General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NASA CR-
144512

SINDA/SINFLO COMPUTER ROUTINE

Report No. 2-53002/4R-3167

Revision A

VOLUME II

15 February 1975

Submitted By

VOUGHT SYSTEMS DIVISION
LTV Aerospace Corporation
P.O. Box 5907
Dallas, Texas

To

TRW SYSTEMS GROUP
P.O. Box 58327
Space Park Drive
Houston, Texas

VOUGHT SYSTEMS DIVISION
LTV AEROSPACE CORPORATION

APPENDIX D

SUBROUTINE LISTINGS

Listings of subroutines which have been added and modified during the development of SINDA/SINFLO are presented in alphabetical order in this Appendix. A description of the capabilities is contained in Volume I of this report.

```
@ELT,L ABS1
ELTOTY RLIB70 02/28-03:18:21-(0,)
000001      000         FUNCTION ABS1(X)
000002      000         ABS1 = ABS(X)
000003      000         RETURN
000004      000         END

-END ELT.


@HDG,P ACOMB
```

```
*ELT,L ACOMB
ELTOT7 RLSBTO 02/28-03;18;22-(0,)
000001      000          SUBROUTINE ACOMB(AC,F1,A1,F2,A2)
000002      000    C
000003      000          DIMENSION AC(1), A1(1), A2(1)
000004      000    C
000005      000          EQUIVALENCE (AC1,NC), (A11,N1), (A21,N2)
000006      000    C
000007      000    C
000008      000          AC1 = AC(1)
000009      000          A11 = A1(1)
000010      000          A21 = A2(1)
000011      000          IF(NC .NE. N1) GO TO 910
000012      000          IF(MOD(NC,2) .NE. 0) GO TO 930
000013      000          DO 100 I=2,NC,2
000014      000          CALL DIDEG1(A1(I),A2,V)
000015      000          AC(I) = A1(I)
000016      000          AC(I+1) = F1*A1(I+1) + F2*V
000017      00       100 CONTINUE
000018      000          CALL LINECK(2)
000019      000          CALL GENOUT(AC(2),1,NC,'0COMBINED ARRAY')
000020      000          RETURN
000021      000      910 WRITE(6,920) NC, N1
000022      000      920 FORMAT(57H0ARRAYS ARE NOT OF EQUAL LENGTH IN SUBROUTINE ACOMB, NC
000023      000         1= I5, 6H N1 = I5)
000024      000          GO TO 950
000025      000      930 WRITE(6,940) NC
000026      000      940 FORMAT(46H0WRONG ARRAY LENGTH FOR SUBROUTINE ACOMB, IC = I5)
000027      000      950 CALL WLKBCK
000028      000          CALL EXIT
000029      000          END

END ELT.


*HDG,P ATOR
```

```
@ELT,L ATOR
ELTOTT RLIBTO 02/28-03:18:44-(2,)
000C01      001          SUBROUTINE ATOR(KODE,IA,ILOC,ILEN,ISW)
000002      000    C
000003      000          COMMON /BUCKET/ IB(1)
000004      000          COMMON /POINT / LOC(20), LEN(20)
000005      000          COMMON /DATA  / DUM1(6), NGT, NUC, DUM2(1), ERR
000006      000          COMMON /TAPE  / NIN, NOUT
000007      001          DIMENSION CODE(5)
000008      000    C
000009      001          DATA CODE/1HA,1HT,2HVP,1HW,2HPR/
000010      000    C
000011      000    C
000012      000          ISW = 1
000013      000          IF(KODE .LT. 0) GO TO 500
000014      001          IF (KODE.GT.2) GO TO 30
000015      000          GO TO (10,20), KODE
000016      000    C
000017      000       10 L = 1
000018      000          LL = LOC(14)
000019      000          IST = LOC(13)
000020      000          IEND = IST + LEN(13) - 1
000021      000          DO 14 JJ=IST,IEND
000022      000          IF(IA .EQ. IB(JJ)) GO TO 490
000023      000          L = L + IB(LL)
000024      000          LL = LL + 1
000025      000       14 CONTINUE
000026      000          GO TO 480
000027      000    C
000028      000       20 NLOC = LOC(1)
000029      000          NLEN = LEN(1)
000030      000          GO TO 450
000031      001    C
000032      001       30 NLOC=ILOC
000033      001          NLEN=ILEN
000034      000    C
000035      000      450 CALL SEARCH(IA,IB(NLOC),NLEN,L)
000036      000          IF(L) 480,480,490
000037      000      480 ERR = 1.0
000038      000          WRITE(NOUT,485) CODE(KODE), IA
000039      001      485 FORMAT(15H0* * *  ACTUAL A2,I5,26H IS NOT IN THE LIST  * * *)
000040      000          IA = 0
000041      000          ISW = 2
000042      000          RETURN
000043      000      490 IA = L
000044      000          RETURN
000045      000    C
000046      000      500 CONTINUE
000047      000          RETURN
000048      000          END

END ELT.


@HOG,P CABIN
```

```
@ELT,L CABIN
ELT077 RL18Y0 02/28-03:18:46-(0,)
000001      000          SUBROUTINE CABIN(NLOC,TC,SUMQL,SUMWL)
000002      000    C
000003      000          LOGICAL EXPLCT
000004      000    C
000005      000          COMMON /ARRAY / DATA(1)
000006      000          COMMON /FIXCON/ CON(1)
000007      000          COMMON /TEMP  / T(1)
000008      000          COMMON /SOURCE/ Q(1)
000009      000          COMMON /XSPACE/ NDIM, NTH, EXT(1)
000010      000          COMMON /DIMENS/ NND, NNA
000011      000    C
000012      000          DIMENSION NLOC(1)
000013      000          DIMENSION NDATA(1)
000014      000          DIMENSION NEXT(1)
000015      000    C
000016      000          EQUIVALENCE (CON(1),TIME), (CON(2),TINC), (CON(22),DTIME1)
000017      000          EQUIVALENCE (DATA,NDATA), (EXT,NEXT)
000018      000    C
000019      000          DEFINE DTAU(I) = EXT(NNC+I)
000020      000    C
000021      000          NNT = NNA + NND
000022      000          NNC = NTH - NNT
000023      000          EXPLCT = .TRUE.
000024      000          IF(DTIME1 .GT. 0.0) EXPLCT = .FALSE.
000025      000          IF(NLOC(1) .EQ. 6) GO TO 102
000026      000          CALL TOPLIN
000027      000          WRITE(6,101) NLOC(1)
000028      000      101 FORMAT(57H0* * *  INCORRECT NUMBER OF ELEMENTS INPUT TO CABIN, IC
000029      000         1= I5, 7H  * * *)
000030      000          CALL WLKBCK
000031      000          CALL EXIT
000032      000    C
000033      000      102 NST  = NLOC(2)
000034      000          NCRV = NLOC(3)
000035      000          NCON = NLOC(4)
000036      000          LHC  = NLOC(5)
000037      000          LHFP = NLOC(6)
000038      000          LHTB = NLOC(7)
000039      000          LAR  = NTH + 1
000040      000    C
000041      000          NSPT = 0
000042      000          NL1 = 0
000043      000          NL2 = 0
000044      000          NL3 = 0
000045      000          IF(LHTB .GT. 0) NL1 = NDATA(LHTB)
000046      000          IF(LHFP .GT. 0) NL2 = NDATA(LHFP)
000047      000          IF(LHC  .GT. 0) NL3 = NDATA(LHC )
000048      000          NSPT = (NL1/4 + NL2/5 + NL3/2) * 3
000049      000          NEXT(LAR) = NSPT
000050      000          IF(NDIM .GE. NSPT) GO TO 104
000051      000          NEED = NSPT - NDIM
000052      000          CALL TOPLIN
000053      000          WRITE(6,103) NEED
000054      000      103 FORMAT(83H0* * *  INSUFFICIENT DYNAMIC STORAGE AVAILABLE FOR CABIN
000055      000         1 ANALYSIS SUBROUTINE  * * * // 8X 5HSHORT 15, 10H LOCATIONS)
```

```
000056  000        CALL WLKBCK
000057  000        CALL EXIT
000058  000  C
000059  000    104 CONTINUE
000060  000  C
000061  000        NS = NDATA(NST+1)
000062  000        NV1 = IABS(NDATA(NST)) - 1
000063  000        IF (NDATA(NST) .LT. 0) GO TO 2
000064  000        IF(NS .NE. NV1/3) CALL ERR(3H1ST)
000065  000        NDATA(NST) = -NDATA(NST)
000066  000      2 IF (NDATA(NCRV) .LT. 0) GO TO 4
000067  000        IF(NDATA(NCRV) .NE. 8) CALL ERR(3H2ND)
000068  000        NDATA(NCRV) = -NDATA(NCRV)
000069  000      4 IF(NDATA(NCON) .NE. 11) CALL ERR(3H3RD)
000070  000        NCPA = NDATA(NCRV+4)
000071  000        NCPV = NDATA(NCRV+5)
000072  000        LAMDA = NDATA(NCRV+8)
000073  000        RA = DATA(NCON+1)
000074  000        RV = DATA(NCON+2)
000075  000        VC = DATA(NCON+3)
000076  000        PC = DATA(NCON+4)
000077  000        XC = DATA(NCON+5)
000078  000        WV = DATA(NCON+6)
000079  000        PSICAB = DATA(NCON+7)
000080  000        PO = DATA(NCON+8)
000081  000        TO = DATA(NCON+9)
000082  000        CONV = DATA(NCON+10)
000083  000        TZ = DATA(NCON+11)
