```
C
C
C     TRANSIENT-STATE SIMULATION PROGRAM FOR MINE VENTILATION
C                   UNDER THE INFLUENCE OF FIRES
C
C                     VERSION OF DECEMBER 1994
C                             (PART 2)
C                           Version 1.30
C
      INCLUDE 'CMMN1.DAT'
C
      CHARACTER*50 MD1,OPTION
C
C
      OPEN (10,FILE='MFCTL1',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (10) NETWW
      IF (NETWW.EQ.1) GO TO 310
      OPEN (21,FILE='MFCTL2',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (21) OPTION
      READ (21) MD1,TACC
      IF (OPTION.EQ.'DONE') GO TO 310
      WRITE (6,760)
      OPEN (20,FILE='MFCTL3',STATUS='UNKNOWN',FORM='UNFORMATTED')
      READ (20) NB,NJ,NFNUM,NETW,NTEMP,ITN,DR,TR,TINC,SPAN,IOUT,TOUT,
     .          CONCT,NSTOP
      IF (NSTOP.GE.1) GO TO 310
      READ (20) NAV,MAXJ,INFLOW,CRITSM,CRITGS,CRITHT,MNO
      READ (20) TIME,TSTART,JSTART,NSFLOW,WRNPR,WRNGS,WRNSM,WRNHT
      READ (20) MARKX,MARKY,NCOMT1,NCOMTS,NCOMT2,LLINE
      READ (20) JAN,JAJ
      DO 3200 I=1,NB
         READ (20) NO(I),JS(I),JF(I),NWTYP(I),R(I),Q(I),KF(I),LA(I)
         READ (20) A(I),O(I),RSTD(I),DZRD(I),FRNVP(I),NREV(I),RDCH4(I)
         READ (20) RDPROP(I),TRD(I),TJS(I),FFRNVP(I),NNREV(I),JSB(I)
         READ (20) TROCK(I),TMRD(I),HA(I),HK(I)
         READ (20) XNEW(I),DCOAGE(I),CH4V(I),CH4PA(I)
         READ (20) NGIN(I),MMIN(I),NGOUT(I),LOUT(I)
         IF (CONCT.EQ.1) THEN
            QQQ(I)=Q(I)
            TRDD(I)=TRD(I)
            TJSS(I)=TJS(I)
         ENDIF
3200  CONTINUE
      DO 3202 I=1,IMX
         READ (20) NOF(I),NFREG(I),RGRAD(I),NFCW(I),MPTS(I),
     .             NSKP(I),NEGQ(I)
         READ (20) NCENT(I),CONT(I),CONC(I),HEAT(I),O2MIN(I),SMPO2(I),
     .              HTPO2(I),TPR(I),QCENT(I)
         DO 3201 J=1,IMY
            READ (20) QF(I,J),PF(I,J)
            DO 3205 K=1,4
               READ (20) FKQ(I,J,K)
3205        CONTINUE
3201     CONTINUE
3202  CONTINUE
      do 3215 i=1,nfnum
         read (20) ncof(i),nd(i),nswt(i)
         ii=ncof(i)
         do 3215 j=1,ii
            read (20) cof(i,j)
3215  continue
      READ (20) NPLOT
      DO 3203 I=1,NJ
         READ (20) JNO(I),T(I),Z(I),CH4C(I),JNOL(I),PROP(I),
     .             PRCH4(I),JLR(I)
         IF (CONCT.EQ.1) TTT(I)=T(I)
3203  CONTINUE
      DO 3204 K=1,JAJ
         READ (20) JSTAR(K), TSTAR(K)
3204  CONTINUE
      DO 3206 I=1,MNO
         READ (20) MEND(I),FNVP(I)
3206  CONTINUE
      IYY=MEND(MNO)
      DO 3207 I=1,IYY
         READ (20) MSL(I)
3207  CONTINUE
      DO 3209 I=1,50
         READ (20) ITITLE(I)
         DO 3208 J=1,20
            READ (20) TITLE(I,J)
3208     CONTINUE
3209  CONTINUE
      DO 3210 I=1,NB
         READ (20) INU(I)
3210  CONTINUE
C
      NSTOP=0
      INIFAN=0
      SSP=SPAN*60.0
      REWIND (21)
      WRITE (21) OPTION
      WRITE (21) MD1,SSP
C
1     OPEN (9,FILE=MD1)
      OPEN (8,FILE='MFCTL4',STATUS='UNKNOWN')
5     IF (LLINE.GT.0) THEN
         READ (9,740) ROW(1)
         LLINE=LLINE-1
         GO TO 5
      ENDIF
C
C     MARKR: 0: OUTPUT WAS GIVEN IN THE LAST INTERVAL.
C            1: OUTPUT WILL BE OMITTED IN THE PRESENT INTERVAL DUE TO
C            INSIGNIFICANCY OF CHANGE. CHANGES IN Q WILL BE CUMULATED.
C     MH: CONTROLLING THE WRITING OF THE HEADLINE FOR THE CONVERGENCE
C         INFORMATION OF OMITTED OUTPUT.
C     ISQC: COUNTER OF AIRWAYS WHICH HAVE THEIR DETAILED DATA RECORDS.
C
      MARKR=0
      TCUMU=0.0
      IF (TOUT.LT.1.E-5) TOUT=1.0
      MH=0
      MARKX=1
      JDP=0
      TACC=0.0
      ISQC=0
      QS=100.0
      QSUM=0.0
      WRITE (8,560)
      WRITE (8,550)
      IF (CONCT.EQ.1) WRITE (8,555)
      CALL INIT (1)
C
C     TINC: USER-SPECIFIED SPAN OF TIME INTERVAL IN SEC..
C     QSTD: A CERTAIN PERCENTAGE OF THE SUM OF ABSOLUTE AIRFLOW CHANGES
C           IN AIRWAYS IN THE FIRST INTERVAL.
C     QS: VARIATION IN SUM OF ABS. AIRFLOW CHANGES IN SUCCESSIVE
C         INTERVALS, CRITERION FOR OUTPUT DURING DYNAMIC SIMULATION.
C     DELTAT: TIME INCREMENT THAT IS ACTUALY ADOPTED IN THE CURRENT
```

```
C             SIMULATION UPDATING.
C
      IF (TINC.LE.0.0) TINC=15.0
      DELTAT=TINC
      DO 290 MASTER=1,1000
         NTACC=0
         IF (MARKR.EQ.0) WRITE (8,560)
         CALL CDCH (NSTOP)
         IF (NSTOP.GT.0) GO TO 310
C
C     MARKP: 0: CONVERGENT RESULTS FOR THE PRESENT INTERVAL HAVE NOT
C            BEEN REACHED.  1: CONVERGENT RESULTS OBTAINED.
C     ITT: COUNTER OF ITERATION IN NETWORK BALANCING.
C     IBB: COUNTER OF ITERATION IN DATA PREPARATION.
C
         MARKP=0
         ITT=0
         IBB=0
         CALL INIT (2)
C
C     MARKS: TIMER FOR DATA HANDLING.
C     MARKQ: 0: NORMAL CONDITION.  1: SIGNIFICANT CONDITION CHANGES
C            IN JUNCTIONS ARE DETECTED, MORE ITERATIONS ARE NEEDED.
C
160      MARKS=0
C
C     MARKL: 0: NORMAL CONDITION.  1: AIRFLOW REVERSAL FOUND IN THE
C            SYSTEM, RECONSTRUCTION OF MESH FORMATION IS NEEDED.
C
         MARKL=0
         CALL  DTTR (MARKL,MARKP)
         IF (MARKL.NE.0) THEN
C           CALL ARR (MARKX)
            CALL BASE
            CALL MSLIST
         ENDIF
         DO 165 I=1,NJ
            TLEFT(I)=-0.1
165      CONTINUE
170      MARKQ=0
         CALL CDJN (ISQC,MARKS,NSTOP)
         IF (NSTOP.GT.0) GO TO 310
         DO 180 I=1,NJ
            SUMQ(I)=0.0
            SUMC(I)=0.0
            SUMM(I)=0.0
            SMHEAT(I)=0.0
180      CONTINUE
         DO 240 I=1,NB
            IQ=IBTN(I)
            IS=NSACB(I,2)
            IF (IBTN(I).NE.0) THEN
               TM=TMRD(I)
               IF (MARKS.GE.1) THEN
                  CALL ADPT (IS,IQ,MARKS,I)
               ELSE
                  TMST=TACC+DELTAT
                  CALL KALPHA (TMST,TM,FX,CP,I)
                  COAGE=BI(I)-FX*BI(I)**2/(0.375+BI(I))
                  DCOAGE(I)=COAGE
                  XNEW(I)=HK(I)*O(I)**2.*COAGE/(120.*DR*QQ(I)*A(I))
                  CALL ADPT (IS,IQ,MARKS,I)
               ENDIF
C               do 210 jj=1,is
C                  jja=is-jj+1
C                  if (dppa(iq,ja,3).ne.0.) go to 211
C210           continue
C              rdprop(i)=0.
C211           continue
            ELSE
               DO 215 IJ=1,NJ
                  KN=JLR(IJ)
                  IF(JS(I).EQ.JNO(KN)) THEN
                     DST=QQ(I)*DELTAT/(A(I)*60.0)
                     IF (DST.GT.LA(I)) DST=LA(I)
                     CH4B=(RCH4(I)-PRCH4S(KN))*(LA(I)-DST)/LA(I)+
     .                    PRCH4S(KN)
                     RDCH4(I)=(CH4B*QQ(I)+CH4V(I)*DST/LA(I))/
     .                        (QQ(I)+CH4V(I)*DST/LA(I))
                  ENDIF
215            CONTINUE
            ENDIF
            TTB=TRD(I)
            JJ=IABS(JF(I))
            DO 220 J=1,NJ
               IF (JNO(J).EQ.JJ) THEN
                  SUMQ(J)=SUMQ(J)+QQ(I)
                  SUMC(J)=SUMC(J)+QQ(I)*RDPROP(I)
                  SUMM(J)=SUMM(J)+QQ(I)*RDCH4(I)
                  SMHEAT(J)=SMHEAT(J)+(0.2376*TTB+1.2E-5*TTB**2)*QQ(I)
                  GO TO 240
               ENDIF
220         CONTINUE
240      CONTINUE
         DO 245 I=1,NJ
            JCH(I)=0
245      CONTINUE
C
C
      DO 250 I=1,NJ
         IF (JNOL(I).EQ.JSTART) GO TO 250
C       UPDATING THERMAL STATES IN JUNCTIONS EXCEPT FOR THE USER-
C     SPECIFIED STARTING JUNCTION (USUALLY THE ATMOSPHERE).
C
         K=JLR(I)
         MARKW=0
         TPP=T(K)
         PRPP=PROP(K)
         PRCPP=PRCH4(K)
C
C     PROP(K): CONTENT OF CONTAMINANTS IN AIRFLOW.
C     PRCH4(K): CONTENT OF CH4 IN AIRFLOW.
C     T(K): TEMPERATURE OF AIRFLOW.
C     THE ABOVE PARAMETERS ARE EXPRESSED AS MASS RATIOS WITH RESPECT
C     TO THAT OF AIR CURRENT.
C
         IF (SUMQ(K).GT.1.E-5) THEN
            PROP(K)=SUMC(K)/SUMQ(K)
            PRCH4(K)=SUMM(K)/SUMQ(K)
            PTCONT=9.801E7+2.0*SMHEAT(K)/(SUMQ(K)*2.4E-5)
            T(K)=-9900.+SQRT(PTCONT)
         ENDIF
         IF (JAN.LE.1) GO TO 248
         DO 247 II=1,JAJ
            IF (JNO(K).EQ.JSTAR(II)) THEN
               T(K)=TSTAR(II)
               PROP(K)=0.0
               PRCH4(K)=0.0
            ENDIF
247      CONTINUE
```

```
248      IF (CONCT.EQ.1) T(K)=TTT(K)
C
C       MARKW=1 INDICATES THAT ALL THE DOWN-STREAM AIRWAYS OF JUNCTION
C     JNO(K) NEED TO HAVE THEIR DETAILED DATA RECORDS.
C
         IF (ABS(TPP-T(K)).GE.1.0) MARKW=1
         IF (ABS(PRPP-PROP(K)).GE.1.E-6) MARKW=1
         IF (ABS(PRCPP-PRCH4(K)).GE.1.E-3) MARKW=1
         IF (MARKW.EQ.1) JCH(K)=1
         IF (MARKW.EQ.1) MARKQ=1
250   CONTINUE
C
C
         IF (MARKS.LT.1.OR.MARKQ.NE.0) THEN
            IBB=IBB+1
            IF (IBB.LE.100) THEN
               MARKS=MARKS+1
               GO TO 170
            ELSE
               IF (NTACC.EQ.0) WRITE (8,580) TACC
               NTACC=1
            ENDIF
         ENDIF
         CALL PARAM
         CALL NVP2
         ITT=ITT+1
         ITDT=0
         IF (INIFAN.EQ.1) THEN
            CALL ITR1 (MARKX,NSFLOW,ITT,ITDT)
         ELSE IF (INIFAN.EQ.0) THEN
            CALL ITR (MARKX,NSFLOW,ITT)
         ENDIF
C         do 249 jj=1,nfnum
C            if (nswt(jj).eq.2) then
C            mf=mpts(jj)
C            do 246 ii=1,mf-1
C246        continue
C            elseif (nswt(jj).eq.1) then
C            n1a=ncof(jj)
C            endif
C249        continue
            IF (ITT.GE.(ITN-9)) THEN
            J=ITT-ITN+10
            HRM(J)=SUMFNV/MNO
            DO 260 I=1,NB
               IF (CONCT.EQ.1) Q(I)=QQQ(I)
               TMRCD(I,J)=TMRD(I)
               QRCD(I,J)=ABS(Q(I))
               IF (JSB(I).NE.IABS(JS(I)).AND.Q(I).GT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
               IF (JSB(I).EQ.IABS(JS(I)).AND.Q(I).LT.0.0)
     .            QRCD(I,J)=-ABS(Q(I))
260         CONTINUE
         ENDIF
         WRITE (6,770) ITT,SUMFNV/MNO
         IF (ITT.GE.ITN.OR.(SUMFNV/MNO.LE.2.E-4.AND.ITT.GE.2)) THEN
            CALL EVHT (MARKX,MARKP,MARKL,MARKR,QSUM,QS)
         ELSE
            IF (ITT.GE.(ITN-2).AND.IOUT.LT.0) THEN
               WRITE (8,720) ITT
               TACC=TACC+TINC
               CALL OUTPUT (4,MARKX,MARKY,ITT)
               IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
               TACC=TACC-TINC
               WRITE (8,730)
            ENDIF
            CALL DISP (MARKX,ISQC,IBB)
            GO TO 160
         ENDIF
         IF (MASTER.LE.1) QSTD=QSUM/7.
         WRITE (6,530) TACC
         TCUMU=TCUMU+DELTAT
         IF (TACC.LT.(SPAN*60.)) THEN
            IF (QSUM.LE.QSTD.AND.ITT.LT.ITN.AND.TCUMU.LT.TOUT*60.) THEN
               MARKR=MARKR+1
               TACC1=TACC/60.0
               IF (MH.EQ.0) THEN
                  MH=1
                  WRITE (8,590)
               ENDIF
               WRITE (8,600) SUMFNV/MNO,ITT,TACC1
            ELSE
               MARKR=0
               TCUMU=0.0
               MH=0
               CALL OUTPUT (4,MARKX,MARKY,ITT)
               IF (IOUT.LT.0) CALL FWCTA
               IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
               IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,ITT)
            ENDIF
         ELSE
            MARKX=5
            CALL OUTPUT (4,MARKX,MARKY,ITT)
            MARKX=1
            CALL FWCTA
            IF (IOUT.LE.(-2)) CALL OUTPUT (6,MARKX,MARKY,ITT)
            IF (IOUT.NE.0) CALL OUTPUT (5,MARKX,MARKY,ITT)
            GO TO 300
         ENDIF
290   CONTINUE
C
300   WRITE (6,640)
310   STOP
C
530   FORMAT (/,1X,'*** SIMULATION GOES ON, ',F7.0,' SEC. PASSED.',/)
550   FORMAT (T19,'NON-STEADY STATE SIMULATION OUTPUT')
555   FORMAT (T16,'---FOR CONCENTRATION CALCULATION ONLY---')
560   FORMAT (///,1X,71('*'),///)
580   FORMAT (//,1X,'* ATTENTION *  SIMULATION STEP DELTAT NEEDS TO ',
     .        'BE SHORTENED AT',/,' ABOUT',F7.0,' SEC. AFTER EVENT ',
     .        'IF DETAILED ANALYSIS IS WANTED.')
590   FORMAT (//,1X,'* ATTENTION *   OUTPUT FOR  THE FOLLOWING TIME ',
     .       'INCREMENTS WERE',/,1X,'OMITTED DUE TO INSIGNIFICANT ',
     .       'CONDITION CHANGES ( THE THRESHOLD ',/,1X,'IN ACCURACY: ',
     .       'NVP CORR. PER MESH < 0.0002 IN.W.G. )',//,5X,'SUM OF ',
     .       'NVP CORR',T30,'ITERATIONS',T45,'AT TIME IN MIN.',/)
600   FORMAT (7X,F9.5,T31,I5,T47,F6.2)
630   FORMAT (///,1X,71('*'),///,T20,'STEADY-STATE SIMULATION OUTPUT',
     .       ///,1X,71('*'))
640   FORMAT (/,1X,'DYNAMIC SIMULATION PART COMPLETED.',//)
720   FORMAT (//,1X,71('-'),///,' * ATTENTION *  THE FOLLOWING ',
     .        'IS OUTPUT FOR ITERATION ',I4,/,' "DELTA Q" HAS ',
     .        'NO MEANING HERE.',/)
730   FORMAT (//,1X,71('-'))
740   FORMAT (80A1)
760   FORMAT (/,1X,'ENTERING DYNAMIC SIMULATION PART OF THE PROGRAM.',/
     .        ,1X,'PREFIXED ACCURACY CRITERION = 0.0002 IN.W.G. PER ',
     .        'MESH.',/)
770   FORMAT (10X,'ITERATION ',I3,'  COMPLETED, CURRENT ERROR = ',
     .        F8.5)
```

```
C
      END
C
C
      BLOCK DATA
C
C     ----------------------------------------------------------------
C
C     SUBPROGRAM PURPOSE:
C     INITIATION OF THE VARIABLES AND ARRAYS IN COMMON BLOCKS.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
      PARAMETER ( K1=NMX*4,  K2=NMX*3,  K3=NMX*6,  K4=NMX*8,  K5=NMY*3,
     .            K6=IMX*2,  K7=IMX*IMY*2,    K8=LMX+NMY+1, K9=NMY+NMX,
     .            K10=NMY*2, K11=NMX*2+IMX*11, K16=NMZ*10*5, K22=NMX*2,
     .            K17=NMZ*10*5+NMX*2, K19=NMZ*22, K21=IMX*3, K23=NMY*4,
     .            K24=IMY*5+IMX*IMY*4, K25=NXX*3)
C
C
      DATA  NO,JS,JF,NWTYP,R,Q,P,KF,LA,A,O,RSTD,DZRD,FRNVP,NREV,RDCH4,
     .      RDPROP,TRD,TJS,RDOP,RCH4,FFRNVP,RA,NNREV
     .      /K1*0,K2*0.0,NMX*0,K3*0.0,NMX*0,K4*0.0,NMX*0/
      DATA  NOF,NFREG,RGRAD,NFCW,MPTS,QF,PF,NSKP,NEGQ,JSB
     .      /K6*0,IMX*0.0,K6*0,K7*0.0,IMX*0,IMX*0,NMX*0/
      DATA  FC1,FC2,FC3,DK,FK,FKQ
     .      /K24*0.0/
      DATA  MNO,MEND,MSL,FNVP,RQ,INU,KNO,KJS,KJF
     .      /K8*0,K9*0.0,NMX*0,K25*0/
      DATA  JNO,T,Z,CH4C,JNOL,PROP,PRCH4,JLR
     .      /NMY*0,K5*0.0,NMY*0,K10*0.0,NMY*0/
      DATA  TROCK,TMRD,HA,HK
     .      /K1*0.0/
      DATA  NCENT,CH4V,CH4PA,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,TFSI,
     .      O2BEH,TPR,HTAD,QCENT
     .      /NMX*0,K11*0.0/
      DATA  NGIN,MMIN,NGOUT,LOUT
     .      /K1*0/
      DATA  MEMREC,NOREC,ESTPR,ESTCH4,ESTTR
     .      /K22*0,K2*0.0/
      DATA  XNEW,DCOAGE
     .      /K22*0.0/
      DATA  TIME,TSTART,JSTART,TLEFT
     .      /0.0,0.0,0,NMX*0.0/
      DATA  SUMQ,SUMC,SUMM,SMHEAT
     .      /K23*0.0/
      DATA  PROPS,PRCH4S,QQ,QTP,TTJS,TTRD,TAS,BI
     .      /K4*0.0/
      DATA  DELTAT,TACC,SUMFNV
     .      /3*0.0/
      DATA  IBTN,DPPA,NSAC,TMRDA,DELTAQ,DPPB,NSACB,GBTN,JCH,TAUXA,
     .      TAUXB,TAUXC,TAUXD,FMASS
     .      /NMX*0,K16*0.0,K22*0,K17*0.0,K22*0,50*0.0,NMY*0,K19*0.0,
     .       NMZ*0.0/
      DATA  WRNPR,WRNGS,WRNSM,WRNHT,IOMIT,ROMIT,IAC
     .      /4*0.0,IMX*0,K21*0.0,0/
      DATA  ITITLE,NCOMT1,NCOMTS,NCOMT2,LLINE
     .      /54*0/
C
      END
C
C
      SUBROUTINE ARR (MARKX)
C
C     ------------------------------------------------------------
C
C
C       SUBROUTINE PURPOSES:
C       1) GROUPING AIRWAYS ACCORDING TO THEIR TYPES.
