y+.07,.12,N(1,KK,JJ),0.,3)
      CALL SYMBOL(X,Y-.13,.12,3HNGT,0.,3)
      CALL NUMBER(X+.5,Y-.13,.12,N(3,KK,JJ),0.,3)
      GO TO 80
   70 IND=FLIGHT(J,I)
      CALL SYMBOL(X,Y,.15,PLANE(1,IND),0.,18)
   80 CONTINUE
      X=XTEMP
  800 CONTINUE
      CALL PLOT(XCTR*2.+2.,0.,-3)
      RETURN
      END
      SUBROUTINE RWYLEG(AINCH)
C  THIS SUBROUTINE PLOTS THE RUNWAY LEGEND
      COMMON/RYNME/RYNAM
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1  XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/NN/N(3,20,10),NRW,NET(10)
      REAL*8 RYNAM(10)
      CALL PLOT(AINCH+4.,2.,-3)
      CALL PLOT(2.,2.,3)
      CALL PLOT(-2.,2.,2)
      CALL PLOT(-2.,-2.,2)
      CALL PLOT(2.,-2.,2)
      CALL PLOT(2.,2.,2)
      CALL SYMBOL(-1.63,1.70,.25,13HRUNWAY LEGEND,0.,13)
      DO 1111 I=1,NRW
      SX1=XLM1(1,I)/5000.
      SY1=XLM1(2,I)/5000.
      SX2=XLM2(1,I)/5000.
      SY2=XLM2(2,I)/5000.
      CALL RUNWAY(SX1,SY1,SX2,SY2)
      SLOPE=(SY2-SY1)/(SX2-SX1)
      ANG=ATAN(SLOPE)*180./3.14159
      IF(SLOPE .GT. 0. .AND. SY2 .GT. SY1) GO TO 1000
      CALL SYMBOL(SX1+.125,SY1+.125,.063,RYNAM(I),ANG,3)
```

```
      SAVF=YTOP-2.*DFLY
      YTOP=YTOP-2.*DFLY
      CALL SYMBOL(X,YTOP-4.*DFLY,.15,15HTAKEOFF/LANDING,0.,15)
      CALL SYMBOL(X,YTOP-6.*DELY,.15,15HAIRCRAFT WEIGHT      ,0.,15)
      CALL SYMBOL(X+2.25,YTOP-6.*DFLY,.15,LBMF(INDEX),0.,4)
      CALL SYMBOL(X,YTOP-8.*DELY,.15,14HPERCENT THRUST,0.,14)
      CALL SYMBOL(X,YTOP-10.*DFLY,.15,13HAIRCRAFT TYPF,0.,13)
      CALL SYMBOL(X,YTOP-12.*DFLY,.15,17HNO. OF OPFRATIONS,0.,17)
      CALL SYMBOL(X,YTOP-14.*DFLY,.15,17HNO. OF FXTFNSIONS,0.,17)
      IF(AMAX.LT.1.) GO TO 50
      MAX=AMAX
      DO 15 K=1,MAX
      IF(K.FQ.2.OR.K.FQ.4)YTOP=YTOP-2.
      CALL SYMBOL(X+.3,YTOP-16.*DELY,.15,9HFXTENSION,0.,9)
      IF(K.FQ.3)YTOP=YTOP+2.
      IF(K.GT.2)GO TO 15
      CALL SYMBOL(X,YTOP-18.*DFLY,.15,16HPROJFCTED LENGTH       ,0.,16)
      CALL SYMBOL(X+2.4,YTOP-18.*DFLY,.15,FTMD(INDFX),0.,6)
      CALL SYMBOL(X,YTOP-20.*DELY,.15,20HFLEVATION ANGLE(DFG),0.,20)
      CALL SYMBOL(X,YTOP-22.*DFLY,.15,14HPFRCFNT THRUST,0.,14)
      CALL SYMBOL(X,YTOP-24.*DFLY,.15, 6HRADIUS      ,0., 6)
      CALL SYMBOL(X+1.,YTOP-24.*DELY,.15,FTKG(INDFX),0.,4)
   15 CONTINUF
   50 X=X+3.5-FLFN
      JJ=1
      KK=0
      DO 800 I=1,NUM
      Y=SAVF-4.*DFLY
      X=X+FLFN
      IF(KK .LT. NFT(JJ)) GO TO 55
      JJ=JJ+1
      KK=0
   55 CONTINUF
      KK=KK+1
      IF(KK.FQ.1)CALL SYMBOL( X,Y+4.*DFLY,.15,6HRUNWAY,0.,6)
      IF(KK.EQ.1)CALL NUMBFR(X+1.1,Y+4.*DFLY,.15,FLOAT(JJ),0.,-1)
      CALL SYMBOL(X,Y+2.*DELY,.15,6HFLIGHT,0.,6)
      CALL NUMBER(X+1.1,Y+2.*DELY,.15,FLOAT(KK),0.,-1)
      IF(FLIGHT(1,I).FQ.1.) GO TO 60
      CALL SYMBOL(X,Y,.15,7HTAKEOFF,0.,7)
      GO TO 65
   60 CALL SYMBOL(X,Y,.15,7HLANDING,0.,7)
   65 K=6+INT(FLIGHT(6,I))*4
      XTFMP=X
      DO 80 J=2,K
      NK=-1
      IF(J .FQ. 16 .OR. J .FQ. 20) NK=1
      IF(J.FQ.8.OR.J.FQ.12)NK=1
      Y=Y-2.*DFLY
      IF(J.EQ.5.AND.WFCPFG)GO TO 67
      IF(J .FQ. 5 .AND. NFFFLG) GO TO 670
      IF(J.EQ.4)GO TO 70
      IF(J.FQ.7.OR.J.FQ.11) Y=Y-2.*DFLY
      IF(J.FQ.5)NK=3
      IF(J.FQ.7)SAY7=Y
```

```
      RETURN
      END
      SUBROUTINE VSUB(C,A,B)
      COMMON/NCOORD/NC
      DIMENSION A(1),B(1),C(1)
C THIS SUBROUTINE PERFORMS VECTOR SUBTRACTION  C=A-B
      DO 10 I=1,NC
      C(I)=A(I)-B(I)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE VSCL(C,F,A)
      COMMON/NCOORD/NC
      DIMENSION C(1),A(1)
C THIS SUBROUTINE PERFORMS SCALAR MULTIPLICATION ,C=FA,WHERE
C F IS A SCALAR AND A,C ARE VECTORS
      DO 10 I=1,NC
   10 C(I)=F*A(I)
      RETURN
      END
      SUBROUTINE VMAG(A,XXX)
      COMMON/NCOORD/NC
      DIMENSION A(1)
C THIS FUNCTION FINDS THE MAGNITUDE OF A VECTOR A
      XXX=SQRT(VDOT(A,A))
      RETURN
      END
      SUBROUTINE BNOISE(TAC,HX)
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1 PTGDFG,PTCOFG,PTALFG
      COMMON/BB/R0(3),FLIGHT,PFN,EPNDB
      COMMON/BK/DIAG,YES,MAYBE
      DIMENSION TOC(20),GTO(20),TOT(20),CL(20),
     1GL(20),TL(20),THC(20)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1 PTCOFG,PTALFG
      REAL MAYBE
      DATA TOC/1.,3.,1.,1.,3.,3.,3.,5.,5.,5.,2*16.,7.,7*0./,
     1GTO/-4.,-3.,3*0.,2*-2.,3.,-10.,2*0.,3.0,8*0./,
     2TOT/4*100.,3*93.,3*100.,2*93.,100.,7*0./,
     2CL/2.,4.,2*2.,3*4.,3*6.,2*17.,9.,7*0./,
     3GL/-8.,-4.,3*0.,2*-2.,3.,3*0.,3.,8*0./,
     3TL/42.,33.,5*42.,3*100.,3*42.,7*0./,
     4THC/11.,13.,2*10.,3*12.,13*0./
      Z=0.
C TAC IS THE A/C TYPE NUMBER
      I=TAC
C FLIGHT=1 IMPLIES A LANDING
      IF(FLIGHT.EQ.1.)GO TO 50
      NFN=TOC(I)
      COR=GTO(I)
      THR=TOT(I)
      GO TO 100
   50 NFN=CL(I)
      COR=GL(I)
```

```
      THR=TL(I)
  100 CONTINUE
      M=THC(I)
C COR+GENFN(NFN,HX,1) WILL GIVE THE EPNL AT A DISTANCE
C HX,NEGLECTING GROUND ATTENUATION,SHIELDING,+CHANGE FROM
C THE ASSUMED THRUST(THR).M IS THE THRUST CORRECTION CURVE.
      CALL GENFN(NFN,HX,1,XQX)
      EPNDB=XQX+COR
C NO THRUST CORRECTION IS MADE FOR A/C TYPE GT 7, SINCE NO
C DATA ARE AVAILABLE
      IF(I.GE.8)GO TO 900
      CALL GENFN(M,PFN,2,X)
      CALL GENFN(M,THR,3,Y)
      Z=X-Y
      EPNDB=EPNDB+Z
  900 IF(DGL1FG) CALL FNT4(NFN,COR,Z,XQX)
 1000 RETURN
      END
      SUBROUTINE DATA(J)
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/CC/LAMBD1(3),LAMBD2(3),TAU1,TAU2,WEIGHT,ACTYPE,
     1DIST,SLOPE,UG(3),SINB,XDIST,GLUMP
      REAL LAMBD1,LAMBD2
C THIS SUBROUTINE DETERMINES CERTAIN QUANTITIES PERTAINING TO
C THE RUNWAY GEOMETRY FOR RUNWAY NO. J.LAMBD1 + LAMBD2 ARE
C TWO POINTS ON THE RUNWAY.FLIGHT TAKES PLACE IN THE DIRECTION
C LAMBD1 GT LAMBD2,WHICH DEFINES THE UNIT VECTOR UG.GLUMP IS
C THE DISTANCE FROM LAMBD1(ALONG UG) AT WHICH THE A/C BEGINS
C THE TAKE-OFF ROLL.TAU1 IS THE DISTANCE TO TOUCHDOWN.
      DO 10 M=1,3
      LAMBD1(M)=XLM1(M,J)
      LAMBD2(M)=XLM2(M,J)
   10 CONTINUE
      GLUMP=STO(J)
      TAU1=SL(J)
   20 CALL VSUB(UG,LAMBD2,LAMBD1)
   30 CALL VUNT(UG,UG)
      RETURN
      END
      SUBROUTINE SLPDS(NZ,ACWT,DIST,IF,XXX)
C INPUT
C     NZ,THE AIRCRAFT TYPE (TAKEOFF IF  ,LANDING IF -).
C     ACWT,THE AIRCRAFT WEIGHT IN POUNDS.
C OUTPUT
C     SLPDS,THE TANGENT OF THE CLIMBOUT(OR DESCENT )ANGLE.
C     DIST THE ROLL DISTANCE FOR TAKFOFF.
C     IF AND ERROR INDICATOR.
      COMMON/WTPEV/WECTR,PEGWT(2,2,11),NACT
C INITIALIZE SLPDS FOR THE LANDING CASE.
      COMMON/TOOPRF/DECTR,TOPRF(2,8),NTOP
      COMMON/TPNS/PI
      COMMON/LN/VLN
      COMMON/ZXX/IPREL,IPR(150)
      LOGICAL IBND,BND
```

```
      TOL=10./WFCTR
      IF=0
C DETERMINE IF THE AIRCRAFT TYPE IS ACCOMODATED BY THE PROGRAM.
      DIST=0.
      XXX=TAN(DGTRD(3.))
      IF(IBND(IABS(NZ),1,NACT))GO TO 10
      IF=IF+1
C PRINT THE ERROR MESSAGE AND CHANGE THE ERROR INDICATOR.
      WRITE(6,100)
      GO TO 40
   10 CONTINUE
      IF(NZ.LT.0)GO TO 40
C IF THIS IS A LANDING RETURN.
      IPRFL=NZ-4
C SET UP THE INDEX NZ FOR THE PROPER WT-PROFILE CURVE.
      IF(NZ.GE.11.AND.NZ.LE.12)GO TO 30
      IF(NZ.GE.13)NZ=NZ-2
      GEW=ACWT/WFCTR
      L1=INT(PFGWT(1,1,NZ)+2.0001)
      L2=INT(PFGWT(1,2,NZ)+1.9999)
      IF(BND(GEW,PFGWT(2,1,NZ)-TOL,PFGWT(2,2,NZ)+TOL))GO TO 20
      IF=IF+2
      L1=1
      L2=NTOP
   20 CONTINUE
C FIND THE INDEX FOR THE TAKEOFF PROFILE CURVE.
      IPRFL=INT(XINT(PFGWT(1,1,NZ),GEW)+2.0)
      IPRFL=MAX0(L1,MIN0(L2,IPRFL))
   30 CONTINUE
C FIND THE TAKEOFF SLOPE AND ROLL DISTANCE.
      XXX=TOPRF(1,IPRFL)
C ZERO SLOPE INDICATES AN ERROR.
      IF(XXX .EQ. 0.) GO TO 50
      DIST=-TOPRF(2,IPRFL)/XXX
   40 CONTINUE
      RETURN
   50 CONTINUE
      IF=IF+4
      WRITE(6,400)
      GO TO 40
  100 FORMAT(33H ERROR- THERE IS NO CURVE FOR A/C )
  200 FORMAT(27H  ERROR-WT IS OUT OF RANGE )
  400 FORMAT(20H ERROR-SLOPE IS ZERO )
      END
      LOGICAL FUNCTION IBND(IV,IS,IH)
      IBND=.FALSE.
      IF(IV.GE.IS.AND.IV.LE.IH)IBND=.TRUE.
      RETURN
      END
      LOGICAL FUNCTION BND(V,S,H)
      BND=.FALSE.
      IF(V.GE.S.AND.V.LE.H)BND=.TRUE.
      RETURN
      END
      FUNCTION DGTRD(DG)
```

```
      COMMON/TRNS/PI
      DGTRD=(DG*PI)/180.
      RETURN
      END
      FUNCTION XINT(PT,Y)
      COMMON/LN/VLN
      DIMENSION PT(4)
      YI=PT(4)-PT(2)
      IF(YI.EQ.0.)GO TO 10
      XINT=(PT(3)-PT(1))*(Y-PT(2))/YI+PT(1)
      RETURN
   10 CONTINUE
      IF(Y.NE.PT(2))GO TO 20
      XINT=PT(1)
      RETURN
   20 CONTINUE
      WRITE(6,100) Y,PT
      XINT=VLN
  100 FORMAT(12H INT. ERROR ,F10.3,6X,2F10.3,3X,2F10.3)
      RETURN
      END
      FUNCTION NACM (FL,ACT)
      NACM=INT(SIGN(ACT,0.5-FL))
      RETURN
      END
C THE SUBROUTINES ENT2 THROUGH ENT5 THAT FOLLOW BELOW
C ARE CALLED TO WRITE OPTIONAL OUTPUT TO FILE NO. 1 (OUT).
C ENT2,ENT3,AND ENT5 ARE CALLED FROM FUNCTION EXPOSE,
C AND ENT4 IS CALLED FROM SUBROUTINE BNOISE.THE ENT
C SUBROUTINES ARE CALLED ON THE CONDITION THAT
C DIAG=YES OR DIAG=MAYBE WHERE YES IS  Y  AND
C MAYBE IS  M .DIAG OBTAINS IT VALUE IN SUBROUTINE READIN
C WHERE THE QUESTION  VERIFICATION AT A POINT   IS POISED
C TO THE USER.THE PURPOSE OF THESE SUBROUTINES IS
C SHOW THE INTERMEDIATE VALUES OBTAINED IN THE CALCULATION
C OF NE AT A CONTOUR POINT.THESE RESULTS ARE ESPECIALLY
C USEFULL IN CHECKING HAND CALCULATIONS.
      SUBROUTINE ENT2(CAT,TW,PNF,IXX,FLT)
      COMMON/JOE2/NEFFLG,NOISFG
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL NEFFLG,NOISFG
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      DIMENSION AX(3),BX(5),CX(5),DX(9),IA(12),IB(12),IC(12)
      DATA SSX/4HSST /
      DATA AX/4H2ENG,4H3ENG,4H4ENG/,BX/4HHBPR,4HLBPR,
     14HPROP,4HEXEC,4H    /,CX/4HTFAN,4HTJET,4HJET ,
     24HFANS,4H    /,DX/2HAA,2H A,2H B,2H C,2H D,2H E,
     32HPA,2HPB,2HLD/,IA/3,2,3,3,2,1,1,3,1,1,1,3/,
     4IB/1,1,2,5,2,2,2,4,2,4,3,3/,IC/1,1,1,2,1,1,4,3,4,3,5,5/
 3    FORMAT(/5X,13H A/C TYPE =   ,3A5)
    4 FORMAT(7X,10H WEIGHT = ,1X,I6)
    5 FORMAT(5X,11H THRUST =   ,I3)
    6 FORMAT(6X,12H PROFILE =   ,A2)
      IQ=CAT
C THIS SECTION ADDED TO ALLOW FOR NEW AIRCRAFT TYPE NUMBERS READ IN WITH
```

```
C USER-SPECIFIED EPNDB VALUES AT EXECUTION TIME. IT IS ASSUMED THAT THE
C TRUE A/C TYPE NUMBER IS DIGIT Y IN NUMBERS OF FORM XYX.
      IF(.NOT. NOISEG) GO TO 100
      IQ=IQ/10
      IQ=MOD(IQ,10)
  100 CONTINUE
      IF(IQ .NE. 13) GO TO 8
      WRITE(6,3) SSX
      GO TO 9
    8 ITEMP1=IA(IQ)
      ITEMP2=IB(IQ)
      ITEMP3=IC(IQ)
      WRITE(6,3) AX(ITEMP1),BX(ITEMP2),CX(ITEMP3)
    9 IQ=TW
      IF(MKSFLG)IQ=TW*.453592+.5
      WRITE(6,4)IQ
      IQ=PNF
      WRITE(6,5)IQ
      IF(FLT.EQ.1.)GO TO 10
      IX=IXX
      GO TO 20
   10 IX=9
   20 WRITE(6,6)DX(IX)
      RETURN
      END
      SUBROUTINE ENT3(XQ,NQ,DQ)
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      DIMENSION NQ(3)
      REAL NQ
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
    7 FORMAT(1X,17HDAY OPERATIONS=   ,F7.3/1X,17HEVN OPERATIONS=   ,F7.3/1
     1X,17HNGT OPERATIONS=   ,F7.3)
    8 FORMAT(12X,6H H =  ,I6)
    9 FORMAT(9X,9H BETA =  ,F6.2)
      WRITE(6,7) (NQ(IJ),IJ=1,3)
      IQ=DQ
      IF(MKSFLG)IQ=DQ*.3048+.5
      WRITE(6,8)IQ
      WRITE(6,9)XQ
      RETURN
      END
      SUBROUTINE ENT4(IFN,XCOR,XTH,XEPN)
   10 FORMAT(//5X,8H CURVE  ,I2,8HGIVES     ,F6.2,6HEPNDB )
   16 FORMAT(5X,19H THRUST CORR IS     ,F7.2)
   20 FORMAT(5X,19H CORRECTION  IS     ,F7.2)
      WRITE(6,10)IFN,XEPN
      WRITE(6,20)XCOR
      WRITE(6,16)XTH
      RETURN
      END
      SUBROUTINE ENT5N(X1,Y1,Z1,XNF1,AXF1)
   13 FORMAT(5X,16H ATTENUATION IS ,3X,F7.2)
   14 FORMAT(7X,14H SHIELDING IS ,3X,F7.2)
   15 FORMAT(5X,19H OPERATIONS  IS     ,F7.2)
   17 FORMAT(/22X,7(1H*)/7X,16H NET EPNDB IS      ,F6.2//5X,
```

```
     1 15H EPNDB SUBTOTAL,4X,F6.2)
   18 FORMAT(44H AXE1 WAS 0 IN ENT5...SUBTOTALS ARE SPURIOUS )
      X1=-X1
      Y1=-Y1
      WRITE(6,13)X1
      WRITE(6,14)Y1
      WRITE(6,15)Z1
      IF(AXE1.NE.0.)GO TO 30
      AXE1=10.
      WRITE(6,18)
   30 XE1=10.*ALOG10(AXE1)
      WRITE(6,17)XNE1,XE1
      RETURN
      END
      SUBROUTINE ENT5W(X1,Y1,Z1,XNE1,AXE1)
      DIMENSION XE1(3)
      DIMENSION Z1(3),AXE1(3),XNE1(3)
 13   FORMAT(5X,16H ATTENUATION IS ,3X,F7.2)
 14   FORMAT(7X,14H SHIELDING IS ,3X,F7.2)
   18 FORMAT(44H AXE1 WAS 0 IN ENT5...SUBTOTALS ARE SPURIOUS )
      X1=-X1
      Y1=-Y1
      WRITE(6,13)X1
      WRITE(6,14)Y1
      WRITE(6,15) Z1(1)
      WRITE(6,150) Z1(2)
      WRITE(6,1500) Z1(3)
   15 FORMAT(24H  DAY OPERATIONS   IS     ,F7.2)
  150 FORMAT(24H  EVN OPERATIONS   IS     ,F7.2)
 1500 FORMAT(24H  NGT OPERATIONS   IS     ,F7.2)
      IF(AXE1(1) .NE. 0. .OR. AXE1(2) .NE. 0. .OR. AXE1(3) .NE. 0.)
     1   GO TO 30
      WRITE(6,18)
   30 WRITE(6,35) XNE1
   35 FORMAT(/15H NET EPNDB IS  ,3F15.2)
      DO 1600 I=1,3
      IF(AXE1(I) .EQ. 0.) GO TO 1600
      XE1(I)=10.*ALOG10(AXE1(I))
 1600 CONTINUE
      WRITE(6,36) XE1
   36 FORMAT(15H EPNDB SUBTOTAL,3F15.2)
      RETURN
      END
      SUBROUTINE EXPWE(DUMY,XXX)
      COMMON/JOE2/NEFFLG,NOISEG
      COMMON/EPRIN/JEICS
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGLOFG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1 PTGDFG,PTCOFG,PTALFG
      COMMON/DENOP/CONARR(3),S,NOPARR(3)
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IET(400),MEL,MXX,MXSG
```

```
      COMMON/BK/DIAG,YES,MAYBE
      COMMON/TRNS/PI
      COMMON/BB/RO(3),FLIGHT,PEN,EPNDB
      COMMON/CC/LAMBD1(3),LAMBD2(3),TAU1,TAU2,WEIGHT,ACTYPE,
     1DIST,SLOPE,UG(3),SINB,XDIST,GLUMP
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/ZXX/IPRFL,IPR(150)
      DIMENSION XRO(3)
      DIMENSION AXFARR(3),YIARR(3),ZARR(3),ECPNLA(3),XNFARR(3)
      LOGICAL NEFFLG,NOISFG
      REAL NOPARR
      REAL MAYBE
      REAL LAMBD1,LAMBD2,N
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      DATA XLN10/2.302585/
      TT1=0.
      TT2=0.
      TT3=0.
      DO 5 IJ=1,3
      ZARR(IJ)=0.
      XNFARR(IJ)=0.
      ECPNLA(IJ)=0.
    5 AXFARR(IJ)=0.
      NFL=0
      MXX=0
      XNEM=0.
      DO 100 J=1,NRW
      IF(DGL1FG) WRITE(6,1) J
    1 FORMAT(/////12X,12H RUNWAY NO.=,I3/12X,1H ,12(1H*)//)
    2 FORMAT(12H FLIGHT NO.=,2X,I2/1H ,12(1H*))
      M=NFT(J)
      DO 100 I=1,M
      FLIGHT=TP(I,J)
      WEIGHT=W(I,J)
      ACTYPE=TY(I,J)
      NFL=NFL+1
      IF(DGL1FG) WRITE(6,2) I
      MX=IFXT(I,J)
      IF(MX.EQ.0)GO TO 20
      ISFG=JEXT(MX)+2
      GO TO 25
   20 ISEG=2
   25 IF(.NOT. DGL2FG) GO TO 26
      DO 27 IJ=1,3
   27 XRO(IJ)=RO(IJ)
      IF(MKSFLG) CALL FTTOM(XRO,3)
      WRITE(6,120) (XRO(IL),IL=1,3),ISEG,NFL
   26 CONTINUE
C AT THIS POINT,MX IS ZERO AND ISEG IS 2 IF THIS
C FLIGHT HAS NO EXTENSIONS.IF IT DOES HAVE SOME EXTENSIONS
C THEN MX IS THE APPROPRIATE POSITION IN THE ARRAYS IN
C COMMON/EXTS/TO LOOK FOR INFORMATION REGARDING THE
C OF SEGMENTS ON THIS FLIGHT.IF ISEG IS 4 THEN THE
```

```
C INFORMATION FOR THE SECOND EXTENSION MAY BE FOUND
C IN POSITION MX+1 OF THE ARRAYS OF COMMON/EXTS/.
      CALL HBETA(XDIST,SINB,PFN,MXS,RO,ISEG,NFL)
      IF(JFICS .EQ. 99) GO TO 100
      IF(.NOT. DGL1FG) GO TO 28
C HBETA DETERMINED H(=XDIST),SIN(BETA)(=SINB),THE PERCENT
C THRUST(=PFN),AND THE SEGMENT USED IN CALCULATING THESE
C (=MXS).
      IPRF=IPR(NFL)
      CALL FNT2(ACTYPE,WEIGHT,PFN,IPRF,FLIGHT)
   28 X=ARSIN(SINB)*180./PI
C N IS THE NUMBER OF IDENICAL OPERATIONS
      NOPARR(1)=N(1,I,J)
      NOPARR(2)=N(2,I,J)
      NOPARR(3)=N(3,I,J)
      Y1=NOPARR(1)+NOPARR(2)+NOPARR(3)
      IF(DGL1FG) CALL FNT3(X,NOPARR,XDIST)
C BNOISE FINDS THE OBSERVED EPNL NEGLECTING SHIELDING,
C GROUND ATTENUATION ,AND CHANGES FOR IDENICAL OPERATIONS.
   30 IF(NOISEG) GO TO 32
      CALL BNOISE(ACTYPE,XDIST)
      GO TO 33
   32 CALL BNOIIN(ACTYPE,XDIST,NFL,MXS)
   33 CONTINUE
C THE CORRECTION FOR EXTRA GROUND ATTENUATION IS-
      CALL ATTENU(SINB,XDIST,FLIGHT,X)
C THE SHIELDING TERM IS -
      Y=0.
C CALLING THE CORRECTED EPNL EXPOSE ,WE GET -
      XXX=EPNDB-(X+Y)
 9000 CONTINUE
      IF(XXX .LT. XNEM) GO TO 90
      MFL=NFL
      MXSG=MXS
      MXX=MX
      XNEM=XXX
   90 CONTINUE
      DO 95 IJ=1,3
      Y1ARR(IJ)=NOPARR(IJ)
      IF(Y1ARR(IJ) .EQ. 0.) GO TO 95
      ZARR(IJ)=10.*ALOG10(Y1ARR(IJ))
      XNEARR(IJ)=XXX+ZARR(IJ)
      AXEARR(IJ)=AXEARR(IJ)+EXP(XLN10*(XNEARR(IJ)/10.))
   95 CONTINUE
      IF(DGL1FG) CALL FNT5W(X,Y,ZARR,XNEARR,AXEARR)
      DO 96 IJ=1,3
   96 XNEARR(IJ)=0.
  100 CONTINUE
      DO 97 IJ=1,3
      IF(AXEARR(IJ) .EQ. 0.) GO TO 97
      ECPNLA(IJ)=10.*ALOG10(AXEARR(IJ))-CONARR(IJ)
   97 CONTINUE
      IF(AXEARR(1) .EQ. 0.) GO TO 300
      TT1=.5*EXP(XLN10*(ECPNLA(1)/10.))
  300 IF(AXEARR(2) .EQ. 0.) GO TO 301
```

```
      TT2=.125*EXP(XLN10*((ECPNLA(2) + 5.)/10.))
  301 IF(AXEARR(3) .EQ. 0.) GO TO 302
      TT3=.375*EXP(XLN10*((ECPNLA(3) + 10.)/10.))
  302 CONTINUE
      WECPNL=10.*ALOG10(TT1+TT2+TT3)+S
      XXX=WECPNL
      IF(DGL1FG) WRITE(6,110) ECPNLA,XXX
  110 FORMAT(/22H SUBTRACTIVE CONSTANTS,8H    36.35,10X,5H30.33,10X,5H35.
     111/16H ECPNL FOR D,E,N,F14.2,2F15.2/15H ***WECPNL IS  ,F15.2,4H **
     2*)
  120 FORMAT(21H FROM EXPOSE TO HBETA /12H RO,ISEG,NFL /1H ,3F12.3,2I4)
 9100 CONTINUE
      RETURN
      END
      SUBROUTINE EXPNF(DUMY,XXX)
      COMMON/JOE2/NEFFLG,NOISFG
      COMMON/ERRIN/JEICS
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IFT(400),MFL,MXX,MXSG
      COMMON/BK/DIAG,YES,MAYBE
      COMMON/TRNS/PI
      COMMON/BB/RO(3),FLIGHT,PFN,EPNDB
      COMMON/CC/LAMBD1(3),LAMBD2(3),TAU1,TAU2,WEIGHT,ACTYPE,
     1DIST,SLOPE,UG(3),SINB,XDIST,GLUMP
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/ZXX/IPRFL,IPR(150)
      LOGICAL NEFFLG,NOISFG
      REAL N
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      REAL MAYBE
      REAL LAMBD1,LAMBD2
      DATA XLN10/2.302585/
      AXE=0.
      NFL=0
      MXX=0
      XNEM=0.
      DO 100 J=1,NRW
      IF(DGL1FG) WRITE(6,1) J
    1 FORMAT(/////12X,12H RUNWAY NO.=,I3/12X,1H ,12(1H*)//)
    2 FORMAT(12H FLIGHT NO.=,2X,I2/1H ,12(1H*))
      M=NFT(J)
      DO 100 I=1,M
      FLIGHT=TP(I,J)
      WEIGHT=W(I,J)
      ACTYPE=TY(I,J)
```