000084  000        FLOIN = 0.0
000085  000        PSIIN = 0.0
000086  000        TIN = 0.0
000087  000        FLOCP = 0.0
000088  000        DO 5 I=1,NV1,3
000089  000        LOC = NST + I + 1
000090  000        LOC1 = LOC + 1
000091  000        LOC2 = LOC + 2
000092  000        FLO = DATA(LOC)
000093  000        IF(IABS(NDATA(LOC)).LE. 99999 .AND. IABS(NDATA(LOC)).GT. 0)
000094  000       X  FLO = POL(NDATA(LOC),TIME)
000095  000        PSI = DATA(LOC1)
000096  000        IF(IABS(NDATA(LOC1)).LE. 99999 .AND. IABS(NDATA(LOC1)).GT. 0)
000097  000       X  PSI = POL(NDATA(LOC1),TIME)
000098  000        TEMP = DATA(LOC2)
000099  000        IF(IABS(NDATA(LOC2)).LE. 99999 .AND. IABS(NDATA(LOC2)).GT. 0)
000100  000       X  TEMP = POL(NDATA(LOC2),TIME)
000101  000        FLOIN = FLOIN + FLO
000102  000        PSIIN = PSIIN + FLO*PSI
000103  000        CPIN  = (POL(NCPA,TEMP)+PSI*POL(NCPV,TEMP))/(1.0+PSI)
000104  000        TIN   = TIN + FLO*CPIN*TEMP
000105  000        FLOCP = FLOCP + FLO*CPIN
000106  000      5 CONTINUE
000107  000        PSIIN = PSIIN/FLOIN
000108  000        TIN = TIN/FLOCP
000109  000        FLOC = POL(NDATA(NCRV+1),TIME)
000110  000        WVIN = TINC*FLOIN*PSIIN/(1.0+PSIIN)
000111  000        RG = RA*(1.0+PSIIN/XC)/(1.0+PSIIN)
000112  000        RHOIN = PC/(RG*(TIN-TZ))
```

```
000113  000          FLOUT = FLOIN*(WV+WV/PSICAB)/VC/RHOIN
000114  000          WVOUT = TINC*FLOUT*PSICAB/(1.0+PSICAB)
000115  000          WV = WV + WVIN - WVOUT - SUMWL
000116  000          DATA(NCON+6) = WV
000117  000          PV = WV+RV*(TC-TZ)/VC
000118  000          PA = PC - PV
000119  000          WA = VC*PA/RA/(TC-TZ)
000120  000          PSICAB = WV/WA
000121  000          DATA(NCON+7) = PSICAB
000122  000          UA = POL(NDATA(NCRV+2),TC)
000123  000          UV = POL(NDATA(NCRV+3),TC)
000124  000          CPA = POL(NCPA,TC)
000125  000          CPV = POL(NCPV,TC)
000126  000          CA = POL(NDATA(NCRV+6),TC)
000127  000          CV = POL(NDATA(NCRV+7),TC)
000128  000          UC = (XC*UA+PSICAB*UV)/(XC+PSICAB)
000129  000          CPC = (CPA+PSICAB*CPV)/(1.0+PSICAB)
000130  000          CC = (XC*CA+PSICAB*CV)/(XC+PSICAB)
000131  000          RHOC = (WV+WA)/VC
000132  000          TC = TC + (FLOIN*(TIN-TC) - SUMQL/CPC)/(WV+WA)*TINC
000133  000          PRC = CC/(RHOC*CPC)
000134  000          SUMQL = 0.0
000135  000          SUMWL = 0.0
000136  000          LL = LAR
000137  000    C
000138  000          IF(LHTB .EQ. 0) GO TO 25
000139  000          PRC31 = PRC**.31
000140  000          IF(MOD(NDATA(LHTB),4) .NE. 0) CALL ERR(3H7TH)
000141  000          DO 20 I=1,NL1,4
000142  000          LOC = LHTB + I
000143  000          J = NDATA(LOC)
000144  000          DI = DATA(LOC+1)
000145  000          AI = DATA(LOC+2)
000146  000          VIWO = DATA(LOC+3)
000147  000          VI = VIWO*FLOC
000148  000          RE = VI*DI*RHOC/UC
000149  000          IF(IFIX((RE-22000)/18000)) ,7,10
000150  000          XNU = .43 + .533*SQRT(RE)*PRC31
000151  000          GO TO 15
000152  000        7 XNU = .43 + .193*RE**.618*PRC31
000153  000          GO TO 15
000154  000       10 XNU = .43 + .0265*RE**.805*PRC31
000155  000       15 HA = AI*CC*XNU/DI
000156  000          CALL QSUM
000157  000       20 CONTINUE
000158  000    C
000159  000       25 IF(LHFP .EQ. 0) GO TO 35
000160  000          PRC33 = CBRT(PRC)
000161  000          IF(MOD(NDATA(LHFP),5) .NE. 0) CALL ERR(3H6TH)
000162  000          DO 30 I=1,NL2,5
000163  000          LOC = LHFP + I
000164  000          J = NDATA(LOC)
000165  000          XX = DATA(LOC+1)
000166  000          XI = DATA(LOC+2)
000167  000          AI = DATA(LOC+3)
000168  000          VIWO = DATA(LOC+4)
000169  000          VI = VIWO*FLOC
```

```
000170  000          VRU = VI*RHOC/UC
000171  000          XNU = .664*PRC3**(SQRT(VRU*(XX+XI)) - SQRT(VRU*XX))
000172  000          HA = AI*CC*XNU/XI
000173  000          CALL QSUM
000174  000       30 CONTINUE
000175  000    C
000176  000       35 IF(LHC .EQ. 0) GO TO 45
000177  000          IF(MOD(NDATA(LHC),2) .NE. 0) CALL ERR(3H5TH)
000178  000          DO 40 I=1,NL3,2
000179  000          LOC = LHC + I
000180  000          J = NDATA(LOC)
000181  000          HA = DATA(LOC+1)
000182  000          CALL QSUM
000183  000       40 CONTINUE
000184  000    C
000185  000       45 WVPRME = WV + WVIN - WVOUT - SUMWL
000186  000          PVPRME = WVPRME*RV*(TC-TZ)/VC
000187  000          SUMWL = 0.
000188  000          KK = NEXT(LAR)/3*2 + LAR
000189  000          LL = LAR
000190  000          IF(LHTB .GT. 0) CALL CONCK(NL1,4,LHTB)
000191  000          IF(LHFP .GT. 0) CALL CONCK(NL2,5,LHFP)
000192  000          IF(LHC  .GT. 0) CALL CONCK(NL3,2,LHC)
000193  000          RETURN
000194  000    C
000195  000          SUBROUTINE CONCK(NN,NUM,IND)
000196  000          DO 60 I=1,NN,NUM
000197  000          J = NDATA(IND+I)
000198  000          LL = LL + 1
000199  000          PWI = EXT(LL)
000200  000          LL = LL + 1
000201  000          DWI = EXT(LL)
000202  000          PVPW = PV - PWI
000203  000          IF((PVPRME-PWI)/PVPW .LT. 0.) DWI = DWI*PVPW/(PV-PVPRME)
000204  000          KK = KK + 1
000205  000          IF(EXT(KK)+DWI .LT. 0.0) DWI = -EXT(KK)
000206  000          EXT(KK) = EXT(KK) + DWI
000207  000          SUMWL = SUMWL + DWI
000208  000          QL = DWI*POL(LAMDA,T(J))/TINC
000209  000          Q(J) = Q(J) + QL
000210  000       60 CONTINUE
000211  000          RETURN
000212  000    C
000213  000          SUBROUTINE ERR(NUMB)
000214  000          CALL TOPLIN
000215  000          WRITE(6,100) NUMB
000216  000      100 FORMAT(1HO 131(1H=)//1X'THE 'A3,' ARGUMENT IN THE CALL DOES NOT HA
000217  000         XVE THE CORRECT NUMBER OF VALUES. EXECUTION TERMINATED IN SUBROUTIN
000218  000         XE CABIN'//1X 131(1H=))
000219  000          CALL WLKBCK
000220  000          CALL EXIT
000221  000    C
000222  000          SUBROUTINE QSUM
000223  000          IF(EXPLCT) DTAU(J) = DTAU(J) + HA
000224  000          Q(J) = Q(J) + HA*(TC-T(J))
000225  000          QL = HA*(TC-T(J))*TINC
000226  000          SUMQL = SUMQL + QL
```

```
000227      000          XLAM = POL(LAMDA,T(J))
000228      000          PWI = PO*EXP(XLAM/RV/(TO-TZ)*(T(J)-TO)/(T(J)-TZ)*CONV)
000229      000          LL = LL + 1
000230      000          EXT(LL) = PWI
000231      000          DWI = HA/CPC/PC*(PV-PWI)*TINC
000232      000          LL = LL + 1
000233      000          EXT(LL) = DWI
000234      000          SUMWL = SUMWL + DWI
000235      000          RETURN
000236      000          END

END ELT.


#HDG,P CARDIN
```

```
@ELT,L CARDIN
ELTOT7 RLI870 02/28-03:18:48-(2,)
000001      000          SUBROUTINE CARDIN(ISW)
000002      000          COMMON /TAPE  / NIN, NOUT
000003      000          COMMON /CARD  / KRD, KOL, MXKOL
000004      000          COMMON /CIMAGE/ KARD(80)
000005      000          COMMON /SRDCOM/ KOL1, KOL27, ALPH(14), KODE, N, KOL11, B81(81),
000006      000         1                IWRDS, FMT(70), B62(62), K4TRAN(2?- KB, NRNSFR(22)
000007      000    C
000008      000          DATA  KC, KREM, KD , KEND / 1HC, 3HREM, 1H$, 3HEND /
000009      000    C
000010      000    C
000011      000        5 CALL SREADC(5)
000012      000          CALL SREADC(11)
000013      000          WRITE(NOUT,6) (KARD(I),I=2,80), KOL1
000014      000        6 FORMAT(1X, 80A1)
000015      000          IF(KOL1 .EQ. KC) GO TO 5
000016      000          IF(KODE .EQ. KREM) GO TO 5
000017      000          DO 10 I=12,80
000018      000          IF(KARD(I) .EQ. KD) GO TO 20
000019      000       10 CONTINUE
000020      000          I = 81
000021      000       20 MXKOL = I - 1
000022      000          ISW = 1
000023      000          IF(KODE .EQ. KEND) ISW = 2
000024      001          KOL=12
000025      000          RETURN
000026      000          END

END ELT.