C       2) ARRANGING REGULAR AIRWAYS ACCORDING TO THE PRODUCT OF
C          Q AND R.
C
C     ------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      NBU=NB
      NBL=1
      DO 10 K=1,NB
         IF (NWTYP(K).LT.0) THEN
            INU(NBU)=K
            NBU=NBU-1
         ELSE IF (NWTYP(K).EQ.0.OR.NWTYP(K).EQ.10) THEN
            RQ(K)=ABS(R(K)*Q(K))
            NWTYP(K)=2
         ELSE
            IF (NFNUM.GT.0) THEN
               DO 5 J=1,NFNUM
                  IF (NOF(J).EQ.NO(K)) GO TO 10
5              CONTINUE
            ENDIF
            INU(NBL)=K
            NBL=NBL+1
         ENDIF
10    CONTINUE
      IF (NFNUM.GT.0) THEN
         DO 25 K=1,NB
            DO 20 J=1,NFNUM
               IF (NOF(J).EQ.NO(K)) THEN
                  RGRAD(J)=0.0
                  NFCW(J)=0
                  NFREG(J)=K
                  INU(NBU)=K
                  NBU=NBU-1
                  GO TO 25
               ENDIF
20          CONTINUE
25       CONTINUE
      ENDIF
C
C
      IF (NBU.GT.(NBL-1)) THEN
30       NRETU=0
         DO 40 K=1,NB
            IF (NWTYP(K).GE.2) THEN
               IF (NRETU.LE.0) THEN
                  NMIN=K
                  NRETU=1
               ENDIF
               IF (RQ(NMIN).GT.RQ(K)) NMIN=K
            ENDIF
40       CONTINUE
         INU(NBL)=NMIN
         NBL=NBL+1
         NWTYP(NMIN)=0
         IF ((NBU+1).GT.NBL) GO TO 30
      ENDIF
      IF (MARKX.GT.0) THEN
         DO 60 I=1,INFLOW
            DO 50 J=1,NB
```

```
            IF (NCENT(I).EQ.NO(J)) THEN
               NWTYP(J)=10
               GO TO 60
            ENDIF
50        CONTINUE
60      CONTINUE
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE BASE
C
C
C     -----------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       FORMATION OF BASE SYSTEM.
C
C     -----------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      IND=INU(1)
      KJF(NJ)=JS(IND)
      KJS(NJ-1)=JS(IND)
      JS(IND)=-JS(IND)
      KJF(NJ-1)=JF(IND)
      KNO(NJ-1)=IND
      KNUM=NJ-1
10    DO 30 NUC=2,NB
        IND=INU(NUC)
        IF (JS(IND).LT.0) GO TO 30
        N1=0
        N0=0
        DO 20 K=KNUM,NJ
          IF (JS(IND).EQ.KJF(K)) N0=1
          IF (JF(IND).EQ.KJF(K)) N1=1
20      CONTINUE
        IF ((N1+N0).EQ.1) THEN
          IF (NWTYP(IND).LT.0) WRITE (8,100) NO(IND)
          KNUM=KNUM-1
          IF (N0.LE.0) THEN
            KJS(KNUM)=JF(IND)
            KJF(KNUM)=JS(IND)
            KNO(KNUM)=-IND
            JS(IND)=-JS(IND)
          ELSE
            KJS(KNUM)=JS(IND)
            KJF(KNUM)=JF(IND)
            KNO(KNUM)=IND
            JS(IND)=-JS(IND)
          ENDIF
          GO TO 10
        ELSE IF ((N1+N0).GT.1) THEN
          JS(IND)=-JS(IND)
          JF(IND)=-JF(IND)
        ENDIF
30    CONTINUE
      DO 40 K=1,NB
        JS(K)=IABS(JS(K))
40    CONTINUE
C
100   FORMAT (/,1X,'* ATTENTION *  REGULATOR',I5,' IN BASE SYST.')
C
      RETURN
      END
C
C
      SUBROUTINE MSLIST
C
C
C     ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       SET-UP OF MESH LIST.
C
C     ---------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MESC=0
      MNO=0
      DO 60 K=1,NB
        IF (JF(K).LT.0) THEN
          JF(K)=-JF(K)
          MNO=MNO+1
          JBM=JS(K)
          JEM=JF(K)
          NK=K
10        MESC=MESC+1
          MEND(MNO)=MESC
          MSL(MESC)=NK
          DO 20 KCO=1,NJ
            IF (JBM.EQ.KJF(KCO)) THEN
              KB=KCO
              GO TO 30
            ENDIF
20        CONTINUE
30        DO 40 KCO=1,NJ
            IF (JEM.EQ.KJF(KCO)) THEN
              KE=KCO
              GO TO 50
            ENDIF
40        CONTINUE
50        IF (KB.LT.KE) THEN
            NK=KNO(KB)
            JBM=KJS(KB)
            GO TO 10
          ELSE IF (KB.GT.KE) THEN
            NK=-KNO(KE)
            JEM=KJS(KE)
            GO TO 10
          ENDIF
        ENDIF
60    CONTINUE
      IF (MEND(MNO).GT.LMX) WRITE (8,100)
C
      RETURN
100   FORMAT (/,1X,'* ERROR *  NETWORK TOO BIG, INCREASE LMX.')
C
      END
C
C
      SUBROUTINE MBLNC
C
C
C     -----------------------------------------------------------
C
```

```
C       SUBROUTINE PURPOSE:
C       REALIZATION OF BALANCE OF MASS FLOW RATES.
C
C     ------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=2
      DO 20 K=1,MNO
         MENDW=MEND(K)
         DO 10 J=MBEGW,MENDW
            N=MSL(J)
            IF (N.LT.0) N=-N
            Q(N)=0.
10       CONTINUE
         MBEGW=MENDW+2
20    CONTINUE
C
C
      MBEGW=1
      DO 40 K=1,MNO
         MENDW=MEND(K)
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         Q1=Q(N)
         M=MBEGW+1
         DO 30 J=M,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
            N=-N
            FACT=-1.
            ENDIF
            Q(N)=Q(N)+Q1*FACT
30       CONTINUE
         MBEGW=MENDW+1
40    CONTINUE
C
      RETURN
      END
C
C
      SUBROUTINE ITR (MARKX,NSFLOW,ITT)
C
C     -------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     -------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
5     IT=0
10    DQSUM=0.
      MBEGW=1
      DO 50 K=1,MNO
         MENDW=MEND(K)
         DPSUM=0.
         RQSUM=0.
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         IF (NWTYP(N).EQ.(-1)) THEN
            MBEGW=MENDW+1
            GO TO 50
         ENDIF
         DO 30 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            IF(NWTYP(N).EQ.1) THEN
               IF (NFNUM.GT.0) THEN
                  DO 20 L=1,NFNUM
                     IF (NFREG(L).EQ.N) THEN
                        RQSUM=RQSUM-(RGRAD(L)*100000)
                        GO TO 25
                     ENDIF
20                CONTINUE
               ENDIF
25             IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
               DPSUM=DPSUM-FACT*R(N)
            ELSE
               RQ2=R(N)*ABS(Q(N))*2.E-5
               RQSUM=RQSUM+RQ2
               DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
               DPSUM=DPSUM+FACT*DP
            ENDIF
30       CONTINUE
         IF (ABS(RQSUM).GT.1.E-10) THEN
            DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
         ELSE
            DQ=0.0
         ENDIF
         DO 40 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            Q(N)=Q(N)-(DQ*FACT)
40       CONTINUE
         DQSUM=DQSUM+ABS(DQ)
         MBEGW=MENDW+1
50    CONTINUE
C
      DO 100 KI=1,NB
         IF (NFNUM.GT.0.AND.NWTYP(KI).EQ.1) THEN
            DO 90 J=1,NFNUM
               IF (NOF(J).EQ.NO(KI)) THEN
                  NFCW(J)=0
                  NEGQ(J)=0
                  IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .               NEGQ(J)=1
                  IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .               NEGQ(J)=1
                  NABF=JS(KI)
                  DO 70 L=1,NJ
                     IF (NABF.EQ.JNO(L)) THEN
                        TABF=T(L)
                        GO TO 80
                     ENDIF
70                CONTINUE
80                IF (NEGQ(J).EQ.0) THEN
                     IF (NSWT(J).EQ.0) NSWT(J)=2
                     IF (NSKP(J).EQ.0) THEN
                        IF (NSWT(J).EQ.2) THEN
```

```
                     CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                  ELSEIF (NSWT(J).EQ.1) THEN
                     CALL LSFAN (J,Q(KI),R(KI),TABF,0,0)
                  ENDIF
                  NSKP(J)=1
               ELSE
                  IF (NSWT(J).EQ.2) THEN
                     CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                  ELSEIF (NSWT(J).EQ.1) THEN
                     CALL LSFAN (J,Q(KI),R(KI),TABF,1,0)
                  ENDIF
               ENDIF
               GO TO 100
            ELSE
               R(KI)=PF(J,1)
               R(KI)=R(KI)*(TR+460.)/(TABF+460.)
C              *********************************
               RGRAD(J)=0.0
            ENDIF
         ENDIF
90       CONTINUE
      ENDIF
100   CONTINUE
C
      IT=IT+1
      IF ((DQSUM/FLOAT(MNO)).LT.10.) RETURN
      IF (IT.GT.100) THEN
         WRITE (8,210) DQSUM/FLOAT(MNO),ITT,IT,IBB
         RETURN
      ENDIF
      GO TO 10
C
C200   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
C     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 100'
C     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
C     .        ' CURRENT ERROR:',F8.0)
210   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 500'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 CFM PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0,/,' IT WAS HAPPENED IN DATA ',
     .        'PREPARATION CYCLE ',I5,/' IT= ',I4,2X,' IBB= ',I4)
C
      END
C
C
      SUBROUTINE ITR1 (MARKX,NSFLOW,ITT,ITDT)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       APPLICATION OF HARDY CROSS METHOD TO NETWORK BALANCING.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
5     IT=0
10    DQSUM=0.
      MBEGW=1
      DO 50 K=1,MNO
         MENDW=MEND(K)
         DPSUM=0.
         RQSUM=0.
         N=MSL(MBEGW)
         IF (N.LT.0) N=-N
         IF (NWTYP(N).EQ.(-1)) THEN
            MBEGW=MENDW+1
            GO TO 50
         ENDIF
         DO 30 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
            ENDIF
            IF(NWTYP(N).EQ.1) THEN
               IF (NFNUM.GT.0) THEN
                   DO 20 L=1,NFNUM
                     IF (NFREG(L).EQ.N) THEN
                        RQSUM=RQSUM-(RGRAD(L)*100000)
                        GO TO 25
                     ENDIF
20                CONTINUE
               ENDIF
25             IF (IABS(JS(N)).NE.JSB(N)) FACT=-FACT
               DPSUM=DPSUM-FACT*R(N)
            ELSE
               RQ2=R(N)*ABS(Q(N))*2.E-5
               RQSUM=RQSUM+RQ2
               DP=R(N)*Q(N)*ABS(Q(N))*1.E-10
               DPSUM=DPSUM+FACT*DP
            ENDIF
30       CONTINUE
         IF (ABS(RQSUM).GT.1.E-10) THEN
            DQ=(DPSUM-FNVP(K))*1.E5/RQSUM
         ELSE
            DQ=0.0
         ENDIF
         DO 40 J=MBEGW,MENDW
            N=MSL(J)
            FACT=1.
            IF (N.LT.0) THEN
               N=-N
               FACT=-1.
           ENDIF
            Q(N)=Q(N)-(DQ*FACT)
40       CONTINUE
         DQSUM=DQSUM+ABS(DQ)
         MBEGW=MENDW+1
50    CONTINUE
C
      DO 100 KI=1,NB
         IF (NWTYP(KI).EQ.1.AND.NFNUM.GT.0) THEN
            DO 90 J=1,NFNUM
               IF (NOF(J).EQ.NO(KI)) THEN
                  NFCW(J)=0
                  NEGQ(J)=0
                  IF (Q(KI).LT.0.0.AND.IABS(JS(KI)).EQ.JSB(KI))
     .               NEGQ(J)=1
                  IF (Q(KI).GT.0.0.AND.IABS(JS(KI)).NE.JSB(KI))
     .               NEGQ(J)=1
                  IF (NVPN.GE.1.AND.MADJC.LE.0) THEN
                     TABF=TR
                  ELSE
                     NABF=JS(KI)
                     DO 70 L=1,NJ
                        IF (NABF.EQ.JNO(L)) THEN
                           TABF=T(L)
                           GO TO 80
```

```
                       ENDIF
70                  CONTINUE
                 ENDIF
80               IF (NEGQ(J).EQ.0) THEN
                    IF (NSWT(J).EQ.0) NSWT(J)=2
                    IF (NSKP(J).EQ.0) THEN
                       IF (NSWT(J).EQ.2) THEN
                          CALL SPLINE (J,Q(KI),R(KI),TABF,0)
                       ELSEIF (NSWT(J).EQ.1) THEN
                          CALL LSFAN (J,Q(KI),R(KI),TABF,0,0)
                       ENDIF
                       NSKP(J)=1
                    ELSE
                       IF (NSWT(J).EQ.2) THEN
                          CALL SPLINE (J,Q(KI),R(KI),TABF,1)
                       ELSEIF (NSWT(J).EQ.1) THEN
                          CALL LSFAN (J,Q(KI),R(KI),TABF,1,0)
                       ENDIF
                    ENDIF
                    GO TO 100
                 ELSE
                    R(KI)=PF(J,1)
                    R(KI)=R(KI)*(TR+460.)/(TABF+460.)
C                   *********************************
                    RGRAD(J)=0.0
                 ENDIF
              ENDIF
90         CONTINUE
        ENDIF
100   CONTINUE
C
C
      IT=IT+1
         ITDT=ITDT+1
         IF (IT.LE.1) THEN
            GO TO 10
         ELSE IF ((DQSUM/FLOAT(MNO)).LT.10.) THEN
            RETURN
         ELSE IF (ITDT.GT.200) THEN
            NSFLOW=0
            WRITE (8,200) DQSUM/FLOAT(MNO)
            WRITE (8,205) MNO
            RETURN
         ELSE IF (IT.GT.50) THEN
            CALL ARR (MARKX)
            CALL BASE
            CALL MSLIST
            CALL MBLNC
            CALL NVP1
            CALL NVP2
            GO TO 5
         ENDIF
      GO TO 10
C
200   FORMAT (//,1X,'* ATTENTION *  ACCURACY CRITERION IN NETWORK',
     .        ' BALANCING',/,1X,'( SUB. ITR ) WAS NOT SATISFIED IN 200'
     .        ,'ITERATIONS. THE',/,' CRITERION: 10 PER MESH,',6X,
     .        ' CURRENT ERROR:',F8.0)
205   FORMAT (//1X,'NUMBER OF MESHES=',I5)
C
      END
C
C
C
      SUBROUTINE LSFAN (L,QKI,RKI,TABF,NTW,NEXP)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       USING LEAST SQUARE METHOD FOR FAN CURVE FITTING.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
      DIMENSION ZZ(IMX),S(IMX),ALPHA(IMX),BETA(IMX),SIGMA2(IMX)
     .,PPP(IMX,IMX)
      DOUBLE PRECISION W(IMX)
C
      NFCW(L)=0
      NP=MPTS(L)
      IF (NTW.EQ.1) GO TO 145
      MARKXX=0
      DO 1 I=1,10
1     ALPHA(I)=0.0
      IF (NP.GT.2) GO TO 35
      ND(L)=1
      N1A=2
      NCOF(L)=N1A
      IF (NP.LT.1) THEN
         STOP 10
      ELSEIF (NP.EQ.1) THEN
         COF(L,1)=PF(L,1)
         COF(L,2)=0.
         GO TO 145
      ENDIF
      IF (QF(L,1).EQ.QF(L,2)) THEN
         STOP 25
      ELSE
         COF(L,1)=(PF(L,1)*QF(L,2)-PF(L,2)*QF(L,1))/(QF(L,2)-QF(L,1))
         COF(L,2)=(PF(L,2)-PF(L,1))/(QF(L,2)-QF(L,1))
         GO TO 145
      ENDIF
35    DSQ=0.0
      DO 40 J=1,NP
         DSQ=DSQ+PF(L,J)*PF(L,J)
         PPP(1,J)=0.
40       PPP(2,J)=1.
      W(1)=NP
      BETA(1)=0.
      N1A=4
      IF (NP.LT.5) N1A=NP-1
      IF (NEXP.GT.0) N1A=NEXP
      DO 85 I=1,N1A
         K=I+1
         ZZ(I)=0.
         DO 60 J=1,NP
60          ZZ(I)=ZZ(I)+PF(L,J)*PPP(K,J)
         S(I)=ZZ(I)/W(I)
         DSQ=DSQ-S(I)*S(I)*W(I)
         SIGMA2(I)=DSQ/FLOAT(NP-I)
         IF(SIGMA2(I).LE.1.D-6) GO TO 105
         IF (I.GE.N1A) GO TO 90
            ALPHA(I)=0.
            DO 75 J=1,NP
75          ALPHA(I)=ALPHA(I)+QF(L,J)*PPP(K,J)*PPP(K,J)
            ALPHA(I)=ALPHA(I)/W(I)
            W(K)=0.