```
      NFL=NFL+1
      IF(DGL1FG) WRITE(6,2) I
      MX=IFXT(I,J)
      IF(MX.EQ.0)GO TO 20
      ISEG=JEXT(MX)+2
      GO TO 25
   20 ISEG=2
   25 IF(DGL2FG) WRITE(6,120) (RO(IL),IL=1,3),ISEG,NFL
C AT THIS POINT,MX IS ZERO AND ISEG IS 2 IF THIS
C FLIGHT HAS NO EXTENSIONS.IF IT DOES HAVE SOME EXTENSIONS
C THEN MX IS THE APPROPRIATE POSITION IN THE ARRAYS IN
C COMMON/EXTS/TO LOOK FOR INFORMATION REGARDING THE
C OF SEGMENTS ON THIS FLIGHT.IF ISEG IS 4 THEN THE
C INFORMATION FOR THE SECOND EXTENSION MAY BE FOUND
C IN POSITION MX+1 OF THE ARRAYS OF COMMON/EXTS/.
      CALL HBETA(XDIST,SINB,PFN,MXS,RO,ISEG,NFL)
C HBETA DETERMINED H(=XDIST),SIN(BETA)(=SINB),THE PERCENT
C THRUST(=PFN),AND THE SEGMENT USED IN CALCULATING THESE
C (=MXS).
      IF(JFICS .EQ. 99) GO TO 100
      IPRF=IPR(NFL)
      IF(.NOT. DGL1FG) GO TO 28
      CALL FNT2(ACTYPE,WEIGHT,PFN,IPRF,FLIGHT)
   28 X=ARSIN(SINB)*180./PI
C N IS THE NUMBER OF IDENICAL OPERATIONS
C     Y1=N(I,J)
C     THIS IS A TEMPORARY PATCH TO RESTRICT THE EVENTUAL 3-TYPE
C     OPERATION AS INDICATED IN INPUT TO THE SINGLE TYPE OPERATION OF
C     THE MOD1 VERSION.  N REPLACED BY NTEMP
      Y1=N(1,I,J)+16.67*N(3,I,J)
C     END OF PATCH
      IF(DGL1FG) CALL FNT3(X,N(1,I,J),XDIST)
C BNOISE FINDS THE OBSERVED EPNL NEGLECTING SHIELDING,
C GROUND ATTENUATION ,AND CHANGES FOR IDENICAL OPERATIONS.
   30 IF(NOISEG) GO TO 32
      CALL BNOISE(ACTYPE,XDIST)
      GO TO 33
   32 CALL BNOIIN(ACTYPE,XDIST,NFL,MXS)
   33 CONTINUE
C THE CORRECTION FOR EXTRA GROUND ATTENUATION IS-
      CALL ATTENU(SINB,XDIST,FLIGHT,X)
C THE SHIELDING TERM IS -
      Y=0.
C CALLING THE CORRECTED EPNL EXPOSE ,WE GET -
      XXX=EPNDB-(X+Y)
C THE CORRECTION FOR Y1 IDENICAL OPERATIONS IS -
      Z=10.*ALOG10(Y1)
C THE OBSERVED EPNL IS GIVEN BY XNE -
      XNE=XXX+Z
 9000 CONTINUE
      IF(XNE.LT.XNEM)GO TO 90
      MFL=NFL
      MXSG=MXS
      MXX=MX
      XNEM=XNE
```

```
C ONCE ALL FLIGHTS ARE CONSIDERED AXE WILL CONTAIN THE
C SUM OF THE ANTILOGS OF ONE TENTH THE OBSERVED EPNLS
C CONTRIBUTED FROM EACH FLIGHT.
   90 AXE=AXE+EXP(XLN10*(XNE/10.))
      IF(DGL1FG) CALL ENT5N(X,Y,Z,XNE,AXE)
  100 CONTINUE
C CALLING THE NOISE EXPOSURE EXPOSE WE GET -
      XXX=10.*ALOG10(AXE)-88.01
      IF(DGL1FG) WRITE(6,110) XXX
  110 FORMAT(/22X,10H NO. 88.01/22X,1H ,7(1H*)//9X,6H NE IS,21X,F7.2)
  120 FORMAT(21H FROM EXPOSE TO HBETA /12H RO,ISEG,NEL /1H ,3F12.3,2I4)
 9100 CONTINUE
      RETURN
      END
      INTEGER FUNCTION IEXT(IS,JS)
C THIS FUNCTION GIVES THE EXTENSION NO. IF ONE EXISTS AND IS 0 OTHERWISE
C THIS VALUE THEN SHOULD BE USED WHEN REFERENCING THE
C THE ARRAYS OF COMMON/EXTS/FOR INFORMATION ABOUT
C THE 1ST EXTENSION(3RD SEGMENT) OF THAT FLIGHT.IF
C JEXT(IEXT) IS TWO THEN THE FLIGHT HAS TWO EXTENSIONS
C AND THE IEXT+1 POSITION OF THE ARRAYS IN COMMON/EXTS/
C SHOULD BE USED TO GET INFORMATION ABOUT THE SECOND EXT.
C SHOULD BE USED TO GET INFORMATION ABOUT THE SECOND EXT.
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IFT(400),MFL,MXX,MXSG
      IEXT=0
      DO 100 KS=1,NEXT
      IF(JRW(KS).NE.JS.OR.IFT(KS).NE.IS)GO TO 100
      IEXT=KS
      GO TO 120
  100 CONTINUE
  120 RETURN
      END
      SUBROUTINE VCTPRP(POS,UNIT,PT,PRP)
C THIS SUBROUTINE GIVES A VECTOR PRP THAT IS PERPENDICULAR
C TO THE LINE DEFINED AS PASSING THROUGH THE POINT POS
C AND HAVING UNIT DIRECTION UNIT.FURTHERMORE THE
C MAGNITUDE OF PRP IS THE DISTANCE FROM THE POINT HAVING
C POSITION VECTOR PT TO THAT LINE.
      DIMENSION POS(3),UNIT(3),PT(3),A(3),PRP(3)
      CALL VSUB(A,POS,PT)
      UCOF=VDOT(A,UNIT)
      CALL VSCL(PRP,UCOF,UNIT)
      CALL VSUB(PRP,A,PRP)
      RETURN
      END
      FUNCTION THRST(TH1,TH2,X)
C AN A/C HAS THRUST T1 ON ONE SEGMENT AND T2 ON THE NEXT,
C THE TRANSITION WILL BE SMOOTHED IN SUCH A WAY THAT IT WILL
C BE 99 PERCENT COMPLETE AFTER THE A/C HAS TRAVELED ABOUT ONE MILE
C AND APPROACHES 100 PERCENT AS A LIMIT.
C THE TRANSITION ALWAYS TAKES PLACE ON THAT SEGMENT
C WHICH IS FARTHER FROM THE RUNWAY,WHETHER ON TAKE-OFF
C OR LANDING.
      THETA=ABS(X)/5000.
```

```
      IF(ABS(X).GT.5000.)THETA=1.
      THRST=TH1+THETA*(TH2-TH1)
      RETURN
      END
      SUBROUTINE DSTORE
C THIS SUBROUTINE DEVELOPES THE VALUES FOR THE ARRAYS
C IN COMMON/XNEW/.
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG2/DGLOFG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/HLX/HFLCN(3,10,150)
      COMMON/CC/LAMBD1(3),LAMBD2(3),TAU1,TAU2,WEIGHT,ACTYPE,
     1DIST,SLOPE,UG(3),SINB,XDIST,GLUMP
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IFT(400),MFL,MXX,MXSG
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),
     1XLM1(3,10),XLM2(3,10),STO(10),SL(10)
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/TRNS/PI
      COMMON/ZXX/IPRFL,IPR(150)
      COMMON/BK/DIAG,YES,MAYBE
      DIMENSION XGIM1(3),XGIM2(3),XGIM3(3),XGIM4(3)
      DIMENSION UV(3)
      DIMENSION XLMBD1(3)
      DIMENSION VEC(3),VEC1(3),VEC2(3)
      DIMENSION VOBR(3)
      REAL N
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      LOGICAL DGLOFG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      REAL LAMBD1,LAMBD2
      REAL MAYBE
      NFL=0
      DO 1000 JS=1,NRW
      CALL DATA(JS)
      M=NFT(JS)
      DO 1000 IS=1,M
      KS=IFXT(IS,JS)
      MEXT=JEXT(KS)
      NFL=NFL+1
      FLIGHT=TP(IS,JS)
      WEIGHT=W(IS,JS)
      ACTYPE=TY(IS,JS)
      ITEMP1=NACM(FLIGHT,ACTYPE)
C     TEST TO SEE IF TAKEOFF ANGLE AND DIST ARE INPUT
      TEMP1=THETAI(2,NFL)
      IF(TEMP1 .EQ. 0.) GO TO 38
C THEY ARE INPUT
      DIST=ZETI(1,NFL)
      SLOPE=TAN(TEMP1)
      GO TO 39
   38 CONTINUE
```

```
      CALL SLPDS(ITEMP1,WEIGHT,DIST,IF,SLOPE)
   39 CONTINUE
      IPR(NFL)=IPREL
C FLIGHT=1 IMPLIES A LANDING.
      IF(FLIGHT.EQ.1.)GO TO 40
      IU=1
      GO TO 45
   40 IU=-1
C UG IS IN THE DIRECTION OF FLIGHT.ON BOTH TAKEOFF
C AND LANDINGS THE U AND GIM ARRAYS SHOULD INCREASE
C IN THE DIRECTION AWAY FROM THE RUNWAY.
C HENCE FOR LANDING LET UV=-UG.
 45   CONTINUE
      FFIU=IU
C     SET UP POSITION AND UNIT VECTORS FOR FIRST TWO SEGMENTS OF FLIGHT
      CALL VSCL(UV,FFIU,UG)
      DO 100 LX=1,3
      IF (FLIGHT.EQ.1.)GO TO 50
C ON TAKEOFF GIM1 IS GLUMP FEET DOWN THE RUNWAY.
C GLUMP IS THE DISTANCE TO TAKEOFF ROLL.
      XLMBD1(LX)=LAMBD1(LX)
      GIMC(LX,1,NFL)=XLMBD1(LX)+GLUMP*UV(LX)
C DIST IS THE ROLL DISTANCE FOR TAKEOFF.
      GIMC(LX,2,NFL)=GIMC(LX,1,NFL)+DIST*UV(LX)
      GO TO 60
   50 CONTINUE
C TAU1 IS THE DISTANCE TO TOUCHDOWN.
C THE AIRCRAFT IS ASSUMED TO ROLL 1 MILE AFTER TOUCHDOWN.
      XLMBD1(LX)=LAMBD1(LX)
      GIMC(LX,1,NFL)=XLMBD1(LX)-(TAU1+5280.)*UV(LX)
      GIMC(LX,2,NFL)=GIMC(LX,1,NFL)+5280.*UV(LX)
   60 CONTINUE
C SLOPE IS THE TANGENT OF THE CLIMBOUT(OR DESCENT) ANGLE
C FOUND FROM THE AIRCRAFT WEIGHT AND TYPE.
      THETA1=ATAN(SLOPE)
C     STORE CLIMBOUT(DESCENT) ANGLE IN INPUT ELEVATION ARRAY
      THETAI(2,NFL)=THETA1
      SINX=SIN(THETA1)
      UNC(LX,1,NFL)=UV(LX)
      UNC(LX,2,NFL)=COS(THETA1)*UV(LX)
      IF(LX .EQ. 3) UNC(LX,2,NFL)=UNC(LX,2,NFL)+SINX
  100 CONTINUE
C IF NO EXTENSIONS EXIST, GO TO STATEMENT 1000
      IF(KS .EQ. 0)GO TO 1000
      NOSG=NSG(NFL)
      NOXT=NOSG-2
      DO 900 LX=3,NOSG
C DETERMINE IF PRECEDING SEGMENT LINEAR. IF NOT GO TO STATEMENT 700
      IF(RADI(LX-1,NFL) .NE. 0.) GO TO 700
C PRECEDING SEGMENT LINEAR
      TEMP=ZETI(LX-1,NFL)
      TEMP1=THETAI(LX-1,NFL)
      TEMP1=TEMP*TAN(TEMP1)
      TEMP=SQRT(TEMP**2+TEMP1**2)
      CALL VSCL(VEC,TEMP,UNC(1,LX-1,NFL))
```

```
      CALL VADD(GIMC(1,LX,NFL),VEC,GIMC(1,LX-1,NFL))
      CALL FLUV(LX,NFL)
      GO TO 750
C PRECEDING SEGMENT NON-LINEAR
  700 TH=ZETI(LX-1,NFL)
C     SET INDICATOR TO INDICATE OBSERVER NOT RELEVANT
      VOBR(1)=-9999.
      CALL PARSU1(VOBR,LX-1,NFL)
      CALL FNHEV(TH,VEC)
      CALL VADD(VEC,VEC,HFLCN(1,LX-1,NFL))
      DO 701 IJ=1,3
  701 GIMC(IJ,LX,NFL)=VEC(IJ)
      CALL FUDVH(TH,VEC)
      DO 702 IJ=1,3
  702 UNC(IJ,LX-1,NFL)=VEC(IJ)
      CALL FLUV(LX,NFL)
C DETERMINE IF CURRENT SEGMENT IS NON-LINEAR
  750 IF(RADI(LX,NFL) .EQ. 0.) GO TO 800
C CURRENT SEGMENT NON-LINEAR. AT THIS POINT THE POSITION VECTOR OF THE
C INITIAL POINT, THE INITIAL UNIT VECTOR OF THE SEGMENT, AND THE ELEVA-
C TION ANGLE ARE ALL KNOWN.
C FIND CENTER OF HELIX
      CALL CENHEL(LX,NFL)
  800 CONTINUE
  900 CONTINUE
 1000 CONTINUE
C IF DIAGNOSTIC LEVEL 1 WANTED, OUTPUT     SEGMENT POS. AND UNIT VECTORS
      IF(.NOT. DGLIFG .AND. .NOT. PTTBFG) GO TO 2000
      WRITE(6,1600)
      WRITE(6,1604)
      NFL=0
      DO 1570 IJ=1,NRW
      IJK=NFT(IJ)
      DO 1570 JI=1,IJK
      NFL=NFL+1
      K1=NSG(NFL)-2
      WRITE(6,1601) IJ,JI,K1
      WRITE(6,1602)
      K1=K1+2
      DO 1570 K2=1,K1
      DO 1565 K3=1,3
      VEC1(K3)=GIMC(K3,K2,NFL)
 1565 VEC2(K3)=HELCN(K3,K2,NFL)
      IF(MKSFLG) CALL FTTOM(VEC1,3)
      WRITE(6,1605) K2,PTHRI(K2,NFL),VEC1,(UNC(K3,K2,NFL),K3=1,3)
      IF(RADI(K2,NFL) .EQ. 0.) GO TO 1560
      IF(MKSFLG) CALL FTTOM(VEC2,3)
      WRITE(6,1606) VEC2
 1560 CONTINUE
 1570 CONTINUE
      PTTBFG=.FALSE.
 1600 FORMAT(1H1)
 1601 FORMAT(/8H RUNWAY ,I2,10H   FLIGHT ,I2,20H   NO. OF EXTENSIONS,I3)
 1602 FORMAT(120H SEG  THRUST       GAMMA X      GAMMA Y      GAMMA Z      U
     1NIT X      UNIT Y      UNIT Z HELIX CEN X HELIX CEN Y HELIX CEN Z)
```

```
 1604 FORMAT(120H ***POSITION VECTORS OF END POINTS OF SEGMENTS, UNIT VE
     1CTORS OF SEGMENTS, AND POSITION VECTORS OF HELIX CENTERS***       )
 1605 FORMAT(1H ,I3,F8.1,6F12.3)
 1606 FORMAT(1H+,83X,3F12.3)
 2000 RETURN
      END
      SUBROUTINE FALLIN(NSEGM,NFL,X)
C THIS SUBROUTINE DETERMINES IF A THE PROJECTED OBSERVER POINT LIES
C WITHIN,TO THE LEFT, TO THE RIGHT OF A SEGMENT OR WHETHER IT IS UNDE-
C FINED. IT SETS UP THE ARRAY IDTOSI AS FOLLOWS- 1 LEFT OF SEGMENT, 2 ON
C SEGMENT, 3 RIGHT OF SEGMENT, 4 UNDEFINED.
C
      COMMON/TRNS/PI
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/MNDIS/DTSXYZ (3,10),DTSMIN(10),IDTOSI(10)
      DIMENSION VEC(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
C TEST TO SEE IF LINEAR SEGMENT
      IF(X .NE. -100.) GO TO 100
C LINEAR SEGMENT
C     TEST TO SEE IF FINAL SEGMENT OF FLIGHT
      IF(NSEGM .NE. NSG(NFL)) GO TO 12
C     FINAL SEGMENT
      CALL VSCL(VEC,100000.,UNC(1,NSEGM,NFL))
      CALL VADD(GIMC(1,NSEGM+1,NFL),GIMC(1,NSEGM,NFL),VEC)
   12 CONTINUE
      DO 10 I=1,3
      S1=GIMC(I,NSEGM,NFL)
      S2=GIMC(I,NSEGM+1,NFL)
      IF(ABS(S1-S2) .GT. .1) GO TO 50
   10 CONTINUE
   50 T=DTSXYZ(I,NSEGM)
      IF(T .EQ. -99999.) GO TO 93
      IF(S1 .LT. S2) GO TO 60
      SR=S1
      SL=S2
      K1=2
      GO TO 70
   60 SL=S1
      SR=S2
      K1=1
   70 IF(T .GE. SL .AND. T .LE. SR) GO TO 90
      IF(T .GE. SL .AND. T .GE. SR) GO TO 91
      IF(T .LE. SL .AND. T .LE. SR) GO TO 92
   90 K=2
      GO TO 1000
   91 IF(K1 .EQ. 1) K=3
      IF(K1 .EQ. 2) K=1
      GO TO 1000
   92 IF(K1 .EQ. 1) K=1
```

```
      IF(K1 .EQ. 2) K=3
      GO TO 1000
   93 K=4
      GO TO 1000
C HELICAL SEGMENT
C TEST TO SEE IF FINAL SEGMENT OF FLIGHT
  100 TH2PI=2.*PI
      THMX=.75*PI
      IF(NSEGM .NE. NSG(NFL)) THMX=ZETI(NSEGM,NFL)
      IF(X .NE. -99999.) GO TO 900
      K=4
      GO TO 1000
  900 XRD=AMOD(X,TH2PI)
      IF(XRD .GE. 0. .AND. XRD .LE. THMX) K=2
      IF(XRD .LT. 0.) K=1
      IF(XRD .GT. THMX) K=3
 1000 IDTOSI(NSEGM)=K
C TEST FOR DIAGNOSTIC
      IF(.NOT. DGL3FG) GO TO 1500
      WRITE(6,300)
  300 FORMAT(12H FROM FALLIN)
      WRITE(6,310) I,NSEGM,NFL,K1,K
  310 FORMAT(17H I,NSEGM,NFL,K1,K/5I10)
  320 FORMAT(31H S1,S2,T,SR,SL,THMX,X,XRD,TH2PI/10F12.3)
      WRITE(6,320) S1,S2,T,SR,SL,THMX,X,XRD,TH2PI
 1500 RETURN
      END
      SUBROUTINE CENHEL(NSEGM,NFL)
C THIS SUBROUTINE FINDS THE COORDINATES OF THE CENTER OF A HELIX(CIRCU-
C LAR).ASSUMED KNOWN IS THE POSITION VECTOR OF THE INITIAL POINT, INI-
C TIAL UNIT VECTOR, AND ELEVATION ANGLE(ABOVE X-Y PLANE) OF HELICAL
C SEGMENT.
C
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/HLX/HELCN(3,10,150)
      DIMENSION PUV(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
C Z-COORDINATE SAME AS THAT OF INITIAL POINT
      HELCN(3,NSEGM,NFL)=GIMC(3,NSEGM,NFL)
C PUV IS PROJECTED UNIT VECTOR ONTO X-Y PLANE
      DO 10 I=1,3
      PUV(I)=UNC(I,NSEGM,NFL)
      IF(I .EQ. 3) PUV(I)=0.
   10 CONTINUE
      RAD=RADI(NSEGM,NFL)
      ARAD=ABS(RAD)
C YA - YH
      TEMP=PUV(1)*ARAD/SQRT(PUV(1)**2+PUV(2)**2)
C XA - XH
      TEMP3=ABS(PUV(1))
```

```
      IF(TEMP3.LT..001) GO TO 15
      TEMP1=-TEMP*PUV(2)/PUV(1)
      GO TO 16
   15 TEMP1=-ARAD
   16 CONTINUE
      CP=PUV(1)*TEMP-PUV(2)*TEMP1
C SIGNS OF CP AND RADIUS AGREE
      IF(CP .GT. 0..AND.RAD .GT. 0. .OR. CP .LT. 0..AND. RAD .LT. 0.)
     1  GO TO 20
C SIGNS DISAGREE
      TEMP=-TEMP
      TEMP1=-TEMP1
      GO TO 20
   20 HELCN(2,NSEGM,NFL)=GIMC(2,NSEGM,NFL)-TEMP
      HELCN(1,NSEGM,NFL)=GIMC(1,NSEGM,NFL)-TEMP1
C DIAGNOSTIC TEST
      IF(.NOT. DGL3EG) GO TO 100
      WRITE(6,30)
   30 FORMAT(12H FROM CENHEL)
      WRITE(6,35) PUV,(HELCN(I,NSEGM,NFL),I=1,3),RAD,ARAD,TEMP,TEMP1,CP
   35 FORMAT(33H PUV,HELCN,RAD,ARAD,TEMP,TEMP1,CP/(8F15.3))
      WRITE(6,40) NSEGM,NFL
   40 FORMAT(10H NSEGM,NFL/2I10)
  100 RETURN
      END
      SUBROUTINE READOU
C DUMMY
      RETURN
      END
      SUBROUTINE RUNWAY(X1,Y1,X2,Y2)
      COMMON/IANG/SINAN,COSAN,XW,YW
      XW=0.
      YW=0.
      DELX=.075
      DELY=.05
      CALL PLOT(X1,Y1,-3)
      XX2=SQRT((Y2-Y1)**2+(X2-X1)**2)
      SINAN=(Y2-Y1)/XX2
      COSAN=(X2-X1)/XX2
      CALL PLOTA(0.,DELY,3)
      CALL PLOTA(0.,-DELY,2)
      CALL PLOTA(XX2,DELY,3)
      CALL PLOTA(XX2,-DELY,2)
      YY=DELX
      DO 10 I=1,5
      YY=YY-DELX/3.
      CALL PLOTA(0.,YY,3)
   10 CALL PLOTA(XX2,YY,2)
      CALL PLOT(-X1,-Y1,-3)
      RETURN
      END
      SUBROUTINE HBETA(HX,BETAS,PEN,MAXSEG,OBR,ISEG,NFL)
C THIS SUBROUTINE DETERMINES THE DISTANCE HX FROM OBSERVER OBR TO FLIGHT
C PATH TOGETHER WITH SINE    OF ELEVATION ANGLE BETAS AND PERCENT THRUST
C PEN. OTHER ARGUMENTS ARE!
```

```
C          MAXSEG    SEGMENT NUMBER THAT YIELDS MINIMUM DISTANCE
C          ISEG      NUMBER OF SEGMENTS IN FLIGHT
C          NFL       NO. OF FLIGHT(CUMULATIVE)
      COMMON/EPRIN/JEICS
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZFTI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/MNDIS/DTSXYZ (3,10),DTSMIN(10),IDTOSI(10)
      DIMENSION VEC(3),VECSM(3),VEC1(3),OBR(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      NSEG=ISEG
      DO 50 I=1,10
   50 IDTOSI(I)=0
      DO 100 I=1,NSEG
C     LINEAR SEGMENT TEST
      IF(RADI(I,NFL) .NE. 0.) GO TO 80
C LINEAR SEGMENT
      CALL VCTPRP(GIMC(1,I,NFL),UNC(1,I,NFL),OBR,VEC)
      CALL VMAG(VEC,TEMP)
      DTSMIN(I)=TEMP
      CALL VADD(VECSM,VEC,OBR)
      DO 70 II=1,3
   70 DTSXYZ(II,I)=VECSM(II)
C DETERMINE IF PROJECTED POINT WITHIN LINEAR SEGMENT
      CALL FALLIN(I,NFL,-100.)
      GO TO 100
C     HELICAL SEGMENT
   80 CALL HELMN(OBR,I,NFL,VEC,TH)
      CALL VADD(VECSM,VEC,OBR)
      DO 700 II=1,3
  700 DTSXYZ(II,I)=VECSM(II)
C DETERMINE IF TH FALLS WITHIN HELICAL SEGMENT
      CALL FALLIN(I,NFL,TH)
  100 CONTINUE
C CHOOSE CORRECT SEGMENT OF FLIGHT
      JEICS=0
      CALL SEGCH(ISN,NFL,OBR)
      IF(JEICS .EQ. 99) RETURN
      MAXSEG=ISN
      CALL VSUB(VEC,DTSXYZ(1,ISN),GIMC(1,ISN,NFL))
      CALL VMAG(VEC,X)
C IF CHOSEN SEGMENT IS 1 OR 2, NO DISCONTINUITY EXISTS BETWEEN THEM IN
C THRUST VALUES
      IF(ISN .EQ. 1 .OR. ISN .EQ. 2) GO TO 120
C CALCULATE CORRECT THRUST FOR HELICAL PROJECTED POINT
      RXX=RADI(ISN,NFL)
      IF(RXX .EQ. 0.) GO TO 118
      CALL HELMN(OBR,ISN,NFL,VEC,TH)
      X=ABS(RXX)*TH
  118 CONTINUE
      TH1=PTHRI(ISN-1,NFL)
      TH2=PTHRI(ISN,NFL)
```

```
C USE THRST TO SMOOTH DISCONTINUITY BETWEEN SEGMENT AND PRECEDING SEGMN
      PFN= THRST(TH1,TH2,X)
      GO TO 125
  120 PFN=PTHRI(ISN,NFL)
  125 CONTINUE
C DETERMINE HX AND BETAS
      IF(ISN .EQ. 1 .AND. IDTOSI(1) .EQ. 1) GO TO 140
      CALL VSUB(VEC1,DTSXYZ(1,ISN),OBR)
      CALL VMAG(VEC1,HX)
C     CALCULATE SINE OF ELEVATION ANGLE BETAS
      BETAS=VEC1(3)/HX
      GO TO 145
  140 CALL VSUB(VEC1,GIMC(1,1,NFL),OBR)
      CALL VMAG(VEC1,HX)
      BETAS=0.
  145 CONTINUE
C DIAGNOSTICS FOR SUBROUTINE HBETA
  130 IF(.NOT. DGL3FG) GO TO 250
      WRITE(6,190)
  190 FORMAT(11H FROM HBETA)
      WRITE(6,195) HX,BETAS,PFN,OBR,VEC,VECSM,VEC1,TH
  195 FORMAT(35H HX,BETAS,PFN,OBR,VEC,VECSM,VEC1,TH/(8F15.3))
      WRITE(6,196) MAXSEG,ISEG,NFL,NSEG
  196 FORMAT(21H MAXSEG,ISEG,NFL,NSEG/4I15)
  250 RETURN
      END
      FUNCTION F1(X)
C FIRST COMPONENT OF VECTOR FROM OBSERVER TO HELIX
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      F1=R+ABS(RAD)*(E*COS(X)+Q*F*SIN(X))
      RETURN
      ENTRY F1P(X)
      F1P=ABS(RAD)*(-E*SIN(X)+Q*F*COS(X))
      F1=F1P
      RETURN
      ENTRY F1PP(X)
      F1PP=-F1+R
      F1=F1PP
      RETURN
      END
      FUNCTION F2(X)
C SECOND COMPONENT OF VECTOR FROM OBSERVER TO HELIX
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      F2=S+ABS(RAD)*(F*COS(X)-Q*E*SIN(X))
      RETURN
      ENTRY F2P(X)
      F2P=ABS(RAD)*(-F*SIN(X)-Q*E*COS(X))
      F2=F2P
      RETURN
      ENTRY F2PP(X)
      F2PP=-F2+S
      F2=F2PP
      RETURN
      END
      FUNCTION F3(X)
```

```
C THIRD COMPONENT OF VECTOR FROM OBSERVER TO HELIX
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      F3=T+ABS(RAD)*X*TAN(ALPHA)
      RETURN
      ENTRY F3P(X)
      F3P=ABS(RAD)*TAN(ALPHA)
      F3=F3P
      RETURN
      ENTRY F3PP(X)
      F3PP=0.
      F3=F3PP
      RETURN
      END
      SUBROUTINE SEGCH(NSEGM,NFL,OBR)
C THIS SUBROUTINE CHOOSES THE SEGMENT ONTO WHICH THE PROJECTED DISTANCE
C FALLS AND WHICH YIELDS THE MINIMUM DISTANCE.
C         NFL    NUMBER OF FLIGHT(CUMULATIVE)
C         NSEGM RETURNED SEGMENT NUMBER
C
      COMMON/ERRIN/JEICS
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/MNDIS/DTSXYZ (3,10),DTSMIN(10),IDTOSI(10)
      DIMENSION VEC(3),OBR(3)
      DIMENSION NTAB(5)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      D=1000000.
      K1=0
      DO 50 I=1,5
   50 NTAB(I)=0
      J=1
      K2=NSG(NFL)
      DO 100 I=1,K2
      IF(IDTOSI(I) .NE. 2) GO TO 100
      NTAB(J)=I
      J=J+1
  100 CONTINUE
  101 IF(J .EQ. 1) GO TO 200
C PROJECTED DISTANCE FALLS WITHIN AT LEAST ONE SEGMENT
C SPECIAL TEST
      IF(IDTOSI(1) .NE. 1) GO TO 140
      CALL VSUB(VEC,GIMC(1,1,NFL),OBR)
      CALL VMAG(VEC,D)
      NSEGM=1
  140 CONTINUE
      JJJ=J-1
      DO 150 I=1,JJJ
      K1=NTAB(I)
      IF(DTSMIN(K1) .GT. D) GO TO 150
      D=DTSMIN(K1)
      NSEGM=K1
```

```
  150 CONTINUE
      GO TO 300
C PROJECTED DISTANCE NOT WITHIN ANY SEGMENT
  200 IF(IDTOSI(1) .EQ. 1 .AND. IDTOSI(K2) .EQ. 3) GO TO 250
      IF(IDTOSI(1) .EQ. 1) GO TO 225
      IF(IDTOSI(K2) .EQ. 3) GO TO 230
C     WRITE(6,205)
C 205 FORMAT(50H PROGRAM CANNOT DETERMINE MINIMUM DISTANCE SEGMENT)
      JEICS=99
      GO TO 305
  250 IF(DTSMIN(K2) .GT. DTSMIN(1)) GO TO  252
      D=DTSMIN(K2)
      NSEGM=K2
      GO TO 280
  225 D=DTSMIN(1)
      NSEGM=1
      GO TO 280
  230 D=DTSMIN(K2)
      NSEGM=K2
      GO TO 280
  252 D=DTSMIN(1)
      NSEGM=1
  280 CONTINUE
  300 IF(.NOT. DGL3FG) GO TO 305
      WRITE(6,301)
  301 FORMAT(11H FROM SEGCH)
      WRITE(6,302) NSEGM,NFL,K1,K2,J,IDTOSI
  302 FORMAT(25H NSEGM,NFL,K1,K2,J,IDTOSI/15I7)
      WRITE(6,303) DTSMIN
  303 FORMAT(13H DTSMIN ARRAY/10F12.3)
  305 RETURN
      END
      SUBROUTINE PARSU1(OBR,NSEGM,NFL)
C THIS SUBROUTINE CALCULATES THE VALUES IN THE CHELX COMMON ARRAY
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/HLX/HELCN(3,10,150)
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      DIMENSION VEC(3),VEC1(3),OBR(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      R1=RADI(NSEGM,NFL)
      IF(R1 .EQ. 0.) GO TO 100
      RAD=R1
      ZH=HELCN(3,NSEGM,NFL)
      ALPHA=THETAI(NSEGM,NFL)
      IF(R1 .LT. 0.) Q=-1.
      IF(R1 .GT. 0.) Q=1.
      CALL VSUB(VEC,GIMC(1,NSEGM,NFL),HELCN(1,NSEGM,NFL))
      CALL VUNT(VEC1,VEC)
      E=VEC1(1)
      F=VEC1(2)
```