@HOG,P CINOSL
```

```
*ELT,L SICOUM
ELTDT7 RLIB70 02/28-03:21:28-(0,)
000001      000    COUM     PROC
000002      000          IF(FLD(1,1,NSQ1(K1+1)).EQ.0) GO TO 200
000003      000          NTYPE = FLD(0,5,NSQ2(K2))
000004      000          GO TO (199,198,198,198,199,198,198,198,199), NTYPE
000005      000      198 K2 = K2+1
000006      000      199 K2 = K2+1
000007      000      200 CONTINUE
000008      000     END

END ELT.


*HDG,P SICOMM
```

```
@ELT,L CINDSL
ELTOT7 RL1870 02/28-03:18:49-(1,)
000001      000          SUBROUTINE CINDSL
000002      000    C     STEADY STATE EXECUTION SUBROUTINE FOR SINDA     FORTRAN V
000003      000    C     THE LONG PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000004      000    C     DIFFUSION NODES RECEIVE A SUCCESSIVE POINT ITERATION
000005      000    C     ARITHMETIC NODES RECEIVE A SUCESSIVE POINT ITERATION
000006      000    C     OVER-RELAXATION IS ALLOWED, THE DAMPING FACTORS ARE ADDRESSABLE
000007      001          LOGICAL FLOW
000008      001          COMMON /FDIMNS/ NTYP, NSYS
000009      000          INCLUDE COMM,LIST
000010      000          INCLUDE DEFF,LIST
000011      000          IF(KON(5).LE.0) GO TO 999
000012      000          IF(CON(9).LE.0.) CON(9) = 1.0
000013      000          IF(CON(10).LE.0.) CON(10) = 1.0
000014      000          IF(NNA.GT.0.AND.CON(19).LE.0.) GO TO 998
000015      000          IF(NND.GT.0.AND.CON(26).LE.0.) GO TO 997
000016      000          IF(KON(31).NE.1) GO TO 994
000017      000          IF (CON(50) .LE. 0.) CON(50)=1.0
000018      000          PASS = -1.0
000019      000          NN = NND+1
000020      000          NNC = NNA+NND
000021      001          FLOW = .FALSE.
000022      001          TZERO = -460.
000023      001          NSP = 0
000024      001          IXF = NTH
000025      001          IF(NSYS .LT. 1) GO TO 2
000026      001          FLOW = .TRUE.
000027      001          NSP = NNT
000028      001          DO 1 I=1,NNT
000029      001          NX(IXF+I)=0
000030      001        1 CONTINUE
000031      001        2 IE1=IXF+NSP
000032      000          IE2 = IE1+NNC
000033      000          NLA = NDIM
000034      001          JJ = 2*(NNC+NSP)
000035      000          NTH = NTH+JJ
000036      000          NDIM = NDIM-JJ
000037      000          IF(NDIM.LT.0) GO TO 996
000038      000          CON(1) = CON(13)
000039      000          CON(14) = CON(13)
000040      000          GO TO 10
000041      000        5 CON(1) = CON(13)+CON(18)
000042      000          IF(CON(1)-CON(3).GT.0.) CON(1) = CON(3)
000043      000          CON(14) = (CON(1)+CON(13))/2.0
000044      000          CON(2) = CON(1)-CON(13)
000045      000    C     COMPUTE STEADY STATE TEMPERATURES
000046      000       10 LAX = KON(5)
000047      000          JJ = 0
000048      000          DO 145 K1 = 1,LAX
000049      000          JJ = JJ+1
000050      000          KON(20) = K1
000051      000    C     ZERO OUT ALL SOURCE LOCATIONS
000052      000          DO 15 I = 1,NNC
000053      000       15 Q(I) = 0.0
000054      000          CALL VARBLI
000055      000          IF(PASS.GE.0.) GO TO 20
```

```
000056  000          CALL OUTCAL
000057  000          PASS = 1.0
000058  000       20 J1 = 0
000059  000          J2 = 1
000060  000          RLXD = 0.0
000061  000          RLXA = 0.0
000062  000          IF(NND.LE.0) GO TO 75
000063  001          CON(30) = 0.0
000064  001          IF(FLOW) CALL FLUID(5,0,IXF,0.0,KOP)
000065  001          RLXCA = CON(30)
000066  000          DN = CON(10)
000067  000          DD = 1.0-DN
000068  000    C     DO A SUCCESSIVE POINT ITERATION ON THE DIFFUSION NODES
000069  000          DO 70 I = 1,NND
000070  000          GSUM = 0.0
000071  000          INCLUDE DUMC,LIST
000072  000          INCLUDE VARD,LIST
000073  000       25 J1 = J1+1
000074  000          LG = FLD(5,16,NSD1(J1))
000075  001          IF(LG.EQ.0) GO TO 36
000076  000          LTA = FLD(22,14,NSD1(J1))
000077  000          INCLUDE VARG,LIST
000078  000    C     CHECK FOR RADIATION CONDUCTOR
000079  000          IF(FLD(3,1,NSD1(J1)).EQ.0) GO TO 30
000080  000          T1 = T(I)+460.0
000081  000          T2 = T(LTA)+460.0
000082  000          GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000083  000          GV = GV*CON(50)
000084  000          GO TO 35
000085  000       30 GV = G(LG)
000086  000       35 GSUM = GSUM+GV
000087  000          Q(I) = Q(I)+GV*T(LTA)
000088  000    C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000089  000          IF(NSD1(J1).GT.0) GO TO 25
000090  001       36 IF(.NOT. FLOW) GO TO 40
000091  001          LMP = NX(IXF+I)
000092  001          IF(LMP .EQ. 0) GO TO 40
000093  001          HA = X(IXF+LMP)
000094  001          Q(I) = Q(I) + HA*T(LMP)
000095  001          GSUM = GSUM + HA
000096  001       40 T2 = DD*T(I) + DN*Q(I)/GSUM
000097  000    C     OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000098  000          T1 = ABS(T(I)-T2)
000099  000    C     STORE THE NEW TEMPERATURES AND EXTRAPOLATION FACTORS
000100  000          GO TO(65,60,55),JJ
000101  000       55 LE1 = IE1+I
000102  000          LE2 = IE2+I
000103  000          R1 = T2-T(I)
000104  000          X(LE1) = T(I)
000105  000          X(LE2) = R1/(R1-X(LE2))
000106  000          GO TO 65
000107  000       60 LE2 = IE2+I
000108  000          X(LE2) = T2-T(I)
000109  000       65 T(I) = T2
000110  000    C     SAVE THE MAXIMUM DIFFUSION RELAXATION CHANGE
000111  000          IF(RLXD.GT.T1) GO TO 70
000112  000          RLXD = T1
```

```
000113      000          N1 = I
000114      000       70 CONTINUE
000115      000          CON(27) = RLXD
000116      000          IF(NNA.LE.0) GO TO 130
000117      000       75 DN = CON(9)
000118      000          DD = 1.0-DN
000119      000          JJ1 = J1
000120      000          JJ2 = J2
000121      000    C     DO SUCCESSIVE POINT ITERATION ON ARITHMETIC NODES
000122      000          DO 125 I = NN,NNC
000123      000          GSUM = 0.0
000124      000          L = I
000125      000          INCLUDE VR02,LIST
000126      000       80 JJ1 = JJ1+1
000127      000          LG = FLD(5,16,NSQ1(JJ1))
000128      000          LTA = FLD(22,14,NSQ1(JJ1))
000129      000          INCLUDE VR02,LIST
000130      000    C     CHECK FOR RADIATION CONDUCTOR
000131      000          IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 85
000132      000          T1 = T(I)+460.0
000133      000          T2 = T(LTA)+460.0
000134      000          GV = G(LG)*(T1+T1+T2+T2)*(T1+T2)
000135      000          GV = GV*CON(50)
000136      000          GO TO 90
000137      000       85 GV = G(LG)
000138      000       90 Q(I) = Q(I)+GV*T(LTA)
000139      000          GSUM = GSUM+GV
000140      000    C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000141      000          IF(NSQ1(JJ1).GT.0) GO TO 80
000142      000          T2 = DD*T(I)+DN*Q(I)/GSUM
000143      000          T1 = ABS(T(I)-T2)
000144      000    C     STORE THE NEW TEMPERATURES AND EXTRAPOLATION FACTORS
000145      000          GO TO(120,115,110),JJ
000146      000      110 LE1 = IE1+I
000147      000          LE2 = IE2+I
000148      000          R1 = T2-T(I)
000149      000          X(LE1) = T(I)
000150      000          X(LE2) = R1/(R1-X(LE2))
000151      000          GO TO 120
000152      000      115 LE2 = IE2+I
000153      000          X(LE2) = T2-T(I)
000154      000      120 T(I) = T2
000155      000          IF(RLXA.GT.T1) GO TO 125
000156      000          RLXA = T1
000157      000          N2 = I
000158      000      125 CONTINUE
000159      000          CON(30) = RLXA
000160      000    C     SEE IF THE RELAXATION CRITERIA ARE MET
000161      000      130 IF(RLXA.LE.CON(19).AND.RLXD.LE.CON(26)) GO TO 150
000162      000          IF(JJ.LE.2) GO TO 140
000163      000          JJ = 0
000164      000          DO 135 I = 1,NNC
000165      000          LE2 = IE2+I
000166      000    C     SEE IF THE EXTRAPOLATION CRITERIA ARE MET
000167      000          IF(X(LE2).GE.0.) GO TO 135
000168      000          IF(X(LE2).LT.-8.) X(LE2) = -8.