            DO 80 J=1,NP
               PPP(K+1,J)=(QF(L,J)-ALPHA(I))*PPP(K,J)-BETA(I)*PPP(I,J)
80             W(K)=W(K)+PPP(K+1,J)*PPP(K+1,J)
```

```
85     BETA(K)=W(K)/W(I)
90     SMALL=SIGMA2(1)
       I=1
       DO 100 J=2,N1A
          IF (SMALL.LE.SIGMA2(J)) GO TO 100
          I=J
          SMALL=SIGMA2(J)
100    CONTINUE
105    WRITE (8,195) SIGMA2(I)
       N1A=I
       NCOF(L)=N1A
       N=N1A-1
       IF (N.LE.0) THEN
          ND(L)=1
          COF(L,1)=S(1)
          COF(L,2)=0.
          GO TO 145
       ENDIF
       ND(L)=N
       DO 125 I=1,N1A
          DO 120 J=1,I
120          PPP(I,J)=0.0
125       PPP(I,I+1)=1.
       DO 130 J=1,N
130       PPP(1,J+2)=-PPP(1,J+1)*ALPHA(J)-PPP(1,J)*BETA(J)
       DO 135 I=2,N
          DO 135 J=I,N
135          PPP(I,J+2)=PPP(I-1,J+1)-PPP(I,J+1)*ALPHA(J)-PPP(I,J)*BETA(J)
       DO 140 I=1,N1A
          COF(L,I)=0.
          DO 140 J=1,N1A
140          COF(L,I)=COF(L,I)+PPP(I,J+1)*S(J)
       NSKP(L)=1
       write (8,*) 'ENTER FAN CURVE FITTING'
145    IF (MARKXX.EQ.0) WRITE (8,200) (COF(L,I),I=1,N1A)
       IF (ABS(QKI).LT.1.D-6) GO TO 160
       MARKXX=1
       IF (NPLOT.EQ.1) THEN
C          IF (QKI.LT.QF(L,1).OR.QKI.GT.QF(L,NP)) NFCW(L)=1
C          FANQ=QKI
C          RGRAD(L)=0.0
C          II=1
C          GO TO 165
          IF (QKI.LE.QF(L,1)) THEN
             DQKI=0
             DQKI1=QKI-QF(L,1)
             FANQ=QKI
             RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
             RKI=PF(L,1)-DQKI1*RGRAD(L)
             RKI=RKI*(TR+460.)/(TABF+460.)
             NFCW(L)=1
             RETURN
          ELSEIF (QKI.GT.QF(L,NP)) THEN
C               DQKI=QF(L,NP)-QF(L,NP-1)
             DQKI=QF(L,NP)
             DQKI1=QKI-QF(L,NP)
             FANQ=QKI
             RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
             RKI=PF(L,NP)+DQKI1*RGRAD(L)
             RKI=RKI*(TR+460.)/(TABF+460.)
             NFCW(L)=1
             RETURN
          ELSE
             FANQ=QKI
             RGRAD(L)=0.0
                II=1
                GO TO 165
             ENDIF
          ELSE IF (NPLOT.EQ.2) THEN
             IF (QKI.LT.QF(L,NP)) GO TO 146
             RKI=PF(L,NP)
             RGRAD(L)=0.0
             NFCW(L)=1
             RETURN
146          IF (QKI.GT.QF(L,1)) GO TO 147
                DQKI=0
                DQKI1=QKI-QF(L,1)
                FANQ=QKI
                RGRAD(L)=COF(L,2)+2*COF(L,3)*DQKI+3*COF(L,4)*DQKI**2
                RKI=PF(L,1)+DQKI1*RGRAD(L)
                RKI=RKI*(TR+460.)/(TABF+460.)
                NFCW(L)=1
                RETURN
147          FANQ=QKI
             RGRAD(L)=0.0
             II=1
             GO TO 165
          ELSE IF (NPLOT.EQ.3) THEN
             IF (QKI.LT.QF(L,NP)) GO TO 150
             RKI=PF(L,NP)
             RGRAD(L)=0.0
             NFCW(L)=1
             RETURN
150          IF (QKI.GT.QF(L,1)) GO TO 155
             RKI=PF(L,1)
             RGRAD(L)=0.0
             NFCW(L)=1
             RETURN
155          FANQ=QKI
             RGRAD(L)=0.0
             II=1
             GO TO 165
          ENDIF
          GO TO 165
160       QCS=QF(L,NP)-QF(L,1)
          QCSA=QCS/120.
          RGRAD(L)=0.0
          Q0=QF(L,1)-5.*QCSA
          Q2(1)=Q0
          FANQ=Q0
          II=130
165       NE=ND(L)
          DO 180 I=1,II
             H2(I)=COF(L,NE+1)*FANQ
             IF (NE.LE.1) GO TO 175
             KT=NE
             DO 170 J=2,NE
                H2(I)=(H2(I)+COF(L,KT))*FANQ
                IF (MARKXX.EQ.1) RGRAD(L)=(RGRAD(L)+KT*COF(L,KT+1))*FANQ
170             KT=KT-1
175          H2(I)=H2(I)+COF(L,1)
             RGRAD(L)=RGRAD(L)+COF(L,2)
             IF (MARKXX.EQ.1) RKI=H2(1)
             Q2(I+1)=FANQ+QCSA
             FANQ=Q2(I+1)
180       CONTINUE
          IF (MARKXX.EQ.1) RETURN
C          WRITE (6,190) (Q2(I),H2(I),I=1,130)
190       FORMAT (1X,5(F8.0,F6.2))
195       FORMAT (/18X,8HSIGMA2= ,1PE14.7)
```

```
200   FORMAT (1H0,17X,12HCOEFFICIENTS,//18X,1P6E18.7)
      RETURN
      END
C
C
C
      SUBROUTINE SPLINE (J,QKII,RKI,TABF,NTW)
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       1) FAN CURVE FITTING.
C       2) FAN PRESSURE INTERPOLATION.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     FKQ(J,I,1-4): COEFFS. OF 3RD ORDER POLYNOMIAL FOR FAN NOF(J),
C                   SEGMENT I.
C     NSKP(J)=1: MARKER INDICATING THAT FAN CURVE FITTING FOR FAN
C                NOF(J) HAS BEEN PROCESSED.
C
      MF = MPTS(J)
      IF (NTW.EQ.0) THEN
C
         DO 5 I=1,MF-1
            DK(I)=QF(J,I+1)-QF(J,I)
5        CONTINUE
C
         DO 10 I=2,MF-1
            FC3(I)=3.0*(PF(J,I+1)*DK(I-1)-PF(J,I)*(QF(J,I+1)-QF(J,I-1))
     .             +PF(J,I-1)*DK(I))/(DK(I-1)*DK(I))
10       CONTINUE
C
         FK(1)=1.0
         FC1(1)=0.0
         FC2(1)=0.0
C
         DO 15 I=2,MF-1
            FK(I)=2.0*(QF(J,I+1)-QF(J,I-1))-DK(I-1)*FC1(I-1)
            FC1(I)=DK(I)/FK(I)
            FC2(I)=(FC3(I)-DK(I-1)*FC2(I-1))/FK(I)
15       CONTINUE
C
         FK(MF-1)=1.0
         FC3(MF-1)=0.0
         FKQ(J,MF-1,3)=0.0
C
         DO 20 I=MF-1,1,-1
            FKQ(J,I,3)=FC2(I)-FC1(I)*FKQ(J,I+1,3)
            FKQ(J,I,2)=(PF(J,I+1)-PF(J,I))/DK(I)-
     .                 DK(I)*(FKQ(J,I+1,3)+2.0*FKQ(J,I,3))/3.0
            FKQ(J,I,4)=(FKQ(J,I+1,3)-FKQ(J,I,3))/(3.0*DK(I))
20       CONTINUE
C
         DO 25 I=1,MF-1
            FKQ(J,I,1)=PF(J,I)
25       CONTINUE
         NSKP(J)=1
      ENDIF
C
C
      QKI=ABS(QKII)
      IF (NPLOT.EQ.1) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            dqki=0.0
            dqki1=qki-qf(j,1)
            m=1
            RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,1)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            dqki=qf(j,mf)-qf(j,mf-1)
            dqki1=qki-qf(j,mf)
            m=mf-1
            rgrad(j) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,mf)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ENDIF
      ELSEIF (NPLOT.EQ.2) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            dqki=0.0
            dqki1=qki-qf(j,1)
            m=1
            RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .                 3*FKQ(J,M,4)*DQKI**2
            rki=pf(j,1)+dqki1*rgrad(j)
            rki=rki*(tr+460.)/(tabf+460.)
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ENDIF
      ELSEIF (NPLOT.EQ.3) THEN
         IF (QKI.LE.QF(J,1).OR.NEGQ(J).GE.1) THEN
            RKI = PF(J,1)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ELSE IF (QKI.GE.QF(J,MF)) THEN
            RKI = PF(J,MF)*(TR+460.)/(TABF+460.)
            RGRAD(J) = 0.0
            NFCW(J) = 1
            RETURN
         ENDIF
      ENDIF
      IF (QF(J,1).GE.QKI) THEN
         M = 1
      ELSE
         DO 40 K = 2,MF
            IF (QF(J,K).GE.QKI) THEN
               M = K-1
               GO TO 45
            ENDIF
40       CONTINUE
      ENDIF
      M = MF-1
45    DQKI=QKI-QF(J,M)
      RKI = FKQ(J,M,1)+FKQ(J,M,2)*DQKI+FKQ(J,M,3)*DQKI**2
     .        +FKQ(J,M,4)*DQKI**3
```

```
      RKI=RKI*(TR+460.)/(TABF+460.)
      RGRAD(J) = FKQ(J,M,2)+2*FKQ(J,M,3)*DQKI+
     .            3*FKQ(J,M,4)*DQKI**2
C
      RETURN
      END

C
      SUBROUTINE  KALPHA (TMST,TM,FX,CP,I)
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF "COEFF. OF AGE" ASSUMING DRY CASE.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     WT: TEMP. CORRECTED AIR DENSITY, LBM/FT**3.
C     VISC: KINETIC VISCOSITY OF AIR, FT/HR.
C     HKA: THERMAL CONDUCTIVITY OF ROCK, BTU/HR*FT*F.
C     FO: FOURIER NUMBER.  BI: BIOT NUMBER.  RN: REYNOULD NUMBER.
C     HC: HEAT TRANSFER COEFFICIENT.
C
      PI=3.1415936
      CP=0.24
      TMX=TM
      QX=QQ(I)
      TX=TMST/3600.
      WT=DR*(TR+460.)/(TMX+460.)
      VISC=0.000145*((460.+TMX)/492.)**1.75
      HKA=0.014*((460.+TMX)/492.)**0.81
      RN=QX*DR/(15.*WT*O(I)*VISC)
      FR0=0.0032+0.221/RN**0.237
      POT=(100./RN)**0.125
      COR=((KF(I)*0.075)/(809*DR*FR0))**POT
      HC=0.005*HKA*O(I)*RN**0.8*COR/A(I)
      FO=TX*HA(I)*O(I)**2/(4.*A(I)**2)
      BI(I)=HC*2.*A(I)/(O(I)*HK(I))
      X=(0.375+BI(I))*SQRT(FO)
C
      IF (X.LT.2.5) THEN
         N=0
         SUMT=0.0
         ADDT=X
10       SUMT=SUMT+ADDT
         N=N+1
         ADDT=-ADDT*X**2*(2.*N-1)/(N*(2.*N+1))
         IF (ABS(ADDT).GE.0.00001) GO TO 10
         FX=1.0-(EXP(X**2))*(1.0-(2.0/SQRT(PI))*SUMT)
      ELSE
         N=0
         SUMT=0.0
         ADDT=1.0
20       SUMT=SUMT+ADDT
         N=N+1
         OLADDT=ADDT
         ADDT=-ADDT*(2*N-1)/(2.0*X**2)
         IF (ABS(ADDT).LT.0.00001) GO TO 30
         IF (ABS(OLADDT).GT.ABS(ADDT)) GO TO 20
30       FX=1.0-1.0*SUMT/(SQRT(PI)*X)
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE NVP1
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF NVP THROUGH MANIPULATION OF INPUT JUNCTION DATA
C       IN TEMP. AND ELEVATION.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=1
      DO 30 K=1,MNO
         IF (NVPN.GE.1) GO TO 30
         MENDW=MEND(K)
         TZ=0.0
         ZOT=0.0
C
         DO 20 J=MBEGW,MENDW
            N=MSL(J)
            NX=N
            IF (NX.LT.0) NX=-NX
            DO 10 L=1,NJ
               IF (JS(NX).EQ.JNO(L)) THEN
                  T0=T(L)+460.
                  Z0=Z(L)
               ELSE IF (JF(NX).EQ.JNO(L)) THEN
                  T1=T(L)+460.
                  Z1=Z(L)
               ENDIF
10          CONTINUE
            DZZ=(Z1-Z0)
            IF (N.LT.0) DZZ=-DZZ
            TZ=TZ+0.5*(T0+T1)*DZZ
            ZOT=ZOT+DZZ/(0.5*(T0+T1))
20       CONTINUE
C
         IF (ABS(ZOT).LE.0.000001) THEN
            TM=TR+460.0
         ELSE
            TM=ABS(SQRT(ABS(TZ/ZOT)))
         ENDIF
C
         IF (ABS(TM).LE.0.000001) TM=TR+460.0
         FNVP(K)=TZ*DR/(5.2*TM)
         MBEGW=MENDW+1
30    CONTINUE
C
      RETURN
      END
C
CC
C

      SUBROUTINE  NVP2
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       EVALUATION OF NVP ACCORDING TO CALCULATED FRNVP OF EACH AIRWAY
C     (REFER TO SUBS. TEVAL AND PARAM).
C
```

```
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      MBEGW=1
      SUMFNV=0.0
      DO 20 K=1,MNO
         MENDW=MEND(K)
C
C     FNVP(K): NVP IN MESH K. IN IN.W.G.
C     SUMFNV: SUM OF NVP VARIATION IN MESHES IN THE SUCCESSIVELY
C           PERFORMED ITERATIONS, SERVING AS A CRITERION TO JUDGE IF
C           THE CONVERGENT RESULTS ARE REACHED.
C
         ONVP=FNVP(K)
         FSM=0.0
         DO 10 J=MBEGW,MENDW
            N=MSL(J)
            IF (N.LT.0) THEN
               FACT=-1.0
               NX=-N
            ELSE
               FACT=1.0
               NX=N
            ENDIF
            FSM=FSM+FACT*FRNVP(NX)
10       CONTINUE
         FNVP(K)=-FSM/5.2
         SUMFNV=SUMFNV+ABS(ONVP-FNVP(K))
         MBEGW=MENDW+1
20    CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE  INIT (IPT)
C
C     --------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       INITIATION OF DATA STORAGE ARRAYS BEFORE DYNAMIC SIMULATION OR
C     BEFORE EACH TIME INCREMENT.
C
C     --------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      IF (IPT.EQ.1) THEN
         DO 30 I=1,NB
            RA(I)=R(I)
            TMRDA(I)=TMRD(I)
            IF (JS(I).LT.0) JS(I)=-JS(I)
            IF (JF(I).LT.0) JF(I)=-JF(I)
30       CONTINUE
         DO 50 I=1,INFLOW
            DO 40 J=1,NB
               IF (NCENT(I).NE.NO(J)) GO TO 40
               NWTYP(J)=10
               GO TO 50
40          CONTINUE
50       CONTINUE
      ELSE
C
         DO 70 I=1,NJ
            TAS(I)=T(I)
            PROPS(I)=PROP(I)
            PRCH4S(I)=PRCH4(I)
70       CONTINUE
         DO 80 I=1,NMZ
            FMASS(I)=-1.0
80       CONTINUE
         DO 90 I=1,INFLOW
            HTAD(I)=0.0
90       CONTINUE
         DO 100 I=1,NB
            IF (NNREV(I).EQ.(NNREV(I)/2)*2) THEN
               NREV(I)=0
            ELSE
               NREV(I)=1
            ENDIF
            TTJS(I)=TJS(I)
            TTRD(I)=TRD(I)
            RDOP(I)=RDPROP(I)
            RCH4(I)=RDCH4(I)
            QQ(I)=Q(I)
            QTP(I)=Q(I)
            QQ2(I)=Q(I)
            IF (CONCT.EQ.1) THEN
               TTJS(I)=TJSS(I)
               TTRD(I)=TRDD(I)
               QQ(I)=QQQ(I)
               QTP(I)=QQQ(I)
               QQ2(I)=QQQ(I)
            ENDIF
100      CONTINUE
      ENDIF
C
      RETURN
      END
C
C
      SUBROUTINE CDCH (NSTOP)
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       UPDATING CONDITION CHANGES ACCORDING TO USER-SPECIFIED TIME TABL
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION  QFX(20),PFX(20),DAL(15)
C
C
      IF (TACC.LT.DELTAT) IREAD=1
C
C      ICODE: CODE NUMBER. 1: CHANGE TO ORDINARY AIRWAY.    2: CHANGE
C             TO FAN BRANCH.  3: CHANGE TO FIRE SOURCE. 4: FIRE CEESED.
C             5: CHANGE SIMULATION STEPS. 6: JUNCTION WHOSE DOWN-STREAN
C             SIDE AIRWAYS HAVE DETAILED DATA RECORDS.
C      NBRR: AIRWAY CALLING NUMBER IN WHICH CONDITION CHANGE HAPPENS.
C      INPUT SCHEME:
C        TS,ICODE,(MAY BE FOLLOWED BY OTHER PARAMETERS)
C        TS,1,NBRR,RSTD
C        TS,2,NBRR,INDEX,QF1,PF1,QF2,PF2,QF3,PF3,QF4,PF4,QF5,PF5
C        TS,3,NBRR,CONT,CONC,HEAT,O2MIN,SMPO2,HTPO2,QCENT,TPR
C        TS,4,NBRR
C        TS,5,DELTAT
```

```
C         TS,6,TOUT
C         TS,7,JDPP
C
1      IF (IREAD.EQ.1) THEN
4         CALL READIN (DAL,14,ISTOP,0)
          IF (ISTOP.EQ.1) THEN
             ICODE=DAL(2)
             WRITE (8,300)
             IF (ICODE.EQ.1) WRITE (8,310)
             IF (ICODE.EQ.2) WRITE (8,320)
             IF (ICODE.EQ.3) WRITE (8,330)
             IF (ICODE.EQ.4) WRITE (8,340)
             IF (ICODE.EQ.5) WRITE (8,350)
             IF (ICODE.EQ.6) WRITE (8,360)
             IF (ICODE.EQ.7) WRITE (8,365)
             WRITE (8,370) (ROW(IE),IE=1,80)
             NSTOP=5
             RETURN
          ENDIF
          IF (DAL(15).LT.(-1.E20)) GO TO 4
          IF (NCOMTS.GT.NCOMT2) THEN
             NCOMT2=NCOMTS
             GO TO 4
          ENDIF
          IREAD=0
          TS=DAL(1)
       ENDIF
       IF (TS*60.0.GT.TACC) RETURN
       ICODE=DAL(2)
C
       IF (ICODE.EQ.5) THEN
          DELT=DAL(3)
          IF (DELT.GT.1.E-3.AND.DELT.LT.30.0) THEN
             WRITE (8,190) DELTAT,DELT,TS
             DELTAT=DELT
          ELSE
             WRITE (8,200) DELT
          ENDIF
          GO TO 100
C
       ELSE IF (ICODE.EQ.6) THEN
          TOUT=DAL(3)
          IF (TOUT.LT.1.E-5) TOUT=1.0
          GO TO 100
C
       ELSE IF (ICODE.EQ.7) THEN
          JDP=JDP+1
          IF (JDP.GT.IMX) THEN
             WRITE (8,240)
             JDP=JDP-1
             GO TO 100
          ENDIF
          JDPP(JDP)=DAL(3)
          GO TO 100
       ENDIF
       NBRR=DAL(3)
       DO 5 I=1,NB
          IF (NO(I).EQ.NBRR) THEN
             N=I
             GO TO 6
          ENDIF
5      CONTINUE
       WRITE (8,210) NBRR
       GO TO 100
C
C        ICODE=1 INDICATES THAT AIRWAY NBRR BECOMES AN ORDINARY AIRWAY
C      WITH RESISTANCE EQUAL TO RSTD AT TS MIN. AFTER EVENT.