```
      IF(OBR(1).NE.-9999.)  GO TO 80
      R=0.
      S=0.
      T=0.
      GO TO 100
   80 R=HELCN(1,NSEGM,NEL)-OBR(1)
      S=HELCN(2,NSEGM,NEL)-OBR(2)
      T=HELCN(3,NSEGM,NEL)-OBR(3)
  100 IF(.NOT. DGL3FG) GO TO 300
      WRITE(6,120)
  120 FORMAT(12H FROM PARSU1)
      WRITE(6,125) R,S,T
  125 FORMAT(1X,3G12.3)
      WRITE(6,127) R,S,T
  127 FORMAT(1X,3Z8)
      WRITE(6,130) R,S,T,RAD,E,F,ALPHA,Q,ZH
  130 FORMAT(25H R,S,T,RAD,E,F,ALPHA,Q,ZH/9F12.3)
  300 RETURN
   .  END
      SUBROUTINE FNHEV(TH,VEC)
C THIS SUBROUTINE FINDS THE VECTOR FROM CENTER OF HELIX TO HELIX FOR ANY
C GIVEN TH(RADIANS) IN I,J,K COORDINATES
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      DIMENSION VEC(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      VEC(1)=F1(TH)-R
      VEC(2)=F2(TH)-S
      VEC(3)=F3(TH)-T
      IF(.NOT. DGL3FG) GO TO 100
      WRITE(6,50)
   50 FORMAT(11H FROM FNHEV)
      WRITE(6,125) R,S,T
  125 FORMAT(1X,3G12.3)
      WRITE(6,127) R,S,T
  127 FORMAT(1X,3Z8)
      WRITE(6,60) VEC,TH,R,S,T
   60 FORMAT(18H VEC ARRA,TH,R,S,T/7F12.3)
  100 RETURN
      END
      SUBROUTINE FUDVH(TH,VEC1)
C THIS SUBROUTINE FINDS THE UNIT DERIVATIVE VECTOR TO A HELIX AT A GIVEN
C TH(RADIANS) VALUE.
      COMMON/CHELX/R,S,T,RAD,E,F,ALPHA,Q,ZH
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      DIMENSION VEC(3),VEC1(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      VEC(1)=F1P(TH)
      VEC(2)=F2P(TH)
      VEC(3)=F3P(TH)
      CALL VUNT(VEC1,VEC)
```

```
      IF(.NOT. DGL3FG) GO TO 100
      WRITE(6,50)
   50 FORMAT(11H FROM FUDVH)
      WRITE(6,60) VEC,VEC1,TH
   60 FORMAT(12H VEC,VEC1,TH/7F12.3)
  100 RETURN
      END
      SUBROUTINE FLUV(NSEGM,NFL)
C THIS SUBROUTINE CALCULATES THE UNIT VECTOR AT THE INITIAL POINT OF A
C SEGMENT GIVEN THE UNIT VECTOR AT THE TERMINAL POINT OF THE PRECEDING
C SEGMENT AND THE ELEVATION ANGLE OF THE CURRENT SEGMENT.
C           NSEGM     CURRENT SEGMENT NUMBER
C           NFL       FLIGHT NUMBER(CUMULATIVE)
C
      COMMON/XNEW/GIMC(3,10,150),UNC(3,10,150)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      DIMENSION VEC(3),VEC1(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
      K1=NSEGM-1
C FIND PROJECTED VECTOR OF PRECEDING UNIT VECTOR ONTO XY PLANE
      VEC(1)=UNC(1,K1,NFL)
      VEC(2)=UNC(2,K1,NFL)
      VEC(3)=0.
C TAKE ITS UNIT VECTOR
      CALL VUNT(VEC1,VEC)
      TH=THETAI(NSEGM,NFL)
C DEVELOP CURRENT UNIT VECTOR
      UNC(1,NSEGM,NFL)=COS(TH)*VEC1(1)
      UNC(2,NSEGM,NFL)=COS(TH)*VEC1(2)
      UNC(3,NSEGM,NFL)=SIN(TH)
      IF(.NOT. DGL3FG) GO TO 100
      WRITE(6,50)
   50 FORMAT(10H FROM FLUV)
      WRITE(6,55)
   55 FORMAT(60H PRECEDING UNIT VECTOR, NEW FLEVATION ANGLE, NEW UNIT VE
     1CTOR)
      WRITE(6,60) (UNC(K2,K1,NFL),K2=1,3),TH,(UNC(K2,NSEGM,NFL),K2=1,3)
   60 FORMAT(7F12.3)
  100 RETURN
      END
      SUBROUTINE EXTARR
C THIS SUBROUTINE SETS UP THE EXTENSION ARRAYS(OF DIMENSION 100 EACH)
C FOR ALL THE EXTENSIONS IN ONE PROCESSING PASS.
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IFT(400),MFL,MXX,MXSG
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
```

```
     1  PTCOFG,PTALFG
        REAL N
        K=0
        KS=0
        DO 200 J=1,NRW
        K1=NFT(J)
        DO 100 I=1,K1
C K IS RUNNING TOTAL OF NUMBER OF FLIGHTS(MAXIMUM ALLOWED 25)
        K=K+1
        NOS=NSG(K)
        NOEXT=NOS-2
        IF(NOEXT .EQ. 0) GO TO 100
        DO 50 L=1,NOEXT
        KS=KS+1
C KS IS RUNNING TOTAL OF NUMBER OF EXTENSIONS(MAXIMUM 100)
        ALT(KS)=ZETI(L+2,K)
        CANGLE(KS)=THETAI(L+2,K)
        THRUST(KS)=PTHRI(L+2,K)
        CURV(KS)=RADI(L+2,K)
        JEXT(KS)=NOEXT
        JRW(KS)=J
        IFT(KS)=I
   50   CONTINUE
  100   CONTINUE
  200   CONTINUE
        NEXT=KS
C TEST FOR DIAGNOSTICS
        IF(.NOT. DGL3FG) GO TO 400
        WRITE(6,300)
  300   FORMAT(12H FROM EXTARR)
        WRITE(6,310) (ALT(L),L=1,KS)
  310   FORMAT(10H ALT ARRAY/(8F15.3))
        WRITE(6,320) (JEXT(L),JRW(L),IFT(L),L=1,KS)
  320   FORMAT(20H JEXT,JRW,IFT ARRAYS/(8I15))
  400   RETURN
        END
        SUBROUTINE HMIND(THET1,THET2,THET,VEC)
C THIS SUBROUTINE FINDS THE VALUE OF THET YIELDING THE MINIMUM DISTANCE
C FROM A POINT(THE OBSERVER) TO A HELIX, THE COMPONENTS OF THE HELIX ARE
C DEFINED IN THE FUNCTION SUBPROGRAMS F1,F2,F3 TOGETHER WITH THEIR DE- V.
C RIVATIVES. THET=-99999. IMPLIES FAILURE TO CONVERGE.
C MIDWAY ANGLE IS INITIAL APPROXIMATION
        COMMON/NERAP/EPS1,EPS2,NOITT
        COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
        DIMENSION VEC(3)
        DIMENSION XARR(4)
        LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
        XARR(1)=ABS(THET2-THET1)/2.
        XARR(2)=THET1
        XARR(3)=THET2
        XARR(4)=2./3.*XARR(1)
        DO 150 J=1,4
        X=XARR(J)
```

```
      DO 100 I=1,NOITT
      G=F1(X)*F1P(X)+ F2(X)*F2P(X) + F3(X)*F3P(X)
      G1=F1(X)*F1PP(X) + F1P(X)**2
      G2=F2(X)*F2PP(X) + F2P(X)**2
      G3=F3(X)*F3PP(X) + F3P(X)**2
      GP=G1+G2+G3
      IF(ABS(GP) .LT. EPS2) GO TO 150
      XNEW=X-G/GP
C TEST FOR CONVERGENCE
      IF(ABS(XNEW-X) .LE. EPS1) GO TO 200
C TEST FOR DIAGNOSTICS
      IF(.NOT. DGL3FG) GO TO 75
      WRITE(6,70)
   70 FORMAT(11H FROM HMIND)
      WRITE(6,71) X,G,G1,G2,G3,GP,XNEW
   71 FORMAT(21H X,G,G1,G2,G3,GP,XNEW/8F15.3)
   75 CONTINUE
      X=XNEW
  100 CONTINUE
  150 CONTINUE
      THET=-99999.
      GO TO 400
  200 THET=XNEW
      VEC(1)=F1(XNEW)
      VEC(2)=F2(XNEW)
      VEC(3)=F3(XNEW)
  400 RETURN
      END
      SUBROUTINE HELMN(OBR,I,NFL,VEC,TH)
C THIS SUBROUTINE FINDS THE MINIMUM VECTOR FROM THE OBSERVER POSITION TO
C THE HELIX AND ALSO THE ANGLE TH WHICH YIELDS THIS MINIMUM.
C
      COMMON/TRNS/PI
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/MNDIS/DTSXYZ (3,10),DTSMIN(10).IDTOSI(10)
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1  PTGDFG,PTCOFG,PTALFG
      DIMENSION OBR(3),VEC(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1  PTCOFG,PTALFG
C SET THE VALUES IN CHELX COMMON BLOCK TO APPROPRIATE VALUES
      CALL PARSU1(OBR,I,NFL)
C CALL MINIMIZATION ROUTINE
      THET2=ZETI(I,NFL)
      IF(THET2 .EQ. 0. .AND. I .EQ. NSG(NFL)) THET2=.75*PI
      CALL HMIND(0.,THET2,TH,VEC)
      IF(TH .EQ. -99999.) GO TO 280
      CALL VMAG(VEC,TMNDS)
C STORE MAGNITUDE IN DTSMIN ARRAY
      DTSMIN(I)=TMNDS
  280 CONTINUE
C TEST FOR DIAGNOSTICS
      IF(.NOT. DGL3FG) GO TO 400
      WRITE(6,300)
```

```
  300 FORMAT(11H FROM HELMN)
      WRITE(6,310) OBR,VFC,TH,TMNDS,THET2
  310 FORMAT(23H OBR,VFC,TH,TMNDS,THET2/(8F15.3))
      WRITE(6,320) I,NFL
  320 FORMAT(10H I,NFL     /2I10)
  400 RETURN
      END
      SUBROUTINE PLOTA(X,Y,I)
      COMMON/IANG/SINAN,COSAN,XW,YW
      XX=X-XW
      YY=Y-YW
      XP=XX*COSAN+YY*SINAN+XW
      YP=XX*SINAN-YY*COSAN+YW
      CALL PLOT(XP,YP,I)
      RETURN
      END
      SUBROUTINE CONTR
      RETURN
      END
      SUBROUTINE AUTCNT(VAL)
      DIMENSION V1(3),V2(3),V3(3),P1(3),P2(3)
      DIMENSION C(4,3),COEF(8,4)
      COMMON/ARP/ZZ(20,25)
      COMMON/GDPAR/CX,CY,XX1,YY1,NS,MS
      COMMON/WM/X1S,Y1S,SF,XO,YO
      COMMON/WARP/Z(27,22)
      COMMON/CAREA/AREA
      AREA=0.
      DX=ABS(CX)
      DY=ABS(CY)
      M=MS+2
      N=NS+2
      X1=X1S-DX
      Y1=Y1S+DY
      DO 3 I=1,NS
      DO 3 J=1,MS
      ZTEMP=ZZ(I,J)
      IP1=I+1
      JP1=J+1
    3 Z(JP1,IP1)=ZTEMP
      DO 6 J=3,N
      L=J-1
      ZTEMP=2.*Z(2,L)-Z(3,L)
      Z(1,L)=ZTEMP
      ZTEMP=2.*Z(M-1,L)-Z(M-2,L)
    6 Z(M,L)=ZTEMP
      DO 10 I=3,M
      L=I-1
      ZTEMP=2.*Z(L,2)-Z(L,3)
      Z(L,1)=ZTEMP
      ZTEMP=2.*Z(L,N-1)-Z(L,N-2)
   10 Z(L,N)=ZTEMP
      Z(1,1)=(2.*(Z(2,1)+Z(1,2)+Z(2,2))
     X        -(Z(3,1)+Z(1,3)+Z(3,3)))/3.0
      ZTEMP=(2.*(Z(1,N-1)+Z(2,N)+Z(2,N-1))
```

```
     X         -(Z(1,N-2)+Z(3,N)+Z(3,N-2)))/3.0
      Z(1,N)=ZTEMP
      ZTEMP=(2.*(Z(M,2)+Z(M-1,2)+Z(M-1,1))
     X         -(Z(M,3)+Z(M-2,3)+Z(M-2,1)))/3.0
      Z(M,1)=ZTEMP
      ZTEMP=(2.*(Z(M,N-1)+Z(M-1,N-1)+Z(M-1,N))
     X         -(Z(M,N-2)+Z(M-2,N-2)+Z(M-2,N)))/3.0
      Z(M,N)=ZTEMP
      ACC=.1
      NFINE=ABS(DX*SF/ACC)+1.0
      MFINE=ABS(DY*SF/ACC)+1.0
      DO 9 I=2,M
      DO 9 JQ=2,N
      J=JQ
      J14=MOD(I,2)
      IF(J14.NE.0) J=N+2-JQ
      IF(NFINE.LE.2.OR.MFINE.LE.2) GO TO 30
      IF((I.GT.2.AND.I.LT.M).AND.(J.GT.2.AND.J.LT.N)) GO TO 8
      GO TO 9
   30 DO 4 L=1,4
      JM=J+MOD(L/2,2)-1
      IM=I+MOD((L+3)/2,2)-1
      C(L,1)=X1+DX*FLOAT(JM-1)
      C(L,2)=Y1-DY*FLOAT(IM-1)
    4 C(L,3)=Z(IM,JM)
      CALL CRECT(C,VAL)
      GO TO 9
    8 CALL CUBICS(I,J,COEFF)
      DO 7 II=2,MFINE
      DO 7 JK=2,NFINE
      JJ=JK
      J24=MOD(II,2)
      IF(J24.NE.0) JJ=NFINE+2-JK
      DO 5 L=1,4
      JM=JJ+MOD(L/2,2)-1
      IM=II+MOD((L+3)/2,2)-1
      C(L,1)=FLOAT(JM-1)/FLOAT(NFINE-1)-1.
      C(L,2)=FLOAT(IM-1)/FLOAT(MFINE-1)-1.
      CALL INTPT(COEFF,C(L,1),C(L,2),C(L,3))
      TCC=C(L,1)*DX+X1+DX*FLOAT(J-1)
      C(L,1)=TCC
      TCC=-C(L,2)*DY+Y1-DY*FLOAT(I-1)
    5 C(L,2)=TCC
      CALL CRECT(C,VAL)
    7 CONTINUE
    9 CONTINUE
      DO 24 I=1,MS
   24 WRITE(6,25)(ZZ(J,I),J=1,NS)
   25 FORMAT(1X,10F7.1/1X,10F7.1)
      WRITE(6,26)CX,CY,XX1,YY1,NS,MS
   26 FORMAT(1H1,2F17.5///1X,2F17.5////1X,2I10)
      WRITE(6,27)X1S,Y1S,SF,XO,YO
   27 FORMAT(////1X,5F17.5)
      DO 28 I=1,M
   28 WRITE(6,25)(Z(I,J),J=1,N)
```

```
      RETURN
      END
      SUBROUTINE CRECT(C,VAL)
      DIMENSION C(4,3),V1(3),V2(3),V3(3),P1(3),P2(3)
      COMMON/ARP/ZZ(20,25)
      COMMON/GDPAR/CX,CY,XX1,YY1,N,M
      COMMON/WM/X1,Y1,SF,XO,YO
      COMMON/WARP/Z(27,22)
      DX=ABS(CX)
      DY=ABS(CY)
      V2(1)=(C(1,1)+C(2,1))/2.0
      V2(2)=(C(2,2)+C(3,2))/2.0
      V2(3)=(C(1,3)+C(2,3)+C(3,3)+C(4,3))/4.0
      DO 81 K=5,8
      IL=MOD((K-1),4)+1
      IT=MOD((K-2),4)+1
      V1(1)=C(IL,1)
      V1(2)=C(IL,2)
      V1(3)=C(IL,3)
      V3(1)=C(IT,1)
      V3(2)=C(IT,2)
      V3(3)=C(IT,3)
      CALL STRIKE(V1,V2,V3,VAL,P1,P2,IER)
      IF(IER.NE.2) GO TO 81
      U1=P1(1)*SF+XO
      V11=P1(2)*SF+YO
      U2=P2(1)*SF+XO
      V22=P2(2)*SF+YO
      CALL PLOT(U1,V11,3)
      CALL PLOT(U2,V22,2)
   81 CONTINUE
      RETURN
      END
      SUBROUTINE CUBICS(I,J,COEF)
      DIMENSION COEF(8,4)
      COMMON/WARP/Z(27,22)
      DO 3 IEQ=1,4
      IB=I+IEQ-3
      COEF(IEQ,1)=Z(IB,J)
      CF=(Z(IB,J-2)-6.*Z(IB,J-1)+3.*Z(IB,J)
     X    +2.*Z(IB,J+1))/6.
      COEF(IEQ,2)=CF
      CF=(Z(IB,J-1)+Z(IB,J+1))/2.0-Z(IB,J)
      COEF(IEQ,3)=CF
      CF=(-Z(IB,J-2)+3.*Z(IB,J-1)-3.*Z(IB,J)
     X    +Z(IB,J+1))/6.0
    3 COEF(IEQ,4)=CF
      DO 4 IEQ=5,8
      JB=J+IEQ-7
      COEF(IEQ,1)=Z(I,JB)
      CF=(Z(I-2,JB)-6.*Z(I-1,JB)+3.*Z(I,JB)
     X    +2.*Z(I+1,JB))/6.0
      COEF(IEQ,2)=CF
      CF=(Z(I-1,JB)+Z(I+1,JB))/2.0-Z(I,JB)
      COEF(IEQ,3)=CF
```

```
      CF=(-Z(I-2,JB)+3.*Z(I-1,JB)-3.*Z(I,JB)
     X   +Z(I+1,JB))/6.0
    4 COEFF(IEQ,4)=CF
      RETURN
      END
      SUBROUTINE INTPT(COFF,XREL,YREL,HT)
      DIMENSION COEF(8,4),ZH(4)
      DO 30 IEQ=1,4
      IF(XREL .NE. 0.) GO TO 2
      ZH(IEQ)=COFF(IEQ,1)
      GO TO 30
    2 ZH(IEQ)=0.
      DO 3 J=1,4
      ZTEMP=ZH(IEQ)+COFF(IEQ,J)*XREL**(J-1)
    3 ZH(IEQ)=ZTEMP
   30 CONTINUE
      ZY=ZH(3)
     X      +YREL*(ZH(1)-6.*ZH(2)+3.*ZH(3)+2.*ZH(4))/6.
     X      +YREL**2*((ZH(2)+ZH(4))/2.-ZH(3))
     X      +YREL**3*(-ZH(1)+3.*ZH(2)-3.*ZH(3)+ZH(4))/6.
      DO 50 IEQ=5,8
      I=IEQ-4
      IF(YREL .NE. 0.) GO TO 4
      ZH(I)=COFF(IEQ,1)
      GO TO 50
    4 ZH(I)=0.
      DO 5 J=1,4
      ZTEMP=ZH(I)+COFF(IEQ,J)*YREL**(J-1)
    5 ZH(I)=ZTEMP
   50 CONTINUE
      ZX=ZH(3)
     X      +XREL*(ZH(1)-6.*ZH(2)+3.*ZH(3)+2.*ZH(4))/6.
     X      +XREL**2*((ZH(2)+ZH(4))/2.-ZH(3))
     X      +XREL**3*(-ZH(1)+3.*ZH(2)-3.*ZH(3)+ZH(4))/6.
      HT=(ZX+ZY)/2.
      RETURN
      END
      SUBROUTINE STRIKE(V1,V2,V3,H,P1,P2,IER)
      COMMON/CAREA/AREA
      DIMENSION V1(1),V2(1),V3(1),P1(1),P2(1),Q(3,3)
      ARINC=ABS((V3(1)-V1(1))*(V2(2)-V1(2))-(V3(2)-V1(2))*(V2(1)-V1(1)))
     1/2.
      DO 6 I=1,3
      P1(I)=0.
      P2(I)=0.
      T1=V1(I)
      Q(I,1)=T1
      T1=V2(I)
      Q(I,2)=T1
      T1=V3(I)
    6 Q(I,3)=T1
      DO 7 I=1,3
      DO 1 J=I,3
      IF(Q(3,I)-Q(3,J))1,1,2
    2 DO 8 L=1,3
```

```
      A=Q(L,I)
      T1=Q(L,J)
      Q(L,I)=T1
      Q(L,J)=A
    8 CONTINUE
    1 CONTINUE
    7 CONTINUE
      IER=0
      IF(Q(3,3).LT.H.OR.Q(3,1).GT.H)GO TO 10
      IF(Q(3,1).EQ.H.AND.Q(3,2).EQ.H.AND.Q(3,3).EQ.H) GO TO 20
      A=(H-Q(3,1))/(Q(3,1)-Q(3,3))
      P1(1)=(Q(1,1)-Q(1,3))*A+Q(1,1)
      P1(2)=(Q(2,1)-Q(2,3))*A+Q(2,1)
      P1(3)=H
      IF(Q(3,1).NE.H) GO TO 3
      IF(Q(3,2).NE.H)GO TO 4
      P2(1)=Q(1,2)
      P2(2)=Q(2,2)
      P2(3)=Q(3,2)
      IER=2
      GO TO 20
    3 IF(Q(3,2).GT.H)GO TO 5
      A=(H-Q(3,2))/(Q(3,2)-Q(3,3))
      P2(1)=(Q(1,2)-Q(1,3))*A+Q(1,2)
      P2(2)=(Q(2,2)-Q(2,3))*A+Q(2,2)
      P2(3)=H
      IER=2
      GO TO 20
    4 IER=1
      P2(1)=P1(1)
      P2(2)=P1(2)
      P2(3)=H
      RETURN
    5 A=(H-Q(3,1))/(Q(3,1)-Q(3,2))
      P2(1)=(Q(1,1)-Q(1,2))*A+Q(1,1)
      P2(2)=(Q(2,1)-Q(2,2))*A+Q(2,1)
      P2(3)=H
      IER=2
   20 AREA=AREA+ARINC/2.
      RETURN
   10 IF(Q(3,1).GT.H)AREA=AREA+ARINC
      RETURN
      END
      SUBROUTINE CALPLT
      COMMON/CAREA/AREA
      COMMON/RYNME/RYNAM
      COMMON/FACTR/AXMAX,AXMIN,AYMAX,AYMIN,DELX,DELY,AINCHX,AINCHY,
     1  SETPLT
      COMMON/WM/XULHC,YULHC,XSF,XZIN,YHT
      COMMON/ARR/ARNV(20,25)
      COMMON/LOGFG2/DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,
     1    PTGDFG,PTCOFG,PTALFG
      COMMON/GDPAR/CX,CY,X1,Y1,NX,NY
      COMMON/JOF2/NFFFLG,NOISFG
      COMMON/JOF1/WFCDFG,CANCEL
```

```
      COMMON/TROUPL/FLIGHT(22,20)
      REAL*8 RYNAM(10)
      REAL*8 GEONAM
      REAL*8 PNOIS(3)
      LOGICAL DGL0FG,DGL1FG,DGL2FG,DGL3FG,PTINFG,PTTBFG,PTOUFG,PTGDFG,
     1   PTCOFG,PTALFG
      LOGICAL NEFFLG,NOISFG
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/LOGFG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/EXTS/ALT(400),CANGLE(400),THRUST(400),CURV(400)
      COMMON/EXTS1/NEXT,JEXT(400),JRW(400),IET(400),MFL,MXX,MXSG
      COMMON/JOE/PEELIN,XCOORD(2),YCOORD(2),TITLE(12),SYMFLG
      COMMON/NN/N(3,20,10),NRW,NET(10)
      COMMON/XDATA/ZETI(10,150),THETAI(10,150),PTHRI(10,150),
     1  RADI(10,150),NSG(150),NFLGHT
      COMMON/AA/TP(20,10),PT(20,10),W(20,10),TY(20,10),XLMI(3,10),
     X XLM2(3,10),STO(10),SL(10)
      DIMENSION SCA(4),XREF(30),YREF(30),INX(10),NEV(10)
      LOGICAL WECPEG,CANCEL
      LOGICAL SYMFLG
      LOGICAL SETPLT
      REAL N
      DATA PNOIS/6HNE VAL,6HWECPNL,6HN E F /
      READ(5,910) VAL1,VAL2,VAL3
  910 FORMAT(3F10.1)
C TITLE OF PLOT ON ITS SIDE
      CALL SYMBOL(0.,10.,.5,TITLE,-90.,21)
      CALL PLOT(6.,0.,-3)
      INCP=1
      IF(WECPEG) INCP=2
      IF(NEFFLG) INCP=3
      CONS1=.3048006096
      IF(.NOT. MKSFLG) GO TO 1030
      CX=CX*CONS1
      CY=CY*CONS1
      X1=X1*CONS1
      Y1=Y1*CONS1
 1030 CONTINUE
      IF(.NOT. MKSFLG) GO TO 1020
      DO 1010 J=1,NRW
      DO 1010 I=1,3
      XLM1(I,J)=XLM1(I,J)*CONS1
      XLM2(I,J)=XLM2(I,J)*CONS1
 1010 CONTINUE
 1020 CONTINUE
      IF(NFLGHT .GT. 10) GO TO 211
      DO 1000 J=1,20
      DO 1000 I=1,22
 1000 FLIGHT(I,J)=0.
      IIXT=0
      KI=0
      DO 500 J=1,NRW
      L=NET(J)                                                           35200
      DO 500 I=1,L
      KI=KI+1
```

```
      FLIGHT(1,KI)=TP(I,J)
      FLIGHT(2,KI)=W(I,J)
      FLIGHT(3,KI)=PT(I,J)
      FLIGHT(4,KI)=TY(I,J)
      FLIGHT(5,KI)=N(1,I,J)
      NXT=NSG(KI)-2
      FLIGHT(6,KI)=NXT
      IF(NXT.EQ.0)GO TO 500
      INDEX=IIXT
      IIXT=IIXT+NXT
      DO 550 K=1,NXT
      IAD=(K-1)*4
      KX=INDEX+K
      FLIGHT(IAD+7,KI)=ALT(KX)                                    35550
      FLIGHT(IAD+8,KI)=CANGLE(KX)*57.2958
      FLIGHT(IAD+9,KI)=THRUST(KX)
      FLIGHT(IAD+10,KI)=CURV(KX)
  550 CONTINUE
  500 CONTINUE
      IREF=0
      IF(SYMFLG)IREF=1
C
C        DRAW THE DATA FOR THE PLOT.
C
      NUM=0
      DO 210 I=1,NRW
  210 NUM=NUM+NET(I)
      CALL HEAD(NUM)
  211 CONTINUE
      AINCH=AINCHX
      YINCH=AINCHY
      XD=DELX
      YD=DELY
      IF(SETPLT)GO TO 600
C
C
C        GET ADJUSTED SCALE VALUES FOR VERTICAL AXIS
C                   SCA(3)=ADJUSTED MINIMUM
C                   SCA(4)=INCREMENT ALONG AXIS
C
      XNX=NX-1
      XNY=NY-1
      T1=X1+XNX*CX
      T2=Y1+XNY*CY
      AYMIN=Y1
      AYMAX=T2
      IF(T2 .GT. Y1) GO TO 20
      AYMIN=T2
      AYMAX=Y1
   20 CONTINUE
      YULHC=AYMAX
      SCA(1)=AYMAX
      SCA(2)=AYMIN
      CALL SCALE(SCA,10.,2,1)
      YM=SCA(3)
```