000169      000          LE1 = IE1+I
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
000179  000          T(I) = X(LE2)*X(LE1)+(1.0-X(LE2))*T(I)
000171  000      135 CONTINUE
000172  000      140 IF(KON(7).NE.0) CALL OUTCAL
000173  000      145 CONTINUE
000174  000          WRITE(6,882)
000175  000          WRITE(6,885) LAX
000176  000          KON(28) = KON(28)+2
000177  000      150 KON(37) = N2
000178  000          IF(RLXA.GT.RLXO) GO TO 155
000179  000          CON(30) = RLXO
000180  000          KON(37) = N1
000181  000    C     CHECK THE ENERGY BALANCE OF THE SYSTEM
000182  000      155 CALL NONLIN
000183  000          QOUT = 0.0
000184  000          QIN = 0.0
000185  000          J1 = 0
000186  000          DO 195 I = 1,NNC
000187  000          QIN = QIN+Q(I)
000188  001          IF(.NOT. FLOW) GO TO 165
000189  001          LMP = NX(IXF+I)
000190  001          IF(LMP .EQ. 0) GO TO 165
000191  001          QIN = QIN + X(IXF+LMP)*(T(LMP)-T(I))
000192  000      165 J1 = J1+1
000193  000          LTA = FLD(22,14,NSO1(J1))
000194  000          IF(LTA.LE.NNC) GO TO 175
000195  000          LG = FLD(5,16,NSO1(J1))
000196  001          IF(LG.EQ.0) GO TO 195
000197  000          IF(FLD(3,1,NSO1(J1)).EQ.0) GO TO 170
000198  000          T1 = T(I)+460.0
000199  000          T2 = T(LTA)+460.0
000200  000          QOUT = QOUT +G(LG)*CON(50)*(T1**4 -T2**4)
000201  000          GO TO 175
000202  000      170 QOUT = QOUT+G(LG)*(T(I)-T(LTA))
000203  000    C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000204  000      175 IF(NSO1(J1).GT.0) GO TO 165
000205  000      195 CONTINUE
000206  000          CON(32) = ABS(QIN-QOUT)
000207  000          CALL VARBL2
000208  000          CON(3) = CON(1)
000209  000          CALL OUTCAL
000210  000          WRITE(6,882)
000211  000          WRITE(6,883) KON(20),CON(32)
000212  000          KON(28) = KON(28)+2
000213  000          IF(CON(3).GT.CON(1)*1.000001) GO TO 5
000214  001          NTH = IXF
000215  000          NDIM = NLA
000216  000          RETURN
000217  000      994 WRITE(6,884)
000218  000          GO TO 1000
000219  000      996 WRITE(6,886) NDIM
000220  000          GO TO 1000
000221  000      997 WRITE(6,887)
000222  000          GO TO 1000
000223  000      998 WRITE(6,888)
000224  000          GO TO 1000
000225  000      999 WRITE(6,889)
000226  000     1000 CALL OUTCAL
```

```
000227      000          CALL EXIT
000228      000      882 FORMAT(1H )
000229      000      883 FORMAT(10H LOOPCT = I6,10H ENGBAL = E12.5)
000230      000      884 FORMAT(45H CINDSL REQUIRES LONG PSEUDO-COMPUTE SEQUENCE)
000231      000      885 FORMAT(35H ITERATION COUNT EXCEEDED, LOOPCT =,  I10)
000232      000      886 FORMAT(I8,20H LOCATIONS AVAILABLE)
000233      000      887 FORMAT(10H NO DRLXCA)
000234      000      888 FORMAT(10H NO ARLXCA)
000235      000      889 FORMAT(14H NO LOOP COUNT)
000236      000          END

END ELT.


*HDGIPCI**OSS
```

```
*ELT,L CINDSS
ELTOT7 RLIB70 02/28-03:18:51-(6,)
000001      000          SUBROUTINE CINDSS
000002      000    C     STEADY STATE EXECUTION SUBROUTINE FOR SINDA     FORTRAN V
000003      000    C     THE SHORT PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000004      000    C     DIFFUSION NODES RECEIVE A BLOCK ITERATION
000005      000    C     ARITHMETIC NODES RECEIVE A SUCESSIVE POINT ITERATION
000006      000    C     OVER-RELAXATION IS ALLOWED, THE DAMPING FACTORS ARE ADDRESSABLE
000007      001          COMMON /FDIMNS/ NTYP, NSYS
000008      000          INCLUDE COMM,LIST
000009      000          INCLUDE OEFF,LIST
000010      000          IF(KON(5).LE.0) GO TO 999
000011      000          IF(CON(9).LE.0.) CON(9) = 1.0
000012      000          IF(CON(10).LE.0.) CON(10) = 1.0
000013      000          IF(NNA.GT.0.AND.CON(19).LE.0.) GO TO 998
000014      000          IF(NND.GT.0.AND.CON(26).LE.0.) GO TO 997
000015      004           IF(CON(50) .LE. 0) CON(50)= 1.
000016      000          IF(KON(31).NE.0) GO TO 994
000017      000          PASS = -1.0
000018      000          NN = NND+1
000019      000          NNC=NNA+NND
000020      000          IE=NTH
000021      000          NLA = NOIM
000022      000          NTH=NTH+NND
000023      000          NOIM=NOIM-NND
000024      000          IF(NOIM.LT.0) GO TO 996
000025      000          CON(1) = CON(13)
000026      000          CON(14) = CON(13)
000027      000          GO TO 15
000028      000       10 CON(1) = CON(13)+CON(18)
000029      000          IF(CON(1)-CON(3).GT.0.) CON(1) = CON(3)
000030      000          CON(14) = (CON(1)+CON(13))/2.0
000031      000    C     COMPUTE STEADY STATE TEMPERATURES
000032      000       15 LAX = KON(5)
000033      000          DO 120 K1 = 1,LAX
000034      000          KON(20) = K1
000035      000    C     ZERO OUT ALL SOURCE LOCATIONS
000036      000          DO 20 I = 1,NNC
000037      000       20 Q(I)=0.
000038      000          CALL VARBLE
000039      000          IF(PASS.GE.0.)GO TO 25
000040      000          CALL OUTCAL
000041      000          PASS = 1.0
000042      000       25 J1 = 0
000043      000          J2 = 1
000044      000          RLXD = 0.0
000045      000          RLXA = 0.0
000046      000          IF(NND.LE.0) GO TO 75
000047      000          ON = CON(10)
000048      000          OD = 1.0-ON
000049      000    C     ZERO OUT EXTRA LOCATIONS
000050      000          DO 30 I = 1,NND
000051      000          LE=IE+I
000052      000       30 X(LE)=0.0
000053      000    C     DO A BLOCK ITERATION ON THE DIFFUSION NODES
000054      000          DO 70 I = 1,NND
000055      000          LE=IE+I
```

```
000056  000        INCLUDE DUMC,LIST
000057  000        INCLUDE VARQ,LIST
000058  000     35 J1 = J1+1
000059  000        LG = FLD(5,16,NSQ1(J1))
000060  000        IF(LG.EQ.0) GO TO 50
000061  000        LTA = FLD(22,14,NSQ1(J1))
000062  000        INCLUDE VARG,LIST
000063  000  C     CHECK FOR RADIATION CONDUCTOR
000064  000        IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 40
000065  000        T1 = T(I)+460.0
000066  000        T2 = T(LTA)+460.0
000067  000        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000068  004         GV= GV *CON(50)
000069  000        GO TO 45
000070  000     40 GV = G(LG)
000071  000     45 X(LE) = X(LE)+GV
000072  000        Q(I) = Q(I)+GV*T(LTA)
000073  000  C     CHECK FOR ADJOINING DIFFUSION NODE, WATCH FOR ONE WAY CONDUCTOR
000074  000        IF(LTA.GT.NND.OR.FLD(21,1,NSQ1(J1)).EQ.1) GO TO 50
000075  000        LE1 = IE+LTA
000076  000        X(LE1) = X(LE1)+GV
000077  000        Q(LTA) = Q(LTA)+GV*T(I)
000078  000  C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000079  000     50 IF(NSQ1(J1).GT.0) GO TO 35
000080  000     70 CONTINUE
000081  001        KOP = CON(7)
000082  005        CON(30) = 0.0
000083  005        KON(37) = 0
000084  005        IF(NSYS.NE.0) CALL FLUID(3,IE,0,0.,KOP)
000085  005        RLXD = CON(30)
000086  005        N1 = KON(37)
000087  005        DO 74 I=1,NND
000088  006        LE = IE+I
000089  005        IF(.NOT.X(LE).GT.0.0) GO TO 74
000090  005        T2 = OD*T(I)+DN*Q(I)/X(LE)
000091  005        T1 = ABS(T(I)-T2)
000092  005        T(I) = T2
000093  005        IF(RLXD.GT.T1) GO TO 74
000094  005        RLXD = T1
000095  005        N1 = I
000096  005     74 CONTINUE
000097  005     76 CON(27) = RLXD
000098  000        IF(NNA.LE.0) GO TO 115
000099  000     75 DN = CON(9)
000100  000        OD = 1.0-DN
000101  000        JJ1 = J1
000102  000        JJ2 = J2
000103  000  C     DO A SUCCESSIVE POINT ITERATION ON THE ARITHMETIC NODES
000104  000        DO 110 I = NN,NNC
000105  000        GSUM = 0.0
000106  000        L = I
000107  000        INCLUDE VRQ2,LIST
000108  000     80 JJ1 = JJ1+1
000109  004         LG= FLD(5,16,NSQ1(JJ1))
000110  000        LTA = FLD(22,14,NSQ1(JJ1))
000111  000        INCLUDE VRG2,LIST
000112  000  C     CHECK FOR RADIATION CONDUCTOR
```

```
000113  000          IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 85