C
6      IF (ICODE.EQ.1) THEN
          RSTD(N)=DAL(4)
          R(N)=RSTD(N)*(TMRD(N)+460.0)**2/(TR+460.0)**2
          NWTYP(N)=0
          IF (NFNUM.GT.0) THEN
             DO 8 J=1,NFNUM
                IF (NOF(J).EQ.NBRR) THEN
                   IF (NFNUM.GT.J) THEN
                      DO 14 K=J,NFNUM-1
                         NOF(K)=NOF(K+1)
                         MPTS(K)=MPTS(K+1)
                         NSKP(K)=NSKP(K+1)
C                        ********************
                         RGRAD(K)=RGRAD(K+1)
                         NFREG(k)=NFREG(K+1)
C                        *******************
                         DO 7 KK=1,IMX
                            QF(K,KK)=QF(K+1,KK)
                            PF(K,KK)=PF(K+1,KK)
7                        CONTINUE
                         IF (NSWT(K).EQ.2) THEN
                            MF=MPTS(K+1)
                            DO 12 II=1,MF-1
                               FKQ(K,II,1)=FKQ(K+1,II,1)
                               FKQ(K,II,2)=FKQ(K+1,II,2)
                               FKQ(K,II,3)=FKQ(K+1,II,3)
                               FKQ(K,II,4)=FKQ(K+1,II,4)
12                          CONTINUE
                         ELSEIF (NSWT(K).EQ.1) THEN
                            NCOF(K)=NCOF(K+1)
                            N1A=NCOF(K)
                            DO 13 II=1,N1A
                               COF(K,II)=COF(K+1,II)
13                          CONTINUE
                         ENDIF
14                    CONTINUE
                   ENDIF
                   INIFAN=1
                   NOF(NFNUM)=0
                   MPTS(NFNUM)=0
                   NSKP(NFNUM)=0
                   NFNUM=NFNUM-1
                   GO TO 9
                ENDIF
8            CONTINUE
          ENDIF
9         WRITE (8,110) NO(N),RSTD(N),TS
C
       ELSE IF (ICODE.EQ.2) THEN
          INDEX=DAL(4)
          IF (INDEX.LE.0) THEN
             WRITE (8,120) INDEX,NBRR
             GO TO 100
          ELSE IF (INDEX.EQ.1) THEN
             RX=DAL(5)
             IF (NFNUM.GT.0) THEN
                DO 50 J=1,NFNUM
                   IF (NOF(J).EQ.NBRR) THEN
                      DO 40 JJ=J,NFNUM-1
                         NOF(JJ)=NOF(JJ+1)
                         MPTS(JJ)=MPTS(JJ+1)
                         DO 40 KK=1,IMX
```

```
                        QF(JJ,KK)=QF(JJ+1,KK)
                        PF(JJ,KK)=PF(JJ+1,KK)
40                   CONTINUE
                     NFNUM=NFNUM-1
                  ENDIF
50             CONTINUE
            ENDIF
            NWTYP(N)=1
            R(N)=RX
            WRITE (8,150) NBRR,R(N)
            GO TO 100
         ELSE IF (INDEX.GT.5) THEN
            WRITE (8,120) INDEX,NBRR
            INDEX=5
         ENDIF
         DO 55 I=1,INDEX
            II=5+(I-1)*2
            QFX(I)=DAL(II)
            PFX(I)=DAL(II+1)
55       CONTINUE
         IF (NFNUM.GT.0) THEN
            DO 20 J=1,NFNUM
               IF (NOF(J).EQ.NO(N)) THEN
                  MPTS(J)=INDEX
                  DO 10 K=1,INDEX
                     QF(J,K)=QFX(K)
                     PF(J,K)=PFX(K)
10                CONTINUE
                  JN=J
                  GO TO 35
               ENDIF
20          CONTINUE
         ENDIF
         IF (NFNUM.EQ.0) INIFAN=1
         NFNUM=NFNUM+1
         IF (NFNUM.GT.IMX) THEN
            WRITE (8,130) NO(N),PFX(1)
            NFNUM=NFNUM-1
            NWTYP(N)=1
            R(N)=PFX(1)
            GO TO 100
         ENDIF
         NWTYP(N)=1
         NOF(NFNUM)=NO(N)
         MPTS(NFNUM)=INDEX
C        **************
         NSKP(NFNUM)=0
         NFREG(NFNUM)=N
C        **************
         DO 30 K=1,INDEX
            QF(NFNUM,K)=QFX(K)
            PF(NFNUM,K)=PFX(K)
30       CONTINUE
C         INIFAN=1
         JN=NFNUM
35       WRITE (8,140) NBRR,TS,(QF(JN,K),PF(JN,K),K=1,INDEX)
C
      ELSE IF (ICODE.EQ.3) THEN
         CONTX=DAL(4)
         CONCX=DAL(5)
         HEATX=DAL(6)
         O2MINX=DAL(7)
         SMPO2X=DAL(8)
         HTPO2X=DAL(9)
         QCENTX=DAL(10)
         TPRX=DAL(11)
         IF (TPRX.LT.1.E-5) TPRX=1.E-5
         IF (INFLOW.GT.0) THEN
            DO 60 J=1,INFLOW
               IF (NCENT(J).EQ.NBRR) THEN
                  CONT(J)=CONTX
                  CONC(J)=CONCX
                  HEAT(J)=HEATX
                  O2MIN(J)=O2MINX
                  SMPO2(J)=SMPO2X
                  HTPO2(J)=HTPO2X
                  QCENT(J)=QCENTX
                  TPR(J)=TPRX
                  JN=J
                  GO TO 70
               ENDIF
60          CONTINUE
         ENDIF
         INFLOW=INFLOW+1
         IF (INFLOW.GT.IMX) THEN
            WRITE (8,260) NBRR
            GO TO 100
         ENDIF
         NCENT(INFLOW)=NBRR
         CONT(INFLOW)=CONTX
         CONC(INFLOW)=CONCX
         HEAT(INFLOW)=HEATX
         O2MIN(INFLOW)=O2MINX
         SMPO2(INFLOW)=SMPO2X
         HTPO2(INFLOW)=HTPO2X
         QCENT(INFLOW)=QCENTX
         TPR(INFLOW)=TPRX
         JN=INFLOW
70       NWTYP(N)=10
         DO 75 K=1,NB
            IF (NO(K).EQ.NBRR.AND.LA(K).GT.30.0)
     .         WRITE (8,220) NO(K),LA(K),TS
75       CONTINUE
         WRITE (8,160) NBRR,TS,TPR(JN),CONT(JN),CONC(JN),HEAT(JN),
     .                 O2MIN(JN),SMPO2(JN),HTPO2(JN),QCENT(JN)
C
      ELSE IF (ICODE.EQ.4) THEN
         IF (INFLOW.GT.0) THEN
            DO 90 I=1,INFLOW
               IF (NCENT(I).EQ.NBRR) THEN
                  DO 80 J=I,INFLOW
                     NCENT(J)=NCENT(J+1)
                     CONT(J)=CONT(J+1)
                     CONC(J)=CONC(J+1)
                     HEAT(J)=HEAT(J+1)
                     O2MIN(J)=O2MIN(J+1)
                     SMPO2(J)=SMPO2(J+1)
                     HTPO2(J)=HTPO2(J+1)
                     QCENT(J)=QCENT(J+1)
                     TPR(J)=TPR(J+1)
80                CONTINUE
                  INFLOW=INFLOW-1
                  NWTYP(N)=0
                  WRITE (8,170) NBRR,TS
                  GO TO 100
               ENDIF
90          CONTINUE
         ENDIF
      ELSE
         WRITE (8,180) ICODE, NBRR
```

```
      ENDIF
100   IREAD=1
      GO TO 1
C
C
110   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' BECAME AN ORDINARY ',
     .        'AIRWAY WITH',/,1X,'R = ',F8.3,' AT',F6.1,' MIN. AFTER ',
     .        'EVENT.')
120   FORMAT (/,1X,'* WARNING *  INVALID NUMBER OF DATA POINTS FOR A ',
     .        'FAN CURVE,',/,1X,'WHICH IN TIME TABLE IS 1 TO 5.')
130   FORMAT (/,1X,'* WARNING *  NUMBER OF FANS BECAME EXCESSIVE. ',
     .        'FAN IN',/,1X,'BRANCH ',I5,' IS TAKEN AS FIXED-P FAN OF',
     .        F7.3,' IN.W.G.')
140   FORMAT (/,1X,'* ATTENTION * THE FOLLOWING CHARACTERISTICS FOR ',
     .        'FAN ',I4,/,' ACTIVATED AT',F6.2,' MIN. AFTER EVENT.',//,
     .        5('  Q/1000   PF '),//,5(F8.0,F6.2))
150   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' BECAME A FAN BRANCH ',
     .        'WITH PRESSURE',/,' FIXED AT',F6.2,' IN. W.G. AT',F6.2,
     .        ' MIN. AFTER EVENT.')
160   FORMAT (/,1X,'* ATTENTION *  BRANCH',I5,' CAUGHT IN FIRE AT ',
     .        F6.2,' MIN.',/,1X,'LEADING TIME: ',F5.1,'  PARAMETERS AS',
     .        ' FOLLOWS:',//,T7,'FIXED-HEAT-INPUT',T30,'OXYGEN-RICH',
     .        T48,'FUEL-RICH',/,1X,'FLOWRATE',T14,'CONC.'
     .        ,T23,'HEAT',T34,'(%)',T43,'FUME P O2',T54,'HEAT PO2',T66,
     .        'REF. Q',//,1X,F8.0,T11,F8.2,T21,E8.3,T30,F7.2,T43,
     .        F7.2,T53,F7.2,T63,F9.0)
170   FORMAT (/,1X,'* ATTENTION *  FIRE',I5,' CEESED AT',F6.2,' MIN.',
     .        ' AFTER EVENT.')
180   FORMAT (/,' * ATTENTION *  INVALID CODE',I5,' FOR AIRWAY',I5,
     .        ' IGNORED. (CDCH)')
190   FORMAT (/,1X,'* ATTENTION *  SIMULATION STEP WAS CHANGED ',
     .        'FROM ',F6.2,' SEC.',/,' TO ',F6.2,' SEC. AT ',F6.2,
     .        ' MINS. AFTER EVENT.')
200   FORMAT (/,1X,'* WARNING *  INVALID SIMULATION STEP (',F6.2,' SEC',
     .        ') IGNORED.',/,1X,'IT MUST BE LESS THAN 30.0 SEC. (CDCH)')
210   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' NOT FOUND, DATA ',
     .        'IGNORED. (CDCH)')
220   FORMAT (/,1X,'* ATTENTION *  AIRWAY',I5,' OF LENGTH',F8.0,
     .        ' FT CONTAINS A',/,' HEAT/CONTAMINANT SOURCE AT ',F6.2,
     .        ' MIN. AFTER EVENT.',/,' IT IS ADVISED TO DIVIDE THE ',
     .        'AIRWAY INTO TWO AND LOCATE THE',/,' FIRE SOURCE IN ',
     .        'THE ONE SHORTER THAN 30 FT.')
230   FORMAT (/,1X,'* ATTENTION *  FIRE BRANCH ',I5,' NOT FOUND  AT ',
     .        F6.2,' MIN.',/,' AFTER SIMULATION, CHANGE IGNORED.')
240   FORMAT (/,1X,'* ATTENTION *  TOO MANY JDPP INPUT, DATA IGNORED.')
250   FORMAT (/,1X,'* ATTENTION *  JUNCTION',I6,' NOT FOUND AT',F6.2,
     .        ' MIN.',/,' AFTER EVNT, CHANGE IGNORED.')
260   FORMAT (/,1X,'* WARNING *  NUMBER OF FIRE SOURCES BECAME ',
     .        'EXCESSIVE. SPECIFIED',/,' CONDITION CHANGE IN TIME ',
     .        'TABLE IGNORED.')
300   FORMAT (/,1X,'THE FOLLOWING VARIABLES WERE EXPECTED TO BE READ',
     .        ' IN:',//)
310   FORMAT (1X,'TS,1,NBRR,R')
320   FORMAT (1X,'TS,2,NBRR,INDEX,QF,PF,QF,PF,QF,PF,QF,PF,QF,PF')
330   FORMAT (1X,'TS,3,NBRR,CONT,CONC,HEAT,02MIN,SMPO2,HTPO2,QCENT')
340   FORMAT (1X,'TS,4,NBRR')
350   FORMAT (1X,'TS,5,DELTAT')
360   FORMAT (1X,'TS,6,TOUT')
365   FORMAT (1X,'TS,7,JDPP')
370   FORMAT (/,1X,'THE CURRENT INPUT WAS:',//,80A1)
C
      END
C
C
      SUBROUTINE  DTTR (MARKL,MARKP)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       REARRANGING DETAILED DATA RECORDS FOR THE AIRWAY WHOSE AIRFLOW
C     REVERSED.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DO 90 I=1,NB
         DO 1 J=1,NJ
            IF (JNO(J).EQ.JS(I)) JU=J
1        CONTINUE
C
C     MARKP=0: A THERMALLY BALANCED SOLUTION HAS NOT BEEN REACHED YET
C              WITHIN A TIME INTERVAL.
C     MARKP=1: THE CALCULATION OF THE PRESENT TIME INTERVAL IS DONE.
C     Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C           INTERVAL.
C     QQ(I): TIME AVERAGED AIRFLOW RATE.
C
         IF (MARKP.EQ.0) QRS=QQ(I)
         IF (MARKP.EQ.1) QRS=Q(I)
         IF (QRS.LT.0.0) THEN
            MARKL=1
            IQ=IBTN(I)
            IF (IQ.GT.0) THEN
               IF (MARKP.EQ.1) THEN
                  IS=NSAC(I,2)
                  IF (IS.GT.0) THEN
                     DO 2 J=1,IS
                        DO 2 K=1,5
                           GBTN(J,K)=DPPA(IQ,J,K)
2                    CONTINUE
                     DO 6 J=1,IS
                        JP=IS-J+1
                        DO 4 K=1,5
                           DPPA(IQ,JP,K)=GBTN(J,K)
4                       CONTINUE
                        DPPA(IQ,JP,1)=LA(I)-DPPA(IQ,JP,1)
                        IF (J.EQ.1) THEN
                           DPPA(IQ,JP,2)=TAUXC(IQ)
                        ELSE
                           DPPA(IQ,JP,2)=TAUXA(IQ,J-1)
                        ENDIF
6                    CONTINUE
                     DO 7 J=1,IS
                        JP=IS-J+1
                        IF (J.EQ.1) THEN
                           TAUXA(IQ,JP)=TRD(I)
                        ELSE
                           TAUXA(IQ,JP)=GBTN(J-1,2)
                           IF (J.EQ.IS) TAUXC(IQ)=GBTN(J,2)
                        ENDIF
7                    CONTINUE
                     FMASS1=DPPA(IQ,1,5)
                     IF (IS.GE.2) THEN
                        DO 8 J=1,IS-1
                           DPPA(IQ,J,5)=DPPA(IQ,J+1,5)
8                       CONTINUE
                     ENDIF
                     IF (FMASS(IQ).LT.0.0) THEN
                        FK1=535.0/(460.0+(DPPA(IQ,IS,2)+TRD(I))/2.0)
```

```
                FMASS(IQ)=A(I)*DPPA(IQ,IS,1)*0.075*FK1
             ENDIF
             DPPA(IQ,IS,5)=FMASS(IQ)
             FMASS(IQ)=FMASS1
             RDPROP(I)=DPPA(IQ,1,3)
              RDCH4(I)=DPPA(IQ,1,4)
             RDOP(I)=DPPA(IQ,1,3)
             RCH4(I)=DPPA(IQ,1,4)
          ELSE
             IF (NWTYP(I).NE.10) THEN
                RDPROP(I)=PROP(JU)
                RDCH4(I)=PRCH4(JU)
                RDOP(I)=PROP(JU)
                RCH4(I)=PRCH4(JU)
             ENDIF
          ENDIF
       ELSE
          IS=NSACB(I,2)
          IF (IS.GT.0) THEN
             DO 10 J=1,IS
                DO 10 K=1,5
                   GBTN(J,K)=DPPB(IQ,J,K)
10           CONTINUE
             DO 30 J=1,IS
                JP=IS-J+1
                DO 20 K=1,5
                   DPPB(IQ,JP,K)=GBTN(J,K)
20              CONTINUE
                DPPB(IQ,JP,1)=LA(I)-DPPB(IQ,JP,1)
                IF (J.EQ.1) THEN
                   DPPB(IQ,JP,2)=TAUXD(IQ)
                ELSE
                   DPPB(IQ,JP,2)=TAUXB(IQ,J-1)
                ENDIF
30           CONTINUE
             DO 32 J=1,IS
                JP=IS-J+1
                IF (J.EQ.1) THEN
                   TAUXB(IQ,JP)=TTRD(I)
                ELSE
                   TAUXB(IQ,JP)=GBTN(J-1,2)
                   IF (J.EQ.IS) TAUXD(IQ)=GBTN(J,2)
                ENDIF
32           CONTINUE
             FMASS1=DPPB(IQ,1,5)
             IF (IS.GE.2) THEN
                DO 35 J=1,IS-1
                   DPPB(IQ,J,5)=DPPB(IQ,J+1,5)
35              CONTINUE
             ENDIF
             IF (FMASS(IQ).LT.0.0) THEN
                FK1=535.0/(460.0+(DPPB(IQ,IS,2)+TTRD(I))/2.0)
                FMASS(IQ)=A(I)*DPPB(IQ,IS,1)*0.075*FK1
             ENDIF
             DPPB(IQ,IS,5)=FMASS(IQ)
             FMASS(IQ)=FMASS1
             DO 40 J=1,IS
                DPPA(IQ,J,5)=DPPB(IQ,J,5)
40           CONTINUE
             RDPROP(I)=DPPB(IQ,1,3)
             RDCH4(I)=DPPB(IQ,1,4)
             RDOP(I)=DPPB(IQ,1,3)
             RCH4(I)=DPPB(IQ,1,4)
          ELSE
             IF (NWTYP(I).NE.10) THEN
                      RDPROP(I)=PROPS(JU)
                      RDCH4(I)=PRCH4S(JU)
                      RDOP(I)=PROPS(JU)
                      RCH4(I)=PRCH4S(JU)
                   ENDIF
                ENDIF
             ENDIF
          ENDIF
          TRDA=TRD(I)
          TTRDA=TTRD(I)
          TRD(I)=TJS(I)
          TTRD(I)=TTJS(I)
          TJS(I)=TRDA
          TTJS(I)=TTRDA
          JE=JS(I)
          JS(I)=JF(I)
          JF(I)=JE
          DZRD(I)=-DZRD(I)
          FRNVP(I)=-FRNVP(I)
          FFRNVP(I)=-FFRNVP(I)
          QTP(I)=-QTP(I)
          QQ(I)=-QQ(I)
          Q(I)=-Q(I)
          NUMCT=0
          DO 60 J=1,INFLOW
             IF (NO(I).EQ.NCENT(J)) THEN
                DO 50 JJ=1,NB
                   IF (JF(JJ).EQ.JNO(JU)) THEN
                      NUMCT=NUMCT+1
                      SFUME=SFUME+RDPROP(JJ)
                   ENDIF
50              CONTINUE
                PROP(JU)=SFUME/NUMCT
             ENDIF
60        CONTINUE
C          DO 70 J=1,JAJ
C             IF (JS(I).EQ.JSTAR(J)) THEN
C                PROP(JU)=0.