```
      YD=SCA(4)
      AXMIN=X1
      AXMAX=T1
      IF(T1 .GT. X1) GO TO 10
      AXMIN=T1
      AXMAX=X1
   10 CONTINUE
      XULHC=AXMIN
      SCA(1)=AXMAX
      SCA(2)=AXMIN
      CALL SCALE(SCA,10.,2,1)
C        COMPUTE THE NUMBER OF INCHES FOR X-AXIS USING
C        THE Y-AXIS INCREMENT.IF LESS THAN 20,USE THE
C        Y-INCREMENT ON THE X-AXIS(AND Y-AXIS)
C
      SCMX=SCA(3)+10.*SCA(4)
      NINCH=(SCMX-SCA(3))/YD+.9
      IF(NINCH.LE.20)GO TO 40
C
C        IF THE NUMBER OF INCHES IS GREATER THAN 20,
C        SCALE THE X-AXIS TO 20 INCHES AND USE THIS
C        INCREMENT FOR THE Y-AXIS.
C
      CALL SCALE(SCA,20.,2,1)
      NINCH=20
      YD=SCA(4)
   40 IF(NINCH.LT.10)NINCH=10
      AINCH=NINCH
C
C        COMPUTE THE ZERO POINT ON THE Y-AXIS,IF ANY,
C        AND START THE X-AXIS THERE.
C
      YHT=-YM/YD
      IF(YHT.GT.10..OR.YHT.LT.0.)YHT=0.
C
C        DRAW THE AXES.
C                  Y-AXIS = 10 INCHES LONG
C                  X-AXIS = BETWEEN 10 AND 20 INCHES LONG
C
      IF(.NOT.MKSFLG)CALL AXIS(-.5,0.,35HLATERAL DISTANCE(THOUSANDS OF F
     XEET) ,35,10.,90.,YM/1000.,YD/1000.)
      IF(MKSFLG)CALL AXIS(-.5,0.,37HLATERAL DISTANCE(THOUSANDS OF METERS
     X) ,37,10.,90.,YM/1000.,YD/1000.)
      IF(.NOT.MKSFLG)CALL AXIS(0.,YHT,27HDISTANCE(THOUSANDS OF FEET) ,
     X -27,AINCH,0.,SCA(3)/1000.,YD/1000.)
      IF(MKSFLG)CALL AXIS(0.,YHT,29HDISTANCE(THOUSANDS OF METERS) ,
     X -29,AINCH,0.,SCA(3)/1000.,YD/1000.)
C
C
      XM=SCA(3)
      XD=YD
      XSF=XD
      XSF=1./XSF
  600 IF(.NOT.SETPLT) GO TO 650
      XNX=NX-1
```

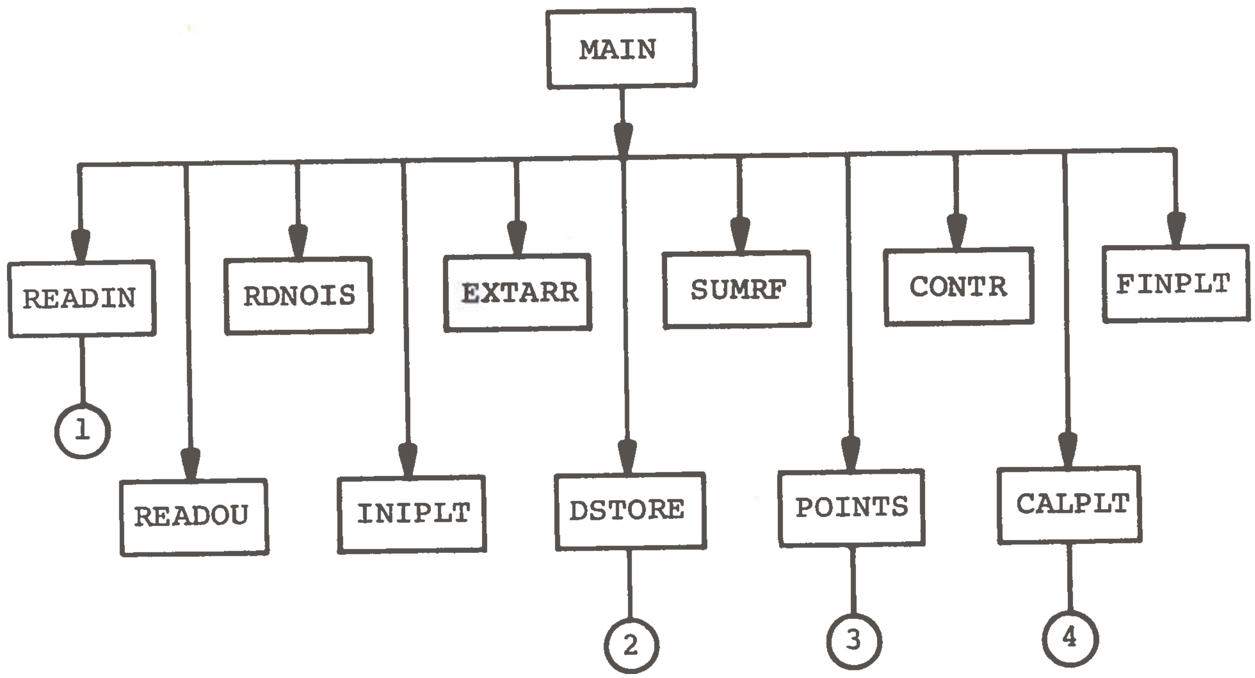
```
      XNY=NY-1
      T1=X1+XNX*CX
      T2=Y1+XNY*CY
      XULHC=X1
      IF(T1 .LT. X1) XULHC=T1
      YULHC=Y1
      IF(T2 .GT. Y1) YULHC=T2
      XM=AXMIN
      IF(.NOT. MKSFLG)CALL AXIS2(-.5,0.,AYMAX,AYMIN,YD,YINCH,
     X 35HLATERAL DISTANCE(THOUSANDS OF FEET) ,35,-1,DELNY)
      IF(MKSFLG)CALL AXIS2(-.5,0.,AYMAX,AYMIN,YD,YINCH,
     X 37HLATERAL DISTANCE(THOUSANDS OF METERS) ,37,-1,DELNY)
      YHT=-AYMIN/DELNY
      IF(YHT.GT.YINCH .OR. YHT .LT. 0.) YHT=0.
      IF(.NOT.MKSFLG)CALL AXIS2(0.,YHT,AXMAX,AXMIN,XD,-AINCH,
     X 27HDISTANCE(THOUSANDS OF FEET) ,27,-1,DELNX)
      IF(MKSFLG)CALL AXIS2(0.,YHT,AXMAX,AXMIN,XD,-AINCH,
     X 29HDISTANCE(THOUSANDS OF METERS) ,29,-1,DELNX)
      YD=DELNY*1000.
      YM=AYMIN*1000.
      XM=AXMIN*1000.
      XD=DELNX*1000.
      XSF=1./XD
  650 CONTINUE
      DO 111 I=1,NRW
      X1=(XLM1(1,I)-XM)/XD
      Y1=(XLM1(2,I)-YM)/YD
      X2=(XLM2(1,I)-XM)/XD
      Y2=(XLM2(2,I)-YM)/YD
  111 CALL RUNWAY(X1,Y1,X2,Y2)
      CALL RWYLEG(AINCH)
      TEMP1=AINCH-4.5
      XUB=XM+AINCH*XD
      XZIN=-AINCH*XM/(XUB-XM)
      CALL NOMEN(XM,YM,XD,YD)
      WRITE(6,8010)
      WRITE(6,8020) XM,XUB,XSF,YM,YD,YHT,XZIN,AINCH,SCMX,T1,T2,XULHC,
     1  YULHC
      WRITE(6,8030) NINCH
 8000 CONTINUE
      IF(VAL1 .EQ. 0.) GO TO 900
      IF(MKSFLG) GO TO 8075
      CALL SYMBOL(11.3,9.82,.08,7HAREA IN,0.,7)
      CALL SYMBOL(10.8,9.70,.08,20HTHOUSANDS OF SQ. FT.,0.,20)
      GO TO 8076
 8075 CALL SYMBOL(11.,9.7,.08,15HAREA IN SQ. M. ,0.,15)
 8076 CONTINUE
      CALL SYMBOL(9.2,9.5,     .15,PNOIS(INCP),0.,6)
      CALL NUMBER(10.3,9.5,.15,VAL1,0.,-1)
      CALL AUTCNT(VAL1)
      IF(.NOT.MKSFLG) AREA=AREA/1000.
      CALL NUMBER(11.,9.5,.15,AREA,0.,-1)
      CALL SYMBOL(TEMP1,.25,.2,TITLE,0.,23)
      CALL SYMBOL(TEMP1,0.,.1,43HNUMBER OF FLIGHTS      NUMBER OF OPERAT
     1IONS,0.,43)
```

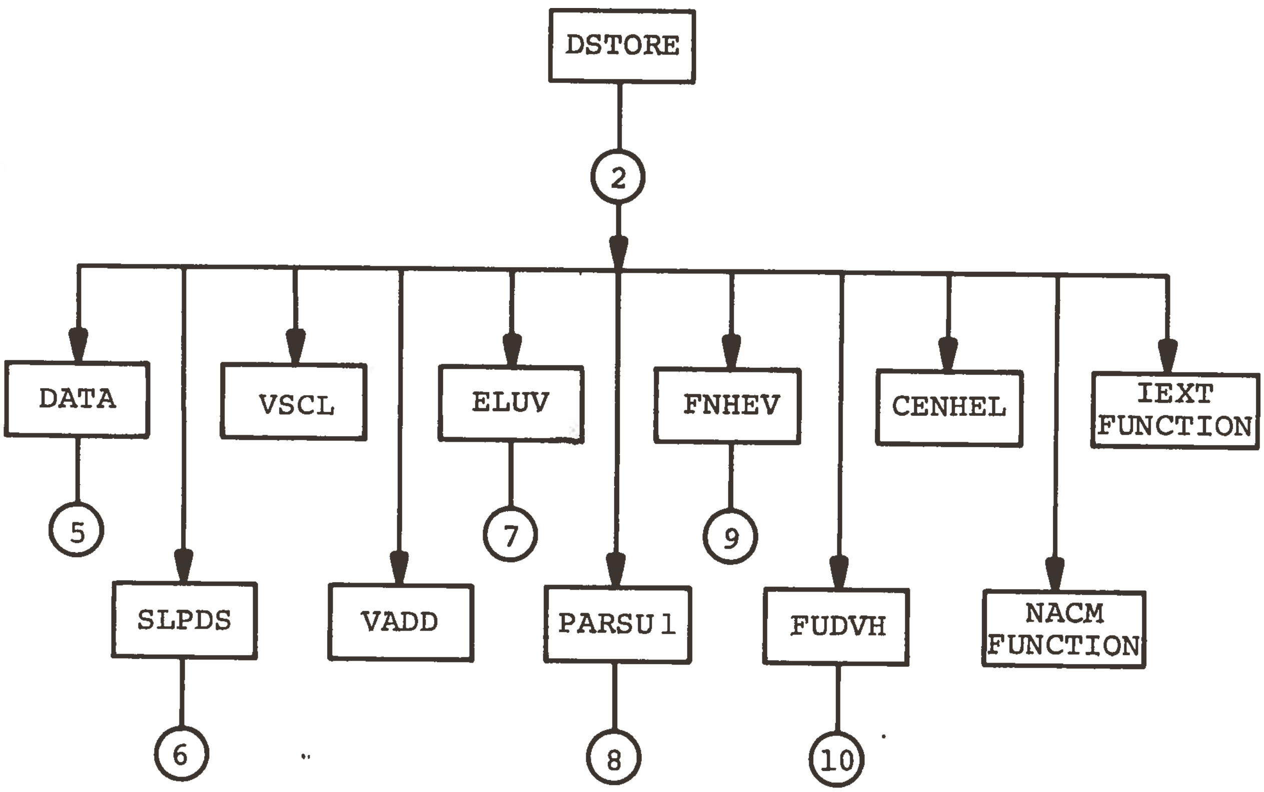
```
      TEMP2=TEMP1+1.85
      XNFLG=NFLGHT
      CALL NUMBER(TEMP2,0.,.1,XNFLG,0.,-1)
      TOTOP=0.
      DO 8200 I=1,3
      DO 8200 J=1,20
      DO 8200 K=1,10
 8200 TOTOP=TOTOP+N(I,J,K)
      TEMP2=TEMP1+4.45
      CALL NUMBER(TEMP2,0.,.1,TOTOP,0.,-1)
      IF(VAL2 .EQ. 0.) GO TO 900
      CALL SYMBOL(+36.,1.,.2,45HCALCOMP OPERATOR--PLEASE CHANGE PEN COLO
     1R NOW,45.,45)
      CALL SYMBOL(+36.,.7,.2,45H****************************************
     1*****,45.,45)
      CALL WAIT
      CALL SYMBOL(36.,8.,1.,9HTHANK YOU,-45.,9)
      CALL SYMBOL(9.2,9.2,.15,PNOIS(INCP),0.,6)
      CALL NUMBER(10.3,9.2,.15,VAL2,0.,-1)
      CALL AUTCNT(VAL2)
      IF(.NOT.MKSFLG) AREA=AREA/1000.
      CALL NUMBER(11.,9.2,.15,AREA,0.,-1)
      IF(VAL3 .EQ. 0.) GO TO 900
      CALL SYMBOL(+38.,1.,.2,28HMAY I IMPOSE UPON YOU AGAIN ,45.,28)
      CALL WAIT
      CALL SYMBOL(+38.,.7,.2,21H*****MANY THANKS*****,45.,21)
      CALL SYMBOL(9.2,8.9,.15,PNOIS(INCP),0.,6)
      CALL NUMBER(10.3,8.9,.15,VAL3,0.,-1)
      CALL AUTCNT(VAL3)
      IF(.NOT.MKSFLG) AREA=AREA/1000.
      CALL NUMBER(11.,8.9,.15,AREA,0.,-1)
  900 CONTINUE
C        END THE PLOT.
C
      CALL PLOT(ABS(AINCH)+12.,0.,-3)
      CALL SYMBOL(0.,0.,.5,TITLE,90.,21)
      CALL PLOT(6.,0.,-3)
      RETURN
 8010 FORMAT(12H1FROM CALPLT)
 8020 FORMAT(55H XM,XUB,XSF,YM,YD,YHT,XZIN,AINCH,SCMX,T1,T2,XULHC,YULHC/
     1(8F15.5))
 8030 FORMAT(6H NINCH/I10)
      END
      SUBROUTINE HEAD(NUM)
      COMMON/JOE2/NEFFLG,NOISFG
      COMMON/TROUBL/FLIGHT(22,20)
      COMMON/JOE/REFLIN,XCOORD(2),YCOORD(2),TITLE(12),SYMFLG
      COMMON/JOE1/WECPEG,CANCEL
      COMMON/LOGEG1/MKSFLG,CALFLG,SSIFLG,LSIFLG
      COMMON/NN/N(3,20,10),NRW,NFT(10)
      REAL*8 FTKG(2),LBMF(2),FTMD(2)
      REAL*8 PLANE(3,13)
      REAL N
      LOGICAL NEFFLG,NOISFG
      LOGICAL SYMFLG
```

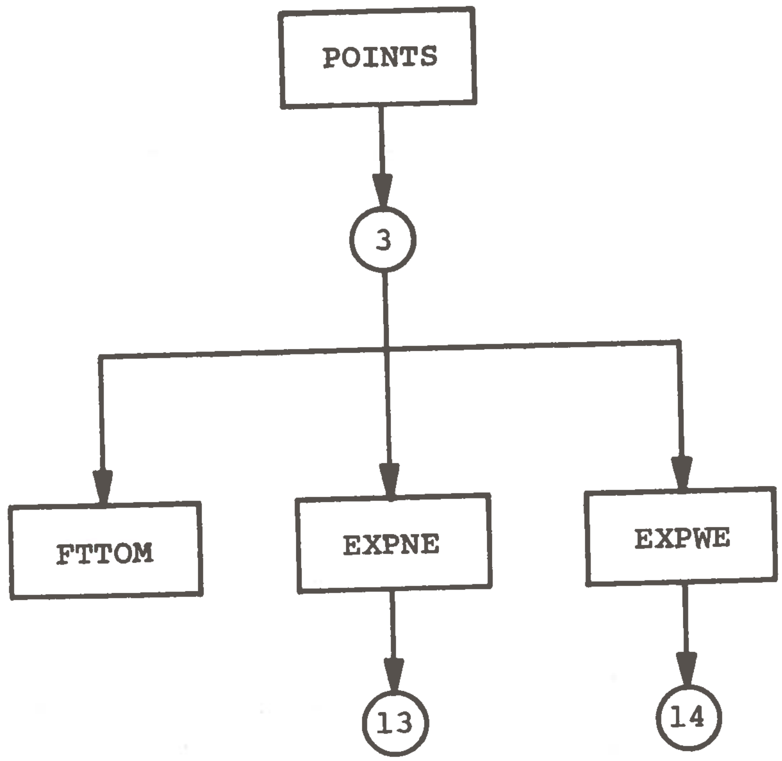
```
      LOGICAL LL
      LOGICAL CANCFL,WECPFG
      LOGICAL MKSFLG,CALFLG,SSIFLG,LSIFLG
      DATA FTKG,LBMF,FTMD/4H(FT),4H(M) ,4H(LB),4H(KG),6H FT/DG,6H M/DG /
      DATA PLANE/
     18H4 ENG. H,8HBPR TF   ,8H        ,8H3 ENG. H,8HBPR TF  ,8H        ,
     28H4 ENG. L,8HBPR TF   ,8H        ,8H4 ENG. T,8HURBOJET ,8H        ,
     38H3 ENG. L,8HBPR TF   ,8H        ,8H2 ENG. L,8HBPR TF  ,8H        ,
     48H2 ENG. L,8HBPR FANS,8H         ,8H4 ENG. F,8HX. JETS ,8H        ,
     38H2 ENG. L,8HBPR FANS,8H         ,8H2 ENG. F,8HX. JETS ,8H        ,
     68H2 ENG. P,8HROP.     ,8H        ,8H4 ENG. P,8HROP.    ,8H        ,
     78HSST     ,8H        ,8H         /
      FLEN=2.5
      X=0.
      AMAX=FLIGHT(6,1)
      IF(NUM.EQ.1) GO TO 5
      DO 6 I=2,NUM
      IF(FLIGHT(6,I).GT.AMAX) AMAX=FLIGHT(6,I)
    6 CONTINUE
    5 XCTR=(3.5+FLOAT(NUM)*FLEN)/2.
      AMX=AMAX
      IF(AMAX .GE. 2.) AMX=2.
      YTOP=(AMX*2.+3.2+2.1)
      YTOP=YTOP+(10.-YTOP)/2.
      INDEX=1
      IF(.NOT.MKSFLG)GO TO 41
      INDEX=2
      DO 40 I=1,NUM
      CALL LBTOKG(FLIGHT(2,I),1)
      IND=IFIX(FLIGHT(6,I))
      IF(IND.EQ.0)GO TO 40
      DO 400 K=1,IND
      JI=4*(K-1)
      CALL FTTOM(FLIGHT(JI+10,I),1)
  400 CONTINUE
   40 CONTINUE
   41 DO 42 I=1,NUM
      IND=FLIGHT(6,I)
      IF(IND.EQ.0)GO TO 42
      DO 420 K=1,IND
      JI=4*(K-1)
      LL=FLIGHT(JI+10,I).EQ.0.
      IF(LL.AND.MKSFLG)CALL FTTOM(FLIGHT(JI+7,I),1)
      IF(.NOT.LL)FLIGHT(JI+7,I)=FLIGHT(JI+7,I)*57.2958
  420 CONTINUE
   42 CONTINUE
      IF(WECPFG)CALL SYMBOL(XCTR-1.3,YTOP-.2,.2,13HWECPNL VALUES ,0.,13)
      IF(.NOT.WECPFG)CALL SYMBOL(XCTR-2.1,YTOP-.2,.2,21HNOISE EXPOSURE V
     XALUES,0.,21)
      CALL SYMBOL(XCTR-2.3,YTOP-.6,.2,TITLE,0.,23)
      CALL SYMBOL(XCTR-3.2,YTOP-1.1,.2,
     1 32HRUNWAYS AND GRID POINTS AS SHOWN,0.,32)
      CALL SYMBOL(XCTR-1.7,YTOP-2.3,.2,17HFLIGHT STATISTICS,0.,17)
      YTOP=YTOP-2.3
      DELY=.2
```

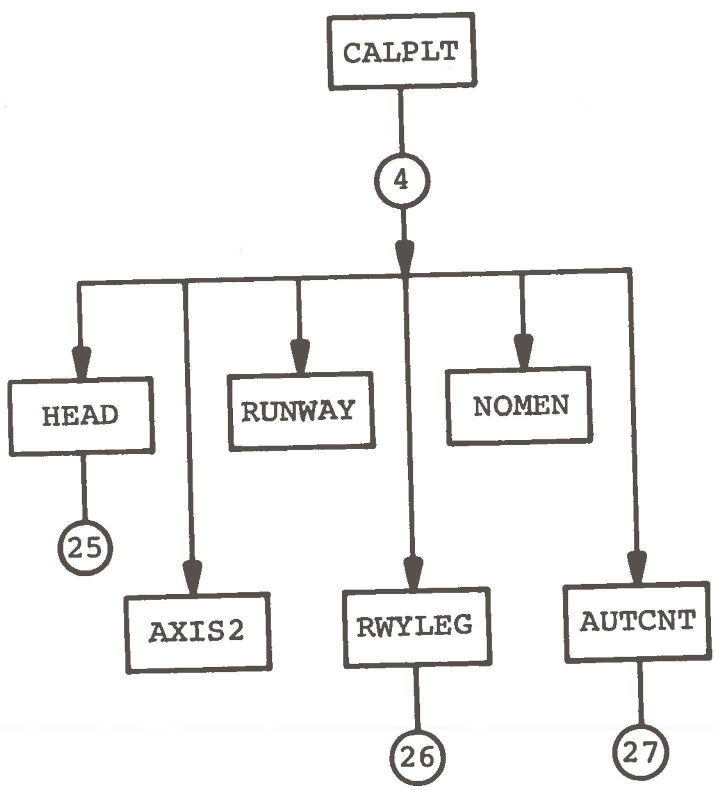
APPENDIX C

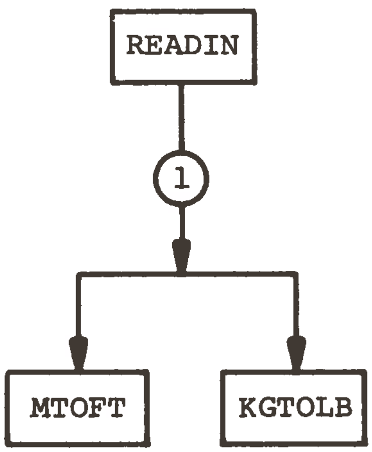
SUBROUTINE CALL CHARTS

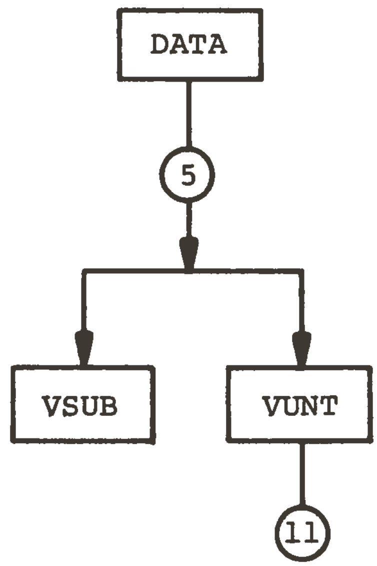

MAIN
READIN
RDNOIS
EXTARR
SUMRF
CONTR
FINPLT
1
READOU
INIPLT
DSTORE
POINTS
CALPLT
2
3
4

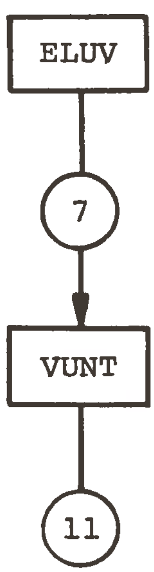

DSTORE
2
DATA
VSCL
ELUV
FNHEV
CENHEL
IEXT FUNCTION
5
SLPDS
VADD
7
PARSU1
9
FUDVH
NACM FUNCTION
6
8
10

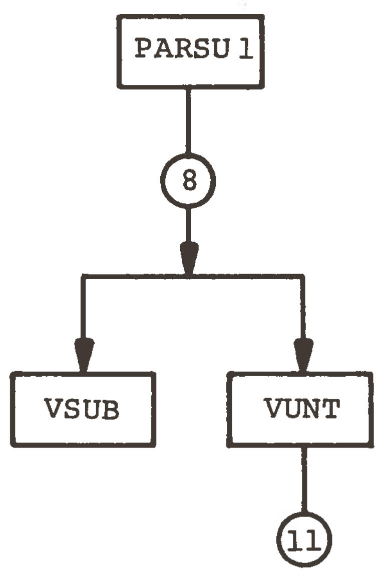

POINTS
3
FTTOM
EXPNE
EXPWE
13
14

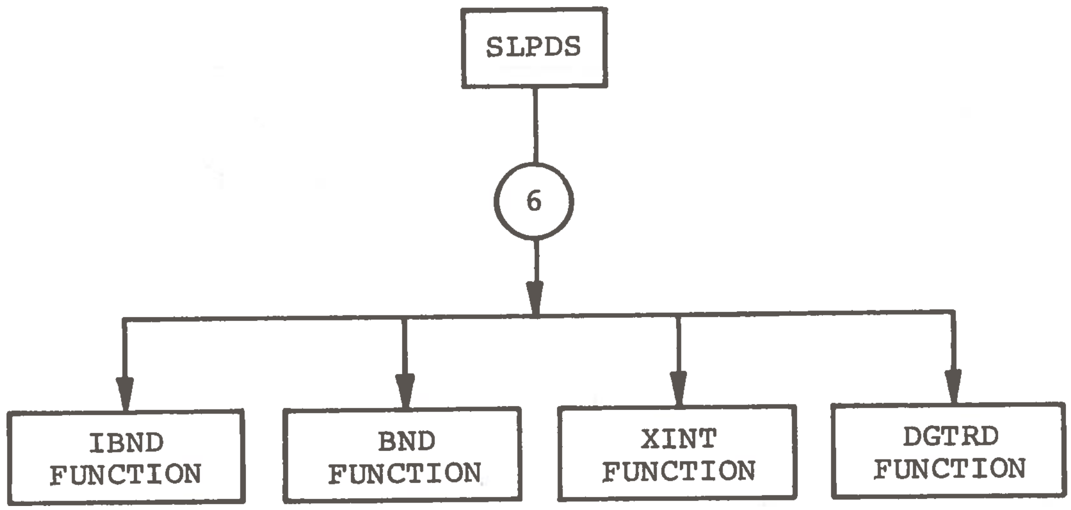

CALPLT
4
HEAD
RUNWAY
NOMEN
25
AXIS2
RWYLEG
AUTCNT
26
27

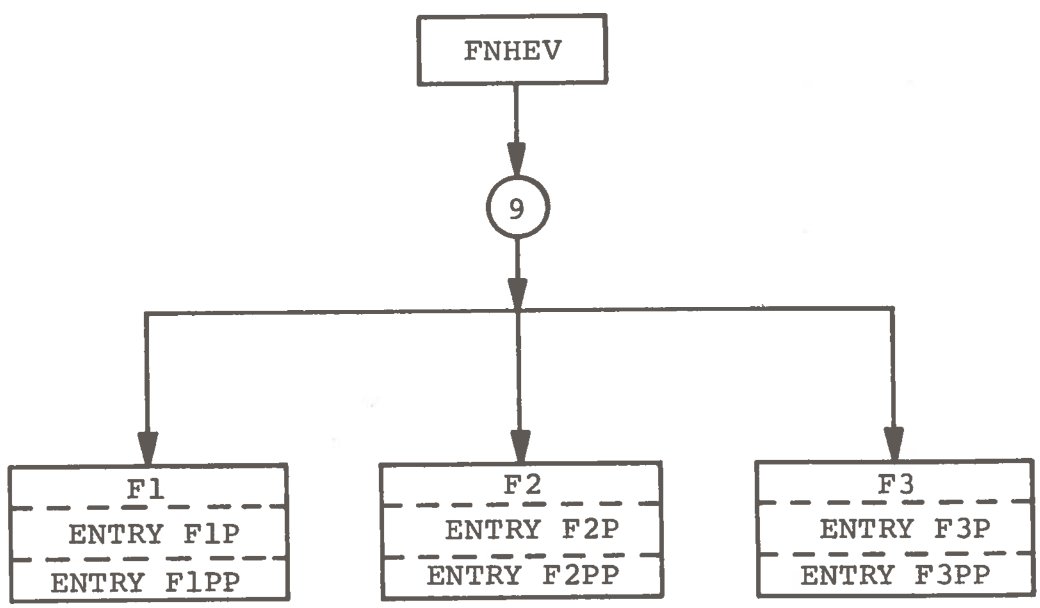

READIN
1
MTOFT
KGTOLB

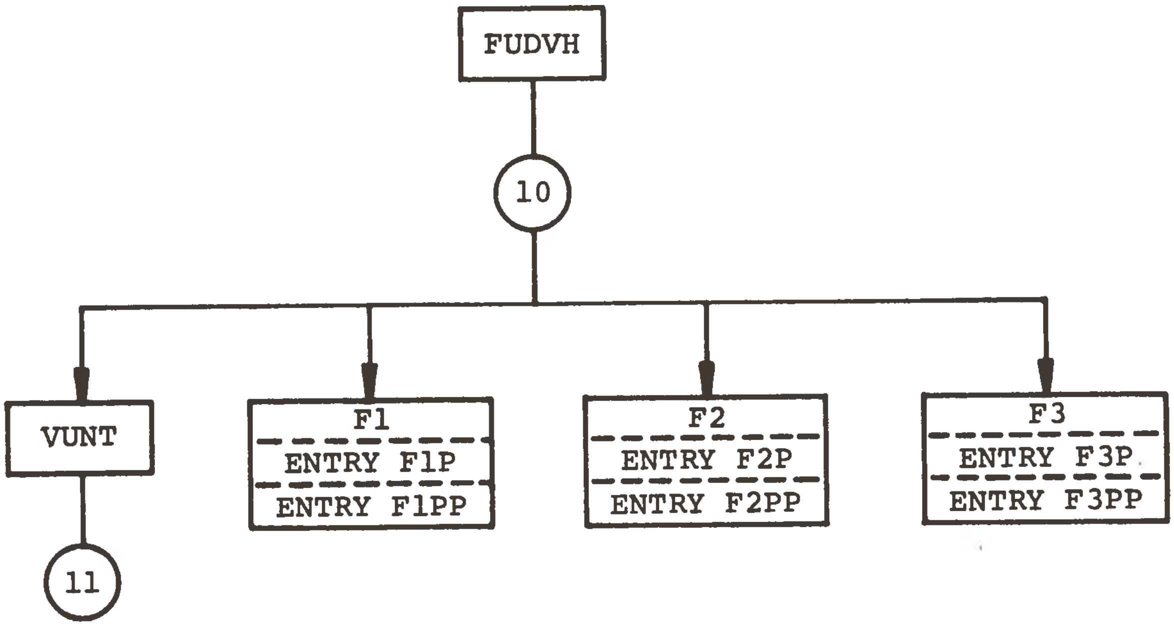

DATA
5
VSUB
VUNT
11

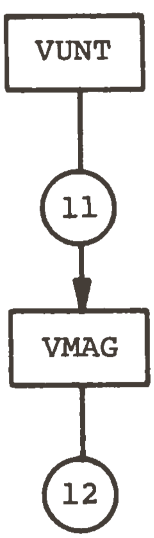

ELUV
7
VUNT
11

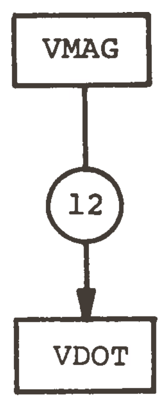

PARSU1
8
VSUB
VUNT
11

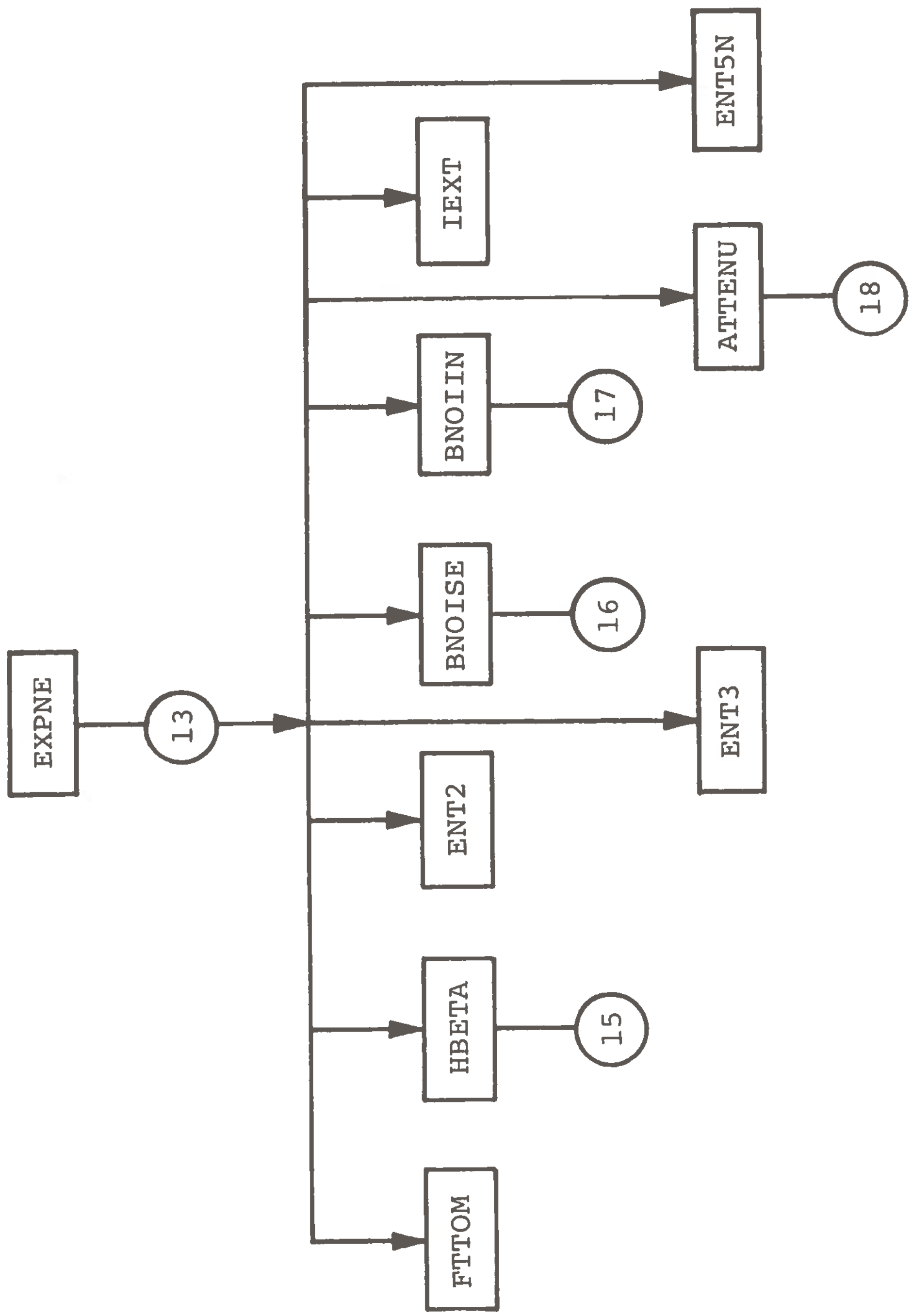

SLPDS
6
IBND FUNCTION
BND FUNCTION
XINT FUNCTION
DGTRD FUNCTION

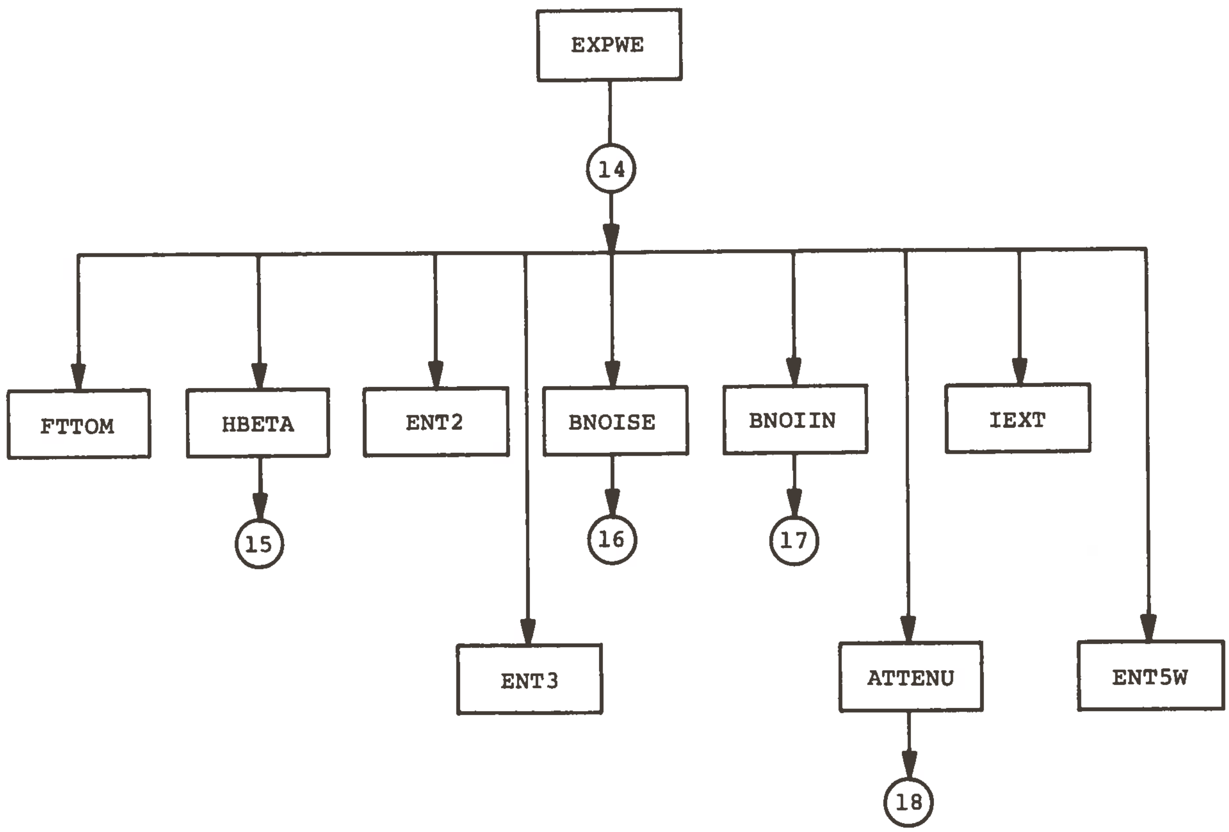

FNHEV
9
F1
ENTRY F1P
ENTRY F1PP
F2
ENTRY F2P
ENTRY F2PP
F3
ENTRY F3P
ENTRY F3PP

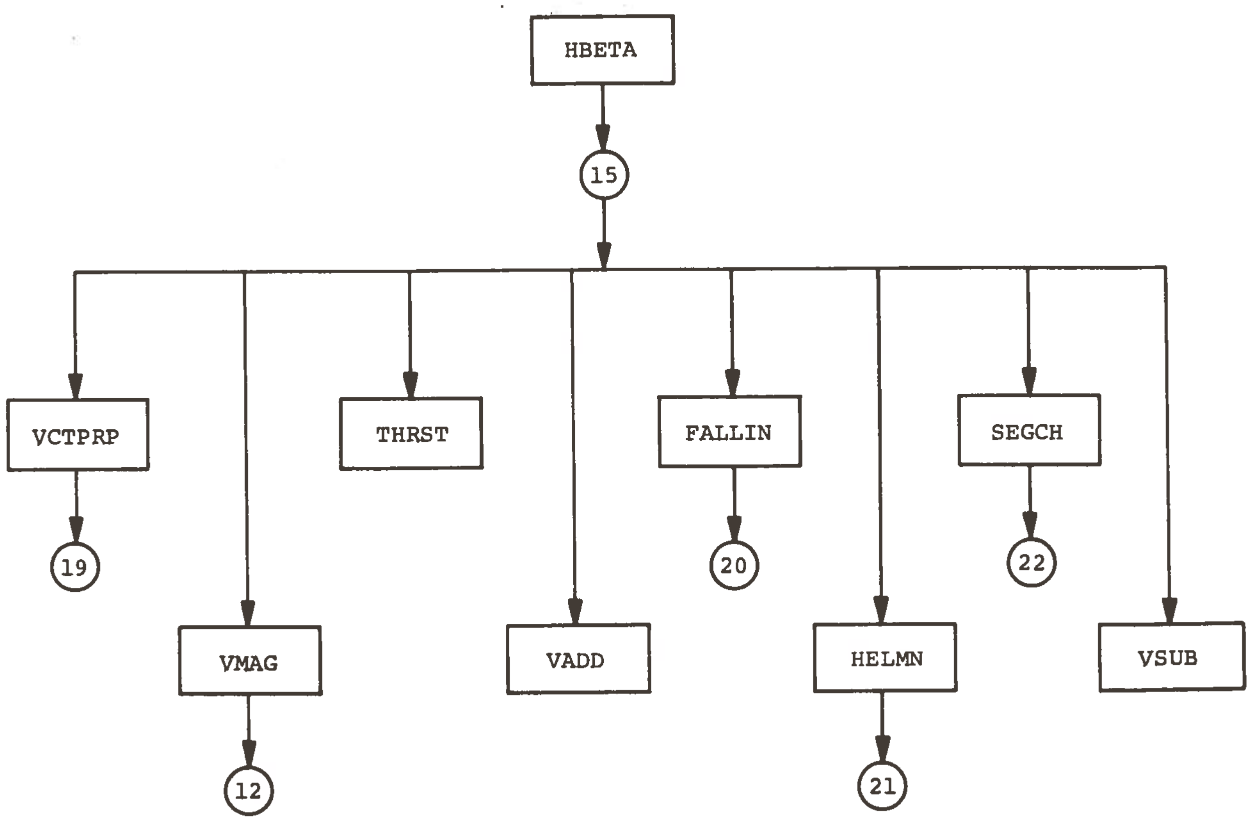

FUDVH
10
VUNT
11
F1
ENTRY F1P
ENTRY F1PP
F2
ENTRY F2P
ENTRY F2PP
F3
ENTRY F3P
ENTRY F3PP

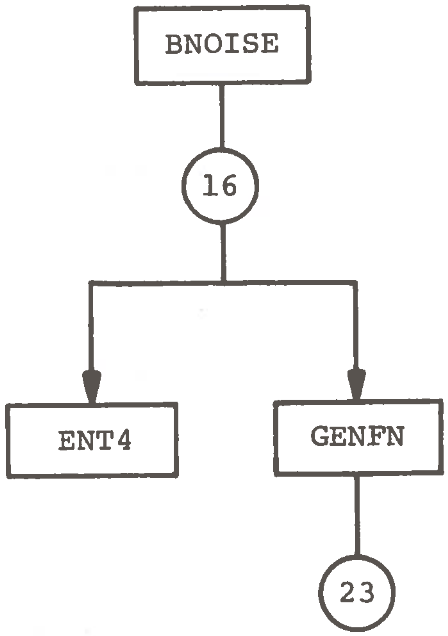

VUNT
11
VMAG
12

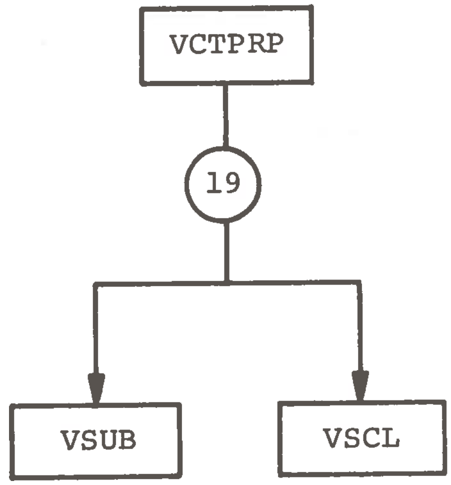

VMAG
12
VDOT

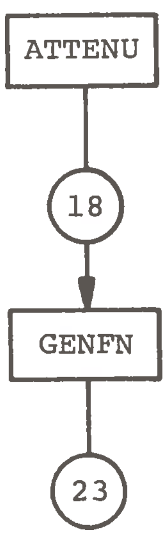

EXPNE
13
FTTOM
HBETA
15
ENT2
ENT3
BNOISE
16
BNOIIN
17
ATTENU
18
IEXT
ENT5N

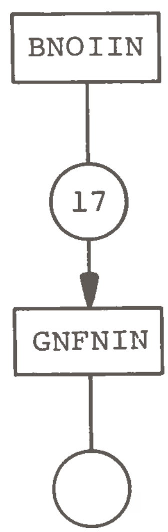

EXPWE
14
FTTOM
HBETA
ENT2
BNOISE
BNOIIN
IEXT
15
16
17
ENT3
ATTENU
ENT5W
18

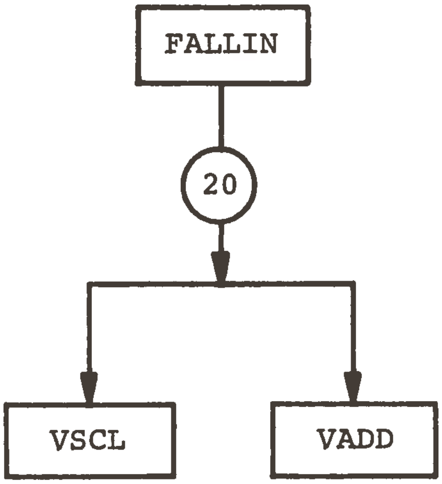

HBETA
15
VCTPRP
THRST
FALLIN
SEGCH
19
20
22
VMAG
VADD
HELMN
VSUB
12
21

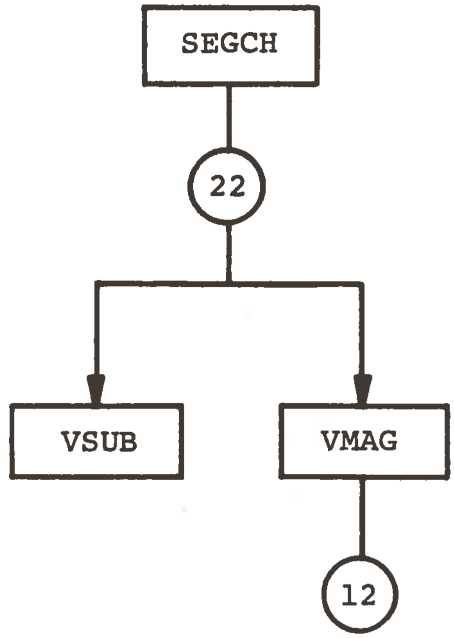

BNOISE
16
ENT4
GENFN
23

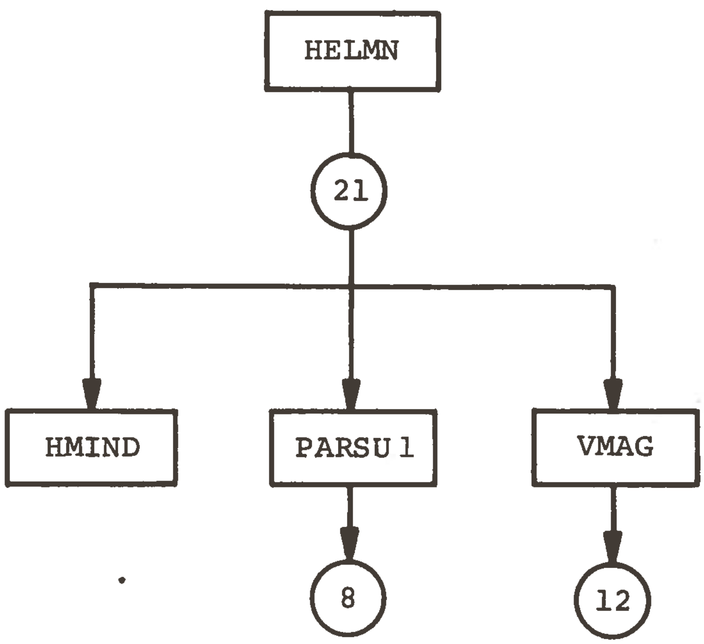

VCTPRP
19
VSUB
VSCL

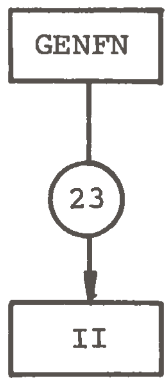

ATTENU
18
GENFN
23

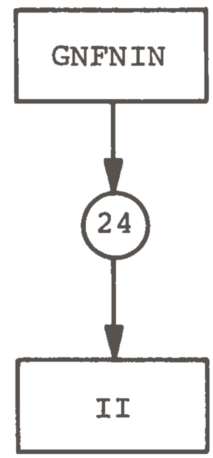

BNOIIN
17
GNFNIN

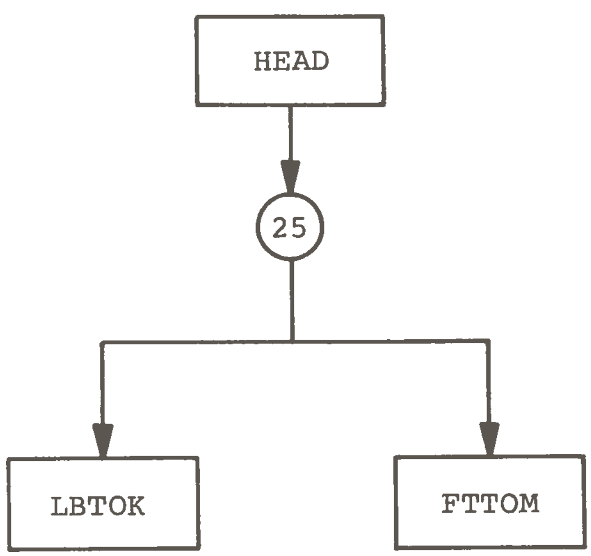

FALLIN
20
VSCL
VADD

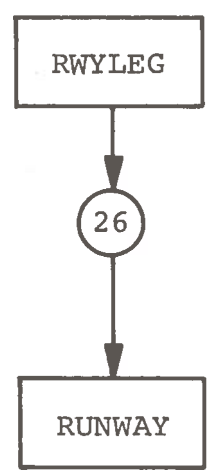

SEGCH
22
VSUB
VMAG
12

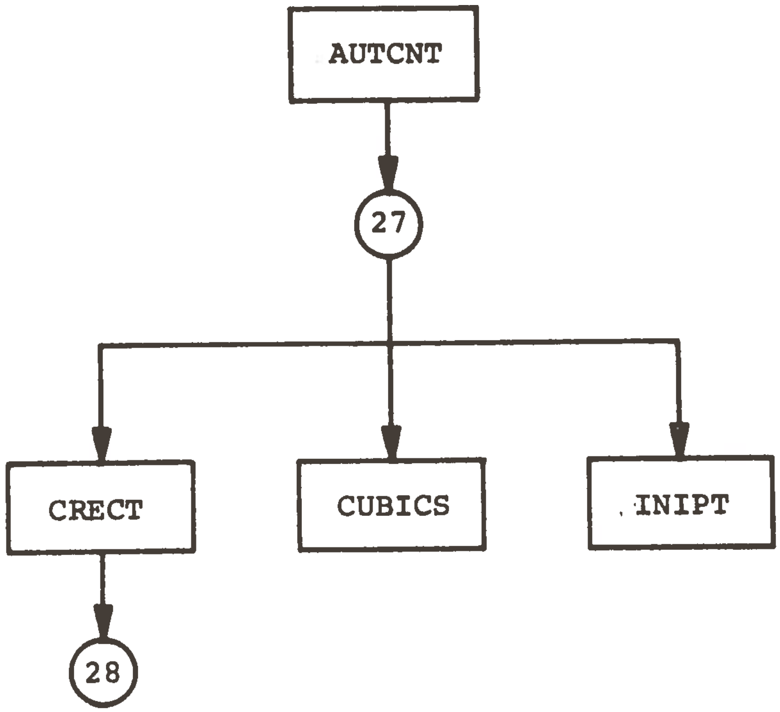

HELMN
21
HMIND
PARSU1
VMAG
8
12

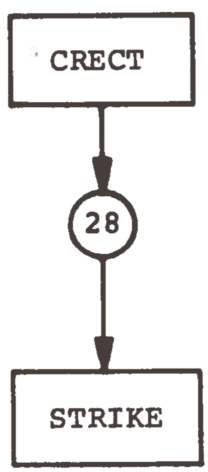

GENFN
23
II


GNFNIN
24
II


HEAD
25
LBTOK
FTTOM


RWYLEG
26
RUNWAY


AUTCNT
27
CRECT
CUBICS
INIPT
28


CRECT
28
STRIKE

APPENDIX D

SUBROUTINE DESCRIPTIONS

NEM-5 SIMULATION COMPUTER PROGRAM ROUTINES

MAIN

SUBROUTINE READIN

SUBROUTINE RDNOIS

SUBROUTINE INIPLT (Entry: FINPLT)

SUBROUTINE EXTARR

SUBROUTINE DSTORE

SUBROUTINE SUMRF

SUBROUTINE POINTS

SUBROUTINE CONTR

SUBROUTINE BNOIIN

SUBROUTINE CALPLT

SUBROUTINE DATA

SUBROUTINE EXPNE

SUBROUTINE EXPWE

SUBROUTINE BNOISE

SUBROUTINE SLPDS

SUBROUTINE ELUV

SUBROUTINE PARSU1

SUBROUTINE FNHEV

SUBROUTINE FUDVH

SUBROUTINE CENHEL

SUBROUTINE GENFN

SUBROUTINE GNFNIN

SUBROUTINE ATTENU

SUBROUTINE HBETA

SUBROUTINE VCTPRP

SUBROUTINE FALLIN

SUBROUTINE HELMN

SUBROUTINE SEGCH

SUBROUTINE THRST

SUBROUTINE HMIND

BLOCK DATA

SUBROUTINE FTTOM

SUBROUTINE MTOFT

SUBROUTINE LBTOKG

SUBROUTINE KGTOLB

FUNCTION DGTRD

SUBROUTINE VADD

SUBROUTINE VSUB

SUBROUTINE VCROS

SUBROUTINE VSCL

FUNCTION VDOT

SUBROUTINE VMAG

SUBROUTINE VUNT

FUNCTION II

FUNCTION XINT

FUNCTION NACM

FUNCTION F1 (Entries: F1P,F1PP)

FUNCTION F2 (Entries: F2P,F2PP)

FUNCTION F3 (Entries: F3P,F3PP)

SUBROUTINE ENT2

SUBROUTINE ENT3

SUBROUTINE ENT4

SUBROUTINE ENT5N

SUBROUTINE ENT5W

FUNCTION IEXT

LOGICAL FUNCTION BND

LOGICAL FUNCTION IBND

SUBROUTINE PLOTA

SUBROUTINE RWYLEG

SUBROUTINE RUNWAY

SUBROUTINE HEAD

SUBROUTINE AXIS2

PROGRAM MAIN

Purpose

MAIN controls the overall flow of the program. It calls subroutine READIN which reads in a data set and then, according to whether the quantity INDEX is 1, 2 or 3, it (1) calls the processing routines, (2) plots the results of processing and calls READIN for a new data set, or (3) plots the result of processing and ends the run. The plots are made only if requested. MAIN also calls subroutine RDNOISE if the user is inputting his own aircraft noise tables.

Subroutines Called

SUBROUTINE READIN (INDEX)

SUBROUTINE READOU (Dummy Subroutine)

SUBROUTINE RDNOIS

SUBROUTINE INIPLT

SUBROUTINE EXTARR

SUBROUTINE DSTORE

SUBROUTINE SUMRF

SUBROUTINE POINTS

SUBROUTINE CONTR (Dummy Subroutine)

SUBROUTINE CALPLT

SUBROUTINE FINPLT

Input

INDEX =	1	PROCES card encountered in subroutine READIN. Proceed with calculations.
	2	NEWSET card encountered in subroutine READIN. Plot results and start new input.
	3	ENDRUN card encountered in subroutine READIN. Plot results and end this run.
PTINFG		If true, call subroutine READOU and print input. From subroutine READIN.
NOISFG		If true, call subroutine RDNOIS and read in noise tables. From subroutine READIN.
CALFLG		If true, call subroutine CALPLT and plot the results. From subroutine READIN.
GRIDFG		If true, call subroutine POINTS and set up grid coordinate loops. From subroutine READIN.
CONTFG		If true, call subroutine CONTR. From subroutine READIN.
ITOM	= 0	INIPLT not yet called.
	= 1	INIPLT called already and FINPLT may be called at end of run.

SUBROUTINE READIN (INDEX)

Purpose

Subroutine READIN reads in all data except the aircraft noise tables. As it reads the data deck, subroutine READIN puts the data into appropriate arrays and sets the logical flags to be used by the MOD-5 program to control options such as plots, units, input noise tables, non-standard output and choice of noise exposure index. Subroutine READIN also sets the variable INDEX which is used by MAIN as a control variable.

Subroutines Called

SUBROUTINE MTOFT (A,I)

SUBROUTINE KGTOLB (W,200)

Output (for each Airport case)

NRW	The number of runways read in for this airport. Left in common NN.
NFLIGHT	The cumulative number (over all runways for the airport) of flight READIN. Left in common XDATA.
NEXT	The cumulative number (over all flights for the airport) of extensions to read in.
DELX	The X increment to be used in plotting. Left in common FACTR.
DELY	The Y increment to be used in plotting. Left in common FACTR.
AINCHX	The length of the x axis to be used in plotting. Left in common FACTR.
AINCHY	The length of the y axis to be used in plotting. Left in common FACTR.
AXMAX AXMIN	The maximum and minimum value of x to be plotted. Left in common FACTR.
AYMAX AYMIN	The maximum and minimum value of Y to be plotted. Left in common FACTR.

XLM1 (I,NUMRUN)	I = 1,2,3; NUMRUN = 1,...NRW. The position vector of the starting point of each runway. Left in common AA.
XLM2 (I,NUMRUN)	I = 1,2,3; NUMRUN = 1,...,NRW. The position vector of the terminal point of each runway. Left in common AA.
SL (NUMRUN)	NUMRUN = 1,...,NRW. The touchdown distance for each runway. Left in common AA.
NFT (NUMRUN)	NUMRUN = 1,...,NRW. The number of flights for each runway. Left in common NN.
TP (NUMFLI,NUMRUN)	NUMFLI = 1,...,NFT (NUMRUN); NUMRUN = 1,...,NRW. The takeoff/landing code for each flight of each runway. Left in common AA.
W (NUMFLI,NUMRUN)	NUMFLI = 1,...,NFT (NUMRUN); NUMRUN = 1,...,NRW. The gross weight of the aircraft for each flight of each runway. Left in common AA.
PT (NUMFLI,NUMRUN)	NUMFLI = 1,...,NFT (NUMRUN); NUMRUN = 1,...,NRW. The percent of thrust on segments one and two for each flight of each runway. Left in common AA.
PTHRI (I,NFL)	I = 1,...,NSG(NFL); NFL = 1,...,NFLIGHT. The percent thrust for each segment of each flight. Left in common XDATA.
TY (NUMFLI,NUMRUN)	NUMFLI = 1,...,NFT (NUMRUN); NUMRUN = 1,...,NRW. The aircraft type number for each flight on each runway. Left in common AA.
N (I,NUMFLI,NUMRUN)	I = 1,2,3; NUMFLI = 1,...,NFT (NUMRUN); NUMRUN = 1,...,NRW. The number of day and evening operations for each flight on each runway. Left in common NN.
NSG (NFL)	NFL = 1,...,HFLGHT. The number of segments (including extensions) for each flight. Left in common XDATA.

THETAI (NEXT,NFL)	NEXT = 1,...,NSG (NFL); NFL = 1,..., NFLGHT.
ZETI (NEXT,NFL)	NEXT = 1,...,NSG (NFL); NFL = 1,..., NFLGHT. The turn angle or length of ground track of each segment of each flight. Left in common XDATA.
RADI (NEXT,NFL)	NEXT = 1,...,NSG; (NFL) = 1,...,NFLGHT. The turn radius of each segment of each flight. Linear segments have a zero turning radius. Left in common XDATA.
CX	The x increment for the observer grid points. Left in common GDPAR.
CY	The y increment for the observer grid points. Left in common GDPAR.
X1	The x coordinate of the initial observer grid point. Left in common GDPAR.
Y1	The y coordinate of the initial observer grid point. Left in common GDPAR.
NX	The number of x elements in the observer grid. Left in common GDPAR.
NY	The number of y elements in the observer grid. Left in common GDPAR.
NEFFLG	Noise Exposure Forecast (NEF) values to be calculated; allows for day and night operations only; input NEFCAL; calls SYMBOL, common JOE2.
NOISFG	User supplies noise input; input NOISIN; calls BNOIIN; common JOE2.
WECPFG	Weighted Equivalent Perceived Noise Level (WECPNL) values to be calculated; allows for day, evening, night operations; input WECPNL; calls EXPWE and SYMBOL; common JOE1.
CANCEL	Causes error messages to be printed in READIN if an error is found in input data.

MKSFLG	Input is in meters and kilograms; input MKSSYS; calls MTOFT, KGROLB, FTTOM, SYMBOL; common LOGFG1.
CALFLG	CALCOMP plot output; calls CALPLT; common LOGFG1.
SSIFLG	Size of CALCOMP plot is 10; input CLCMPS; common LOGFG1.
LSIFLG	Size of CALCOMP plot is 30; input CLCMPL; common LOGFG1.
DGLOFG	No diagnostics are printed; input DIAGLO; common LOGFG2.
DGL1FG	Some diagnostics printed; input DIAGL1; prints noise tables in RDNOIS; prints some computations in EXPNE and EXPWE; prints segments positions and unit vectors in DSTORE; calls ENT3, ENT4, ENT5W, ENT2, ENT5N; common LOGFG2.
DGL2FG	More extensive diagnostics printed in EXPWE and EXPNE; common LOGFG2.
DGL3FG	Complete diagnostics printed in BNOIIN, FALLIN, CENHEL, HBETA, PARSUI, FNHEV, FUDVH, ELUV, EXTARR, HMIND, HELMN; input DIAGL3; common LOGFG2.
PTINFG	Prints input; calls READOU; common LOGFG2; input PRTINP.
PTTBFG	Prints calculated tables; common LOGFG2; input PRTTAB.
PTOUFG	Prints output; common LOGFG2; input PRTOUT.
PTGDFG	Prints grid; prints grid in POINTS; common LOGFG2.
PTOOFG	Prints contour; LOGFG2.
PTALEG	Prints all input, output, some calculated tables; input PRTALL.
RWAYFG	Runway information follows; input RUNWAY.

GRIDFG	Grid information follows; input GRIDCL; calls POINTS.
CONTFG	Contour levels follow; up to 3 permitted; only if CALFLG = TRUE; calls CONTR.
SETPLT	User specifies plot size; calls AXIS2.
CONVRT	Converts output back to meters.
HSEGFG	Helical segment flag.
STORE (IST)	IST = 1,...,6000. Store is an R = 8 array into which all information on each data card is inserted when that card is read.

Input

The data deck.

SUBROUTINE RDNOIS

Purpose

This subroutine reads tables of aircraft noise as a function of distance and thrust for each aircraft type.

The set-up of these tables is described in the User's Manual Section.

Subroutines Called

None.

Input

NOSETS	The number of aircraft typed for which noise tables will be read. From cards.
NACTP	The aircraft type number from which noise tables will be read. From cards.
NOTHRV	The number of thrust values for which a noise table will be read for aircraft type NACTP. From cards.
THR	The thrust value for the noise table to be read.

From cards.

TAKLD	= 0 Noise table is for takeoffs. From cards. = 1 Noise table is for landings. From cards.
TLILOG	= 0 The noise table distance scale in linear. From cards. = 1 The noise table distance scale in logarithmic. From cards.
NOXY	The number of entries in the noise table to be read. From cards.

Output

NCVTOT	The total number of noise tables to be read for all aircraft types and thrust values. Left in common INSIN.
NSEGIN(J)	J = 1,...,NCVTOT. The number of tabular values in the Jth noise table minus one. Left in common INSIN.
NATYTB(J)	J = 1,...,NCVTOT. Aircraft type number associated with Jth noise table. Left in common INSIN.
THRTB(J)	J = 1,...,NCVTOT. The thrust level associated with the Jth noise table. Left in common NSIN.
TKLDTB(J)	J = 1,...,NCVTOT. The takeoff/landing code associated with the Jth noise table. Left in common NSIN.
TYPIN(J)	J = 1,...,NCVTOT. The arithmetic/logarithmic code associated with the Jth noise table. Left in common NSIN.
XXIN(I,J)	I = 1,...,NSEGIN(J); J = 1,...,NCVTOT. The table of distances for the Jth noise table. Left in common NSIN.
YYIN(I,J)	I = 1,...,NSEGIN(J); J = 1,...,NCVTOT. The table of noise levels (EPNdB) for the Jth noise table. Left in common NSIN.

SUBROUTINE INIPLT/FINPLT

Purpose

Subroutine INIPLT initializes the CALCOMP plotter capability. It also calls the CALCOMP system subroutine FACTOR to scale the plotting data for 12 inch or 30 inch plot paper.

Entry FINPLT finalizes the CALCOMP plotter capability.

SUBROUTINE EXTARR

Purpose

Subroutine EXTARR sets up the arrays in common EXTS and EXTS1. These arrays contain the information pertaining to all extensions of all flights on all runways for an airport case stored sequentially from the first extension to the last.

Subroutines Called

None.

Input

NRW	The number of runways for this airport. From common NN.
NFT(J)	J = 1,...,NRW. The number of flights for each runway. From common NN.
NFLIGHT	The cumulative number of flights for this airport. From common XDATA.
NSG(K)	K = 1,...,NFLIGHT. The number of segments in each flight. From common XDATA.
ZETI(L,K)	L = 1,...,NSG(K); K = 1,...,NFLIGHT. The turn angle of each segment of each flight. From common XDATA.
THETAI(L,K)	L = 1,...NSG(K); K = 1,...,NFLIGHT. The climb angle of each segment of each flight. From common XDATA.
RADI(L,K)	L = 1,...,NSG(K); K = 1,...,NFLIGHT. The turn radius of each segment of each flight. From common XDATA.

PTHRI(L,K)	L = 1,...,NSG(K); K = 1,...,NFLIGHT. The percent thrust on each segment of each flight. From common XDATA.

Output

NEXT	Cumulative number of extensions over all flights for this airport. Left in common EXTSI.
ALT(KS)	KS = 1,...,NEXT. The turn angle of each extension. Left in common EXTS.
CANGLE(KS)	KS = 1,...,NEXT. The climb angle of each extension. Left in common EXTS.
CURV(KS)	KS = 1,...,NEXT. The turn radius of each extension. Left in common EXTS.
JEXT(KS)	KS = 1,...,NEXT. The number of extensions in the flight associated with each extension. Left in common EXTS1.
IFT(KS)	KS = 1,...,NEXT. The flight number of the flight associated with each extension. Left in common EXTS1.
JRW(KS)	KS = 1,...,NEXT. The runway number of the runway associated with each extension. Left in common EXTS1.

SUBROUTINE DSTORE

Purpose

Subroutine DSTORE computes the position vector of the starting point and the unit vector in the direction of flight at the starting+point for each segment of each flight. If the segment is helical, DSTORE calls subroutine CENHEL which computes the position vector of the helical center. DSTORE will also print a table of the information described above.

+For both takeoffs and landings, the starting point of a segment is taken as the end point closest in distance along the flight path to the runway.

Subroutines Called

SUBROUTINE DATA (JS)

SUBROUTINE SLPDS (ITEMPI, WEIGHT, DIST,IE,SLOPE)

FUNCTION IEXT (IS,JS)

FUNCTION NSCM (FLIGHT, ACTYPE)

SUBROUTINE VSCL (UV,FFIU,UG)

SUBROUTINE VADD (GIMC (1,LX,NFG),VEC, GIMC(1,LX -1,NFL))

SUBROUTINE ELUV (LX,NFL)

SUBROUTINE PARSUI(VOBR, LX -1,NFL)

SUBROUTINE FNHEV (TH,VEC)

SUBROUTINE CENHEL (LX,NFL)

Input

NRW	The number of runways for this airport. From common NN.
NFT(JS)	JS = 1,...,NRW. The number of flights for each runway. From common NN.
JEXT(KS)	KS = 1,...,NEXT. The number of extensions in the flight associated with each extension. From common EXTS1.
TP(IS,JS)	IS = 1,...,NFT(JS), JS = 1,...,NRW. The takeoff/landing code for each flight of each runway. From common AA.
TY(IS,JS)	IS = 1,...NFT(JS); JS = 1,...,NRW. The aircraft type number for each flight of each runway. From common AA.
THETAI(LX,IS)	LX = 1,...,NSG(IS); IS = 1,...,NFLIGHT. The climb angle for each segment of each flight. From common XDATA.
ZETI(LX,IS)	LX = 1,...,NSG(IS); IS = 1,...,NFLIGHT. The turn angle or length of ground track for each segment of each flight. From common XDATA.

RADI(LX,IS) LX = 1,...,NSG(IS); IS = 1,...,NFLIGHT. The turn radius for each segment of each flight. From common XDATA.

Output

GIMC(LX,NSEGM,NFL) LX = 1,2,3; NSEGM = 1,...,NSG(NFL); NFL = 1,...,NFLIGHT. The position vector of the starting point of each segment of each flight. Left in common XNEW.

UNC(LX,NSEGM,NFL) LX = 1,2,3; NSEGM = 1,...,NSG(NFL); NFL = 1,...,NFLIGHT. The unit vector in the direction of flight from the starting point of each segment of each flight. Left in common XNEW.

Output of flight information is also printed. See Appendix F for sample airport output.

SUBROUTINE SUMRF

Purpose

Subroutine SUMRF prints a summary of the runways and flights associated with each runway including the aircraft type number and landing/takeoff designation.

Subroutines Called

None.

Input

NRW The number of runways for this airport. From common NN.

NFT(I) I = 1,...,NRW. The number of flights for each runway. From common NN.

NFLIGHT The cumulation number of flights for this airport. From common XDATA.

TP(J,I) J = 1,...,NFT(I); I = 1,...,NRW. The takeoff/landing code for each flight of each runway.

From common AA.

TY(J,I)	J = 1,...,NFT(I); I = 1,..., NRW. The aircraft type number for each flight of each runway. From common AA.

Output

Printed output. See Appendix F for sample airport output.

SUBROUTINE POINTS

Purpose

Subroutine POINTS sets up the loops for the x and y coordinates of the observer grid points and calls EXPNE or EXPWE to calculate the appropriate noise values at each such point.

Subroutines Called

SUBROUTINE FTTOM (ARRLOC,NX)

SUBROUTINE EXPNE (DU,X)

SUBROUTINE EXPWE (DU,X)

Input

X1	The x component of the initial observer grid point. From common GDPAR.
Y1	The y component of the initial observer grid point. From common GDPAR.
CX	The x component of the grid increment. From common GDPAR.
CY	The y component of the grid increment. From common GDPAR.
NX	The number of grid elements in the x direction. From common GDPAR.
NY	The number of grid elements in the y direction. From common GDPAR.

Output

RO(I)	I = 1,2,3. The position vector of the observer grid point. This is generated for each observer grid point and left in common BB for other routines to use.
ARNV(IS,JS)	IS = 1,...,NX; JS = 1,...,NY. The value of NE, NER or WECPNL at the observer grid point with position vector (X_{IS}, Y_{JS}). Left in common ARP for output.
ARRLOC(IS)	IS = 1,...,NX. This array is used for printed output directly from POINTS.

Printed output. See Appendix F for example.

SUBROUTINE CONTR

Purpose

Subroutine CONTR is a dummy subroutine. It was inserted in order to allow for a routine which would directly compute the noise exposure contours without first computing noise exposure index at each observer grid point.

SUBROUTINE BNOIIN (TAC, HX, NFL, MXS)

Purpose

Subroutine BNOIIN calculates the noise (EPNdB) at an observer grid point due to a specified aircraft type on a given flight path. The minimum distance from the observer to the flight is used.

Subroutine BNOIIN is called only if the noise tables are supplied by the user.

Subroutines Called

SUBROUTINE GNFNIN(NCV,X,1,XQX)

Input

TAC	The type number of the aircraft for this

flight. From the call list.

NFL	The cumulation flight number of the flight being considered. From the call list.
MXS	The cumulation segment number of the flight path segment which is closest to the observer grid point. From the call list.
HX	The minimum distance from the flight path to the observer grid point. From the call list.
PTHRI(MXS,NFL)	The percent thrust for flight number NFL on segment number MXS. From common XDATA.
THRTB(I)	The percent thrust associated with the Ith noise table where I is determined for TAC and PTHRI(MXS,NFL). From common NSIN.
NATYTB(I)	The aircraft type number associated with the Ith noise table. From common INSIN.

Output

EPNDB	The effective perceived noise level at the observer grid point. From flight number NFL. Left in common BB.

SUBROUTINE CALPLT

Purpose

Subroutine CALPLT controls the CALCOMP plot output. CALPLT reads the contour values and sets up the axes and labels for contour plotting. It also sets up the array "FLIGHT(22,20)" and calls subroutine HEAD in order to plot the text of a detailed description of the flight data if there are not more than 10 flights. CALPLT also calls subroutine RUNWAY which plots the runways for the contour plot and RWYLEG which plots the runway configuration separately.

CALPLT also calls subroutine AUTCNT up to three times in order to obtain one, two or three contours and their plots.

Finally CALPLT notifies the CALCOMP operator when to change pen colors.

Subroutines Called

SUBROUTINE HEAD(NUM)

SUBROUTINE NOMEN (XM,YM,XD,YD) (Dummy Subroutine)

SUBROUTINE AUTCNT (VAL1)

SUBROUTINE RUNWAY (X1, Y1, X2, Y2)

SUBROUTINE RWYLEG (AINCH)

Input

VAL1 VAL2 VAL3	The contour values. Read from data card. Format 3 F10.1.
CX	The x increment for the observer grid points. From common GDPAR.
CY	The y increment for the observer grid points. From common GDPAR.
X1	The x coordinate of the initial observer grid points. From common GDPAR.
Y1	The y coordinate of the initial observer grid point. From common GDPAR.
NX	The number of x elements in the observer grid. From common GDPAR.
NRW	The number of runways in this airport. From common NN.
XLM1 (I,J)	I = 1,2,3; J = 1,...,NRW. The position vector of the starting point of each runway. From common AA.
XLM2(I,J)	I = 1,2,3; J = 1,...,NRW. The position vector of the terminal point of each runway. From common AA.
NFT(J)	J = 1,...,NRW. The number of flights for each runway. From common NN.

TP(I,J)	I = 1,...,NFT(J); J = 1,...,NRW. The takeoff/landing code for each flight of each runway. From common AA.
W(I,J)	I = 1,...,NFT(J); J = 1,...,NRW. The gross weight of the aircraft for each flight of each runway. From common AA.
PT(I,J)	I = 1,...,NFT(J); J = 1,...,NRW. The percent thrust on segments one and two for each flight of each runway. From common AA.
ALT(KX)	KX = 1,...,NEXT. The turn angle of each extension. From common EXTS.
CANGLE(KX)	KX = 1,...,NEXT. The climb angle of each extension. From common EXTS.
THRUST(KX)	KX = 1,...,NEXT. The percent thrust on each extension. From common EXTS.
CURV(KX)	KX = 1,...,NEXT. The turn radius of each extension. From common EXTS.
NSG(KI)	KI = 1,...,NFLIGHT. The number of segments (including extension) for each flight. From common XDATA.
AINCHX	The length of the x axis to be used in plotting. From common FACTR.
AINCHY	The length of the y axis to be used in plotting. From common FACTR.
DELX	The x increment to be used in plotting. From common FACTR.
DELY	The y increment to be used in plotting. From common FACTR.
AXMAX AXMIN	The maximum and minimum values of x to be plotted. From common FACTR.
AYMAX AYMIN	The maximum and minimum values of y to be plotted. From common FACTR.
N(I,J,K)	I = 1,2,3; J = 1,...,NFT(K); K = 1,...,NRW. The number of day, evening and night operations for each flight on each runway. From common NN.

NFLIGHT	The cumulative number (over all runways) of flights for this airport. From common XDATA.

Output

Plots of descriptive text and contours. See Appendix F for sample CALPLT contour data.

SUBROUTINE DATA(J)

Purpose

Subroutine DATA calculates for the Jth runway the unit vector in the direction of takeoff and landing. It also determines the points on the runway at which takeoff roll starts and touchdown occurs.

Subroutines Called

SUBROUTINE VSUB(UG,LAMBD2.LAMBD1)

SUBROUTINE VUNT(UG,UG)

Input

J	The cumulative runway number. From the call sequence.
XLM1(M,J)	M = 1,2,3; J = 1,...,NRW. The position vector of the beginning point of each runway. From common AA.
XLM2(M,J)	M = 1,2,3; J = 1,...,NRW. The position vector of the end point of each runway. From common AA.
STO(J)	J = 1,...,NRW. The distance along each runway from XLM1 at which takeoff roll starts. From common AA.
SL(J)	J = 1,...,NRW. The distance along each runway from XLM1 at which touchdown occurs. From common AA.

Output

UG(M)	M = 1,2,3. The unit vector at XLM1 in the direction XLM1→XLM2 for the Jth runway. Left in common CC.
GUMP	STO(J) for the Jth runway. Left in common CC.
TAU1	SL(J) for the Jth runway. Left in common CC.

SUBROUTINE EXPNE(DUMY,XXX); SUBROUTINE EXPWE(DUMY,XXX)

Purpose

Subroutines EXPWE and EXPNE are the control programs for computation of the noise exposure indices at a specified observer grid point.

Subroutine EXPWE computes WECPNL while subroutine EXPNE computes NE or NEF. Although the algorithms are slightly different, both routines proceed in a similar manner. Essentially, for each flight path ground track from each runway the shortest distance and corresponding elevation angle from the observer grid point to the flight path of a particular aircraft type are obtained. From this, the EPNL at the observer grid point is calculated and modified for ground attenuation, shielding and number of identical operations. The total noise exposure index from all flights on all runways is then computed.

Subroutines Called

SUBROUTINE FTTOM(XRO,3)

SUBROUTINE HBETA(XDIST, SINB, PFN, MXS, RO, ISEG, NFL)

FUNCTION IEXT(I,J)

SUBROUTINE BNOISE(ACTYPE,XDIST)

SUBROUTINE BNOIIN(ACTYPE,XDIST,NFL,MXS)

SUBROUTINE ATTENU(SINB, XDIST, FLIGHT, X)

SUBROUTINE ENT2(ACTYPE, WEIGHT, PFN, IPRF, FLIGHT)

SUBROUTINE ENT3(X, NOPARR, XDIST)

Input

NRW	The number of runways for this airport. From common NN.
NFT(J)	J = 1,...,NRW. The number of flights for each runway. From common NN.
TP(I,J)	I = 1,...,NFT(J); J = 1,...,NRW. The takeoff/landing code for each flight of each runway. From common AA.
W(I,J)	I = 1,...,NFT(J); J = 1,...NRW. The gross weight of the aircraft for each flight of each runway. From common AA.
TY(I,J)	I = 1,...,NFT(J); J = 1,...,NRW. The aircraft type number for each flight of each runway. From common AA.
JEXT(MX)	MX = 1,...,NEXT. The number of extensions in the flight associated with the MXth extension. From common EXTS1.
RO(IJ)	IJ = 1,...,3. The position vector of the grid point at which the noise exposure index is to be calculated.
N(L,I,J)	L = 1,2,3; I = 1,...,M; J = 1,...,NRW. The number of day, evening, and night operations for each flight of each runway. From common NN.

Output

XXX	The values of the noise exposure index at the grid point given by RO(IJ) due to all flights of all runways. Passes through the calling sequence.

SUBROUTINE BNOISE(TAC,HX)

Purpose

Subroutine BNOISE calculates the noise (EPNJB) at an observer grid point due to a specified aircraft type on a given flight path. The minimum distance from the observer to the flight path is used.

Subroutine BNOISE is called only if the built-in noise tables are used.

Subroutines Called

SUBROUTINE GENFN(NFN,HX,1,XQX)

SUBROUTINE ENT4(NFN,COR,Z,XQX)

Input

TAC	The aircraft class type number for the flight being considered. From call sequence.
HX	The minimum distance from the observer grid point to the flight path being considered. From call sequence.
FLIGHT	1 if the flight is a landing, or 0 if the flight is a takeoff. From common BB.

Output

EPNDB	The effective perceived noise level at the observer grid point due to an aircraft of aircraft class type number TAC. Left in common BB.

SUBROUTINE SLPDS(NZ,ACWT,DIST,IE,XXX)

Purpose

Subroutine SLPDS calculates the tangent of the initial climb angle and the roll distance if the flight is a takeoff. SLPDS calculates the tangent of the final descent angle if the flight is a landing.

Subroutines Called

FUNCTION IBND(IV,IS,IH) (Logical)

FUNCTION BND(V,S,H) (Logical)

FUNCTION XINT(PT,Y)

Input

NZ — The aircraft type number for the flight being considered. Assumed positive if a takeoff flight and negative if a landing flight. From call sequence.

ACWT — The aircraft weight, in pounds, designated for the flight being considered. From call sequence.

Output

DIST — The takeoff roll distance for the flight being considered. This is the distance from start of takeoff to lift-off. Computed only for a takeoff flight.

XXX — The tangent of the initial climb angle if the flight being considered is for takeoffs. Otherwise, the tangent of the descent angle for a landing flight. Passes through call sequence.

IE — Error indicator used if NZ is not a valid aircraft type number of if the tangent of the climb (descent) angle is zero. Passes through call sequence.

SUBROUTINE ELUV(NSEGM,NFL)

Purpose

Subroutine ELUV calculates the unit vector in the direction of flight at the starting point of a specified segment of the flight path of a specified flight.

Subroutines Called

SUBROUTINE VUNT(VEC1,VEC)

Input

NSEGM — The segment number of the segment being considered. From call sequence.

NFL — The cumulative flight number of the flight being considered. From call sequence.

UNC(I,K1,NFL) I = 1,2,3; K1 = NSEGM-1. The unit tangent vector in the direction of flight at the end point of the previous segment. From common XNEM.

THETAI(NSEGM,NFL) The climb angle of the segment being considered. From common XDATA.

Output

UNC(I,NSEGM,NFL) I = 1,2,3. The unit vector in the direction of flight at the starting point of the segment being considered. Left in common XNEW.

SUBROUTINE PARSU1(OBR,NSEGM,NFL)

Purpose

Subroutine PARSU1 calculates the parameters necessary for the computation of the coordinates of an aircraft on a specified helical segment of a given flight.

Subroutines Called

SUBROUTINE VSUB(VEC,GIMC,HELCN)

SUBROUTINE VUNT(VEC1,VEC)

Input

OBR(J) J = 1,2,3. The position coordinates of the observer grid point. Passes through call sequence.

NSEGM The segment number of the helical segment being considered. Passes through call sequence.

NFL The cumulative flight number of the flight containing the segment being considered. Passes through call sequence.

RAD1(NSEGM,NFL)	The turning radius of the helical segment being considered. From common XDATA.
HELCN(J,NSEGM,NFL)	J = 1,2,3. The position coordinates of the center of the helical segment being considered. From common HLX.
THETAI(NSEGM,NFL)	The climb angle of the segment being considered. From common XDATA.
GIMC(J,NSEGM, NFL)	J = 1,2,3. The position coordinates of the starting point of the segment being considered. From common XNEW.

Output

RAD	= RADI(NSEGM,NFL). Left in common CHELX.
ZH	= HELCN(3,NSEGM,NFL). Left in common CHELX.
ALPHA	= THETAI(NSEGM,NFL). Left in common CHELX.
Q	= -1, if RAD<0 Left in common CHELX. = 1, if RAD>0
E F	Coordinates of the projection on the XY plane of the unit vector in the direction of flight at the starting point of the segment being considered. Left in common CHELX.
R S T	Coordinates of the helical center of the segment being considered. Left in common CHELX.

SUBROUTINE FNHEV(TH,VEC)

Purpose

Subroutine FNHEV calculates the position vector relative to the helical segment as a function of the angle turned through.

Subroutines Called

FUNCTION F1 (TH)

FUNCTION F2 (TH)

FUNCTION F3 (TH)

Input

TH	The angle through which the aircraft has turned on the helical segment. From call sequence.
R S T	The position coordinates of the center of the helix. From common CHELX.

Output

VEC(I)	I = 1,2,3. The position vector of the aircraft on the helical center. Passes through call sequence.

SUBROUTINE FUDVH(TH,VEC1)

Purpose

Subroutine FUDVH calculates the unit vector in the direction of the derivative vector at a point specified by the angle of turn on a given helical segment.

Subroutines Called

FUNCTION F1P(TH)

FUNCTION F2P(TH)

FUNCTION F3P(TH)

SUBROUTINE VUNT(VEC1,VEC)

Input

TH	The angle of turn. From call sequence.

Output

VEC1	The unit vector in the direction of the derivative vector at the point specified by TH on the given helical segment. Passes through call sequence.

SUBROUTINE CENHEL(NSEGM,NFL)

Purpose

Subroutine CENHEL calculates the position vector of the center of a specified helix.

Subroutines Called

None.

Input

NFL	The cumulative flight number of the flight containing the helical segment under consideration. From call sequence.
NSEGM	The segment number of the helical segment under consideration from the NFLth flight. From call sequence.
GIMC(I,NSEGM,NFL)	I = 1,2,3. The position vector of the starting point of segment number NSEGM of flight number JFL. From common XNEW.
UNC(I,NSEGM,NFL)	I = 1,2,3. The unit vector in the direction of flight from the starting point of segment number NSEGM of flight number NFL. From common XNEW.
RADI(NSEGM,NFL)	The turn radius for segment number NSEGM of flight number NFL. From common XDATA.

Output

HELCN(I,NSEGM,NFL)	I = 1,2,3. The position vector of the center of the helical segment with segment number NSEGM of flight number NFL. Left in common HLX.

SUBROUTINE GENFN (M,X,MVAR,XX)

Purpose

Subroutine GENFN is used for interpolation with the built-in noise table. The interpolation may be linear or logarithmic.

Subroutine GENFN is used to interpolate for the uncorrected perceived noise level and the thrust correction term at the observer grid point only when subroutine BNOISE is used. However, subroutine GENFN is used to obtain the ground attenuation correction term whether the built-in noise tables are used or the user supplies his own noise table.

Subroutines Called

FUNCTION II (IL,IR)

Input

M	The number of the noise curve to be used for interpolation. From call sequence.
X	The variable (minimum distance, percent thrust or standard percent thrust) at which interpolation is to be made. From call sequence.
MVAR	The number which specifies whether X is a value of distance percent thrust or thrust. From call sequence.
NSEGS(M)	The number of tabular values in the Mth curve. Set in DATA statement.
XX(IM,M)	IM = 1,...,NSEGS(M). The table of the independent variable for the Mth curve. Set in DATA statement.

YY(IM,M) — IM = 1,...,NSEGS(M). The table of the dependent variable for the Mth curve. Set in DATA statement.

TYPE(M) — The type of interpolation (linear = 0 or logarithmic = 1) to be used for the Mth curve. Set in DATA statement.

Output

XX — The value of the dependent variable interpolated for. Passes through call sequence.

SUBROUTINE GNFNIN(M,X,MVAR,XXX)

Purpose

Subroutine GNFNIN selects the correct noise curve for a specified aircraft class type number and percent of full thrust and then interpolates for the uncorrected effective perceived noise level, in EPNDB, corresponding to the minimum distance from the observer grid point to the flight path. The interpolation is logarithmic or linear depending on the distance scale of the noise curves.

Subroutine GNFNIN is used only when subroutine BNOISE is used.

Subroutines Called

FUNCTION II(IL,IR)

Input

M — The number of the noise curve to be used for interpolation. From call sequence.

X — The minimum distance from the observer grid point to the flight path. From call sequence.

MVAR — The component of the vector VAR to be used in the event of an error (X = 0) printout. MVAR = 1 from BNOIIN. From call sequence.

NSEGIN(M) — The number tabular values in the Mth noise

	curve. From common INSIN.
XXIN(I,M)	I = 1,...,NSEGIN(M). The distance table for the Mth noise curve. From common NSIN.
YYIN(I,M)	I = 1,...,NSEGIN(M). The noise table for the Mth noise curve. From common NSIN.
TYPIN(M)	The type of interpolation for the Mth noise curve. From noise curve. From common NSIN.

Output

XXX	The noise level at the observer grid point uncorrected for attenuation. Passes through call sequence.

SUBROUTINE ATTENU (SB,HX,FTT,XXX)

Purpose

Subroutine ATTENU calculates the ground attenuation correction term as a function of the distance and elevation angle from the observer grid point to the flight path.

Subroutines Called

SUBROUTINE GENFN (M,HX,YYY)

Input

SB	The sine of the elevation angle from the observer grid point to the closest point of the flight path. From call sequence.
HX	The minimum distance from the observer grid point to the flight path. From call sequence.
FTT	The code which specifies if the flight is a takeoff or landing. From call sequence.

Output

XXX	The ground attenuation correction term in

EPNdB. Passes through call sequence.

SUBROUTINE HBETA (HX,BETAS,PFN,MAXSEG,OBR,ISEG,NFL)

Purpose

Subroutine HBETA calculates the minimum distance and elevation angle from the specified observer grid point to the specified flight path. Subroutine HBETA also smooths the thrust discontinuity between flight path segments when appropriate.

Subroutines Called

SUBROUTINE VCTPRP (GINC(1,I,NFL),UNCO,I,NFW,OBR,VEC)

SUBROUTINE VMAG (VEC,TEMP)

SUBROUTINE VADD (VECSM,VEC,OBR)

SUBROUTINE FALLIN (I,NFL, - 100.)

SUBROUTINE HELMN (OBR,I,NFL,VEC.TH)

SUBROUTINE SEGCH (ISN,NFL,TH)

SUBROUTINE VSUB (VEC,DTSXYZ (1,ISN),GINC(1,ISN,NFL))

FUNCTION THRST (TH1,TH2,X)

Input

OBR(I)	I = 1,2,3. The position vector of the specified observer grid point. From call sequence.
NFL	The cumulative flight number of the specified flight. From call sequence.
ISEG	The number of segments in flight number NFL. From call sequence.
RADI(I,NFL)	I = 1,...,NSEG. The turn radius of each segment of flight number NFL. From common XDATA.
GIMC(L,I,NFL)	L = 1,2,3; I = 1,...,NSEG. The position vector of the endpoint of each segment of flight number

NFL. From common XNEW.

UNC(L,I,NFL) — L = 1,2,3; I = 1,...,NSEG. The unit vector in the direction of flight from the endpoint of each segment of flight number NFL. From common XNEW.

PTHRI(I,NFL) — I = 1,...,NSEG. The percent of full thrust on each segment of flight number NFL. From common XDATA.

Output

HX — The minimum distance from the observer grid point to the flight path. Passes through call sequence.

BETAS — The sine of the elevation angle from the observer grid point to the flight path along the minimum distance vector. Passes through call sequence.

PFN — The percent of full thrust an aircraft uses on the flight path at its minimum distance to the observer grid point. Passes through call sequence.

MAXSEG — The segment number of the segment closest to the observer grid point. Passes through call sequence.

DTSMIN(I) — I = 1,...,NSEG. The distance from the observer grid point to the projected point of each segment of flight number NFL. Left in common MNDIS.

DTSXZ(L,I) — L = 1,2,3; I = 1,...,NSEG. The position vector of the projected point of each segment of flight number NFL. Left in common MNDIS.

IDTOST(I) — I = 1,...,NSEG. The orientation code of the projected point of each segment of flight number NFL. Left in common MNDIS.

SUBROUTINE VCTPRP (POS,UNIT,PT,PRP)

Purpose

Subroutine VCTPRP calculates the point closest to the specified observer grid point and on the line determined by the specified linear segment. This point is referred to as the projected point.

Subroutines Called

Subroutine VSUB (A,POS,PT)

Function VPOT (A,UNIT)

Subroutine VSCL (PRP,UCOF,UNIT)

Input

POS(I)	I = 1,2,3. The position vector of the end-point of the specified linear segment. From call sequence.
UNIT(I)	I = 1,2,3. The unit vector in the direction of the specified linear segment. From call sequence.
PT(I)	I = 1,2,3. The position vector of the specified observer grid point. From call sequence.

Output

PRP(I)	I = 1,2,3. The position vector of the projected point relative to the observer grid point. Passes through call sequence.

SUBROUTINE FALLIN (NSEGM,NFL,X)

Purpose

Subroutine FALLIN determines the orientation of the projected point relative to the segment. The following code is used to present the orientation information:

1. - Projected point is to left of the segment.

2. - Projected point is on the segment.

3. - Projected point is to right of the segment.

4. - Projected point is undefined.

Subroutines Called

SUBROUTINE VSCL (VEC,100000.,UNC (1,NSEGM, NFL))

SUBROUTINE VADD (GIMC (1, NSEGM + 1, NFL), GIMC (1,NSEGM, NFL),VEC)

Input

X	The angle which determines the projected point on the segment if it is helical. X = 100. if the segment is linear. From call sequence.
NFL	The cumulative flight number of the flight being considered. From call sequence.
NSEGM	The number of the segment being tested in flight number NFL. From call sequence.
NSEG(NFL)	The total number of segments in flight number NFL. From common XDATA.
UNC(I,NSEGM,NFL)	I = 1,2,3. The unit vector in the direction of flight from the starting point of the specified segment of flight number NFL. From common XNEW.
GIMC(I,NSEGM,NFL)	I = 1,2,3. The position vector of the starting point of the specified segment of flight number NFL. From common XNEW.
DTSXY2(I,NSEGM)	I = 1,2,3. The position vector of the projection of the specified segment. From common MNDIS.

Output

IDTOSI(NSEGM)	The orientation of the projected point relative to the segment. Left in common MNDIS.

SUBROUTINE HELMN (OBR,I,NFL,VEC,TH)

Purpose

Subroutine HELMN calculates the point closest to the specified observer grid point and on the helix determined by the specified helical segment. This point is referred to as the projected point. Subroutine HELMN also calculates the distance from the projected point to the observer grid point and the turn angle to the projected point.

Subroutines Called

SUBROUTINE PARSU1 (OBR,I,NFL)

SUBROUTINE HMIND (0.,THETA2,TH,VEC)

SUBROUTINE VMAG (VEC,TMNDS)

Input

OBR(J)	J = 1,2,3. The position vector of the observer grid point. From call sequence.
NFL	The cumulative flight number of the flight being considered. From call sequence.
I	The segment number of the specified segment of flight number NFL. From call sequence.
ZETI(I,NFL)	The turn angle of the specified segment. From common XDATA.
NSG(NFL)	The total number of segments in flight number NFL. From common XDATA.

Output

VEC(J)	J = 1,2,3. The position vector of the projected point. Passes through call sequence.
TH.	The turn angle to the projected point. Passes through call sequence.
DISMIN(I)	The distance from the observer grid point to the projected point. Left in common MNDIS.

SUBROUTINE SEGCH (NSEGM, NFL, OBR)

Purpose

Subroutine SEGCH determines the segment of the specified flight for which the projected point is closest to the specified observer grid point.

If the projected points of one or more segments is on the segment, the segment whose projected point is closest to the observer grid point is chosen. Otherwise, the first or last segment is chosen depending on which is closest.

Subroutines Called

SUBROUTINE VSUB(VEC,CIMC(1,1,NFL),OBR)

SUBROUTINE VMAG(VEC,D)

Input

NFL	The cumulative number of the flight being considered. From call sequence.
OBR(I)	I = 1,2,3. The position vector of the observer grid point being considered.
NSG(NFL)	The total number of segments in flight number NFL. From common XDATA.
IDTOSI(I)	I = 1,...,NSG(NFL). The orientation code for the projected point of each segment of flight number NFL. From common MNDIS.
DTSMIN(I)	I = 1,...,NSG(NFL). The distance from the projected point of each segment of flight number NFL to the specified observer grid point. From common MNDIS.

FUNCTION THRST (TH1,TH2,X)

Purpose

Function THRST smooths the thrust transition between segments. This is done by linearly changing the thrust over a distance of about one mile.

Subroutines Called

None.

Input

TH1	The percent of thrust used on the prior segment. From call sequence.
TH2	The percent of thrust used on the present segment. From call sequence.
X	The distance traveled along the present segment. From call sequence.
THRST	The smoothed value of the percentage of thrust used on the present segment. The value of the function THRST.

<u>SUBROUTINE HMIND(THET1,THET2,THET,VEC)</u>

Purpose

Subroutine HMIND iterates for the point closest to the specified observer grid point and on a helix determined by the specified helical segment. The point is called the projected point. Subroutine HMIND also calculates the turn angle to the projected point.

Subroutines Called

FUNCTION F1(X)

FUNCTION F2(X)

FUNCTION F3(X)

Input

THET1	The initial turn angle of the specified helical segment. From call sequence.
THET2	The final turn angle of the specified helical segment. From call sequence.

EPS1	The convergence criteria on the turn angle for the iteration. From common NERAP.
EPS2	The criteria for bounding the second derivative of the distance function away from zero. This prevents a division by zero during the iteration. From common NERAP.
NOITT	The maximum number of iterations allowed for the iteration. From common NERAP.

Output

THET	The turn angle to the projected point. Passes through call sequence.
VEC(I)	I = 1,2,3. The position vector of the projected point. Passes through call sequence.

BLOCK DATA

Purpose

BLOCK DATA sets data into certain common arrays for use by various subroutines.

Output

JUMP	Control variable used by READIN. Left in common JOE3.
EPS1 EPS2 NOITT	Control parameters used by HMIND. Left in common NERAP.
NACT WFCTR PFGWT	Weight profile used by SLPDS. Left in common TOOPRF.
NTOP DFCTR TOPRF	Takeoff profile data used by SLPDS. Left in common TOOPRF.
PI	The number 3.14159265. Left in common TRNS.
NC	The dimensionality of the frame of reference. Left in common NCOORD.

VLN	Parameter used by XINT. Left in common LN.
CONARR S	Weighting turns used for computation of WECPNL by EXPWE. Left in common DENOP.

SUBROUTINE FTTOM (A,IDIM)

Purpose

Subroutine FTTOM converts a vector the components of which are specified in feet to a vector with components specified in meters.

Subroutines Called

None.

Input

IPIN	The dimension of the input vector. From call sequence.
A(I)	I = 1,...,IDIM. The vector with components in feet. From call sequence.

Output

A(I)	I = 1,...,IDIM. The transformed vector with components in meters. Passes through call sequence.

SUBROUTINE MTOFT (A,IDIM)

Purpose

Subroutine MTOFT converts a vector the components of which are specified in meters to a vector with components specified in feet.

Subroutines Called

None.

Input

IDIM	The dimension of the input vector. From call sequence.
A(I)	I = 1,...,IDIM. The vector with components in meters. From call sequence.

Output

A(I)	I = 1,...,IDIM. The transformed vector with components in feet. Passes through call sequence.

<u>SUBROUTINE LBTOKG (A,IDIM)</u>

Purpose

Subroutine LBTOKG converts a vector the components of which are weights specified in pounds to a vector with components specified in kilograms.

Subroutines Called

None.

Input

IDIM	The dimension of the input vector. From call sequence.
A(I)	I = 1,...,IDIM. The vector with components in pounds. From call sequence.

Output

A(I)	I = 1,...,IDIM. The transformed vector with components in kilograms. Passes through call sequence.

SUBROUTINE KGTOLB (A,IDIM)

Purpose

Subroutine KGTOLB converts a vector whose components are weights specified in kilograms to a vector with components specified in pounds.

Subroutines Called

None.

Input

IDIM	The dimension of the input vector. From call sequence.
A(I)	I = 1,...,IDIM. The vector with components in kilograms. From call sequence.

Output

A(I)	I = 1,...,IDIM. The transformed vector with components in pounds. Passes through call sequence.

FUNCTION DGTRD (DG)

Purpose

Function DGTRD converts an angle specified in degrees to radians.

Subroutines Called

None.

Input

DG	The angle, in degrees, to be transformed. From call sequence.

Output

DGTRD	The transformed angle in radians. Left as function value.

SUBROUTINE VADD (C,A,B)

Purpose

Subroutine VADD calculates the vector sum, C = A + B, of two vectors.

Subroutines Called

None.

Input

NC	The dimension of the vectors to be summed.
A(I) B(I)	I = 1,...,NC. The vectors to be summed. From call sequence.

Output

C(I)	I = 1,...,NC. The vector sum of A and B. Passes through call sequence.

SUBROUTINE VSUB (C,A,B)

Purpose

Subroutine VSUB calculates the vector difference, C = A - B, of two vectors.

Subroutines Called

None.

Input

NC	The dimension of the vectors to be differenced. From common NCOORD.

A(I) B(I)	I = 1,...,NC. The vectors to be differenced. From call sequence.

Output

C(I)	I = 1,...,NC. The vector difference, A - B, of A and B. Passes through call sequence.

SUBROUTINE VCROS (C,A,B)

Purpose

Subroutine VCROS calculates the cross product, C = A x B, of two vectors. It should be noted that the input vectors must be of dimension 3.

Subroutines Called

None.

Input

A(I) B(I)	I = 1,2,3. The vectors to be crossed. From call sequence.

Output

C(I)	I = 1,2,3. The cross product, A x B, of A and B. Passes through call sequence.

SUBROUTINE VSCL (C,F,A,)

Purpose

Subroutine VSCL calculates the scalar product, C = F•A, of a scalar F and a vector A.

Subroutines Called

None.

Input

NC	The dimension of the input vector. From common NCOORD.
A(I)	I = 1,...,NC. The vector to be multiplied. From call sequence.
F	The scalar quantity to be multiplied. From call sequence.

Output

C(I)	I = 1,...,NC. The product F•A of the scalar, F, and the vector, A. Passes through call sequence.

FUNCTION VDOT (A,B)

Purpose

Function VDOT calculates the dot product, A•B, of two vectors.

Subroutines Called

None.

Input

NC	The dimension of the input vectors. From common NCOORD.
A(I) B(I)	I = 1,...,NC. The vectors to be dotted. From call sequence.

Output

VDOT	The dot product A•B of the input vectors A and B. Left as function value.

SUBROUTINE VMAG (A,XXX)

Purpose

Subroutine VMAG computes the magnitude of a specified vector.

Subroutines Called

FUNCTION VDOT (A,A)

Input

NC	The dimension of the input vector. From common NCOORD.
A(I)	I = 1,...,NC. The vector for which the magnitude is to be calculated. From call sequence.

Output

XXX	The magnitude, $(A \cdot A)^{1/2}$, of the input vector A. Passes through call sequence.

SUBROUTINE VUNT (C,A)

Purpose

Subroutine VUNT determines the unit vector in the direction of a specified vector.

Subroutines Called

SUBROUTINE VMAG (A,VM)

Input

NC	The dimension of the input vector. From common NCOORD.
A(I)	I = 1,...,NC. The vector for which a unit vector is to be determined. From call sequence.

Output

C(I)	I = 1,...,NC. The unit vector determined by the vector A. Passes through call sequence.

FUNCTION II(N1,N2)

Purpose

Function II determines an integer midway between two specified integers. The algorithm is:

$$II = \left[\frac{N2 - N1 + 1}{2} \right]$$

where [ ] means "integer part."

Subroutines Called

None.

Input

N1	The left most (smallest) integer being considered. From call sequence.
N2	The right most (largest) integer being considered. From call sequence.

Output

II	The middle integer between N1 and N2. Left as function value.

FUNCTION XINT (PT,Y)

Purpose

Function XINT calculates the x coordinate which corresponds to a specified y coordinate and is on a line determined by two specified points.

Subroutines Called

None.

Input

PT(1) PT(2)	The x and y coordinates of the first point which determines the line. From call sequence.
PT(3) PT(4)	The x and y coordinates of the second point which determines the line. From call sequence.
Y	The y coordinate of the point on the line for which the x coordinate is to be calculated. From call sequence.

Output

XINT	The x coordinate of the point with y coordinate, y, and on the line determined by the points (PT(1), PT(2)) and (PT(3), PT(4)). Left as function value.

FUNCTION NACM (FL,ACT)

Purpose

Function NACM assumes the value of a specified aircraft class type number with a positive sign if the given flight is a takeoff and a negative sign if it is a landing.

Subroutines Called

None.

Input

FL	The takeoff (=0) or landing (=1) code for the flight. From call sequence.
ACT	The aircraft clsss type number for the flight. From call sequence.

Output

NACM	= ACT if flight is a takeoff

= -ACT if flight is a landing
Left as function value.

FUNCTION F1(X), FUNCTION F2(X), FUNCTION F3(X)

Purpose

Functions F1, F2 and F3 assume the values of the first, second and third position coordinates, respectively, of a point determined by a given turn angle on a specified helical segment. The position coordinates are given relative to a specified observer grid point.

Functions F1, F2 and F3 have entries F1P, F2P and F3P, respectively, which assume the values of the derivative with rsspect to turn angle of the coordinates.

Functions F1, F2 and F3 also have entries F1PP, F2PP and F3PP, respectively, which assume the values of the second derivative with respect to turn angle of the coordinates.

Subroutines Called

None.

Input

X	The turn angle defining the point on the helical segment. From call sequence.
R S T	Coordinates of the helical center of the segment being considered. From common CHELX.
E F	Coordinates of the projection on the xy plane of the unit vector in the direction of flight at the starting point of the segment being considered. From common CHELX.
RAD	= RADI(NSEGM,NFL). From common CHELX.
ZH	= HELCN(3,NSEGM,NFL). From common CHELX.
ALPHA	= THETAI(NEGM,NFL). From common CHELX.
Q	-1 if RAD<0 or 1 if RAD>0. From common CHELX.

Output

F1 F2 F3	The first, second or third coordinate of the specified point relative to the given observer grid point. Left as function value.

or

F1P F2P F3P	The derivative with respect to turn angle of the first, second or third coordinate of the specified point. Left as function value.

or

F1PP F2PP F3PP	The second derivative with respect to turn angle of the first, second or third coordinate of the specified point. Left as function value.

SUBROUTINE ENT2 (CAT,TW,PNF,IXX,FLT)

Purpose

Subroutine ENT2 is a diagnostic subroutine used to print intermediate values.

Subroutines Called

None.

Input/Printed Output

CAT	Aircraft type number.
TW	Aircraft weight.
PNF	Aircraft thrust in percent.
IXX	Aircraft profile number.
FLT	0. takeoff. 1. landing.

SUBROUTINE ENT3 (XQ,NQ,DQ)

Purpose

Subroutine ENT3 is a diagnostic subroutine used to print intermediate values.

Subroutines Called

None.

Input/Printed Output

XQ	Elevation angle from observer to flight path in degrees.
NQ	Number of operations (classified as day operations for NE calculations, as day-evening-night operations for WECPNL calculations, and as day-night operations for NEF calculations).
DQ	Distance from observer to flight path.

SUBROUTINE ENT4 (IFN,XCOR,XTH,XEPN)

Purpose

Subroutine ENT4 is a diagnostic subroutine used to print intermediate values.

Subroutines Called

None.

Input/Printed Output

IFN	Number of particular EPNdB vs distance curve.
XCOR	Correction to curve (of an additive nature).
XTH	Thrust correction to curve (because of varying thrust).
XEPN	Raw EPNdB value from curve.

NEXT	The total number of extensions for all flights and all runways. From common EXTS1.
JRW(KS)	KS = 1,...,NEXT. The array of runway numbers associated with each extension. From common EXTS1.
JFT(KS)	KS = 1,...,NEXT. The array of flight numbers associated with each extension. From common EXTS1.

Output

IEXT	The value of the index to be used with the arrays in common EXTS for information about the extensions of the ISth flight of the JSth runway. Left as function value.

LOGICAL FUNCTION BND(V,S,H)

Purpose

Logical function BND assumes the value TRUE if its first argument falls in the closed interval defined by its second and third arguments. Otherwise BND assumes the value FALSE. All arguments are real.

Subroutines Called

None.

Input

V	The argument to be tested. From call sequence.
S	The lower bound of the testing interval. From call sequence.
H	The upper bound of the testing interval. From call sequence.

Output

BND	TRUE if $V \in [S,H]$; FALSE if $V \notin [S,H]$. Left as function value.

SUBROUTINE ENT5N (XI,YI,ZI,XNEI,AXEI); SUBROUTINE ENT5W (XI, YI,ZI,XNEI,AXEI)

Purpose

Subroutines ENT5N and ENT5W are diagnostic subroutines used for printout of intermediate values.

Subroutines Called

None.

Input/Printed Output

XI	Attenuation value.
Y	Shielding value.
ZI	Noise contribution from identical operations (day operations in ENT5N and day-evening-night operations in ENT5W).
XNEI	Net EPNdB value.
AXEI	EPNdB subtotal.

FUNCTION IEXT(IS,JS)

Purpose

Function IEXT assumes the value of the index required when referencing the arrays in common EXTS for information about the extensions of the flight path of a specified flight for a specified runway.

Subroutines Called

None.

Input

JS	The cumulative number of the runway being considered. From call sequence.
IS	The number of the flight being considered on the specified runway. From call sequence.

LOGICAL FUNCTION IBND(IV,IS,IH)

Purpose

Logical function IBND assumes the value TRUE if its first argument falls in the closed interval defined by its second and third arguments. Otherwise IBND assumes the value FALSE. All arguments are integers.

Subroutines Called

None.

Input

IV	The integer argument to be tested. From call sequence.
IS	The integer lower bound of the testing interval. From call sequence.
IH	The integer upper bound of the testing interval. From call sequence.

Output

IBND	TRUE, if IV ε [IS,IH]; FALSE, if IV $\not\in$ [IS,IH]. Left as function value.

SUBROUTINE PLOTA(X,Y,I)

Purpose

Subroutine PLOTA performs the same function as the CALCOMP subroutine PLOT except that the point to be plotted is first rotated by a specified angle about a specified point.

Subroutines Called

SUBROUTINE PLOT(----) (CALCOMP)

Input

X	The X coordinate of the point to be rotated and plotted. From call sequence.
Y	The Y coordinate of the point to be rotated and plotted. From call sequence.
I	A signed integer used by subroutine PLOT to control pen position during movement. From call sequence.
SINAN	The sine of the angle of rotation. From common IANG.
COSAN	The cosine of the angle of rotation. From common IANG.
XW YW	The coordinates of the point about which the rotation is made. From common IANG.

Output

A plotted point.

SUBROUTINE RWYLEG(AINCH)

Purpose

Subroutine RWYLEG is used to plot the runway configuration and nomenclature.

Subroutines Called

SUBROUTINE RUNWAY(SX1,SY1,SX2,SY2)

SUBROUTINE PLOT(----) } (CALCOMP)

SUBROUTINE SYMBOL(----) } (CALCOMP)

Input

AINCH	The scale in inches. From call sequence.
NRW	The number of runways to be plotted. From common NN.

XLM1(L,I)	L = 1,2,3; I = 1,...,NRW. The coordinates of the starting point of each runway. From common AA.
XLM2(L,I)	L = 1,2,3; I = 1,...,NRW. The coordinates of the end point of each runway. From common AA.
RYNAM(I)	I = 1,...,NRW. The runway name of each runway. From common RYNME.

Output

The plotted runway legend.

SUBROUTINE RUNWAY(X1,Y1,X2,Y2)

Purpose

Subroutine RUNWAY plots a runway.

Subroutines Called

SUBROUTINE PLOTA(o.,DELY,3)

SUBROUTINE PLOT(----) (CALCOMP)

Input

X1 Y1	The coordinates of the starting point of the runway to be plotted. From call sequence.
X2 Y2	The coordinates of the end point of the runway to be plotted. From call sequence.

Output

A plot on the runway.

SUBROUTINE HEAD(NUM)

Purpose

Subroutine HEAD plots a detailed description of the flight models if there are not more than ten flights.

Subroutines Called

SUBROUTINE LBTOKG(FLIGHT(2,I),1)

SUBROUTINE FTTOM(FLIGHT(JI + L),1)

SUBROUTINE SYMBOL(----)

SUBROUTINE NUMBER(----) } (CALCOMP)

SUBROUTINE PLOT(----)

Input

NUM	The total number of flights for the airport. From call sequence.
NFT(I)	I = 1,...,NRW. The number of flights for each runway.
FLIGHT(I,S)	I = 1,...,22; J = 1,...,20. The array of flight information to be plotted. From common TROUBL.

Output

Plotted output.

SUBROUTINE AXIS2(XD,YD,AMAX,AMIN,DELX,AINCH,BCD,NCH,NDEC DELN)

Purpose

Subroutine AXIS2 is similar to the CALCOMP subroutine AXIS except that it is used when data are to be scaled to the axis instead of scaling the axis to the data when plotting.

Subroutines Called

SUBROUTINE PLOT(----)

SUBROUTINE NUMBER(----) } (CALCOMP)

SUBROUTINE SYMBOL(----)

Input

XD YD	The coordinates of the axis origin. From call sequence.
AMAX AMIN	The maximum and minimum values for the axis. From call sequence.
DELX	The increment along the axis at which tic marks and labeling will occur. From call sequence.
AINCH	The length of the axis in inches. If AINCH is negative, a horizontal axis is plotted. If AINCH is positive, a vertical axis is plotted. From call sequence.
NCH	The number of characters in the axis title. From call sequence.
BCD	The array containing the axis title. From call sequence.
NDEC	The number of decimal places in the numerical tic mark labels. If NDEC is -1, no decimal point is drawn and no decimal places are assumed. From call sequence.

Output

DELN	The scale factor used in scaling the data to the axis. Passes through call sequence.

Plotted titled and labeled axis.

APPENDIX E

INPUT LISTING