000114  000          T1 = T(I)+460.0
000115  000          T2 = T(LTA)+460.0
000116  000          GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000117  004           GV= GV *CON(50)
000118  000          GO TO 90
000119  000       85 GV = G(LG)
000120  000       90 Q(I) = Q(I)+GV*T(LTA)
000121  000          GSUM = GSUM+GV
000122  000    C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000123  000          IF(NSQ1(JJ1).GT.0) GO TO 60
000124  000          T2 = DD*T(I)+DN*Q(I)/GSUM
000125  000          T1 = ABS(T(I)-T2)
000126  000    C     STORE THE NEW TEMPERATURES
000127  000          T(I) = T2
000128  000          IF(RLXA.GT.T1) GO TO 110
000129  000          RLXA = T1
000130  000          N2 = I
000131  000      110 CONTINUE
000132  000          CON(30) = RLXA
000133  000    C     SEE IF THE RELAXATION CRITERIA ARE MET
000134  000      115 IF(RLXA.LE.CON(19).AND.RLXD.LE.CON(26)) GO TO 125
000135  000          IF(KON(7).NE.0) CALL OUTCAL
000136  000      120 CONTINUE
000137  000          WRITE(6,882)
000138  000          WRITE(6,885) LAX
000139  000          KON(28) = KON(28)+2
000140  000      125 KON(37) = N2
000141  000          IF(RLXA.GT.RLXD) GO TO 155
000142  000          CON(30) = RLXD
000143  000          KON(37) = N1
000144  000    C     CHECK THE ENERGY BALANCE OF THE SYSTEM
000145  000      155 CALL NONLIN
000146  000          QOUT = 0.0
000147  000          QIN = 0.0
000148  000          J1 = 0
000149  000          DO 195 I = 1,NNC
000150  000          QIN = QIN+Q(I)
000151  000      165 J1 = J1+1
000152  000          LTA = FLD(22,14,NSQ1(J1))
000153  000          IF(LTA.LE.NNC) GO TO 175
000154  000          LG = FLD(5,16,NSQ1(J1))
000155  000          IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 170
000156  000          T1 = T(I)+460.0
000157  000          T2 = T(LTA)+460.0
000158  004           QOUT= QOUT +G(LG)*CON(50)*(T1**4-T2**4)
000159  000          GO TO 175
000160  000      170 QOUT = QOUT+G(LG)*(T(I)-T(LTA))
000161  000    C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000162  000      175 IF(NSQ1(J1).GT.0) GO TO 165
000163  000      195 CONTINUE
000164  000          CON(32) = ABS(QIN-QOUT)
000165  000          CALL VARBL2
000166  000          CON(13) = CON(1)
000167  000          CALL OUTCAL
000168  000          WRITE(6,882)
000169  000          WRITE(6,883) KON(20),CON(32)
```

```
000170  000        KON(28) = KON(28)+2
000171  000        IF(CON(3).GT.CON(1)+1.000001) GO TO 10
000172  000        NTH=IE
000173  000        NDIM = NLA
000174  000        RETURN
000175  000    994 WRITE(6,884)
000176  000        GO TO 1000
000177  000    996 WRITE(6,886) NDIM
000178  000        GO TO 1000
000179  000    997 WRITE(6,887)
000180  000        GO TO 1000
000181  000    998 WRITE(6,888)
000182  000        GO TO 1000
000183  000    999 WRITE(6,889)
000184  000   1000 CALL OUTCAL
000185  000        CALL EXIT
000186  000    882 FORMAT(1H )
000187  000    883 FORMAT(10H LOOPCT = I6,10H ENGBAL = E12.5)
000188  000    884 FORMAT(46H CINDSS REQUIRES SHORT PSEUDO-COMPUTE SEQUENCE)
000189  000    885 FORMAT(35H ITERATION COUNT EXCEEDED, NLOOP = ,  I10)
000190  000    886 FORMAT(I8,20H LOCATIONS AVAILABLE)
000191  000    887 FORMAT(10H NO DRLXCA)
000192  000    888 FORMAT(10H NO ARLXCA)
000193  000    889 FORMAT(14H NO LOOP COUNT)
000194  000        END

END ELT.


$HDG,P CMPRSS
```

```
#ELT,L CMPRSS
ELTOTF RLI870 02/28-03:18:53-(0,)
000001      000          SUBROUTINE CMPRSS(SPR,IPR,A,B,L28,*)
000002      000    C
000003      000    C THIS SUBROUTINE REDUCES THE COEFFICIENT MATRIX FOR PFCS
000004      000    C
000005      000    C
000006      000          DIMENSION A(1), B(1), L28(1)
000007      000    C
000008      000    C
000009      000    C
000010      000    C LOCATE RELATIVE PRESSURE NODE NUMBER (NPR) OF ACTUAL PRESSURE
000011      000    C NODE NUMBER (IPR) WHICH HAS A SPECIFIED PRESSURE (SPR)
000012      000    C
000013      000          MPRN = L28(1)
000014      000          DO 5 NPR=1,MPRN
000015      000          IF(IPR .EQ. L28(NPR+1)) GO TO 10
000016      000        5 CONTINUE
000017      000          RETURN 6
000018      000    C
000019      000    C CALCULATE THE STARTING LOCATION OF COLUMN NPR
000020      000    C
000021      000       10 LOC = (NPR-1)*NPR/2
000022      000          LD = LOC + NPR + NPR
000023      000    C
000024      000    C MODIFY THE RIGHT HAND SIDE
000025      000    C
000026      000          DO 20 J=1,MPRN
000027      000          IF(J-NPR) 12,20,15
000028      000       12 B(J) = B(J) - SPR*A(LOC+J)
000029      000          GO TO 20
000030      000       15 B(J-1) = B(J) - SPR*A(LD)
000031      000          LD = LD + J
000032      000       20 CONTINUE
000033      000    C
000034      000    C CALCULATE THE STARTING LOCATION OF ROW NPR
000035      000    C
000036      000          LD = LOC + NPR
000037      000          NPK = NPR
000038      000    C
000039      000    C DELETE COLUMN NPR
000040      000    C
000041      000          NPR = NPR + 1
000042      000          DO 30 J=NPR,MPRN
000043      000          DO 25 L=1,J
000044      000    C
000045      000    C DELETE ROW NPR
000046      000    C
000047      000          IF(L.EQ. NPK) GO TO 25
000048      000          LOC = LOC + 1
000049      000          A(LOC) = A(LD+L)
000050      000       25 CONTINUE
000051      000          LD = LD + J
000052      000       30 CONTINUE
000053      000    C
000054      000    C DELETE ACTUAL PRESSURE NODE IPR FROM THE LIST OF ACTUAL PRESSURE NODES
000055      000    C
```

```
000056      000          DO 40 J=NFR,MPRN
000057      000          L28(J) = L28(J+1)
000058      000       40 CONTINUE
000059      000          L28(1) = L28(1) - 1
000060      000          RETURN
000061      000          END

END ELT.


#HOG,P CNBACK
```

```
@ELT,L CNBACK
ELTOT7 RL1870 02/28-03:18:55-(7,)
000001      003          SUBROUTINE CNBACK
000002      003    C     IMPLICIT BACKWARD DIFFERENCING EXECUTION SUBROUTINE
000003      003    C     THE LONG PSEUDO-COMPUTE SEQUENCE IS REQUIRED, SINDA  FORTRAN V
000004      003    C     ALL NODES RECEIVE A SUCCESSIVE POINT ITERATION
000005      003    C     RELAXATION CRITERIA MUST BE SPECIFIED
000006      003    C     OVER-RELAXATION IS ALLOWED, THE DAMPENING FACTORS ARE ADDRESSABLE
000007      003          LOGICAL FLOW
000008      003          COMMON /FDIMNS/ NTYP, NSYS
000009      003          COMMON /POINTN/ LNODE
000010      003          INCLUDE COMN,LIST
000011      003          INCLUDE DEFF,LIST
000012      003          IF(KON(5).LE.0) GO TO 999
000013      003          IF(CON(6).LE.0.) CON(6) = 1.E+8
000014      003          IF(CON(8).LE.0.) CON(8) = 1.E+8
000015      003          IF(CON(9).LE.0.) CON(9) = 1.0
000016      003          IF(CON(10).LE.0.) CON(10) = 1.0
000017      003          IF(CON(11).LE.0.) CON(11) = 1.E+8
000018      003          IF(CON(3).LE.CON(13)) GO TO 990
000019      003          IF(CON(18).LE.0.) GO TO 998
000020      003          IF(NNA.GT.0.AND.CON(19).LE.0.) GO TO 997
000021      003          IF(CON(22).LE.0.) GO TO 996
000022      003          IF(NND.GT.0.AND.CON(26).LE.0.) GO TO 995
000023      003          IF(KON(31).NE.1) GO TO 991
000024      005           IF(CON(50) .LE. 0) CON(50)= 1.
000025      003          TZERO = -960.
000026      003          PASS = -1.0
000027      003          NN = NND+1
000028      003          NLA = NDIM
000029      003          NNC = NND + NNA
000030      003          NSP = NND
000031      003          IF(NSYS .NE. 0) NSP = NNT
000032      003          IE1 = NTH
000033      003          IE2 = IE1 + NNT
000034      003          IE3 = IE2 + NSP
000035      003          J = NND + NSP + NNT
000036      003          NTH = NTH+J
000037      003          NDIM =  NDIM-J
000038      003    C     CHECK FOR EXTRA LOCATIONS FOR CALCULATED NODES
000039      003          IF(NDIM.LT.0) GO TO 994
000040      003          FLOW = .FALSE.
000041      003          IF(NSYS .EQ. 0) GO TO 4
000042      003          FLOW = .TRUE.