C             ENDIF
C70        CONTINUE
C
C     NREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (QQ) IN AIRWAY
C              NO(I) IN THE PRESENT TIME INTERVAL.
C     NNREV(I): NUMBER OF CHANGES IN AIRFLOW DIRECTION (Q) IN AIRWAY
C               NO(I) SINCE TIME ZERO.
C
          NREV(I)=NREV(I)+1
       ENDIF
       IF (MARKP.EQ.1.AND.Q(I)*QTP(I).LT.0.0) NNREV(I)=NNREV(I)+1
       IF (ABS(QQ(I)).LT.1.E-5) THEN
          WRITE (8,200) NO(I),QQ(I)
          FACT=1.0
          IF (QQ(I).LT.0.0) FACT=-1.0
          QQ(I)=FACT*1.E-5
       ENDIF
       IF (ABS(Q(I)).LT.1.E-5) THEN
          WRITE (8,200) NO(I),Q(I)
          FACT=1.0
          IF (Q(I).LT.0.0) FACT=-1.0
          Q(I)=FACT*1.E-5
       ENDIF
90    CONTINUE
C
      IF (MARKL.NE.0) THEN
         L=0
```

```
          M=0
          N=1
          DO 120 I=1,MAXJ
             K=L
             DO 100 J=1,NB
                IF (JS(J).EQ.I) THEN
                   L=L+1
                   NGOUT(L)=J
                ENDIF
100          CONTINUE
             LOUT(N)=L
             MM=M
             DO 110 J=1,NB
                IF (JF(J).EQ.I) THEN
                   M=M+1
                   NGIN(M)=J
                ENDIF
110          CONTINUE
             MMIN(N)=M
             IF (MM.NE.M.OR.K.NE.L) THEN
                JNOL(N)=I
                N=N+1
             ENDIF
120       CONTINUE
          DO 140 I=1,NJ
             DO 130 J=1,NJ
                IF (JNOL(I).EQ.JNO(J)) THEN
                   JLR(I)=J
                   GO TO 140
                ENDIF
130          CONTINUE
140       CONTINUE
       ENDIF
C
200    FORMAT (//,1X,'* ATTENTION *  AIRWAY',I5,' HAD ITS Q OR QQ',
     .       ' DOWN TO',F6.3,/,' FT3/MIN. DURING ITERATIONS. IT WAS ',
     .       'SET TO 1E-5 FT3/MIN.',/,' TO AVOID POSSIBLE DIVISION ',
     .       'FAULT DURING ITERATION.')
C
       RETURN
       END
C
C
       SUBROUTINE  CDJN (ISQC,MARKS,NSTOP)
C
C      ------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) EVALUATION OF THERMAL STATES IN JUNCTIONS.
C        2) DEVELOPING A NEW DATA RECORD INTO AIRWAYS (A) WHOSE BEGINNING
C           JUNCTION HAS DRASTIC CONDITION CHANGES; OR (B) WHERE AIRFLOW
C           REVERSAL HAPPENED; OR (C) WHERE A FIRE SOURCE EXISTS; OR (D)
C           WHICH GOT DETAILED RECORDS IN PREVIOUS INTERVALS ALREADY.
C
C      ------------------------------------------------------------------
C
       INCLUDE 'CMMN1.DAT'
C
       CP=0.24
       DO 60 I=1,NJ
          K=JLR(I)
          JY=1
          IF (I.GT.1) JY=LOUT(I-1)+1
          JZ=LOUT(I)
          DO 50 J=JY,JZ
             NW=NGOUT(J)
             IF (NW.LE.0) write (8,*) 'INTERNAL ERROR, NW=',NW
C
C      NWTYP(I)=10 INDICATES THAT AIRWAY NO(I) CONTAINS A FIRE SOURCE.
C      NSAC(I,2): NUMBER OF CONTROL VOLUMES IN AIRWAY NO(I).
C      IBTN(I) STORES THE ADDRESS OF DATA RECORDS FOR AIRWAY NO(I) IN
C               ARRAY DPPA, DPPB ETC.
C
             IF (A(NW).GE.100000.0) GO TO 50
             IF (MARKS.NE.0.AND.NWTYP(NW).NE.10) THEN
                IF (JCH(K).EQ.0) GO TO 50
C
C        WHEN THE NEWLY OBTAINED DATA NEED TO BE UPDATED AGAIN IN THE
C      PRESENT TIME INTERVAL, THE DATA OBTAINED IN THE PREVIOUS SEARCH
C      IN THE INTERVAL ARE ABANDONED TO AVOID DUPLICATED DATA
C      RECORDS.
C
                IF (IBTN(NW).GT.0) THEN
                   IZ=IBTN(NW)
                   NSAC(NW,2)=NSACB(NW,2)
                   DO 5 IX=1,10
                      DO 5 IY=1,5
                         DPPA(IZ,IX,IY)=DPPB(IZ,IX,IY)
5                  CONTINUE
                ENDIF
             ELSE IF (NWTYP(NW).NE.10) THEN
C
C      NON-QUALIFIED AIRWAYS GET NO DETAILED DATA RECORDS AND ARE IGNORED
C
                IF (JDP.GT.0) THEN
                   DO 7 IX=1,JDP
                      IF (NO(NW).EQ.JDPP(IX)) GO TO 8
7                  CONTINUE
                ENDIF
                IF (JCH(K).EQ.0.AND.IBTN(NW).EQ.0) GO TO 50
             ENDIF
C
C   ** CHECK TO SEE IF A FIRE SOURCE EXISTS IN THE AIRWAY.
C
C
8            DO 10 L=1,INFLOW
                IF (NCENT(L).EQ.NO(NW)) GO TO 20
10           CONTINUE
             CONTAM=0.0
             CONTQ=0.0
             HEATAD=0.0
             GO TO 30
C
20           FACT=1.0
             IF (QCENT(L).GT.10.0) THEN
                Q1=QQ2(NW)
                IF (Q1.LT.QCENT(L)) Q1=Q1+QCENT(L)
                FACT=1.0+(QQ2(NW)-QCENT(L))/Q1
             ENDIF
             IF (ABS(CONT(L)).GT.1.E-6) THEN
                CONTAM=CONT(L)*CONC(L)/100.0
                CONTQ=CONT(L)
             ELSE
                CONTAM=0.0
                CONTQ=0.0
C               **********
             ENDIF
             O2BEH(L)=(0.21-PROP(K)-CONTAM/QQ2(NW))*100.0
             HEATAD=HEAT(L)*FACT
C
```

```
            IF (O2MIN(L).GT.1.0.AND.HEAT(L).GE.0.0) THEN
               O2SCL=O2MIN(L)+(21.0-O2MIN(L))*(FACT-1.0)
               TACCA=TACC+DELTAT
               IF (TACCA.GT.TPR(L)*60.0) TACCA=TPR(L)*60.0
               PROPA=PROP(K)
               O2CONS=((0.21-PROPA-O2SCL/100.0)*TACCA/
     .                (TPR(L)*60.0))*QQ2(NW)
               IF (O2CONS.LT.0.0) O2CONS=0.0
               O2BEH(L)=(0.21-PROPA-O2CONS/QQ2(NW))*100.0
               HEATAD=O2CONS*437.0
               CONTAM=PROPA*QQ2(NW)+O2CONS
            ENDIF
C
            IF (SMPO2(L).GT.1.E-6.AND.HEAT(L).GE.0.0) THEN
               O2LT=0.21-0.21*(TACC+DELTAT)/(TPR(L)*60.0)
               IF (O2LT.LT.0.0) O2LT=0.0
               PROPA=PROP(K)
               O2CONS=(0.21-O2LT-PROPA)*QQ2(NW)
               IF (O2CONS.LT.0.0) O2CONS=0.0
               CONTAM=(0.21-O2LT)*QQ2(NW)
               IF ((0.21-O2LT).LT.PROPA) CONTAM=PROPA*QQ2(NW)
               O2BEH(L)=O2LT*100.0
               IF (FACT.LE.1.0) THEN
                  HTSCL=HTPO2(L)
               ELSE
                  HTSCL=HTPO2(L)/FACT
               ENDIF
               CONTQ=0.0
C              *********
               HEATAD=O2CONS*HTSCL
            ENDIF
C
30          TK=T(K)
            IF (NWTYP(NW).EQ.10) TK=TAS(K)
            VART=(9900.0+TK)**2+2.0*HEATAD/(QQ2(NW)*2.4E-5*DR)
            IF (VART.LT.9.4E7) THEN
C                    *****
               WRITE (8,100) NO(NW)
               NSTOP=6
               RETURN
            ENDIF
            TJS(NW)=-9900.0+SQRT(VART)
            IF (CONCT.EQ.1) TJS(NW)=TJSS(NW)
            IF (TRD(NW).LT.(-200.0).OR.TRD(NW).GT.5000.0) THEN
               WRITE (8,100) NO(NW)
               NSTOP=6
               RETURN
            ENDIF
C           ***************************************
            IF (NWTYP(NW).EQ.10) THEN
               TRD(NW)=TJS(NW)
               RDPROP(NW)=(PROP(K)*QQ2(NW)+CONTAM)/(QQ2(NW)+CONTQ)
               RDCH4(NW)=PRCH4(K)
               HTAD(L)=(TJS(NW)-T(K))*0.24*QQ2(NW)*DR
               GO TO 50
            ENDIF
C
C
            NTAUXC=0
            IF (NSAC(NW,1).EQ.0) THEN
C
C     NSAC(I,1)=1 INDICATES THAT THE FUME FRONT IS WITHIN AIRWAY NO(I).
C
               NTAUXC=1
               NSAC(NW,1)=1
```

```
            ENDIF
            NSAC(NW,2)=NSAC(NW,2)+1
            NS=NSAC(NW,2)
            IF (IBTN(NW).EQ.0) THEN
               ISQC=ISQC+1
               IF (ISQC.LT.NMZ) THEN
                  IBTN(NW)=ISQC
               ELSE
C
C       WHEN THE CAPACITIES OF DATA RECORDING ARRAYS ARE SATURATED, LATE
C     REQUEST WILL BE TURNED DOWN AND A MESSAGE WILL BE GIVEN.
C
                  NSAC(NW,1)=0
                  NSAC(NW,2)=0
                  IF (IAC.GT.0) THEN
                     DO 35 I1=1,IAC
                        IF (IOMIT(I1).EQ.NW) GO TO 36
35                   CONTINUE
                  ENDIF
                  IF (IAC.GE.IMX) GO TO 50
                  IAC=IAC+1
                  I1=IAC
36                IOMIT(I1)=NW
                  ROMIT(I1,1)=T(K)
                  ROMIT(I1,2)=PROP(K)
                  ROMIT(I1,3)=PRCH4(K)
                  GO TO 50
               ENDIF
            ENDIF
            NQ=IBTN(NW)
C
            TS=DELTAT/60.0
            IF (TLEFT(K).GE.0.0) TS=TLEFT(K)+0.00001
            DPPA(NQ,NS,5)=DR*QQ2(NW)*TS
            IF (NS.GT.1.AND.DELTAT/60.0.GT.TS) THEN
               DPPA(NQ,NS-1,5)=DPPB(NQ,NS-1,5)+DR*QQ2(NW)*
     .                         (DELTAT/60.0-TS)
            ENDIF
            TAUXA(NQ,NS)=TJS(NW)
            CALL COLUMN (TJS(NW),SEG,NW,NQ,NS)
            FE=SEG/LA(NW)
            SEGZ=QQ(NW)*TS*(TMRD(NW)+460.0)/(A(NW)*(TR+460.0))
            FA=XNEW(NW)*SEGZ/CP
C           **************************************************
C            FA=XNEW(NW)*SEG/CP
            IF (FA.LT.25.0) THEN
               FAS=EXP(-FA)
            ELSE
               FAS=0.0
            ENDIF
C
C  AIR TEMP. OF THE C.V. AT ITS NEWLY ARRIVED POSITION IS ESTIMATED.
C
            DPPA(NQ,NS,2)=TROCK(NW)+(TJS(NW)-TROCK(NW))*FAS-
     .                    (DZRD(NW)*FE/(2.0*778.26*CP))*(1.0+FAS)
C
C     DATA RECORDED IN DPPA: (1) LOCATION, (2) TEMP. (3) CH4,
C                            (4) FUME, (5) AIR MASS IN C.V..
C
            IF (NTAUXC.EQ.1) TAUXC(NQ)=TROCK(NW)+(TTJS(NW)-TROCK(NW))*
     .                         FAS-(DZRD(NW)*FE/(2.0*778.26*CP))*(1.+FAS)
            DPPA(NQ,NS,1)=SEG
            DPPA(NQ,NS,3)=(PROP(K)*QQ2(NW)+CONTAM)/(QQ2(NW)+CONTQ)
            DPPA(NQ,NS,4)=(PRCH4(K)*QQ2(NW)+CH4V(NW)*FE)/
     .                    (QQ2(NW)+CH4V(NW)*FE)
```

```
50        CONTINUE
60     CONTINUE
C
       RETURN
C
100    FORMAT (//,1X,'* ERROR *  ABNORMAL TEMPERATURE WAS RESULTED ',
      .        'IN AIRWAY ',I5,/,' ADJUST HEAT INPUT AND/OR QCENT.')
       END
C
C
       SUBROUTINE COLUMN (T1,X3,I,JQ,JP)
C
C      ---------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C         DETERMINING THE ADVANCING DISTANCE OF AIR COLUMNS.
C
C      ---------------------------------------------------------------
C
       INCLUDE 'CMMN1.DAT'
C
       IF (DPPA(JQ,JP,5).LT.3.0) THEN
          X3=0.1
          RETURN
       ENDIF
       KK=0
       X1=1.0
       X2=100.0
       CP=0.2376+0.000024*TMRD(I)
       C1=TROCK(I)+460.0
       C2=T1-TROCK(I)
       C3=-HK(I)*DCOAGE(I)*O(I)**2/(DR*QQ(I)*CP*120.0*A(I))
       AC=DPPA(JQ,JP,5)*53.34/(144.0*14.7*A(I))
10     EXPR=0.0
       IF (C3*X1.GT.-25.0) EXPR=EXP(C3*X1)
       XXX=(C1+C2*EXPR)/(C1+C2)
       ALOGR=0.0
       IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
       QM1=X1/C1-ALOGR/(C1*C3)
       IF (QM1.GT.AC) THEN
          X1=X1/5.0
          IF (X1.GT.1.E-3) GO TO 10
       ENDIF
20     EXPR=0.0
       IF (C3*X2.GT.-25.0) EXPR=EXP(C3*X2)
       XXX=(C1+C2*EXPR)/(C1+C2)
       ALOGR=0.0
       IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
       QM2=X2/C1-ALOGR/(C1*C3)
       IF (QM2.LT.AC) THEN
          X2=X2*2.0
          IF (X2.LT.10000.0) GO TO 20
       ENDIF
25     X4=X2/5.0
       EXPR=0.0
       IF (C3*X4.GT.-25.0) EXPR=EXP(C3*X4)
       XXX=(C1+C2*EXPR)/(C1+C2)
       ALOGR=0.0
       IF(ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
       QM4=X4/C1-ALOGR/(C1*C3)
       IF (QM4.GT.AC) THEN
          X2=X4
          QM2=QM4
          IF (X2.GT.10.0) GO TO 25
       ENDIF
```

```
30     X3=X1+(X2-X1)*(AC-QM1)/(QM2-QM1)
       EXPR=0.0
       IF (C3*X3.GT.-25.0) EXPR=EXP(C3*X3)
       XXX=(C1+C2*EXPR)/(C1+C2)
       ALOGR=0.0
       IF (ABS(XXX-1.0).GT.1.E-4) ALOGR=ALOG(XXX)
       QM3=X3/C1-ALOGR/(C1*C3)
       IF (ABS(QM3-AC).GT.(5.E-4*AC)) THEN
          IF (QM3.LE.AC) THEN
             X1=X3
             QM1=QM3
          ELSE
             X2=X3
             QM2=QM3
          ENDIF
          KK=KK+1
          IF (KK.LE.40) GO TO 30
       ENDIF
       IF (X3.LT.0.1) X3=0.1
C
C
       RETURN
       END
C
C
       SUBROUTINE  ADPT (IS,IQ,MARKS,I)
C
C      ------------------------------------------------------------------
C
C        SUBROUTINE PURPOSES:
C        1) ADVANCING EXISTING C.V. IN AIRWAYS, UPDATING RELEVANT DATA.
C        2) UPDATING AIRWAY ENDING CONDITIONS.
C        3) CONDENSING DATA STORAGE ARRAYS IF THERE IS A C.V. EXCEEDING
C           THE AIRWAY ENDING.
C
C      ------------------------------------------------------------------
C
       INCLUDE 'CMMN1.DAT'
C
C      IS: NUMBER OF C.V. IN AIRWAY NO(I).
C
       CP=0.24
       IF (NSAC(I,2).EQ.0) RETURN
       DO 5 K=1,NJ
          IF (JS(I).EQ.JNO(K)) GO TO 8
5      CONTINUE
8      IF (MARKS.GT.0.AND.JCH(K).EQ.0) RETURN
       IF (IS.LE.0) GO TO 50
       IF (DPPA(IQ,IS+1,1).GT.LA(I)) GO TO 50
C
C        "MARKS" ACTS AS A TIMER. WHEN MARKS=0, THIS SUB. ADVANCES ALL
C      EXISTING C.V., WHILE MARKS>0, IT SEARCHES FOR THOSE C.VS. WHICH
C      WERE GENERATED IN THE PRESENT INTERVAL AND TRAVELLED MORE THAN
C      ONE AIRWAY IN THE PRESENT INTERVAL THEN UPDATES THE RELEVANT DATA.
C
C   ** MARKS=0, ADVANCING ALL EXISTING DATA POINTS, STARTING WITH THE C.V
C      CLOSEST TO THE BEGINNING JUNCTION OF THE AIRWAY.
C
       SEGZ=QQ(I)*DELTAT*(TMRD(I)+460.0)/(60.0*A(I)*(TR+460.0))
       FA=XNEW(I)*ABS(SEGZ)/CP
       FAS=0.0
       IF (FA.LT.25.0) FAS=EXP(-FA)
C
       DO 40 K=1,IS
          JA=IS-K+1
```

```
         SEG=DPPA(IQ,JA+1,1)-DPPB(IQ,JA+1,1)
         IF (ABS(SEG).LT.1.E-3) SEG=0.001
         VL=SEG/LA(I)
         T0=TAUXB(IQ,JA)
         T2=DPPB(IQ,JA,2)
         IRV=0
         IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .      TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .      TAUXB(IQ,JA).LT.TROCK(I))) THEN
            T0=DPPB(IQ,JA,2)
            T2=TAUXB(IQ,JA)
            IRV=1
         ENDIF
         T1=TROCK(I)+(T0-TROCK(I))*FAS-(DZRD(I)*VL/(2*778.26*CP))
     .      *(1.0+FAS)
         TAUXA(IQ,JA)=T1
         CALL COLUMN (T1,SEG1,I,IQ,JA)
         SEG=DPPA(IQ,JA+1,1)+SEG1-DPPB(IQ,JA,1)
         VL=SEG/LA(I)
         DPPA(IQ,JA,2)=TROCK(I)+(T2-TROCK(I))*FAS-
     .                 (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
         IF (JA.EQ.1) TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-
     .                 (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
         IF (IRV.EQ.1) THEN
            TBW=DPPA(IQ,JA,2)
            DPPA(IQ,JA,2)=TAUXA(IQ,JA)
            TAUXA(IQ,JA)=TBW
         ENDIF
         DPPA(IQ,JA,1)=DPPB(IQ,JA,1)+SEG
         SE=ABS(DPPA(IQ,JA,1)-DPPB(IQ,JA,1))
         DPPA(IQ,JA,4)=(DPPB(IQ,JA,4)*QQ(I)+CH4V(I)*SE/LA(I))/
     .                 (QQ(I)+CH4V(I)*SE/LA(I))
         IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 50
 40   CONTINUE
      VL=(DPPA(IQ,1,1)-DPPB(IQ,1,1))/LA(I)
      TAUXC(IQ)=TROCK(I)+(TAUXD(IQ)-TROCK(I))*FAS-(DZRD(I)*VL/
     .          (2.0*778.26*CP))*(1.0+FAS)
 50   ISS=NSAC(I,2)
      DO 52 K=1,ISS
         JA=ISS-K+1
         IF (DPPA(IQ,JA,1).GE.LA(I)) GO TO 54
 52   CONTINUE
      IF (NSAC(I,1).EQ.1) RETURN
C
C     IF NO C.V. EXCEEDS THE ENDING, THEN SUBROUTINE RETURNS AFTER
C     UPDATING AIRWAY ENDING CONDITIONS.