```
TITLEF
 SAMPLE AIRPORT 1975
MKSSYS
WECPNL
PRTALL
CLCMPS
NOISIN
RUNWAY                                          8
XCOORD          0.0000       3000.000
YCOORD          0.0000          0.000
ZCOORD          0.0000          0.000
TKGDRL          0.0000
TCHDWN        300.0000
NOFLTS         10.
FLIGHT                                          8 G 154
LANDNG
ACTYPE        154.0000
ACWGHT      60000.0000
THRUST         42.0000
NDAYOP         12.
NEVNOP          2.
NNGTOP          2.
NOEXTS          2.0000
CLIMBA          3.0000
PRSEG1       1200.0000
PRSEG2       7100.0000
EXTNSN
EXTENT        170.0000
ELEVAT          3.0000
THRUST         42.0000
RADIUS      -4000.0000
EXTNSN
EXTENT          0.0000
ELEVAT          3.0000
THRUST         42.0000
RADIUS          0.0000
FLIGHT                                          8 G 151
LANDNG
ACTYPE        151.
ACWGHT      65000.
THRUST         35.4
NDAYOP          8.
NEVNOP          2.
NNGTOP          6.
NOEXTS          2.
CLIMBA          3.
PRSEG1       1200.
PRSEG2       7100.
EXTNSN
EXTENT        170.
ELEVAT          3.
THRUST         35.4
RADIUS      -4000.
EXTNSN
EXTENT          0.
```