000043      003          DO 3 I=1,NNC
000044      003          NX(IE2+I) = 0
000045      003        3 CONTINUE
000046      003        4 TPRINT = CON(13)
000047      003    C     INITALIZE TIME SUM BETWEEN OUTPUT INTERVALS
000048      003        5 TSUM = 0.0
000049      003    C     DOES OLD TIME PLUS THE OUTPUT INTERVAL EXCEED THE STOP TIME
000050      003          IF(CON(13)+CON(18).GT.CON(3)) CON(18) = CON(3)-CON(13)
000051      003    C     DONT EXCEED IT
000052      003       10 TSTEPN = CON(22)
000053      003          IF(TSTEPN.LE.CON(8)) GO TO 20
000054      003       15 TSTEPN = CON(8)
000055      003          GO TO 35
```

```
000056  003  C     DOES THE TIME SUM PLUS THE TIME STEP EXCEED OUTPUT INTERVAL
000057  003       20 IF(TSUM+TSTEPN-CON(18)) 30,35,25
000058  003  C     DONT EXCEED IT
000059  003       25 TSTEPN = CON(18)-TSUM
000060  003          GO TO 35
000061  003  C     DOES TIME SUM PLUS TWO TIME STEPS EXCEED OUTPUT INTERVAL
000062  003       30 IF(TSUM+2.0*TSTEPN.LE.CON(18)) GO TO 35
000063  003  C     APPROACH THE OUTPUT INTERVAL GRADUALLY
000064  003          TSTEPN = (CON(18)-TSUM)/2.0
000065  003  C     STORE DELTA TIME STEP IN THE CONSTANTS
000066  003       35 CON(2) = TSTEPN
000067  003  C     CALCULATE THE NEW TIME
000068  003          IF(PASS.GT.0.) GO TO 40
000069  003          CON(1) = TPRINT
000070  003          CON(2) = 0.0
000071  003          GO TO 45
000072  003       40 CON(1) = TPRINT+TSUM+TSTEPN
000073  003  C     COMPUTE THE MEAN TIME BETWEEN ITERATIONS
000074  003       45 CON(19) = (CON(1)+CON(13))/2.0
000075  003          LAX = KON(5)
000076  003          DN = CON(10)
000077  003          DD = 1.0-DN
000078  003          AN = CON(9)
000079  003          AA = 1.0-AN
000080  003  C     DO THE RELAXATION LOOP
000081  003          DO 240 K1 = 1,LAX
000082  003          KON(20) = K1
000083  003          J1 = 0
000084  003          RLXA = 0.0
000085  003          RLXD = 0.0
000086  003          KOP = CON(7)
000087  003          IF (K1 .GT. 1) GO TO 106
000088  003          J2 = 1
000089  003  C     ZERO OUT ALL SOURCE LOCATIONS AND SHIFT TEMPERATURES
000090  003          DO 50 I = 1,NNC
000091  003       50 Q(I) = 0.0
000092  003          DO 55 I = 1,NNT
000093  003          LE1 = IE1+I
000094  003       55 X(LE1) = T(I)
000095  004          IF(FLOW) CALL FLUID(2,IE1,IE2,0.,KOP)
000096  003          KON(12) = 0
000097  003          CALL VARBL1
000098  003  C     CHECK THE BACKUP SWITCH
000099  003          IF(KON(12).NE.0) GO TO 15
000100  003  C     CHECK FOR FIRST PASS
000101  003          IF(PASS.GE.0.) GO TO 60
000102  003          CALL OUTCAL
000103  003          PASS = 1.0
000104  003          GO TO 10
000105  003       60 RC = 1.E+8
000106  003          JJ = 0
000107  003  C     CALCULATE FIRST PASS TEMPERATURES AND CSGMIN
000108  003          DO 105 I = 1,NND
000109  003          INCLUDE VARC,LIST
000110  003  C     FOLD DELTAT INTO THE CAPACITANCES
000111  003          C(I) = C(I)/TSTEPN
000112  003          R1 = 0.0
```

```
000113  003          S  = 0.0
000114  003          G2 = 0.0
000115  003          INCLUDE VARQ,LIST
000116  003          Q(I) = Q(I)+C(I)*(T(I)+460.0)
000117  003          QSUM = Q(I)
000118  003          GSUM = C(I)
000119  003          IF ( .NOT. FLOW) GO TO 70
000120  003          LMP = NX(IE2+I)
000121  003          IF (LMP .EQ. 0) GO TO 70
000122  003          QSUM = QSUM + X(IE2+LMP)*(T(LMP)-TZERO)
000123  003          GSUM = GSUM + X(IE2+LMP)
000124  003       70 J1 = J1+1
000125  003          LG = FLD(5,16,NSQ1(J1))
000126  003          IF (LG .EQ. 0) GO TO 80
000127  003          LTA = FLD(22,14,NSQ1(J1))
000128  003          INCLUDE VARG,LIST
000129  003          T1 = T(I)+460.0
000130  003          T2 = T(LTA)+460.0
000131  003    C     CHECK FOR RADIATION CONDUCTOR
000132  003          IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 75
000133  005           R1= R1 +G(LG)*CON(50)
000134  005           QSUM= QSUM + G(LG)*CON(50)*T2**4
000135  005           G2= G2 + G(LG)*CON(50)*(T1+T1+T2*T2)*(T1+T2)
000136  003          GO TO 80
000137  003       75 GV = G(LG)
000138  003          G2 = G2+GV
000139  003          GSUM = GSUM+GV
000140  003          QSUM = QSUM+GV*T2
000141  003    C     CHECK FOR LAST CONDUCTOR
000142  003       80 IF(NSQ1(J1).GT.0) GO TO 70
000143  003    C     DAMPEN RADIATION ON THIS NODE IF PRESENT
000144  003          IF(R1.LE.0.) GO TO 100
000145  003          R2 = R1*T1**4
000146  003          T2 = (QSUM-R2)/GSUM
000147  003          R1 = R1*T2**4
000148  003          S = (R1+R2)/2.0
000149  003    C     OBTAIN THE NEW TEMPERATURE
000150  003      100 T(I) = (ON*((QSUM-S)/GSUM)+OD*T1)-460.0
000151  003          R1 = C(I)/G2
000152  003          IF(R1.GE.RC) GO TO 105
000153  003          IF ( .NOT. G2 .GT. 0.0) GO TO 105
000154  003          RC = R1
000155  003          KON(35) = I
000156  003      105 CONTINUE
000157  003    C     CONVERT TEMPERATURES TO RANKINE
000158  003          DO 65 I = 1,NNT
000159  003          LE1 = IE1+I
000160  003          T(I) = T(I)+460.
000161  003       65 X(LE1) = X(LE1)+460.
000162  003          CON(17) = RC*TSTEPN
000163  003          GO TO 225
000164  003    C     NOW RELAX THE NETWORK BY SUCCESSIVE POINT AND EXTRAPOLATION
000165  004      106 IF(FLOW) CALL FLUID(2,IE1,IE2,TZERO,KOP)
000166  003      110 JJ = JJ+1
000167  003          DO 165 I = 1,NND
000168  003          R1 = 0.0
000169  003          S = 0.0
```

```
000170  003          QSUM = Q(I)
000171  003          GSUM = C(I)
000172  003          IF(.NOT. FLOW) GO TO 115
000173  003          LMP = NX(IE2+I)
000174  003          IF(LMP .EQ. 0) GO TO 115
000175  003          QSUM = QSUM + X(IE2+LMP)*T(LMP)
000176  003          GSUM = GSUM + X(IE2+LMP)
000177  003      115 J1 = J1+1
000178  003          LG = FLD(5,16,NSQ1(J1))
000179  003          IF (LG .EQ. 0) GO TO 125
000180  003          LTA = FLD(22,14,NSQ1(J1))
000181  003    C     CHECK FOR RADIATION CONDUCTOR
000182  003          IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 120
000183  005           R1= R1 +G(LG)*CON(50)
000184  005           QSUM= QSUM + G(LG)*CON(50)*T(LTA)**4
000185  003          GO TO 125
000186  003      120 GSUM = GSUM+G(LG)
000187  003          QSUM = QSUM+G(LG)*T(LTA)
000188  003    C     CHECK FOR LAST CONDUCTOR
000189  003      125 IF(NSQ1(J1).GT.0) GO TO 115
000190  003    C     DAMPEN RADIATION ON THIS NODE IF PRESENT
000191  003          IF(R1.LE.0.) GO TO 145
000192  003          R2 = R1*T(I)**4
000193  003          T2 = (QSUM+R2)/GSUM
000194  003          R1 = R1*T2**4
000195  003          S = (R1+R2)/2.0
000196  003    C     OBTAIN THE NEW TEMPERATURE
000197  003      145 T2 = DN*((QSUM-S)/GSUM)+DD*T(I)
000198  003    C     OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000199  003          T1 = ABS(T(I)-T2)
000200  003    C     STORE THE NEW AND OLD TEMPERATURES
000201  003          GO TO (160,155,150), JJ
000202  003      150 LE2 = IE2+I
000203  003          LE3 = IE3+I
000204  003          R1 = T2-T(I)
000205  003          X(LE2) = T(I)
000206  003          X(LE3) = R1/(R1-X(LE3))
000207  003          GO TO 160
000208  003      155 LE3 = IE3+I
000209  003          X(LE3) = T2-T(I)
000210  003      160 T(I) = T2
000211  003          IF(RLXD.GE.T1) GO TO 165
000212  003          RLXD = T1
000213  003          KK1 = I
000214  003      165 CONTINUE
000215  003          GO TO (180,180,170), JJ
000216  003    C     PERFORM LINEAR EXTRAPOLATION ON THE ERROR FUNCTION CURVE
000217  003      170 JJ = 0
000218  003          DO 175 I = 1,NND
000219  003          LE3 = IE3+I
000220  007          LE2 = IE2+I
000221  003    C     SEE IF THE EXTRAPOLATION IS ALLOWABLE
000222  006          IF(X(LE3).GE.0.) GO TO 173
000223  003    C     LIMIT THE EXTRAPOLATION
000224  003          IF(X(LE3).LT.-10.) X(LE3) = -10.