C
      do 51 k=1,iss
         ja=iss-k+1
         if (dppa(iq,ja,3).ne.0) go to 53
 51   continue
      rdprop(i)=0.
      return
 53   SEG=LA(I)-DPPA(IQ,1,1)
      SSL=LA(I)-DPPB(IQ,1,1)
      IF (SSL.LE.SEG) SSL=SEG
      FA=XNEW(I)*SEG/CP
      IF (FA.LE.25.0) THEN
         FAS=EXP(-FA)
      ELSE
         FAS=0.0
      ENDIF
      TT=DPPA(IQ,1,2)
      ENDT=TROCK(I)+(TT-TROCK(I))*FAS-(DZRD(I)*SEG/
     .     (2*778.26*CP*LA(I)))*(1.0+FAS)
      TRD(I)=TTRD(I)+(ENDT-TTRD(I))*(SSL-SEG)/SSL
      IF (CONCT.EQ.1) TRD(I)=TRDD(I)
      RDPROP(I)=RDOP(I)+(DPPA(IQ,1,3)-RDOP(I))*(SSL-SEG)/SSL
      RETURN
C
C  ** C.V. EXCEEDS THE ENDING, UPDATE OF AIRWAY ENDING CONDITIONS
C     IS CALLED FOR.
C     TLEFT(J): TIME LEFT AT JUNCTION JNO(J) FOR WAVE FRONT (C.V.)
C               TO TRAVEL INTO THE DOWN-STREAM AIRWAY.
C
 54   DO 56 LL=1,NJ
         IF (IABS(JF(I)).EQ.IABS(JNO(LL))) KL=LL
         IF (IABS(JS(I)).EQ.IABS(JNO(LL))) ML=LL
 56   CONTINUE
      FL=DPPA(IQ,JA,1)-LA(I)
      TK=DELTAT*FL/((DPPA(IQ,JA,1)-DPPB(IQ,JA,1))*60.0)
      IF (TK.GT.TLEFT(KL)) TLEFT(KL)=TK
      IF (NSAC(I,1).NE.(-1)) NSAC(I,1)=-1
C
C     NSAC(I,1)=-1: AIRWAY NO(I) IS A FUME-FILLED ONE.
C     RDPROP(I): FUME CONCENTRATION AT AIRWAY ENDING.
C     RDCH4(I): CH4 CONCENTRATION AT AIRWAY ENDING.
C     DPPA,DPPB: (1) DISTANCE, (2) TEMP. (3) FUME, (4) CH4,
C                (5) AIR MASS IN C.V.
C
      RDCH4A=DPPA(IQ,JA,4)-CH4V(I)*(DPPA(IQ,JA,1)-LA(I))/
     .       (LA(I)*QQ(I))
C
C     NSACB: BACKUP ARRAY OF NSAC. NSACB(I,2) INDICATES THE NUMBER
C            OF C.V. IN AIRWAY NO(I) IN THE LAST INTERVAL.
C
      IF (JA.EQ.NSAC(I,2)) THEN
         SECT=LA(I)
         TLAST=TJS(I)
         RDF=DPPA(IQ,JA,3)-PROP(ML)
         IF (NWTYP(I).EQ.10) RDF=0.0
         SSL=DPPA(IQ,JA,1)
      ELSE
         SECT=LA(I)-DPPA(IQ,JA+1,1)
         TLAST=TAUXA(IQ,JA)
         RDF=DPPA(IQ,JA,3)-DPPA(IQ,JA+1,3)
         SSL=DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1)
      ENDIF
      RDPROP(I)=DPPA(IQ,JA,3)-RDF*FL/SSL
      RDCH4(I)=RDCH4A
C
      IF ((TAUXB(IQ,JA).LT.DPPB(IQ,JA,2).AND.TAUXB(IQ,JA).GT.
     .   TROCK(I)).OR.(TAUXB(IQ,JA).GT.DPPB(IQ,JA,2).AND.
     .   TAUXB(IQ,JA).LT.TROCK(I))) THEN
C
         FL1=DPPB(IQ,JA,1)-DPPB(IQ,JA+1,1)
         FLL=(DPPA(IQ,JA,1)-LA(I))/(DPPA(IQ,JA,1)-DPPA(IQ,JA+1,1))
         TLAST=DPPB(IQ,JA,2)+(TAUXB(IQ,JA)-DPPB(IQ,JA,2))*FLL
         SECT=LA(I)-DPPB(IQ,JA,1)+FL1*FLL
      ENDIF
      FA=XNEW(I)*SECT/CP
      FAS=0.0
      IF (FA.LT.25.0) FAS=EXP(-FA)
      VL=SECT/LA(I)
      TRD(I)=TROCK(I)+(TLAST-TROCK(I))*FAS-
     .       (DZRD(I)*VL/(2*778.26*CP))*(1.0+FAS)
      IF (CONCT.EQ.1) TRD(I)=TRDD(I)
C
C       WHEN A C.V EXCEEDS AIRWAY ENDING, THE DATA RELATED TO THAT C.V.
C     ARE DISCARDED AND DATA RECORDING ARRAYS ARE CONDENSED.
```

```
C
      JAA=JA+1
      DO 80 K=JAA,11
         IF (K.EQ.JAA) TAUXC(IQ)=TAUXA(IQ,K-1)
         IF (K.LE.10) THEN
            KA=K-JAA+1
            TAUXA(IQ,KA)=TAUXA(IQ,K)
            DO 60 L=1,5
               DPPA(IQ,KA,L)=DPPA(IQ,K,L)
60          CONTINUE
         ELSE
            TAUXA(IQ,10)=0.0
            DO 70 L=1,5
               DPPA(IQ,10,L)=0.0
70          CONTINUE
         ENDIF
80    CONTINUE
      NSAC(I,2)=NSAC(I,2)-JA
C
      RETURN
      END
C
C
C
      SUBROUTINE  PARAM
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) EVALUATION OF MEAN AND MEAN-SQUARE TEMP. IN AIRWAYS.
C       2) CALCULATION OF VARIED AIRFLOW RESISTANCE AND THE INTEGRAL
C          OF AIR DENSITY*ELEVATION VARIATION OF EACH AIRWAY FOR
C          ASSESSING NVP.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DO 30 I=1,NB
         IF (NWTYP(I).EQ.10) THEN
            TMRD(I)=TRD(I)
            FRNVP(I)=2116.0*DZRD(I)/(53.352*(TRD(I)+460.0))
            R(I)=RSTD(I)*(TMRD(I)+460.0)**2/(TR+460.0)**2
         ENDIF
         IF (IBTN(I).EQ.0) GO TO 30
         FRNVP(I)=0.0
         IF (NSAC(I,2).EQ.0) THEN
C
C     SHORT AIRWAYS ARE EVALUATED BASED ON CONDITIONS AT ITS TWO ENDS.
C
            F1=ABS(TJS(I)-TROCK(I))
            F2=ABS(TRD(I)-TROCK(I))
            FF=MAX(F1,F2)
            IF (FF.LT.0.1) THEN
               TMRD(I)=TJS(I)
               TMSQR=(TJS(I)+460.0)**2
               FRNVP(I)=2116.0*DZRD(I)/(53.352*(TJS(I)+460.0))
            ELSE
               TTROCK=TROCK(I)
               IF (ABS(TJS(I)-TTROCK).LT.0.01) TTROCK=TJS(I)-5.0
               TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
               IF (TT.LT.0.01) THEN
                  TTROCK=AMIN1(TJS(I),TRD(I))-5.0
                  TT=(TRD(I)-TTROCK)/(TJS(I)-TTROCK)
               ENDIF
               IF (ABS(TT-1.0).LT.1.E-3) THEN
                  TMRD(I)=TJS(I)
                  TMSQR=(TJS(I)+460.0)**2
                  FRNVP(I)=2116.0*DZRD(I)/(53.352*(TJS(I)+460.0))
               ELSE
C
C     TMRD(I): MEAN AIR TEMP. IN AIRWAY NO(I).
C     TMSQR: MEAN-SQUARE TEMP. IN THE SAME AIRWAY.
C
                  A1=TRD(I)-TTROCK
                  A2=TJS(I)-TTROCK
                  TJ=TTROCK+460.0
                  TMSQR=TJ*TJ+2.0*TJ*(TRD(I)-TJS(I))/ALOG(TT)
     .                  +(A1*A1-A2*A2)/(2.0*ALOG(TT))
                  TMRD(I)=TTROCK+(TRD(I)-TJS(I))/ALOG(TT)
                  YQ=ALOG((TRD(I)+460.0)/(TJS(I)+460.0))/ALOG(TT)
                  FRNVP(I)=2116.0*DZRD(I)*(1.0-YQ)/(53.352*TJ)
               ENDIF
            ENDIF
         ELSE
C
C       PARAMETER EVALUATIONS FOR AIRWAYS WITH DETAILED DATA RECORDS
C     ARE PROCESSED POINT BY POINT.
C
            NZ=NSAC(I,2)
            IQ=IBTN(I)
            TMRD(I)=0.0
            TMSQR=0.0
            DO 10 J=1,NZ+1
               IF (J.EQ.1) THEN
                  D2=LA(I)-DPPA(IQ,J,1)
                  C1=TAUXC(IQ)
                  C2=TRD(I)
               ELSE
                  D2=DPPA(IQ,J-1,1)-DPPA(IQ,J,1)
                  C1=TAUXA(IQ,J-1)
                  C2=DPPA(IQ,J-1,2)
               ENDIF
               IF (D2.LT.1.E-5) D2=0.01
               F1=ABS(C1-TROCK(I))
               F2=ABS(C2-TROCK(I))
               FF=MAX(F1,F2)
               IF (FF.LT.0.1) THEN
                  TMRD(I)=TMRD(I)+C1*D2
                  TMSQR=TMSQR+D2*(C1+460.0)**2
                  FRNVP(I)=FRNVP(I)+2116.0*DZRD(I)*D2/
     .                     (53.352*(C1+460.0))
               ELSE
                  TTROCK=TROCK(I)
                  IF (ABS(C1-TTROCK).LT.0.01) TTROCK=C1-5.0
                  TT=(C2-TTROCK)/(C1-TTROCK)
                  IF (TT.LT.0.01) THEN
                     TTROCK=AMIN1(C1,C2)-5.0
                     TT=(C2-TTROCK)/(C1-TTROCK)
                  ENDIF
                  IF (ABS(TT-1.0).LT.1.E-3) THEN
                     TMRD(I)=TMRD(I)+C1*D2
                     TMSQR=TMSQR+D2*(C1+460.0)**2
                     FRNVP(I)=FRNVP(I)+2116.*DZRD(I)*D2/
     .                        (53.352*(C1+460.))
                  ELSE
                     TMRD(I)=TMRD(I)+(TTROCK+(C2-C1)/ALOG(TT))*D2
                     A1=C2-TTROCK
                     A2=C1-TTROCK
                     TJ=TTROCK+460.0
```

```
                  TMSQR=TMSQR+D2*(TJ*TJ+2.0*TJ*(C2-C1)/ALOG(TT)
     .                  +(A1*A1-A2*A2)/(2.0*ALOG(TT)))
                  YQ=ALOG((C2+460.0)/(C1+460.0))/ALOG(TT)
                  FRNVP(I)=FRNVP(I)+2116.0*DZRD(I)*D2*(1.0-YQ)/
     .                      (53.352*TJ)
               ENDIF
            ENDIF
10       CONTINUE
         TMRD(I)=TMRD(I)/LA(I)
         TMSQR=TMSQR/LA(I)
         FRNVP(I)=FRNVP(I)/LA(I)
      ENDIF
C
C
C       AIRFLOW RESISTANCE IS ASSUMED VARYING LINEARLY WITH
C     MEAN-SQUARE TEMP. DUE TO CHANGED AIRFLOW VELOCITY.
C
         IF (NWTYP(I).NE.1) R(I)=RSTD(I)*TMSQR/(TR+460.0)**2
30    CONTINUE
C
      RETURN
      END
C
C
C
      SUBROUTINE  DISP (MARKX,ISQC,IBB)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DISCARDING THE TENTATIVELY OBTAINED DATA IN THE PREVIOUS
C          ITERATION TO AVOID UNNECESSARY RECORDS.
C       2) RESTORATION OF INITIAL CONDITIONS EXCEPT FOR AIRFLOW RATES.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
C     ABANDON THE DATA FOR AIRWAYS WHICH GOT THEIR DETAILED DATA RECORDS
C     IN THE PREVIOUS ITERATION.
C
      DO 40 LRW=1,NB
         IF (IBTN(LRW).GT.0.AND.NSACB(LRW,1).EQ.0) THEN
            ISQCX=IBTN(LRW)
            DO 10 IV=1,NB
               IF (IBTN(IV).GE.ISQCX) THEN
                  IBTN(IV)=IBTN(IV)-1
               ENDIF
10          CONTINUE
            DO 20 IV=ISQCX,ISQC
               TAUXC(IV)=TAUXC(IV+1)
               DO 20 IU=1,10
                  TAUXA(IV,IU)=TAUXA(IV+1,IU)
                  DO 20 IW=1,5
                     DPPA(IV,IU,IW)=DPPA(IV+1,IU,IW)
20          CONTINUE
            IBTN(LRW)=0
            NSAC(LRW,1)=0
            NSAC(LRW,2)=0
            ISQC=ISQC-1
         ENDIF
40    CONTINUE
C
C     RESTORATION OF INITIAL CONDITIONS FOR THE PRESENT TIME INTERVAL
C     EXCEPT FOR AIRFLOW DISTRIBUTION.
C
      DO 60 I=1,NB
         TJS(I)=TTJS(I)
         TRD(I)=TTRD(I)
         RDPROP(I)=RDOP(I)
         RDCH4(I)=RCH4(I)
C
C     Q(I): AIRFLOW RATE IN AIRWAY NO(I) AT THE END OF THE PRESENT
C           INTERVAL.
C     QTP(I): INITIAL AIRFLOW RATE IN AIRWAY NO(I), CONSTANT WITHIN
C             EACH TIME INCREMENT.
C     QQ(I): TIME AVERAGED AIRFLOW RATE IN AIRWAY NO(I).
C     QQ2 (I): THE AVERAGED AIRFLOW RATE IN AIRWAY NO(I) AT THE TIME
C              INTERVAL THE AIRFLOW REVERSAL HAPPENS
C
         QQA=0.55*Q(I)+0.45*QTP(I)
         QQ(I)=0.85*QQA+0.15*QQ(I)
         qqa2=0.55*abs(q(i))+0.45*abs(qtp(i))
         qq2(i)=0.85*abs(qqa2)+0.15*abs(qq(i))
         IF (CONCT.EQ.1) QQ(I)=QQQ(I)
         NSAC(I,1)=NSACB(I,1)
         NSAC(I,2)=NSACB(I,2)
         IQ=IBTN(I)
         IF (IQ.GT.0) THEN
            TAUXC(IQ)=TAUXD(IQ)
            DO 50 J=1,10
               TAUXA(IQ,J)=TAUXB(IQ,J)
               DO 50 JJ=1,5
                  DPPA(IQ,J,JJ)=DPPB(IQ,J,JJ)
50          CONTINUE
         ENDIF
60    CONTINUE
      DO 65 I=1,NJ
         T(I)=TAS(I)
         PROP(I)=PROPS(I)
         PRCH4(I)=PRCH4S(I)
65    CONTINUE
      DO 70 I=1,IMX
         HTAD(I)=0.0
70    CONTINUE
      IBB=0
C
C     CALL ARR (MARKX)
      CALL BASE
      CALL MSLIST
C
      RETURN
      END
C
C
      SUBROUTINE EVHT (MARKX,MARKP,MARKL,MARKR,QSUM,QS)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DATA REARRAGEMENT FOR AIRWAYS IN WHICH AIRFLOW REVERSAL
C          HAPPENED IN THE PRESENT TIME INCREMENT.
C       2) DATA STORAGE ARRAY CONDENSATION FOR ARRAYS WHOSE CAPACITIES
C          ARE SATURATED.
C       3) UPDATING BACKUP ARRAYS TO SET UP THE INITIAL CONDITIONS FOR
C          THE NEXT TIME INCREMENT.
C
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
```

```
C
      MARKP=1
      TACC=TACC+DELTAT
C
      CALL DTTR (MARKL,MARKP)
      IF (MARKL.NE.0) THEN
C        CALL ARR (MARKX)
         CALL BASE
         CALL MSLIST
      ENDIF
C
C     CONDENSATION OF ARRAY DPPA AS ITS CAPACITY IS SATURATED FOR AN
C     AIRWAY. IT IS DONE BY ABANDONING SOME OF THE MOST CLOSELY LOCATED
C     DATA POINTS.
C
      DO 90 I=1,NB
         IQ=IBTN(I)
         IF (IQ.NE.0) THEN
            NS=NSAC(I,2)
            IF (NS.GE.9) THEN
               DO 84 JB=1,2
                  JG=NSAC(I,2)
                  SS1=1.E5
                  JEE=1
                  DO 70 JD=1,JG-1
                     SS2=DPPA(IQ,JD,1)-DPPA(IQ,JD+1,1)
                     IF (SS2.LT.SS1) THEN
                        SS1=SS2
                        JEE=JD
                     ENDIF
70                CONTINUE
                  IF (JEE.EQ.1) THEN
                     JE=2
                  ELSE
                     SS1=DPPA(IQ,JEE-1,1)-DPPA(IQ,JEE,1)
                     SS2=DPPA(IQ,JEE+1,1)-DPPA(IQ,JEE+2,1)
                     JE=JEE
                     IF (SS1.GE.SS2) JE=JEE+1
                  ENDIF
                  NSAC(I,2)=NSAC(I,2)-1
                  NX=NSAC(I,2)
                  TAUXA(IQ,JE-1)=TAUXA(IQ,JE)
                  DPPA(IQ,JE-1,5)=DPPA(IQ,JE-1,5)+DPPA(IQ,JE,5)
                  DO 82 JC=JE,NX+1
                     IF (JC.LE.NX) THEN
                        TAUXA(IQ,JC)=TAUXA(IQ,JC+1)
                        DO 80 JD=1,5
                           DPPA(IQ,JC,JD)=DPPA(IQ,JC+1,JD)
80                      CONTINUE
                     ELSE
                        TAUXA(IQ,JC)=0.0
                        DO 81 JD=1,5
                           DPPA(IQ,JC,JD)=0.0
81                      CONTINUE
                     ENDIF
82                CONTINUE
84             CONTINUE
            ENDIF
         ENDIF
90    CONTINUE
C
C     QSUM: SUM OF ABS. AIRFLOW VARIATION IN AIRWAYS, CRITERION FOR
C           OUTPUT DURING DYNAMIC SIMULATION.