```
ELEVAT          3.
THRUST          35.4
RADIUS          0.
FLIGHT                                  8 A 154
LANDNG
ACTYPE          154.0000
ACWGHT          60000.0000
THRUST          42.0000
NDAYOP          26.4
NEVNOP          7.6
NNGTOP          11.
NOEXTS          0.0000
CLIMBA          3.0000
PRSEG1          1200.0000
FLIGHT                                  8 A 161
LANDNG
ACTYPE          161.0000
ACWGHT          43000.0000
THRUST          40.0000
NDAYOP          31.0000
NEVNOP          7.0000
NNGTOP          9.0000
NOEXTS          0.0000
CLIMBA          3.0000
PRSEG1          1200.0000
FLIGHT                                  8 B 154
TAKOFF
ACTYPE          154.0000
ACWGHT          60000.0000
THRUST          100.0000
NDAYOP          27.2
NEVNOP          6.8
NNGTOP          6.
NOEXTS          2.0000
CLIMBA          8.9200
PRSEG1          2000.0000
PRSEG2          15000.0000
EXTNSN
EXTENT          10.0000
ELEVAT          8.9200
THRUST          100.0000
RADIUS          3000.0000
EXTNSN
EXTENT          0.0000
ELEVAT          8.9200
THRUST          100.0000
RADIUS          0.0000
FLIGHT                                  8 B 151
TAKOFF
ACTYPE          151.0000
ACWGHT          65000.0000
THRUST          100.0000
NDAYOP          7.
NEVNOP          1.
NNGTOP          3.
```