000225  003          T(I) = X(LE3)*X(LE2)+(1.0-X(LE3))*T(I)
000226  006      173 NX(LE2) = 0
```

```
000227  003     175 CONTINUE
000228  003     180 IF(NNA.LE.0) GO TO 220
000229  003         JJ1 = J1
000230  003         JJ2 = J2
000231  003         DO 230 I = 1,NNT
000232  003     230 T(I) = T(I)-460.0
000233  003         DO 215 I = NN,NNC
000234  003         L = I
000235  003         GSUM = 0.0
000236  003         IF(K1.GT.2) GO TO 6000
000237  003         INCLUDE VRG2,LIST
000238  003         QSUM = Q(I)
000239  003     185 JJ1 = JJ1+1
000240  003         LG = FLD(5,16,NSQ1(JJ1))
000241  003         LTA = FLD(22,14,NSQ1(JJ1))
000242  003         IF(K1.GT.2) GO TO 4000
000243  003         INCLUDE VRG2,LIST
000244  003         T1 = T(I)+460.0
000245  003         T2 = T(LTA)+460.0
000246  003   C     CHECK FOR RADIATION CONDUCTOR
000247  003         IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 190
000248  003         GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000249  005          GV= GV *CON(50)
000250  003         GO TO 195
000251  003     190 GV = G(LG)
000252  003     195 GSUM = GSUM+GV
000253  003         QSUM = QSUM+GV*T2
000254  003   C     CHECK FOR LAST CONDUCTOR
000255  003         IF(NSQ1(JJ1).GT.0) GO TO 185
000256  003   C     CALCULATE THE NEW TEMPERATURE
000257  003         T2 = AN*QSUM/GSUM+AA*T1
000258  003         T1 = ABS(T2-T1)
000259  003         T(I) = T2-460.0
000260  003         IF(RLXA.GE.T1) GO TO 215
000261  003         RLXA = T1
000262  003         KK2 = I
000263  003     215 CONTINUE
000264  003         DO 235 I = 1,NNT
000265  003     235 T(I) = T(I)+460.0
000266  003   C     SEE IF THE ARITHMETIC RELAXATION CRITERIA WAS MET
000267  003         IF(RLXA.GT.CON(19)) GO TO 225
000268  003   C     SEE IF THE DIFFUSION RELAXATION CRITERIA WAS MET
000269  003     220 IF(RLXD.LE.CON(26)) GO TO 245
000270  003     225 IF(KON(7).EQ.0) GO TO 240
000271  003         CALL OUTCAL
000272  003     240 CONTINUE
000273  003         IF(KON(28).GE.65) CALL TOPLIN
000274  003         WRITE(6,882)
000275  003         KON(28) = KON(28)+2
000276  003   C     SEE IF THE TEMPERATURE CHANGES WERE TOO LARGE
000277  003     245 TCGD = 0.0
000278  003         TCGA = 0.0
000279  003         DO 250 I = 1,NNO
000280  003         LE = IGI+I
000281  003         C(I) = C(I)*TSTEPN
000282  003         T1 = ABS(T(I)-X(LE))
000283  003         IF(TCGD.GT.T1) GO TO 250
```

```
000284  003          TCGD = T1
000285  003          KON(36) = I
000286  003      250 CONTINUE
000287  003          IF(TCGD.LE.CON(6)) GO TO 265
000288  003          TSTEPN = 0.95*TSTEPN*CON(6)/TCGD
000289  003      255 DO 260 I = 1,NNT
000290  003          LE = IE1+I
000291  003      260 T(I) = X(LE)-460.0
000292  003          GO TO 30
000293  003      265 IF(NNA.LE.0) GO TO 275
000294  003          DO 270 I = NN,NNC
000295  003          LE = IE1+I
000296  003          T1 = ABS(T(I)-X(LE))
000297  003          IF(TCGA.GT.T1) GO TO 270
000298  003          TCGA = T1
000299  003          KON(38) = I
000300  003      270 CONTINUE
000301  003          IF(TCGA.LE.CON(11)) GO TO 275
000302  003          TSTEPN = 0.95*TSTEPN*CON(11)/TCGA
000303  003          GO TO 255
000304  003    C      CONVERT TEMPERATURES BACK TO FARENHEIT
000305  003      275 DO 280 I = 1,NNT
000306  003      280 T(I) = T(I)-460.0
000307  003    C     STORE THE TEMPERATURE AND RELAXATION CHANGES
000308  003          CON(15) = TCGD
000309  003          CON(16) = TCGA
000310  003          CON(27) = RLXD
000311  003          IF(RLXA.GT.RLXD) GO TO 285
000312  003          KK2 = KK1
000313  003          RLXA = RLXD
000314  003      285 KON(37) = KK2
000315  003          CON(30) = RLXA
000316  003          KON(12) = 0
000317  003          CALL VARBL2
000318  003    C     CHECK THE BACKUP SWITCH
000319  003          IF(KON(12).NE.0) GO TO 255
000320  003    C     ADVANCE TIME
000321  003          CON(13) = CON(1)
000322  003          TSUM = TSUM+TSTEPN
000323  003    C     CHECK FOR TIME TO PRINT
000324  003          IF(TSUM.GE.CON(18)) GO TO 290
000325  003    C     CHECK FOR PRINT EVERY ITERATION
000326  003          IF(KON(7).NE.0) CALL OUTCAL
000327  003          GO TO 10
000328  003    C     TRY TO EVEN THE OUTPUT INTERVALS
000329  003      290 TPRINT = TPRINT+TSUM
000330  003          CALL OUTCAL
000331  003    C     IS TIME GREATER THAN END COMPUTE TIME
000332  003          IF(CON(1)*1.000001.LT.CON(3)) GO TO 5
000333  003          NTH = IE1
000334  003          NDIM = NLA
000335  003          RETURN
000336  003      990 WRITE(6,880)
000337  003          GO TO 1000
000338  003      991 WRITE(6,881)
000339  003          GO TO 1000
000340  003      994 WRITE(6,884) NDIM
```

```
000341      003          GO TO 1000
000342      003      995 WRITE(6,885)
000343      003          GO TO 1000
000344      003      996 WRITE(6,886)
000345      003          GO TO 1000
000346      003      997 WRITE(6,887)
000347      003          GO TO 1000
000348      003      998 WRITE(6,888)
000349      003          GO TO 1000
000350      003      999 WRITE(6,889)
000351      003     1000 CALL OUTCAL
000352      003          CALL EXIT
000353      003      880 FORMAT(29H TRANSIENT TIME NOT SPECIFIED)
000354      003      881 FORMAT(45H CNBACK REQUIRES LONG PSEUDO-COMPUTE SEQUENCE)
000355      003      882 FORMAT(29H RELAXIATION CRITERIA NOT MET)
000356      003      884 FORMAT(I6,20H LOCATIONS AVAILABLE)
000357      003      885 FORMAT(10H NO DRLXCA)
000358      003      886 FORMAT(10H NO DTIMEI)
000359      003      887 FORMAT(10H NO ARLXCA)
000360      003      888 FORMAT(19H NO OUTPUT INTERVAL)
000361      003      889 FORMAT(9H NO NLOOP)
000362      003          END

END ELT.


*HDG,P CNFAST
```

```
@ELT,L CNFAST
ELTDT7 RL1870 02/28-03:18:58-(5,)
000001      001          SUBROUTINE CNFAST
000002      001    C     AN EXPLICIT EXECUTION SUBROUTINE FOR SINDA   FORTRAN V
000003      001    C     THE SHORT PSEUDO COMPUTE SEQUENCE IS REQUIRED
000004      001    C     NODES WITH CSG BELOW DTIMEI RECEIVE STEADY STATE SOLUTION
000005      001    C     NO BACKING UP IS DONE OR ALLOWED
000006      001          COMMON /FDIMNS/ NTYP, NSYS
000007      001          INCLUDE COMM,LIST
000008      001          INCLUDE DEFF,LIST
000009      001          IF(KON(5).LE.0) KON(5) = 1
000010      001          IF(CON(8).LE.0.) CON(8) = 1.E+8
000011      001          IF(CON(9).LE.0.) CON(9) = 1.0
000012      001          IF(CON(18).LE.0.) GO TO 999
000013      001          IF(CON(19).LE.0.) CON(19) = 1.E+8
000014      001          IF(CON(21).LE.0.) GO TO 998
000015      005           IF(CON(50) .LE. 0) CON(50)= 1.