C     DELTAQ(I): AIRFLOW VARIATION FOR AIRWAY NO(I) BETWEEN TWO
C                SUCCESSIVE OUTPUT.
C     MARKR=0: REGULAR OUTPUT WAS GIVEN IN THE LAST TIME INTERVAL.
C              DELTAQ(I) MUST BE INITIATED TO ZERO.
C     MARKR=1: THE OUTPUT FOR THE PRESENT INTERVAL HAS BEEN OMITTED.
C              DELTAQ(I) HAS ITS CUMULATIVE EFFECT.
C
      QS=QSUM
      QSUM=0.0
      DO 110 I=1,NB
         NSACB(I,1)=NSAC(I,1)
         NSACB(I,2)=NSAC(I,2)
         IF (MARKR.EQ.0) DELTAQ(I)=0.0
         DELTAQ(I)=DELTAQ(I)+Q(I)-QTP(I)
         QSUM=QSUM+ABS(DELTAQ(I))
         IF (IBTN(I).NE.0) THEN
            IQ=IBTN(I)
            TAUXD(IQ)=TAUXC(IQ)
            DO 100 J=1,10
               TAUXB(IQ,J)=TAUXA(IQ,J)
               DO 100 K=1,5
                  DPPB(IQ,J,K)=DPPA(IQ,J,K)
100         CONTINUE
         ENDIF
110   CONTINUE
      QS=ABS(QS-QSUM)
C
      RETURN
      END
C
C
      SUBROUTINE OUTPUT (IPT,MARKX,MARKY,ITT)
C
C     -----------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       OUTPUT OF NORMAL CALCULATION RESULTS.
C
C     -----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION LST(NMX),TQR(NMX)
C
C
      IF (IPT.NE.1) THEN
         PP=0.0
         DO 10 L=1,NB
            IF (NWTYP(L).LE.0.OR.NWTYP(L).EQ.10) THEN
               P(L)=R(L)*Q(L)*ABS(Q(L))*1.E-10
            ELSE
               P(L)=R(L)
            ENDIF
            PP=PP+P(L)
10       CONTINUE
         PCRIT=PP/FLOAT(NB)
C
C
         IF (ITT.GE.ITN.AND.IPT.EQ.4) THEN
            DO 1520 I=1,NB
               TTMN=0.0
               DO 1500 J=1,10
                  TTMN=TTMN+TMRCD(I,J)
1500           CONTINUE
               TTMN=TTMN/10.0
               TQR(I)=0.0
               DO 1510 J=1,10
```

```fortran
                  TQR(I)=TQR(I)+(TMRCD(I,J)-TTMN)**2
1510           CONTINUE
1520        CONTINUE
            IZ=0
            LL=0
1530        II=0
            DO 1540 I=1,NB
               IF (NO(I).GT.0) THEN
                  IF (TQR(I).GT.TSR.OR.II.EQ.0) THEN
                     TSR=TQR(I)
                     II=I
                  ENDIF
               ENDIF
1540        CONTINUE
            IF (II.EQ.0) GO TO 1590
            IF (LL.GT.0) THEN
               KK=0
               DO 1550 J=1,LL
                  K=LST(J)
                  IF (TQR(II).GT.TQR(K)) KK=KK+1
1550           CONTINUE
               IF (KK.GE.(LL-2).AND.TQR(II).GT.20.0) GO TO 1560
               GO TO 1590
            ENDIF
1560        LL=LL+1
            LST(LL)=II
            NO(II)=-NO(II)
            JSS=JS(II)
            JFF=JF(II)
1570        DO 1580 I=1,NB
               IF (NO(I).GT.0) THEN
                  IF ((JS(I).EQ.JSS.OR.JF(I).EQ.JSS.OR.JS(I).EQ.JFF.OR.
     .               JF(I).EQ.JFF).AND.(TQR(I).GT.20.0)) THEN
                     LL=LL+1
                     LST(LL)=I
                     NO(I)=-NO(I)
                     IF (JS(I).EQ.JSS) THEN
                        JSS=JF(I)
                     ELSE IF (JF(I).EQ.JSS) THEN
                        JSS=JS(I)
                     ELSE IF (JS(I).EQ.JFF) THEN
                        JFF=JF(I)
                     ELSE
                        JFF=JS(I)
                     ENDIF
                     GO TO 1570
                  ENDIF
               ENDIF
1580        CONTINUE
            IZ=IZ+1
            IF (IZ.LE.3) GO TO 1530
1590        CRT1=0.0
            CRT2=0.0
            CRT3=0.0
            DO 1630 I=1,LL
               K=LST(I)
               NO(K)=IABS(NO(K))
               CRT=0.0
               DO 1600 J=10,3,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-2))**2
1600           CONTINUE
               CRT1=CRT1+CRT/8.0
               CRT=0.0
               DO 1610 J=10,4,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J-3))**2
1610           CONTINUE
               CRT2=CRT2+CRT/7.0
               CRT=0.0
               DO 1620 J=10,5,-1
                  CRT=CRT+(TMRCD(K,J)-TMRCD(K,J)-TMRCD(K,J-4))**2
1620           CONTINUE
               CRT3=CRT3+CRT/6.0
1630        CONTINUE
            CRT1=CRT1/LL
            CRT2=CRT2/LL
            CRT3=CRT3/LL
            LSG=LL
            IF (MIN(CRT1,CRT2,CRT3)*10.0.LE.MAX(CRT1,CRT2,CRT3)) THEN
               ICYCLE=2
               IF (CRT2.LT.0.95*CRT1) ICYCLE=3
               IF (CRT3.LT.0.9*CRT1.AND.CRT3.LT.CRT2) ICYCLE=4
            ELSE
               ICYCLE=0
            ENDIF
         ENDIF
      ENDIF
      IF (IPT.EQ.4) THEN
C
C
         IF (IAC.GT.0) THEN
            WRITE (8,860)
            DO 76 IR=1,IAC
               II=IOMIT(I)
               WRITE (8,870) NO(II),JS(II),ROMIT(IR,1),ROMIT(IR,2),
     .                       ROMIT(IR,3)
76          CONTINUE
            IAC=0
         ENDIF
         ITL=ITT
         JTL=ITN
         WRITE (8,320) TACC
         IF (ITL.GE.JTL) THEN
            WRITE (8,690) SUMFNV/MNO
            KTL=0
            DO 78 I=1,NB
               NREVA=NREV(I)
               NREVA=NNREV(I)
               IF (NREVA.GE.ITN/3) THEN
                  IF (KTL.EQ.0) THEN
                     WRITE (8,700)
                     KTL=1
                  ENDIF
                  IF (NWTYP(I).EQ.1) THEN
                     PP=R(I)
                  ELSE
                     PP=R(I)*Q(I)*ABS(Q(I))*1.E-10
                  ENDIF
                  WRITE (8,710) NO(I),Q(I),PP,NREVA,ITL
               ENDIF
78          CONTINUE
            IZ=ITN-9
            IF (ICYCLE.EQ.0) THEN
               WRITE (8,890) (IJ,IJ=IZ,IZ+9)
            ELSE
               WRITE (8,900) ICYCLE,(IJ,IJ=IZ,IZ+9)
            ENDIF
            WRITE (8,905) (HRM(IJ),IJ=1,10)
            DO 80 I=1,LSG
               K=LST(I)
               IF (JSB(K).EQ.JS(K)) THEN
```

```
                  JSS=JS(K)
                  JFF=JF(K)
               ELSE
                  JSS=JF(K)
                  JFF=JS(K)
               ENDIF
               WRITE (8,910) NO(K),JSS,JFF,(TMRCD(K,J),J=1,10)
               WRITE (8,920) (QRCD(K,J)/1000.0,J=1,10)
80          CONTINUE
         ELSE
            IF (SUMFNV/MNO.LE.2.E-4) WRITE (8,770) SUMFNV/MNO,ITL
            IF (SUMFNV/MNO.GT.2.E-4) WRITE (8,780) SUMFNV/MNO,ITL
         ENDIF
         LINE=NCOMT1+1
         WRITE (8,750)
         DO 86 I=1,NB
85          IJ=0
            IF (LINE.LE.50) IJ=ITITLE(LINE)
            IF (I.EQ.IJ) THEN
               IF (LINE.EQ.1) I1=0
               IF (LINE.NE.1) I1=ITITLE(LINE-1)
               IF (LINE.EQ.50) I2=0
               IF (LINE.NE.50) I2=ITITLE(LINE+1)
               IF (I2.NE.ITITLE(LINE).AND.I1.NE.ITITLE(LINE)) THEN
                  WRITE (8,930) (TITLE(LINE,J),J=1,18)
               ELSE IF (I2.EQ.ITITLE(LINE).AND.I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,960) (TITLE(LINE,J),J=1,18)
               ELSE IF (I1.EQ.ITITLE(LINE)) THEN
                  WRITE (8,940) (TITLE(LINE,J),J=1,18)
               ELSE
                  WRITE (8,950) (TITLE(LINE,J),J=1,18)
               ENDIF
               LINE=LINE+1
               GO TO 85
            ENDIF
            IF (CONCT.EQ.0) THEN
            WRITE (8,760) NO(I),IABS(JS(I)),IABS(JF(I)),DELTAQ(I),Q(I),
     .                    TMRD(I),TRD(I),RDPROP(I),RDCH4(I),P(I)
            ELSE
            WRITE (8,765) NO(I),IABS(JS(I)),IABS(JF(I)),DELTAQ(I),Q(I),
     .                                   RDPROP(I),RDCH4(I),P(I)
            ENDIF
86       CONTINUE
         LI=0
         DO 87 I=1,NB
            IF (NSAC(I,1).EQ.1) THEN
               LK=IBTN(I)
               LM=0
               IF (DPPA(LK,2,3).GE.1.E-4) LM=2
               IF (DPPA(LK,1,3).GE.1.E-4) LM=1
               IF (LM.GT.0) THEN
                  IF (LI.NE.1) THEN
                     LI=1
                     WRITE (8,350)
                  ENDIF
                  IF (CONCT.EQ.0) THEN
                     WRITE (8,360) NO(I),DPPA(LK,LM,1),JS(I),
     .                     DPPA(LK,LM,2),DPPA(LK,LM,3),DPPA(LK,LM,4)
                  ELSE
                     WRITE (8,365) NO(I),DPPA(LK,LM,1),JS(I),
     .                                   DPPA(LK,LM,3),DPPA(LK,LM,4)
                  ENDIF
               ENDIF
            ENDIF
87       CONTINUE
         IF (INFLOW.GT.0) THEN
            WRITE (8,820)
            DO 89 L=1,INFLOW
               DO 88 J=1,NB
                  IF (NCENT(L).EQ.NO(J)) THEN
                     WRITE (8,830) NO(J),Q(J),TJS(J),RDPROP(J),
     .                     O2BEH(L),HTAD(L)
                     IF (TJS(J).GT.3200.0) WRITE (8,995) NO(J)
                     GO TO 89
                  ENDIF
88             CONTINUE
89          CONTINUE
         ENDIF
         IF (IOUT.NE.0) THEN
            WRITE (8,650)
            IF (CONCT.EQ.0) THEN
               WRITE (8,660) (JNO(I),T(I),PROP(I),PRCH4(I),I=1,NJ)
            ELSE
               WRITE (8,665) (JNO(I),     PROP(I),PRCH4(I),I=1,NJ)
            ENDIF
         ENDIF
         IF (IOUT.LT.0.OR.MARKX.EQ.5) THEN
            KTL=0
            DO 90 I=1,NB
               NNREVA=NNREV(I)
               IF ((NNREVA/2)*2.NE.NNREVA) THEN
                  IF (KTL.EQ.0) THEN
                     WRITE (8,470)
                     KTL=1
                  ENDIF
                  IF (Q(I).GT.0.0) THEN
                     WRITE (8,480) NO(I),IABS(JS(I)),IABS(JF(I))
                  ENDIF
               ENDIF
90          CONTINUE
            IF (KTL.EQ.0) WRITE (8,482)
         ENDIF
      ELSE IF (IPT.EQ.5) THEN
         WRNSUM=WRNPR+WRNGS+WRNSM+WRNHT
         IF (WRNSUM.LE.0.) THEN
            WRITE (8,410)
         ELSE
            J=0
            WRNP=WRNPR
            IF (PCRIT*0.05.LE.0.5*WRNPR) WRNP=PCRIT*0.05
            DO 95 I=1,Nb
               K=0
               L=0
               M=0
               N=0
               IF ((100.*RDCH4(I)).GE.WRNGS) K=K+1
               IF ((100.*RDPROP(I)).GE.WRNSM) L=L+1
               IF (TRD(I).GE.WRNHT) M=M+1
               IF (P(I).LT.WRNP) N=N+1
               IF ((K+L+M+N).GT.0) THEN
                  IF (J.EQ.0) THEN
                     WRITE (8,420) WRNGS,WRNSM,WRNHT,WRNP
                     J=J+1
                  ENDIF
                  WRITE (8,430) NO(I),IABS(JS(I)),IABS(JF(I)),RDCH4(I),
     .                          RDPROP(I),TRD(I),P(I)
               ENDIF
95          CONTINUE
            IF (J.LE.0) WRITE (8,440)
            J=0
```

```
            KM=0
            DO 100 I=1,NJ
               K=0
               L=0
               M=0
               IF ((100.*PRCH4(I)).GE.WRNGS) K=K+1
               IF ((100.*PROP(I)).GE.WRNSM) L=L+1
               IF (T(I).GE.WRNHT) M=M+1
               IF ((K+L+M).GT.0) THEN
                  IF (J.EQ.0) THEN
                     WRITE (8,450) WRNGS,WRNSM,WRNHT,WRNGS,WRNSM,WRNHT
                  ENDIF
                  J=J+1
                  IF (2*(J/2).NE.J) THEN
                     KM=I
                  ELSE
                     WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM),
     .                             JNO(I),PRCH4(I),PROP(I),T(I)
                     KM=0
                  ENDIF
               ENDIF
100         CONTINUE
            IF (KM.GT.0) WRITE (8,460) JNO(KM),PRCH4(KM),PROP(KM),T(KM)
            IF (J.LE.0) WRITE (8,445)
         ENDIF
      ELSE IF (IPT.EQ.6) THEN
         IF (CONCT.EQ.1) GO TO 107
         NRW=0
         DO 105 L=1,NB
            LL=IBTN(L)
            IF (LL.GT.0) THEN
               IF (NRW.EQ.0) THEN
                  WRITE (8,570)
                  NRW=1
               ENDIF
               LL1=NSAC(L,2)
               IF (LL1.GT.0) THEN
                  WRITE (8,580) NO(L),(DPPA(LL,JQ,1),JS(L),DPPA(LL,JQ,2)
     .                       ,TAUXA(LL,JQ),(DPPA(LL,JQ,JP),JP=3,5),
     .                       JQ=LL1,1,-1)
               ELSE
                  WRITE (8,640) NO(L),TJS(L),TRD(L),RDPROP(L)
               ENDIF
            ENDIF
105      CONTINUE
107      CONTINUE
      ENDIF
C
      RETURN
C
320   FORMAT (///,20X,'TIME AT ',F6.0,' SEC. AFTER EVENT',//)
350   FORMAT (///,T21,'DATA FOR THE FUME FRONT IN AIRWAYS',
     .       ///,1X,'AIRWAY',T11,'POSITION',T21,'FROM',
     .       T37,'TEMPERATURE',T51,'FUME',T61,'METHANE',/)
360   FORMAT (1X,I5,T11,0PF7.2,T19,I5,T38,0PF8.2,T49,2PF7.4,T61,2PF6.2)
365   FORMAT (1X,I5,T11,0PF7.2,T19,I5,           T49,2PF7.4,T61,2PF6.2)
410   FORMAT (///,1X,'NO THRESHOLD FOR CRITICAL STATES ',
     .       'WERE SPECIFIED.')
420   FORMAT (///,9X,'IN THE FOLLOWING AIRWAYS EXIST CRITICAL ',
     .       'CONDITIONS',//,1X,'AIRWAY',T10,'FROM',T16,'TO',T22,'CH4 %'
     .       ,T33,'FUMES %',T45,'TEMPERATURE',T61,'HEADLOSS',/,
     .       T22,'>',F5.3,T33,'>',F7.4,T46,'>',F5.0,' F',T59,
     .       '<',F6.3,' IN.WG.',/)
430   FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.4,T45,0PF8.1,T62,0PF6.3)
435   FORMAT (I5,I7,I5,T21,2PF6.2,T32,2PF8.4,           T62,0PF6.3)
440   FORMAT (///,1X,'NO CRITICAL CONDITIONS AT AIRWAY ENDS ',
     .       'WERE DETECTED')
445   FORMAT (///,1X,'NO CRITICAL CONDITIONS IN JUNCTIONS WERE ',
     .       'DETECTED.')
450   FORMAT (///,10X,'IN THE FOLLOWING JUNCTIONS EXIST CRITICAL ',
     .       'CONDITIONS',//,' JUNCTION',T11,'CH4  %',T19,'FUMES %',T28
     .       ,'TEMP. F',T40,'JUNCTION',T49,'CH4  %',T57,'FUMES %',T66,
     .       'TEMP. F',/,T11,'>',F5.2,T19,'>',F6.3,T29,'>',F5.0,T49,
     .       '>',F5.2,T57,'>',F6.3,T67,'>',F5.0,/)
460   FORMAT (I6,T11,2PF5.2,T17,2PF8.4,T27,0PF7.1,T39,I6,T49,2PF5.2,
     .       T55,2PF8.4,T65,0PF7.1)
465   FORMAT (I6,T11,2PF5.2,T17,2PF8.2,           T39,I6,T49,2PF5.2,
     .       T55,2PF8.2           )
470   FORMAT (///,1X,'REVERSAL OF AIRFLOW HAS OCCURRED IN THE ',
     .       'FOLLOWING PLACES',/)
480   FORMAT (1X,'AIRWAY',I6,' IS NOW CARRYING AIR FROM',I6,' TO',I6)
482   FORMAT (//,' NO AIRFLOW REVERSAL WAS DETECTED IN THE SYSTEM.')
570   FORMAT (///,23X,'DATA RECORD IN AIRWAYS',//,1X,'AIRWAY',T12,
     .       'DIST.',T19,'FROM',T26,'FRONT T',T37,'REAR T',
     .       T47,'FUME',T57,'CH4',T63,'AIR M (LB)',/)
580   FORMAT (1X,I5,11(T8,0PF9.2,T17,I5,T24,0PF9.2,T34,0PF9.2,T44,
     .       2PF7.2,T53,2PF7.2,T61,0PF9.2,/))
640   FORMAT (1X,I5,T9,'NO C.V. T(BGG)=',0PF7.1,', T(END)=',0PF7.1,',
     .       FUME=',2PF8.4,'%.',/)
650   FORMAT (///,19X,'PARAMETERS OF AIR IN JUNCTIONS',//,1X,'JUNCTION'
     .       ,T13,'TEMP.',T20,'FUMES',T30,'METHANE',T41,'JUNCTION',T50,
     .       'TEMP.',T60,'FUMES',T68,'METHANE',/)
660   FORMAT (I6,T9,0PF6.2,T20,2PF8.4,T30,2PF8.2,T40,I6,T49,0PF6.2,T60,
     .       2PF7.4,T68,2PF7.2)
665   FORMAT (I6,          T20,2PF8.4,T30,2PF7.2,T40,I6,          T60,
     .       2PF7.2,T68,2PF7.2)
690   FORMAT (///,1X,'THE CALCULATION WAS NOT COMPLETED SINCE THE ',
     .       'NUMBER OF EXCHANGES BETWEEN',/,' NETWORK AND TEMPERATURE'
     .       ,' PARTS OF THE PROGRAM BECAME EXCESSIVE.',/,' THRESHOLD',
     .       ' IN NVP IS 0.0002 IN.W.G. PER MESH. THE NVP CORRECTION ',
     .       'PER'/,' MESH NOW EQUALS ',F7.5,' IN.W.G.')