```
NOEXTS        2.0000
CLIMBA        8.3700
PRSEG1     1600.0000
PRSEG2    15400.0000
EXTNSN
EXTENT       10.0000
ELEVAT        8.3700
THRUST      100.0000
RADIUS     3000.0000
EXTNSN
EXTENT        0.0000
ELEVAT        8.3700
THRUST      100.0000
RADIUS        0.0000
FLIGHT                          8 P 161
TAKOFF
ACTYPE      161.0000
ACWGHT    43000.0000
THRUST      100.0000
NDAYOP       23.0000
NEVNOP        5.0000
NNGTOP        2.0000
NOEXTS        2.0000
CLIMBA        8.7500
PRSEG1     1525.0000
PRSEG2    15475.0000
EXTNSN
EXTENT       10.0000
ELEVAT        8.7500
THRUST      100.0000
RADIUS     3000.0000
EXTNSN
EXTENT        0.0000
ELEVAT        8.7500
THRUST      100.0000
RADIUS        0.0000
FLIGHT                          8 C 154
TAKOFF
ACTYPE      154.0000
ACWGHT    60000.0000
THRUST      100.0000
NDAYOP        9.4
NEVNOP        4.6
NNGTOP        7.
NOEXTS        2.0000
CLIMBA        8.9200
PRSEG1     2000.0000
PRSEG2    10000.0000
EXTNSN
EXTENT      170.0000
ELEVAT        8.9200
THRUST      100.0000
RADIUS    -3000.0000
EXTNSN
EXTENT        0.0000
```

```
ELEVAT        8.9200
THRUST      100.0000
RADIUS        0.0000
FLIGHT                                     8 C 151
TAKOFF
ACTYPE      151.0000
ACWGHT    65000.0000
THRUST      100.0000
NDAYOP        4.
NEVNOP        1.
NNGTOP        0.
NOEXTS        2.0000
CLIMBA        8.3700
PRSEG1     1600.0000
PRSEG2    10400.0000
EXTNSN
EXTENT      170.0000
ELEVAT        8.3700
THRUST      100.0000
RADIUS    -3000.0000
EXTNSN
EXTENT        0.0000
ELEVAT        8.3700
THRUST      100.0000
RADIUS        0.0000
FLIGHT                                     8 C 161
TAKOFF
ACTYPE      161.0000
ACWGHT    43000.0000
THRUST      100.0000
NDAYOP       10.0000
NEVNOP        3.0000
NNGTOP        4.0000
NOEXTS        2.0000
CLIMBA        8.7500
PRSEG1     1525.0000
PRSEG2    10475.0000
EXTNSN
EXTENT      170.0000
ELEVAT        8.7500
THRUST      100.0000
RADIUS    -3000.0000
EXTNSN
EXTENT        0.0000
ELEVAT        8.7500
THRUST      100.0000
RADIUS        0.0000
RUNWAY                                     21
XCOORD     3000.0000        500.000
YCOORD     1000.0000      -2000.000
ZCOORD        0.0000          0.000
TKGDRL        0.0000
TCHDWN      300.0000
NOFLTS        6.0000
FLIGHT                                     21 D 111
```

```
LANDNG
ACTYPE        111.0000
ACWGHT     350000.0000
THRUST         60.7000
NDAYOP          6.0000
NEVNOP          1.0000
NNGTOP          2.0000
NOEXTS          0.0000
CLIMBA          3.0000
PRSEG1       2000.0000
FLIGHT                              21 D 134
LANDNG
ACTYPE        134.0000
ACWGHT     100000.0000
THRUST         45.0000
NDAYOP         31.0000
NEVNOP          7.0000
NNGTOP         21.0000
NOEXTS          0.0000
CLIMBA          3.0000
PRSEG1       1500.0000
FLIGHT                              21 E 111
TAKOFF
ACTYPE        111.0000
ACWGHT     350000.0000
THRUST        100.0000
NDAYOP          5.0000
NEVNOP          1.0000
NNGTOP          1.0000
NOEXTS          2.0000
CLIMBA          4.9200
PRSEG1       3350.0000
PRSEG2       8650.0000
EXTNSN
EXTENT         60.0000
ELEVAT          4.9200
THRUST        100.0000
RADIUS       4000.0000
EXTNSN
EXTENT          0.0000
ELEVAT          4.9200
THRUST        100.0000
RADIUS          0.0000
FLIGHT                              21 E 134
TAKOFF
ACTYPE        134.0000
ACWGHT     100000.0000
THRUST        100.0000
NDAYOP         19.2000
NEVNOP          4.8000
NNGTOP         11.0000
NOEXTS          2.0000
CLIMBA         14.7300
PRSEG1       1300.0000
PRSEG2      10700.0000
```

EXTNSN		
EXTENT	60.0000	
ELEVAT	14.7300	
THRUST	100.0000	
RADIUS	4000.0000	
EXTNSN		
EXTENT	0.0000	
ELEVAT	14.7300	
THRUST	100.0000	
RADIUS	0.0000	
FLIGHT		21 F 111
TAKOFF		
ACTYPE	111.0000	
ACWGHT	350000.0000	
THRUST	100.0000	
NDAYOP	1.6000	
NEVNOP	0.4000	
NNGTOP	0.0000	
NOFXTS	2.0000	
CLIMBA	4.9200	
PRSEG1	3350.0000	
PRSEG2	8650.0000	
EXTNSN		
EXTENT	120.0000	
ELEVAT	4.9200	
THRUST	100.0000	
RADIUS	-4000.0000	
EXTNSN		
EXTENT	0.0000	
ELEVAT	4.9200	
THRUST	100.0000	
RADIUS	0.0000	
FLIGHT		21 F 134
TAKOFF		
ACTYPE	134.0000	
ACWGHT	100000.0000	
THRUST	100.0000	
NDAYOP	15.0000	
NEVNOP	4.0000	
NNGTOP	5.0000	
NOFXTS	2.0000	
CLIMBA	14.7300	
PRSEG1	1300.0000	
PRSEG2	10700.0000	
EXTNSN		
EXTENT	120.0000	
ELEVAT	14.7300	
THRUST	100.0000	
RADIUS	-4000.0000	
EXTNSN		
EXTENT	0.0000	
ELEVAT	14.7300	
THRUST	100.0000	
RADIUS	0.0000	
GRIDCL		

```
DELTAX        3000.
DELTAY       -3000.
NOOFXS          20.
NOOFYS          18.
FIRSTX      -32500.
FIRSTY       25500.
PROCES
              5
           111           2
 100.         0.           1.
              6
 400.        700.        1000.       4000.       10000.      50000.
 116.9       113.1       110.3       94.7        84.7        66.9
 60.7        1.          1.
              6
 400.        700.        1000.       4000.       10000.      50000.
 105.9       102.1       99.1        83.3        70.9        48.5
           134           2
 100.        0.          1.
              5
 200.        400.        1000.       10000.      50000.
 126.2       124.8       120.5       102.7       89.8
 45.0        1.          1.
              5
 200.        400.        1000.       10000.      50000.
 116.5       114.8       108.7       88.4        74.I
           151           2
 100.        0.          1.
              6
 400.        1000.       2000.       4000.       10000.      50000.
 116.7       110.2       105.0       99.2        91.6        78.4
 35.4        1.          1.
              6
 400.        1000.       2000.       4000.       10000.      40000.
 104.7       96.2        88.6        79.9        67.8        50.0
           154           2
 100.        0.          1.
              5
 200.        400.        1000.       10000.      50000.
 123.5       118.8       112.6       99.0        87.1
 42.0        1.          1.
              5
 200.        400.        1000.       10000.      50000.
 114.1       110.2       105.0       92.5        83.5
           161           2
 100.        0.          1.
              6
 250.        500.        1000.       2000.       8000.       50000.
 119.5       114.4       109.0       103.0       86.0        60.0
 40.0        1.          1.
              6
 250.        500.        1000.       2000.       8000.       50000.
 112.2       107.0       100.3       93.0        75.0        51.4
 8            21
 ENDRUN
 80.         85.         90.
```

APPENDIX F

OUTPUT

SUBROUTINE DSTORE

POSITION VECTORS OF END POINTS OF SEGMENTS, UNIT VECTORS OF SEGMENTS, AND POSITION VECTORS OF HELIX CENTERS

RUNWAY 1 FLIGHT 1 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	42.0	1909.346	0.0	0.0	-1.000	0.0	0.0			
2	42.0	299.999	0.0	0.0	-0.999	0.0	0.052			
3	42.0	-6799.996	0.0	372.094	0.983	-0.173	0.052	-6799.996	-3999.998	372.094
4	42.0	-7494.590	-7939.227	994.082	0.983	-0.173	0.052			

RUNWAY 1 FLIGHT 2 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	35.4	1909.346	0.0	0.0	-1.000	0.0	0.0			
2	35.4	299.999	0.0	0.0	-0.999	0.0	0.052			
3	35.4	-6799.996	0.0	372.094	0.983	-0.173	0.052	-6799.996	-3999.998	372.094
4	35.4	-7494.590	-7939.227	994.082	0.983	-0.173	0.052			

RUNWAY 1 FLIGHT 3 NO. OF EXTENSIONS 0

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	42.0	1909.346	0.0	0.0	-1.000	0.0	0.0			
2	42.0	299.999	0.0	0.0	-0.999	0.0	0.052			

RUNWAY 1 FLIGHT 4 NO. OF EXTENSIONS 0

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	40.0	1909.346	0.0	0.0	-1.000	0.0	0.0			
2	40.0	299.999	0.0	0.0	-0.999	0.0	0.052			

RUNWAY 1 FLIGHT 5 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1999.999	0.0	0.0	0.988	0.0	0.155			
3	100.0	16999.996	0.0	2354.299	0.973	-0.172	0.155	16999.996	-2999.998	2354.299
4	100.0	17520.938	-45.576	2436.478	0.973	-0.172	0.155			

RUNWAY 1 FLIGHT 6 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1600.000	0.0	0.0	0.989	0.0	0.146			
3	100.0	16999.996	0.0	2265.831	0.974	-0.172	0.146	16999.996	-2999.998	2265.831
4	100.0	17520.941	-45.576	2342.888	0.974	-0.172	0.146			

RUNWAY 1 FLIGHT 7 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1525.000	0.0	0.0	0.988	0.0	0.152			
3	100.0	16999.996	0.0	2381.827	0.973	-0.172	0.152	16999.996	-2999.998	2381.827
4	100.0	17520.941	-45.576	2462.416	0.973	-0.172	0.152			

RUNWAY 1 FLIGHT 8 NO. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1999.999	0.0	0.0	0.988	0.0	0.155			
3	100.0	11999.996	0.0	1569.532	-0.973	0.172	0.155	11999.996	2999.998	1569.532
4	100.0	12520.941	5954.418	2966.602	-0.973	0.172	0.155			

RUNWAY 1 FLIGHT 9 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1600.000	0.0	0.0	0.989	0.0	0.146			
3	100.0	11999.996	0.0	1530.172	-0.974	0.172	0.146	11999.996	2999.998	1530.172
4	100.0	12520.941	5954.418	2835.819	-0.974	0.172	0.146			

RUNWAY 1 FLIGHT 10 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	0.0	0.0	0.0	1.000	0.0	0.0			
2	100.0	1525.000	0.0	0.0	0.988	0.0	0.152			
3	100.0	11999.996	0.0	1612.255	-0.973	0.172	0.152	11999.996	2999.998	1612.255
4	100.0	12520.941	5954.418	2982.275	-0.973	0.172	0.152			

RUNWAY 2 FLIGHT 1 NC. OF EXTENSIONS 0

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	60.7	1777.665	-466.799	0.0	0.640	0.768	0.0			
2	60.7	2807.943	769.535	0.0	0.639	0.767	0.052			

RUNWAY 2 FLIGHT 2 NC. OF EXTENSIONS 0

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	45.0	1777.665	-466.799	0.0	0.640	0.768	0.0			
2	45.0	2807.943	769.535	0.0	0.639	0.767	0.052			

RUNWAY 2 FLIGHT 3 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	2999.999	1000.000	0.0	-0.640	-0.768	0.0			
2	100.0	855.382	-1573.539	0.0	-0.638	-0.765	0.086			
3	100.0	-4682.207	-8218.648	744.606	-0.982	0.170	0.086	-7755.090	-5657.914	744.606
4	100.0	-8436.313	-9999.477	1105.185	-0.982	0.170	0.086			

RUNWAY 2 FLIGHT 4 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	2999.999	1000.000	0.0	-0.640	-0.768	0.0			
2	100.0	2167.759	1.313	0.0	-0.619	-0.743	0.254			
3	100.0	-4682.211	-8218.648	2813.079	-0.953	0.165	0.254	-7755.090	-5657.914	2813.079
4	100.0	-8436.313	-9999.477	3914.330	-0.953	0.165	0.254			

RUNWAY 2 FLIGHT 5 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	2999.999	1000.000	0.0	-0.640	-0.768	0.0			
2	100.0	855.382	-1573.539	0.0	-0.638	-0.765	0.086			
3	100.0	-4682.207	-8218.648	744.606	0.982	-0.170	0.086	-1609.327	-10779.387	744.606
4	100.0	-2290.549	-14720.949	1465.763	0.982	-0.170	0.086			

RUNWAY 2 FLIGHT 6 NC. OF EXTENSIONS 2

SEG	THRUST	GAMMA X	GAMMA Y	GAMMA Z	UNIT X	UNIT Y	UNIT Z	HELIX CEN X	HELIX CEN Y	HELIX CEN Z
1	100.0	2999.999	1000.000	0.0	-0.640	-0.768	0.0			
2	100.0	2167.759	1.313	0.0	-0.619	-0.743	0.254			
3	100.0	-4682.211	-8218.648	2813.079	0.953	-0.165	0.254	-1609.329	-10779.387	2813.079
4	100.0	-2290.553	-14720.949	5015.562	0.953	-0.165	0.254			

SUBROUTINE SUMRF