700   FORMAT (///,1X,'THE FOLLOWING UNSTABLE AIRWAYS MAY HAVE',1X,
     .       'CAUSED THE INCOMPLETION',/,' OF NETWOK BALANCING:',/)
710   FORMAT (1X,'AIRWAY',I5,' OF AIRFLOW RATE',F7.0,' (CFM) AND ',
     .       'HEADLOSS',F7.3,' IN.W.G.',/,' HAD ITS AIRFLOW DIRECTION'
     .       ,' REVERSED',I3,' TIMES IN',I3,' ITERATIONS')
750   FORMAT (///,T8,'TEMP. AND CONCENTRA. AT AIRWAY ENDS, HEADLOSS IN'
     .       ,' AIRWAYS',//,' AIRWAY',T9,'FROM',T15,'TO',T19,'DELTA Q',
     .       T29,'AIRFLOW',T38,'AVE. T',T45,'T AT END',T54,'FUMES',T61,
     .       'CH4',T68,'HEADLOSS',/)
760   FORMAT (I5,I6,I5,T17,0PF8.0,T27,0PF9.0,T37,0PF7.2,T45,0PF7.2,T52,
     .       2PF8.4,T60,2PF6.2,T68,0PF6.3)
765   FORMAT (I5,I6,I5,T17,0PF8.0,T27,0PF9.0,                      T52,
     .       2PF8.4,T60,2PF6.2,T68,0PF6.3)
770   FORMAT (/////,' THRESHOLD IN ACCURACY (SUM OF NVP CORRECTIONS ',
     .       'PER MESH < 2.E-4 IN.W.G.)',/,' SATISFIED. CURRENT ',
     .       'SUMFNV PER MESH',F9.6,' IN.W.G.,   ITERATIONS ',I5)
780   FORMAT (/////,1X,'THRESHOLD IN ACCURACY (SUM OF NVP',1X,
     .       'CORRECTIONS PER MESH<2.E-4 IN. W.G.)',/,' NOT ',
     .       'SATISFIED. CURRENT SUMFNV:',F9.6,' IN.W.G., ITER.',I5)
800   FORMAT (/////,12X,'DATA FOR CONCENTRATION AND TEMPERATURE',
     .       ' CALCULATION',//,1X,'AIRWAY  FROM',T16,'TO',T21,
     .       'ELEV. DIFF.',T34,'ROCK T',T43,'CH4 PROD.',T54,'COND.',
     .       T61,'DIFFUSIVITY',/)
810   FORMAT (I5,I7,I5,T21,F9.2,T33,F7.2,T43,F7.1,T52,F7.2,T61,F8.3)
820   FORMAT (///,23X,'DATA FOR HEAT SOURCES',//,1X,'AIRWAY',T11,
     .       'AIRFLOW',T21,'TEMPERATURE',T37,'FUME',T51,'O2 LEFT',T61,
     .       'HEAT INPUT',/)
830   FORMAT (I5,T8,0PF10.0,T22,0PF8.2,T35,2PF6.4,T50,0PF7.2,T60,E10.3)
840   FORMAT (/,1X,'AIRFLOW REVERSED IN FAN BRANCH NO. ',I5)
```

```
860   FORMAT (//,1X,'THE FOLLOWING DATA RECORDS WERE OMITTED DUE TO ',
     .       'ARRAY CAPACITIES.',//,T1,'AIRWAY',T15,'PARAMETERS AT ',
     .       'STARTING JUNCTIONS IN THE CURRENT TIME INTERVAL',//,T22,
     .       'JS',T31,'TEMPERATURE',T46,'FUME %',T61,'CH4 %',/)
870   FORMAT (1X,I5,T19,I5,T31,F8.1,T44,F7.3,T59,F7.3)
890   FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS NOT CAUSED BY A',
     .       ' CYCLIC PATTERN. DATA OF',/,' SIGNIFICANT CONDITION ',
     .       'CHANGE DURING LAST 10 ITERATIONS FOLLOW. FURTHER',/,
     .       ' ANALYSES ARE NEEDED. ("ITR" STANDS FOR ITERATION)',//,
     .       ' UNITS: Q: AIRFLOW RATE IN 1000FT3/MIN;',17X,'T: TEMP. ',
     .       'IN F;',/,8X,'H: NVP CORRECTION PER MESH IN IN.W.G.',//,
     .       ' ITR',T5,I5,9I7,/)
900   FORMAT (///,1X,'THE FAILURE IN CONVERGING WAS CAUSED BY A CYCLIC',
     .       ' PATTERN IN EVERY',I3,/,' ITERATIONS AS SHOWN BELOW:',
     .       ' ("ITR" STANDS FOR ITERATION)',//,' UNITS: Q: AIRFLOW ',
     .       'RATE IN 1000FT3/MIN;',17X,'T: TEMP. IN F;',/,8X,'H: NVP ',
     .       'CORRECTION PER MESH IN IN.W.G.',//,' ITR',T5,I5,9I7,/)
905   FORMAT (1X,'H',T3,10(1X,F6.4),/)
910   FORMAT (1X,'AIRWAY: ',I4,10X,'JS:',I4,6X,'JF:',I4,/,1X,
     .       'T',T3,10F7.1)
920   FORMAT (1X,'Q',T3,10F7.1,/)
930   FORMAT (/,18A4,/)
940   FORMAT (18A4,/)
950   FORMAT (/,18A4)
960   FORMAT (18A4)
970   FORMAT (/////,20X,'TIME AT ',F5.1,' HOUR(S) AFTER EVENT',//)
980   FORMAT (///,T2,'JUNCTION',T12,'TEMP.',T20,'ELEVATION',T32,'CH4',
     .       T40,'JUNCTION',T50,'TEMP.',T58,'ELEVATION',T70,'CH4',/)
990   FORMAT (T2,I5,T10,F7.1,T19,F8.1,T30,F5.2,T40,I5,T48,F7.1,T57,F8.1
     .       ,T68,F5.2)
995   FORMAT (/,'* WARNING *  AIR TEMP. IN BRANCH ',I5,' IS ABNORMAL ',
     .       'UNDER MINING CONDITIONS',/,' THE POSSIBLE CAUSES ARE:',/,
     .       ' 1. "QCENT" WAS NOT SPECIFIED;',/,' 2. "QCENT" DOES NOT',
     .       ' MATCH "HEAT";',/,' 3. "HTPO2" WAS TOO LARGE.')
C
      END
C
C
      SUBROUTINE READIN (DAL,MAX,ISTOP,ICT)
C
C     ------------------------------------------------------------------
C
C       SUBROUTINE PURPOSE:
C       DATA INPUT.
C
C     ------------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      CHARACTER*1 CHK(15),ER(80)
      DIMENSION DAL(15)
      DATA CHK/'0','1','2','3','4','5','6','7','8','9','.',',','+',
     .         '-',' '/
C
C
      KK=0
      KK1=0
      KK2=0
      NEG=0
      ISTOP=0
      DO 5 I=1,15
         DAL(I)=0.0
5     CONTINUE
      DO 7 I=1,80
         ER(I)=' '
         ROW(I)=' '
7     CONTINUE
      READ (9,110,ERR=60) (ROW(I),I=1,80)
      LLINE=LLINE+1
      IDEC=0
      MLEV=1
      JJ=0
      I=15
      LEV=0
10    LEV=LEV+1
      II=I
      IF (LEV.GT.80) GO TO 50
      DO 20 I=1,15
         IF (ROW(LEV).EQ.CHK(I)) THEN
            IF (I.NE.15) KK=1
            GO TO 30
         ENDIF
20    CONTINUE
      KK2=KK2+1
      IF ((KK.EQ.0.AND.ROW(LEV).NE.'$').OR.KK2.GE.4) THEN
         ISTOP=0
         NCOMTS=NCOMTS+1
         IF (NCOMTS.LE.50) THEN
            IF (ICT.EQ.1) THEN
               KK3=0
               KK4=0
               DO 21 I=1,80
                  IF (ROW(I).NE.CHK(15)) GO TO 22
                  KK3=KK3+1
21             CONTINUE
22             DO 23 I=80,1,-1
                  IF (ROW(I).NE.CHK(15)) GO TO 24
                  KK4=KK4+1
23             CONTINUE
24             LR=(KK3+KK4-8)/2
               IF (LR.LT.0) LR=0
               KK5=80-KK4
               IF (LR.EQ.0) KK5=KK3+72
               DO 25 I=KK3+1,KK5
                  ER(I-KK3+LR)=ROW(I)
25             CONTINUE
               DO 26 I=1,80
                  IF (I.LE.LR) THEN
                     ROW(I)=' '
                  ELSE
                     ROW(I)=ER(I)
                  ENDIF
26             CONTINUE
            ENDIF
            DO 27 I=1,20
               J=(I-1)*4+1
               TITLE(NCOMTS,I)=ROW(J)//ROW(J+1)//ROW(J+2)//ROW(J+3)
27          CONTINUE
         ENDIF
         RETURN
      ELSE IF (KK.EQ.0.AND.ROW(LEV).EQ.'$') THEN
         KK1=1
         GO TO 50
      ELSE IF (KK1.NE.1) THEN
         ER(LEV)=ROW(LEV)
         ISTOP=1
         I=1
      ENDIF
30    IF (I.GT.10) GO TO 40
      IF (JJ.EQ.1) MLEV=MLEV+1
```

```
      JJ=0
      IF (IDEC.EQ.0) THEN
         DAL(MLEV)=(DAL(MLEV)*10.0)+((I-1)*1.0)
      ELSE IF (IDEC.GT.0) THEN
         DAL(MLEV)=DAL(MLEV)+((I-1)*1.0)/(10.0**IDEC)
      ENDIF
      IF (IDEC.NE.0) IDEC=IDEC+1
      GO TO 10
40    IF (I.EQ.11) IDEC=1
      IF (I.EQ.14) NEG=1
      IF (I.EQ.12.OR.I.EQ.15) THEN
         IF (NEG.EQ.1) DAL(MLEV)=0.0-DAL(MLEV)
         NEG=0
         IDEC=0
      ENDIF
      IF (JJ.EQ.1.AND.(I.EQ.11.OR.I.EQ.13.OR.I.EQ.14)) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.12) THEN
         MLEV=MLEV+1
         JJ=0
      ENDIF
      IF (I.EQ.15.AND.II.EQ.12) JJ=5
      IF (I.EQ.15.AND.II.NE.15.AND.II.NE.12.AND.JJ.EQ.0) JJ=1
      IF (I.EQ.12) DAL(MLEV)=0.0
      IF (MLEV.LE.MAX) GO TO 10
50    IF (ISTOP.GT.0) WRITE (8,100) (ROW(I),I=1,80),(ER(I),I=1,80)
      IF (KK.EQ.0.OR.KK1.EQ.1) DAL(15)=-1.E25
      RETURN
60    WRITE (8,120)
      ISTOP=1
      RETURN
C
100   FORMAT (/,' * ERROR *  ILLEGAL CHARACTER DETECTED IN ',
     .       'THE LINE WHERE NUMERICAL INPUT',/,' WAS EXPECTED. ',
     .       'TWO LINES FOLLOW: 1 ORIGINAL; 2 ILLEGAL CHARACTERS.',
     .       //,80A1,/,80A1,/)
110   FORMAT (80A1)
120   FORMAT (//,' * ERROR *  INPUT FILE ERROR, POSSIBLY END-OF-FILE.')
C
      END
C
C
      SUBROUTINE  FWCTA
C
C     ----------------------------------------------------------------
C
C       SUBROUTINE PURPOSES:
C       1) DETECTION OF POSSIBLE AIRFLOW REVERSAL PATHS.
C
C     ----------------------------------------------------------------
C
      INCLUDE 'CMMN1.DAT'
C
      DIMENSION LREC(NMX),IINU(NMX)
      DATA LREC,IINU / NMX*0,NMX*0 /
C
         DO 70 J=1,NJ
            IF (JNOL(J).EQ.JSTART) JSTOUT=J
70       CONTINUE
C
C
         DO 90 I=1,NMX
            LREC(I)=0
90       CONTINUE
         DO 92 I=1,NXX
            KJF(I)=0
            KNO(I)=0
92       CONTINUE
         KREC=0
         N1=JSTOUT
         DO 95 I=1,NJ
            IF (JNO(I).EQ.JSTART) THEN
               JNO(I)=-JNO(I)
               GO TO 100
            ENDIF
95       CONTINUE
100      IF (N1.LE.1) THEN
            N2=1
         ELSE
            N2=LOUT(N1-1)+1
         ENDIF
         N3=LOUT(N1)
         DO 110 I1=N2,N3
            I2=NGOUT(I1)
            LREC(I2)=100
110      CONTINUE
115      JREC=0
         DO 130 I1=1,NJ
            IF (JNO(I1).GT.0) THEN
               DO 117 I9=1,NJ
                  IF (IABS(JNOL(I9)).EQ.IABS(JNO(I1))) GO TO 118
117            CONTINUE
               WRITE (6,*) 'SHOULD NOT OCCUR (117) -- MFIRE2'
C              WRITE (12,1176) JNO(I1)
118            IF (I9.LE.1) THEN
                  L1=1
               ELSE
                  L1=MMIN(I9-1)+1
               ENDIF
               L2=MMIN(I9)
               KREADY=0
               DO 120 I3=L1,L2
                  I4=NGIN(I3)
                  IF (LREC(I4).LE.0) THEN
                     KREADY=1
                     JREC=I9
                  ENDIF
120            CONTINUE
               IF (KREADY.EQ.0) THEN
                  JNO(I1)=-JNO(I1)
                  N1=I9
                  GO TO 100
               ENDIF
            ENDIF
130      CONTINUE
         N1=JSTOUT
1000     IF (N1.LE.1) THEN
            N2=1
         ELSE
            N2=MMIN(N1-1)+1
         ENDIF
         N3=MMIN(N1)
         DO 1100 I1=N2,N3
            I2=NGIN(I1)
            LREC(I2)=100
1100     CONTINUE
1150     JREC=0
         DO 1300 I1=1,NJ
```

```
         IF (JNO(I1).GT.0) THEN
            DO 1170 I9=1,NJ
               IF (IABS(JNOL(I9)).EQ.IABS(JNO(I1))) GO TO 1180
1170        CONTINUE
            WRITE (6,*) 'SHOULD NOT OCCUR (1170)'
1180        IF (I9.LE.1) THEN
               L1=1
            ELSE
               L1=LOUT(I9-1)+1
            ENDIF
            L2=LOUT(I9)
            KREADY=0
            DO 1200 I3=L1,L2
               I4=NGOUT(I3)
               IF (LREC(I4).LE.0) THEN
                  KREADY=1
                  JREC=I9
               ENDIF
1200        CONTINUE
            IF (KREADY.EQ.0) THEN
               JNO(I1)=-JNO(I1)
               N1=I9
               GO TO 1000
            ENDIF
         ENDIF
1300  CONTINUE
      DO 135 I1=1,NXX
         KJS(I1)=0
135   CONTINUE
      IF (JREC.GT.0) THEN
         KJSSX=0
         KJSSY=0
         KJSS=0
         N1=JREC
140      IF (N1.LE.1) THEN
            N2=1
         ELSE
            N2=LOUT(N1-1)+1
         ENDIF
         N3=LOUT(N1)
         NREADY=0
142      DO 170 I1=N2,N3
            I2=NGOUT(I1)
            IF (LREC(I2).EQ.NREADY) THEN
               KJSS=KJSS+1
               KJS(KJSS)=I2
               IF (KJSS.GE.2) THEN
               DO 160 I3=1,KJSS-1
                  J1=KJS(I3)
                  IF (JS(J1).EQ.JF(I2)) THEN
                     KREC=KREC+1
                     JUDGE=0
                     DO 143 I4=I3,KJSS
                        II1=KJS(I4)
                        IF (LREC(II1).LE.0) JUDGE=1
143                  CONTINUE
                     IF (JUDGE.EQ.0) KREC=KREC-1
                     DO 145 I4=I3,KJSS
                        IF (JUDGE.EQ.0) THEN
                           II1=KJS(I4)
                           LREC(II1)=100
                        ELSE
                           KJSSX=KJSSX+1
                           KJF(KJSSX)=KJS(I4)
                           II1=KJS(I4)
                           LREC(II1)=1
                        ENDIF
145                  CONTINUE
                     IF (JUDGE.GE.1) KNO(KREC)=KJSSX
                     DO 150 I4=1,KJSS
                        KJS(I4)=0
150                  CONTINUE
                     KJSS=0
                     DO 155 I4=1,NB
                        IF (KJSSY.GT.0) THEN
                           DO 154 II4=1,KJSSY
                              IF (I4.EQ.IINU(II4)) GO TO 155
154                        CONTINUE
                        ENDIF
                        IF (LREC(I4).LE.0) THEN
                           JJ1=JS(I4)
                           DO 153 I5=1,NJ
                           IF (JJ1.EQ.JNO(I5)) THEN
                              KJSSY=KJSSY+1
                              IINU(KJSSY)=I4
                              N1=I5
                              GO TO 140
                           ENDIF
153                        CONTINUE
                        ENDIF
155                  CONTINUE
                     GO TO 175
                  ENDIF
160            CONTINUE
               ENDIF
               DO 165 N1=1,NJ
                  IF (IABS(JNOL(N1)).EQ.IABS(JF(I2))) GO TO 140
165            CONTINUE
               WRITE (6,*) 'SHOULD NOT OCCUR (165)'
            ENDIF
170      CONTINUE
         IF (NREADY.EQ.0) THEN
            NREADY=1
            GO TO 142
         ENDIF
      ENDIF
175   IF (KREC.GT.0) THEN
         WRITE (8,200)
         I2=1
         DO 190 I1=1,KREC
            I3=KNO(I1)
            K1=0
            DO 180 I4=I2,I3
               I5=KJF(I4)
               K1=K1+1
               IINU(K1)=NO(I5)
               K1=K1+1
               IINU(K1)=JF(I5)
180         CONTINUE
            WRITE (8,210) I1,(IINU(K2),IINU(K2+1),K2=1,K1,2)
            I2=I3+1
190      CONTINUE
      ELSE
         WRITE (8,220)
      ENDIF
      DO 195 I=1,NMX
         IINU(I)=0
         LREC(I)=0
195   CONTINUE
      DO 197 I=1,NXX
```

```
            KNO(I)=0
            KJS(I)=0
            KJF(I)=0
197      CONTINUE
         DO 198 I=1,NJ
            IF (JNO(I).LT.0) JNO(I)=-JNO(I)
198      CONTINUE
C
200   FORMAT (//,1X,'THE FOLLOWING AIR RECIRCULATION PATH(S) WERE DETE',
     .        'CTED:',/)
210   FORMAT (/,1X,'RECIRCULATION PATH ',I3,'  "AIRWAY -> (JUNCTION)->"'
     .        ,50(/,1X,4(I5,'->(',I3,')->')))
220   FORMAT (//,1X,'NO AIR RECIRCULATION WAS DETECTED IN THE SYSTEM.')
      RETURN
C
      END
```