```
**************************************************
* SUMMARY OF RUNWAYS AND ASSOCIATED FLIGHTS *
*   (INCLUDING AIRCRAFT TYPE NUMBERS AND    *
*        LANDING/TAKEOFF INFORMATION)       *
**************************************************

   FLIGHTS 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20     TOTAL
RUNWAYS
8       L154 L151 L154 L161 T154 T151 T161 T154 T151 T161                                                      10

21      L111 L134 T111 T134 T111 T134                                                                           6
                                                                                                             -----
                                                                                                        FLIGHTS 16
```

SUBROUTINE POINTS

NE/WECPNL/NEF VALUES IN A GRID

Y VALS \ X VALS	-32500.	-29500.	-26500.	-23500.	-20500.	-17500.	-14500.	-11500.	-8500.	-5500.	-2500.	500.	3500.	6500.	9500.	12500.	15500.	18500.	21500.	24500.
25500.	67.1	66.9	66.8	66.8	66.7	61.3	67.9	67.8	67.8	67.8	67.7	67.5	67.7	67.9	68.5	69.2	70.1	71.5	76.4	77.8
22500.	67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.0	69.0	69.0	68.9	68.8	68.9	69.2	70.0	70.8	71.2	76.0	79.3	74.3
19500.	68.9	68.7	68.6	68.5	68.3	62.9	70.5	70.4	70.4	70.4	70.3	70.2	70.3	70.7	71.6	72.9	76.1	81.1	75.6	72.3
16500.	70.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0	72.0	72.0	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6	72.0
13500.	71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6	73.8	73.9	73.9	74.1	75.0	76.6	82.9	78.3	75.3	73.6	72.4
10500.	72.7	72.8	72.8	72.8	72.3	75.3	74.5	74.7	74.8	75.3	75.7	76.1	76.6	78.0	82.9	81.8	74.7	75.9	74.5	73.3
7500.	74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6	75.7	76.5	77.3	78.6	82.2	85.0	77.1	79.6	77.5	75.8	74.6
4500.	75.8	76.0	76.2	76.3	76.4	72.6	77.8	77.6	77.3	75.5	73.9	75.8	80.3	89.4	81.6	77.7	82.1	79.7	77.7	76.3
1500.	78.7	79.0	79.3	79.6	79.5	80.5	80.8	81.3	82.0	83.3	83.8	81.7	101.2	87.1	86.0	82.5	84.5	82.1	80.1	78.6
-1500.	79.1	79.4	79.8	80.1	80.6	81.0	81.4	82.0	83.0	84.6	88.0	98.9	89.4	87.0	86.0	82.5	83.4	81.9	80.9	80.1
-4500.	77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2	85.5	91.1	90.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5	78.1
-7500.	76.8	77.4	78.0	78.7	79.7	80.5	81.4	82.7	83.8	87.4	88.4	85.0	82.1	79.1	77.8	76.2	77.3	76.9	76.6	76.1
-10500.	75.9	76.5	77.2	78.0	78.8	79.8	81.1	82.7	84.6	86.1	84.5	82.0	80.2	79.2	78.0	76.6	75.7	76.3	75.9	74.8
-13500.	74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	2.4	83.	82.	80.5	79.3	78.4	77.3	76.4	74.5	74.1	75.7	74.0
-16500.	73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0	80.7	80.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8	73.6
-19500.	72.5	72.9	73.4	73.9	74.5	75.2	75.5	76.8	77.7	78.3	78.3	77.8	77.3	76.8	76.3	75.5	74.7	74.6	74.2	73.3

-2250C.	71.5	71.9	72.3	72.8	73.2	73.5	74.5	75.2	75.8	76.2	76.2	76.0	75.7	75.4	75.1	74.8
	73.8	73.8	73.1	72.7												
-2550C.	70.6	71.0	71.4	71.8	72.2	72.7	73.2	73.5	74.2	74.4	74.4	74.3	74.2	74.0	73.8	73.4
	73.0	72.9	72.3	72.0												

```
FROM CALPLT
XM,XLB,XSF,YM,YC,YHT,XZIN,AINCH,SCMX,T1,T2,XLLFC,YLLFC
 -0.40000E 05   0.40000E 05   0.12500E-03  -0.40000E 05   0.80000E 04   0.50000E 01   0.50000E 01   0.1000CE 02
  0.40000E 05   0.24500E 05  -0.25500E 05  -0.32500E 05   0.25500E 05
NINCH
     10
  67.1   66.9   66.8   66.8   66.7   61.3   67.9   67.8   67.8   67.8
  67.7   67.5   67.7   67.9   68.5   69.2   70.1   71.5   76.4   77.8
  67.9   67.8   67.7   67.5   67.4   62.1   69.1   69.0   69.0   69.0
  68.9   68.8   68.9   69.2   70.0   70.8   71.2   76.0   79.3   74.3
  68.9   68.7   68.6   68.5   68.3   62.9   70.5   70.4   70.4   70.4
  70.3   70.2   70.3   70.7   71.6   72.5   76.1   81.1   75.6   72.3
  70.1   69.9   69.7   69.5   69.4   69.1   71.9   71.9   72.0   72.0
  72.0   71.9   72.1   72.5   73.6   76.1   82.4   76.5   73.6   72.0
  71.6   71.5   71.1   70.8   70.6   74.4   73.4   73.6   73.6   73.8
  73.9   73.9   74.1   75.0   76.6   82.9   78.3   75.3   73.8   72.4
  72.7   72.8   72.8   72.8   72.3   75.3   74.5   74.7   74.8   75.3
  75.7   76.1   76.6   78.0   82.5   81.8   74.7   75.9   74.5   73.3
  74.1   74.2   74.3   74.4   74.4   76.3   76.2   75.4   75.6   75.7
  76.5   77.3   78.6   82.2   85.0   77.1   79.6   77.5   75.8   74.6
  75.8   76.0   76.2   76.3   76.4   77.6   77.8   77.6   77.3   75.5
  73.9   75.8   80.3   89.4   81.6   77.7   82.1   79.7   77.7   76.3
  78.7   79.0   79.3   79.6   79.8   80.5   80.8   81.3   82.0   83.3
  83.8   81.7  101.2   87.1   86.0   82.5   84.5   82.1   80.1   78.6
  79.1   79.4   79.8   80.1   80.6   81.0   81.4   82.0   83.0   84.6
  88.0   99.9   89.4   87.0   86.0   82.5   83.4   81.9   80.9   80.1
  77.5   77.9   78.4   78.8   79.5   80.1   80.6   81.9   82.2   85.5
  91.1   90.2   85.3   81.4   80.2   77.9   79.2   78.5   78.5   78.1
  76.8   77.4   78.0   78.7   79.7   80.5   81.4   82.7   83.8   87.4
  88.4   85.0   82.1   79.1   77.8   76.2   77.3   76.9   76.6   76.1
  75.9   76.5   77.2   78.0   78.8   79.8   81.1   82.7   84.6   86.1
  94.5   87.0   80.2   79.2   78.0   76.6   75.7   76.3   75.9   74.8
  74.0   75.4   75.9   76.6   77.4   78.3   79.5   80.8   [illegible]   83.3
  82.1   80.5   79.3   78.4   77.3   76.4   76.5   76.1   75.1   74.0
  73.7   74.1   74.6   75.2   75.9   76.7   77.6   78.7   80.0   80.7
  80.4   79.5   78.6   77.8   77.1   75.8   75.3   75.2   74.8   73.6
  72.5   72.9   73.4   73.9   74.5   75.2   75.9   76.8   77.7   78.3
  79.3   77.9   77.3   75.8   74.3   75.5   74.7   74.6   74.2   73.2
  71.5   71.9   72.3   72.8   73.3   73.9   74.5   75.2   75.8   76.2
  76.2   76.0   75.7   75.4   75.1   74.8   73.8   73.8   73.1   72.7
  70.6   71.0   71.4   71.8   72.2   72.7   73.2   73.9   74.2   74.4
  74.4   74.3   74.2   74.0   73.8   73.4   73.0   72.9   72.3   72.0
```

0.300000E 04 -0.300000E 04

0.537500E 01 0.512500E 01

20 18

-0.325000E 05 0.255000E 05 0.175000E-03 0.500000E 01 0.500000E 01

66.4	66.2	66.1	65.9	66.0	66.0	60.6	66.8	66.7	66.6
66.6	66.5	66.3	65.5	66.7	67.1	67.7	69.0	67.0	73.5
81.3	84.8								
67.2	67.1	66.9	66.8	66.8	66.7	61.3	67.9	67.8	67.8
67.8	67.7	67.5	67.7	67.9	68.5	69.2	70.1	71.5	76.4
77.8	79.1								
68.0	67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.0	69.0
69.0	68.9	68.3	68.9	69.2	70.0	70.8	71.2	76.0	79.3
74.3	69.2								
69.0	68.9	68.7	68.6	68.5	68.3	62.9	70.5	70.4	70.4
70.4	70.3	70.2	70.3	70.7	71.6	72.9	76.1	81.1	75.6
72.3	69.0								
70.3	70.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0
72.0	72.0	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6
72.0	70.3								
71.7	71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6
73.8	73.9	73.9	74.1	75.0	76.6	82.9	78.3	75.3	73.6
72.4	71.3								
72.7	72.7	72.8	72.8	72.8	72.3	75.3	74.5	74.7	74.8
75.3	75.7	76.1	76.6	78.0	82.5	81.8	74.7	75.9	74.5
73.3	72.2								
74.0	74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6
75.7	76.5	77.3	78.6	82.2	85.0	77.1	79.6	77.5	75.8
74.6	73.4								
75.7	75.8	76.0	76.2	76.3	76.4	77.6	77.8	77.6	77.3
75.5	73.9	75.9	80.3	89.4	81.6	77.7	82.1	79.7	77.7
76.3	74.9								
78.4	78.7	79.0	79.3	79.6	79.8	80.5	80.8	81.3	82.0
83.3	83.8	81.7	101.2	87.1	86.0	82.5	84.5	82.1	80.1
78.6	77.1								
78.7	79.1	79.4	79.9	80.1	80.6	81.0	81.4	82.0	83.0
84.6	88.0	98.9	99.4	97.0	86.0	82.5	83.4	91.9	80.9
80.1	79.3								
77.1	77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2
85.5	91.1	90.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5
78.1	77.9								
76.2	76.8	77.4	78.0	78.7	79.7	80.5	81.4	82.7	83.9
87.4	88.4	95.0	92.1	79.1	77.8	76.2	77.3	76.9	76.6
76.1	75.5								
75.3	75.9	76.5	77.2	78.0	78.8	79.8	81.1	82.7	84.6
86.1	84.5	82.0	80.2	79.2	78.0	76.6	75.7	76.3	75.9

74.8	73.7								
74.4	74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	82.4
83.3	82.1	80.5	79.3	78.4	77.3	76.4	76.5	76.1	75.7
74.0	72.2								
73.2	73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0
80.7	80.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8
73.6	72.3								
72.1	72.5	72.9	73.4	73.9	74.5	75.2	75.9	76.8	77.7
78.3	78.3	77.3	77.3	76.8	76.3	75.5	74.7	74.6	74.2
73.2	72.3								
71.1	71.5	71.9	72.3	72.9	73.3	73.9	74.5	75.2	75.8
76.2	76.2	76.0	75.7	75.4	75.1	74.8	73.5	73.8	73.1
72.7	72.4								
70.3	70.6	71.0	71.4	71.8	72.2	72.7	73.2	73.9	74.2
74.4	74.4	74.3	74.2	74.0	73.8	73.4	73.0	72.9	72.3
72.0	71.7								
69.4	69.7	70.1	70.4	70.8	71.2	71.5	71.9	72.6	72.6
72.7	72.7	72.6	72.7	72.7	72.6	72.1	72.3	72.0	71.5
71.3	71.0								
67.1	66.9	66.8	66.8	64.7	61.3	67.9	67.8	67.8	67.8
67.7	67.5	67.7	67.9	68.5	69.2	70.1	71.9	76.4	77.8
67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.0	69.0	69.0
68.9	68.8	68.9	69.2	70.0	70.8	71.2	76.0	79.2	74.3
68.9	68.7	68.6	68.5	68.3	62.9	70.5	70.4	70.4	70.4
70.2	70.2	70.3	70.7	71.6	72.9	76.1	81.1	75.6	72.3
70.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0	72.0
72.0	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6	72.0
71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6	73.8
73.9	73.9	74.1	75.0	76.6	82.9	78.3	75.3	73.6	72.4
72.7	72.8	72.8	72.8	72.3	75.3	74.5	74.7	74.8	75.3
75.7	76.1	76.6	73.0	92.5	81.8	74.7	75.9	74.5	73.3
74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6	75.7
76.5	77.3	78.6	83.2	85.0	77.1	79.6	77.5	75.8	74.6
75.8	76.0	76.2	76.3	76.4	77.6	77.8	77.6	77.3	75.5
73.9	75.8	80.3	89.4	81.6	77.7	82.1	79.7	77.7	76.3
78.7	79.0	79.3	79.6	79.8	80.5	80.8	81.3	82.8	83.3
83.8	81.7	101.3	87.1	86.0	82.5	84.5	82.1	80.1	78.6
79.1	79.4	79.8	80.1	90.6	81.0	81.4	82.0	83.0	84.6
88.0	98.9	89.4	87.0	86.0	82.5	83.4	81.9	80.9	80.1
77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2	85.5
91.1	90.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5	78.1
76.8	77.4	78.0	78.7	79.7	80.5	81.4	82.1	83.8	87.4
88.4	95.0	82.1	79.1	77.8	76.2	77.3	76.9	76.6	76.1
75.9	76.5	77.3	78.0	78.5	79.8	81.1	82.7	84.6	86.1
84.5	82.0	80.3	79.2	78.0	76.6	75.7	76.3	75.9	74.8
74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	82.4	83.3
82.1	80.5	79.3	78.4	77.3	76.4	76.5	76.1	75.7	74.0
73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0	80.7
80.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8	73.6
72.5	72.9	73.4	73.9	74.5	75.2	75.9	76.8	77.7	78.3
78.3	77.8	77.3	76.8	76.3	75.5	74.7	74.6	74.2	73.2
71.5	71.9	72.3	72.8	73.3	73.9	74.5	75.2	75.8	76.2
76.2	76.0	75.7	75.4	75.1	74.8	73.8	73.8	73.1	72.7
70.6	71.0	71.4	71.8	72.2	72.7	73.2	73.9	74.2	74.4
74.4	74.3	74.2	74.0	73.8	73.4	73.0	72.9	72.3	72.0

C.200CCE C4 -C.2COCCE C4

C.5375CE C1 0.5125CE C1

20 19

-C.3250CE C5 0.255CCE C5 0.125CCE-C3 C.50CCCE C1 0.5COCOE 01

66.4	66.2	66.1	65.9	56.0	66.0	60.6	56.8	66.7	66.6
66.6	66.5	66.3	66.5	66.7	67.1	67.7	69.0	67.0	73.5
81.3	84.8								
67.2	67.1	66.9	66.8	66.8	66.7	61.3	67.9	67.8	67.8
67.8	67.7	67.5	67.7	67.9	68.5	69.2	70.1	71.5	76.4
77.8	79.1								
68.0	67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.0	69.0
69.0	68.9	68.8	68.9	69.2	70.0	70.8	71.2	76.0	79.3
74.3	69.2								
69.0	68.9	68.7	68.6	68.5	68.3	62.9	70.5	70.4	70.4
70.4	70.3	70.2	70.3	70.7	71.6	72.9	76.1	81.1	75.6
72.3	69.0								
70.3	70.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0
72.0	72.0	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6
72.0	70.3								
71.7	71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6
73.8	73.9	73.9	74.1	75.0	76.6	82.9	78.3	75.3	73.6
72.4	71.3								
72.7	72.7	72.8	72.8	72.8	72.3	75.3	74.5	74.7	74.8
75.3	75.7	76.1	76.6	76.0	92.5	81.8	74.7	75.9	74.5
73.3	72.2								
74.0	74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6
75.7	76.5	77.3	78.6	82.2	85.0	77.1	79.6	77.5	75.8
74.6	73.4								
75.7	75.8	76.0	76.2	76.3	76.4	77.6	77.8	77.6	77.3
75.5	73.9	75.9	80.3	89.4	81.6	77.7	82.1	79.7	77.7
76.3	74.9								
78.4	78.7	79.0	79.3	79.6	79.8	80.5	80.8	81.3	82.0
83.3	83.8	81.7	101.2	87.1	86.0	82.5	84.5	82.1	80.1
79.6	77.1								
78.7	79.1	79.4	79.8	80.1	80.6	81.0	81.4	82.0	83.0
94.6	89.0	98.9	95.4	87.0	86.0	82.5	83.4	81.9	80.9
80.1	79.3								
77.1	77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2
85.5	91.1	90.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5
78.1	77.9								
76.2	76.8	77.4	78.0	78.7	79.7	80.5	81.4	82.7	83.8
97.4	98.4	85.0	82.1	79.1	77.5	74.2	77.3	76.9	76.6
76.1	75.5								
75.3	75.9	76.5	77.2	78.0	78.8	79.8	81.1	82.7	84.6
86.1	84.5	82.0	80.2	79.2	78.0	76.6	75.7	76.3	75.9

74.8	73.7								
74.4	74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	82.4
83.3	82.1	80.5	79.3	78.4	77.3	76.4	76.5	76.1	75.7
74.0	72.2								
73.2	73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0
80.7	80.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8
72.6	72.3								
72.1	72.5	72.9	73.4	73.9	74.5	75.2	75.9	76.8	77.7
73.3	78.3	77.8	77.3	76.8	76.3	75.5	74.7	74.6	74.2
73.2	72.3								
71.1	71.5	71.9	72.3	72.8	73.3	73.9	74.5	75.2	75.8
76.2	76.2	76.0	75.7	75.4	75.1	74.8	73.8	73.8	73.1
72.7	72.4								
70.3	70.6	71.0	71.4	71.8	72.3	72.7	73.2	73.9	74.2
74.4	74.4	74.3	74.2	74.0	73.8	73.4	73.0	72.9	72.3
72.0	71.7								
69.4	69.7	70.1	70.4	70.8	71.2	71.5	71.9	72.6	72.6
72.7	72.7	72.6	72.7	72.7	72.6	72.1	72.3	72.0	71.5
71.3	71.0								
67.1	66.9	66.8	66.8	66.7	61.3	67.9	67.8	67.8	67.8
67.7	67.5	67.7	67.9	68.5	69.2	70.1	71.5	76.4	77.8
67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.0	69.0	69.0
68.9	68.8	68.9	69.2	70.0	70.8	71.2	76.0	79.3	74.3
68.9	68.7	68.6	68.5	68.3	62.5	70.5	70.4	70.4	70.4
70.3	70.2	70.3	70.7	71.6	72.9	76.1	81.1	75.6	72.3
70.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0	72.0
72.0	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6	72.0
71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6	73.8
73.9	73.9	74.1	75.0	76.6	82.9	78.3	79.3	73.6	72.4
72.7	72.8	72.8	72.8	72.3	75.3	74.3	74.7	74.8	73.3
75.7	76.1	76.6	78.0	82.5	81.8	74.7	75.9	74.5	73.3
74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6	75.7
76.5	77.3	78.6	82.2	85.0	77.1	79.6	77.5	75.8	74.6
75.8	76.0	76.2	76.3	76.4	77.6	77.8	77.6	77.2	75.5
73.9	75.8	80.3	89.4	81.6	77.7	82.1	79.7	77.7	76.3
78.7	79.0	79.3	79.6	79.8	80.5	80.8	81.3	82.0	83.2
83.8	81.7	101.[illegible]	87.1	86.0	82.5	84.5	82.1	80.1	78.6
79.1	79.4	79.8	80.1	80.6	81.0	81.4	82.0	83.0	84.6
98.0	98.9	89.4	87.0	86.0	82.5	83.4	81.9	80.9	80.1
77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2	85.5
91.1	90.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5	78.1
76.8	77.4	78.0	78.7	79.7	80.5	81.4	82.7	83.8	87.4
88.4	85.0	82.1	79.1	77.8	76.2	77.3	76.9	76.6	76.1
75.9	74.5	77.2	78.0	75.8	75.8	81.1	82.7	84.6	86.1
84.5	83.0	80.2	79.2	76.0	76.6	75.7	76.3	75.9	74.8
74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	82.4	83.3
82.1	80.5	79.3	78.4	77.3	76.4	76.5	76.3	75.7	74.0
73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0	80.7
80.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8	73.6
73.5	72.9	73.4	73.9	74.5	75.2	75.9	76.8	77.7	78.3
78.3	77.8	77.3	76.8	76.3	75.5	74.7	74.6	74.2	73.2
71.5	71.9	72.3	72.8	73.3	73.9	74.5	75.2	75.8	76.2
76.2	76.0	75.7	75.4	75.1	74.8	73.8	73.8	73.1	72.7
70.6	71.0	71.4	71.8	72.2	72.7	73.2	73.9	74.2	74.4
74.4	74.3	74.3	74.0	73.8	73.4	73.0	72.9	72.3	72.0

C.30OCCE 04 -C.3CCCCF C4

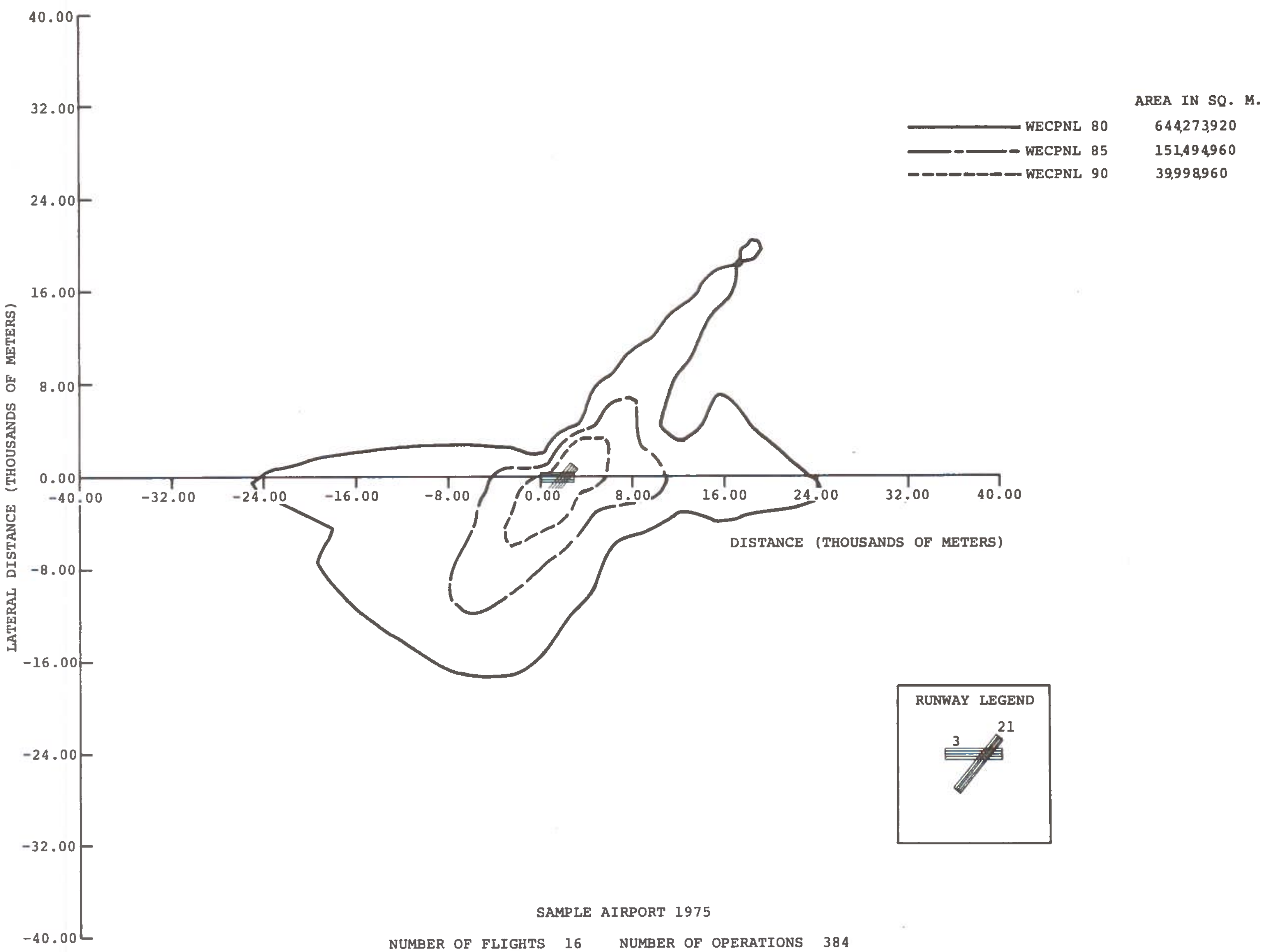
C.5375CE C1 0.5125CE C1

3C 18

-C.32500F 05 0.255CCE CF 0.125CCE-C3 C.500COF C1 C.5CCCOE 01

66.4	66.2	66.1	65.9	66.C	66.0	6C.6	66.8	66.7	66.6
66.6	66.5	66.3	66.5	66.7	67.1	67.7	69.C	67.C	73.5
81.3	84.8								
67.2	67.1	66.9	66.8	66.8	66.7	61.3	67.9	67.8	67.8
67.8	67.7	67.5	67.7	67.9	68.5	69.2	76.1	71.5	76.4
77.6	79.1								
64.0	67.9	67.8	67.7	67.5	67.4	62.1	69.1	69.C	69.0
69.C	68.9	68.9	68.9	69.2	7C.C	7C.8	71.2	76.C	79.3
74.3	69.2								
69.C	68.9	68.7	68.6	68.5	68.3	62.9	70.5	70.4	70.4
70.4	70.3	70.2	70.3	7C.7	71.6	72.9	76.1	81.1	75.6
72.3	69.C								
7C.3	7C.1	69.9	69.7	69.5	69.4	69.1	71.9	71.9	72.0
72.C	72.C	71.9	72.1	72.5	73.6	76.1	82.4	76.5	73.6
72.0	70.3								
71.7	71.6	71.5	71.1	70.8	70.6	74.4	73.4	73.6	73.6
73.8	73.9	73.9	74.1	75.0	76.6	82.9	78.3	75.3	73.6
72.4	71.3								
72.7	72.7	72.9	72.8	72.8	72.3	75.3	74.5	74.7	74.8
75.3	75.7	76.1	76.6	78.C	82.5	81.8	74.7	75.9	74.5
73.3	72.2								
74.C	74.1	74.2	74.3	74.4	74.4	76.3	76.2	75.4	75.6
75.7	76.5	77.3	78.6	82.2	85.0	73.1	78.6	77.5	75.8
74.6	73.4								
75.7	75.8	76.C	76.2	74.7	76.4	77.6	77.8	77.6	77.3
75.5	74.5	75.8	8G.3	85.4	81.6	77.7	87.1	79.7	77.7
76.3	74.9								
78.4	78.7	79.0	79.3	79.6	79.8	80.5	80.8	81.3	82.0
83.7	83.8	81.7	1C1.2	87.1	86.0	82.5	84.5	82.1	80.1
75.6	77.1								
79.7	79.1	79.4	79.8	8C.1	80.6	81.0	81.4	82.C	83.0
84.6	95.C	94.9	85.4	87.C	86.C	82.5	83.4	81.9	8C.9
8C.1	79.3								
77.1	77.5	77.9	78.4	78.8	79.5	80.1	80.6	81.9	82.2
85.5	91.1	9C.2	85.3	81.4	80.2	77.5	79.2	78.5	78.5
78.1	77.3								
74.3	76.5	77.4	78.0	78.7	79.7	8C.5	81.4	82.7	83.8
97.4	88.4	85.0	93.1	79.1	77.8	76.7	77.3	76.9	76.6

76.1	75.5									
75.[illegible]	75.9	76.5	77.2	78.0	78.8	79.8	81.1	82.7	84.6	
86.1	84.5	82.0	80.2	79.2	78.0	76.6	75.7	76.3	75.9	
74.8	73.7									
74.4	74.9	75.4	75.9	76.6	77.4	78.3	79.5	80.8	82.4	
93.3	87.1	80.5	79.3	78.4	77.3	76.4	76.5	76.1	75.7	
74.0	72.2									
73.2	73.7	74.1	74.6	75.2	75.9	76.7	77.6	78.7	80.0	
80.7	90.4	79.5	78.6	77.8	77.1	75.8	75.3	75.2	74.8	
73.6	72.3									
72.1	72.5	72.9	73.4	73.9	74.5	75.2	75.9	76.8	77.7	
78.3	78.3	77.8	77.3	76.8	76.3	75.5	74.7	74.6	74.2	
73.7	72.3									
71.1	71.5	71.9	72.3	72.8	73.3	73.9	74.5	75.2	75.8	
74.2	74.2	74.0	75.7	75.4	75.1	74.8	73.8	73.8	73.1	
73.7	73.4									
70.3	70.6	71.0	71.4	71.9	72.3	72.7	73.2	73.9	74.2	
74.4	74.4	74.3	74.2	74.0	73.8	73.4	73.0	72.9	72.3	
72.0	71.7									
69.4	69.7	70.1	70.4	70.8	71.2	71.5	71.9	72.0	72.6	
72.7	72.7	72.6	72.7	72.7	72.6	72.1	[illegible]	[illegible]	[illegible]	
71.[illegible]	71.0									



SAMPLE AIRPORT 1975

NUMBER OF FLIGHTS 16 NUMBER OF OPERATIONS 384