000016      001          IF(KON(31).NE.0) GO TO 995
000017      001          PASS = -1.0
000018      001          NNC = NNA+NND
000019      001          IE = NTH
000020      001          NLA = NDIM
000021      001          NTH = NTH+NND
000022      001          NDIM = NDIM-NND
000023      001          IF(NDIM.LT.0) GO TO 997
000024      001          NN = NND+1
000025      001          TPRINT = CON(13)
000026      001          TSTEP = CON(21)
000027      001        5 TSUM = 0.0
000028      001          IF(CON(13)+CON(18).GT.CON(3)) CON(18) = CON(3)-CON(13)
000029      001       10 IF(TSTEP.GT.CON(8)) TSTEP = CON(8)
000030      001          IF(TSTEP.LT.CON(21)) TSTEP = CON(21)*1.000001
000031      001          IF(TSUM+TSTEP-CON(18)) 20,25,15
000032      001       15 TSTEP = CON(18)-TSUM
000033      001          GO TO 25
000034      001       20 IF(TSUM+2.0*TSTEP.GT.CON(18)) TSTEP = 0.5*(CON(18)-TSUM)
000035      001       25 CON(2) = TSTEP
000036      001          CON(1) = TPRINT+TSUM+TSTEP
000037      001          CON(14) = 0.5*(CON(1)+CON(13))
000038      001          DO 30 I = 1,NND
000039      001          Q(I) = 0.0
000040      001          LE = IE+I
000041      001       30 X(LE) = 0.0
000042      001          IF(NNA.LE.0) GO TO 40
000043      001          DO 35 I = NN,NNC
000044      001          Q(I) = 0.0
000045      001       35 CONTINUE
000046      001       40 KON(12) = 0
000047      001          CALL VARBL1
000048      001          IF(KON(12).NE.0) GO TO 10
000049      001          IF(PASS.GT.0.) GO TO 45
000050      001          PASS = 1.0
000051      001          CON(1) = TPRINT
000052      001          CON(2) = 0.0
000053      001          CALL OUTCAL
000054      001          CON(1) = TPRINT+TSTEP
000055      001          CON(2) = TSTEP
```

```
000056  001     45 J1 = 0
000057  001        J2 = 1
000058  001        DO 85 I = 1,NND
000059  001        LE = IE+I
000060  001        INCLUDE VARC,LIST
000061  001        INCLUDE VARQ,LIST
000062  001     50 J1 = J1+1
000063  001        LG = FLD(5,16,NSQ1(J1))
000064  001        IF(LG.EQ.0) GO TO 85
000065  001        LTA = FLD(22,14,NSQ1(J1))
000066  001        INCLUDE VARG,LIST
000067  001        IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 55
000068  001        T1 = T(I)+460.0
000069  001        T2 = T(LTA)+460.0
000070  001        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000071  005         GV= GV *CON(50)
000072  001        GO TO 60
000073  001     55 GV = G(LG)
000074  001     60 QDOT = GV*(T(LTA)-T(I))
000075  001        Q(I) = Q(I) + QDOT
000076  001        X(LE) = X(LE)+GV
000077  001        IF(LTA.GT.NND.OR.FLD(21,1,NSQ1(J1)).EQ.1) GO TO 65
000078  001        LEA = IE+LTA
000079  001        X(LEA) = X(LEA)+GV
000080  001        Q(LTA) = Q(LTA) - QDOT
000081  001     65 IF(NSQ1(J1).GT.0) GO TO 50
000082  001     85 CONTINUE
000083  001        CKM = 1.E+8
000084  001        TCGM = 0.0
000085  001        DO 105 I = 1,NND
000086  001        LE = IE+I
000087  004        IF(.NOT.X(LE).GT.0.0)  GO TO 90
000088  001        T1 = C(I)/X(LE)
000089  001        IF(T1.GE.CKM) GO TO 90
000090  001        CKM = T1
000091  001        KON(35) = I
000092  001     90 IF(TSTEP.GT.T1) GO TO 95
000093  001        T1 = T(I) + TSTEP*Q(I)/C(I)
000094  001        GO TO 100
000095  001     95 T1 = T(I) + T1*Q(I)/C(I)
000096  001    100 T2 = ABS(T1-T(I))
000097  001        T(I) = T1
000098  001        IF(T2.LT.TCGM) GO TO 105
000099  001        TCGM = T2
000100  001        KON(15) = I
000101  001    105 CONTINUE
000102  001        CON(15) = TCGM
000103  001        CON(17) = CKM
000104  001        KOP = CON(7)
000105  003        IF(NSYS .NE. 0) CALL FLUID(1,0,0,0.,KOP)
000106  001        IF(NNA.LE.0) GO TO 160
000107  001        LAX = KON(5)
000108  001        DAMPN = CON(9)
000109  001        DAMPO = 1.0-DAMPN
000110  001        DO 150 I = 1,LAX
000111  001        KON(20) = I
000112  001        RLX = 0.0
```

```
000113  001        JJ1 = J1
000114  001        JJ2 = J2
000115  001        DO 145 L = NN,NNC
000116  001        SUMC = 0.0
000117  001        SUMCV = 0.0
000118  001        IF(I.GT.1) GO TO 6000
000119  001        INCLUDE VRG2,LIST
000120  001    110 JJ1 = JJ1+1
000121  001        LG = FLD(5,16,NSQ1(JJ1))
000122  001        LTA = FLD(22,14,NSQ1(JJ1))
000123  001        IF(I.GT.1) GO TO 4000
000124  001        INCLUDE VRG2,LIST
000125  001  C     CHECK FOR RADIATION CONDUCTOR
000126  001        IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 115
000127  001        T1 = T(L)+460.0
000128  001        T2 = T(LTA)+460.0
000129  001        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000130  005         GV= GV *CON(50)
000131  001        GO TO 120
000132  001    115 GV = G(LG)
000133  001        T2 = T(LTA)
000134  001    120 SUMC = SUMC+GV
000135  001        SUMCV = SUMCV+GV*T2
000136  001  C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000137  001        IF(NSQ1(JJ1).GT.0) GO TO 110
000138  001        T1 = DAMPN*(SUMCV+Q(L))/SUMC+DAMPO*T(L)
000139  001        T2 = ABS(T(L)-T1)
000140  001        IF(RLX.GE.T2) GO TO 140
000141  001        RLX = T2
000142  001        KON(37) = L
000143  001    140 T(L) = T1
000144  001    145 CONTINUE
000145  001        IF(RLX.LE.CON(19)) GO TO 155
000146  001    150 CONTINUE
000147  001    155 CON(30) = RLX
000148  001    160 CALL VARBL2
000149  001        CON(13) = CON(1)
000150  001        TSUM = TSUM+TSTEP
000151  001        TSTEP = CKM
000152  001        IF(TSUM.LT.CON(18)) GO TO 10
000153  001        TPRINT = TPRINT+TSUM
000154  001        CALL OUTCAL
000155  001        IF(CON(1)*1.000001.LT.CON(3)) GO TO 5
000156  001        NTH = IE
000157  001        NDIM = NLA
000158  001        RETURN
000159  001    995 WRITE(6,885)
000160  001        GO TO 1000
000161  001    997 WRITE(6,887) NDIM
000162  001        GO TO 1000
000163  001    998 WRITE (6,888)
000164  001        GO TO 1000
000165  001    999 WRITE(6,889)
000166  001   1000 CALL OUTCAL
000167  001        CALL EXIT
000168  001    885 FORMAT(46H CNFAST REQUIRES SHORT PSEUDO-COMPUTE SEQUENCE)
000169  001    887  FORMAT(I8,20H LOCATIONS AVAILABLE)
```

```
000170      001      888 FORMAT(10H NO DTIMEL)
000171      001      889 FORMAT(19H NO OUTPUT INTERVAL)
000172      001          END

END ELT.


@HOG,P CNFRWD
```

```
CNFRWD                                                          DATE 022875        PAGE     1

*ELT,L CNFRWD
ELTOTT RLIB70 02/28-03:19:00-(5,)
000001      000          SUBROUTINE CNFRWD
000002      000    C     EXPLICIT FORWARD DIFFERENCING EXECUTION SUBROUTINE FOR SINDA F-V
000003      000    C     THE SHORT PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000004      001          COMMON /FDIMNS/ NTYP, NSYS
000005      000          INCLUDE COMM,LIST
000006      000          INCLUDE DEFF,LIST
000007      000          IF(CON(4).LT.1.0) CON(4) = 1.0
000008      000          IF(KON(5).LE.0) KON(5) = 1
000009      000          IF(CON(6).LE.0.) CON(6) = 1.E+8
000010      000          IF(CON(8).LE.0.) CON(8) = 1.E+8
000011      000          IF(CON(9).LE.0.) CON(9) = 1.0
000012      000          IF(CON(11).LE.0.) CON(11) = 1.E+8
000013      000          IF(CON(18).LE.0.) GO TO 999
000014      000          IF(CON(19).LE.0.) CON(19) = 1.E+8
000015      005           IF(CON(50) .LE. 0) CON(50)= 1.
000016      000          IF(KON(31).NE.0) GO TO 995
000017      000          PASS = -1.0
000018      000          NNC = NND+NNA
000019      000          IE = NTH
000020      000          NLA = NDIM
000021      004          NTH = NTH+NNT
000022      004          NDIM = NDIM-NNT
000023      000    C     CHECK FOR EXTRA LOCATIONS FOR CALCULATED NODES
000024      000          I = NLA-NNC
000025      000          IF(I.LT.0) GO TO 998
000026      000          L1 = NND+1
000027      000          TSTEP = CON(18)
000028      000          TPRINT = CON(13)
000029      000    C     INITALIZE TIME SUM BETWEEN OUTPUT INTERVALS
000030      000        5 TSUM = 0.0
000031      000    C     DOES OLD TIME PLUS THE OUTPUT INTERVAL EXCEED THE STOP TIME
000032      000          IF(CON(13)+CON(18).LE.CON(3)) GO TO 10
000033      000    C     DONT EXCEED IT
000034      000          CON(18) = CON(3)-CON(13)
000035      000    C     IS THE TIME STEP LARGER THAN ALLOWED
000036      000       10 IF(TSTEP.LE.CON(8)) GO TO 15
000037      000          TSTEP = CON(8)
000038      000    C     DOES THE TIME SUM PLUS THE TIME STEP EXCEED OUTPUT INTERVAL
000039      000       15 IF(TSUM+TSTEP-CON(18)) 25,30,20
000040      000    C     DONT EXCEED IT
000041      000       20 TSTEP = CON(18)-TSUM
000042      000          GO TO 30
000043      000    C     DOES TIME SUM PLUS TWO TIME STEPS EXCEED OUTPUT INTERVAL
000044      000       25 IF(TSUM+2.0*TSTEP.LE.CON(18)) GO TO 30
000045      000    C     APPROACH THE OUTPUT INTERVAL GRADUALLY
000046      000          TSTEP = (CON(18)-TSUM)/2.0
000047      000    C     STORE DELTA TIME STEP IN THE CONSTANTS
000048      000       30 CON(2) = TSTEP
000049      000    C     IS THE TIME STEP USED LESS THAN THE TIME STEP ALLOWED
000050      000          IF(TSTEP.LT.CON(21)) GO TO 997
000051      000    C     CALCULATE THE NEW TIME
000052      000          CON(1) = TPRINT+TSUM+TSTEP
000053      000    C     COMPUTE THE MEAN TIME BETWEEN ITERATIONS
000054      000          CON(14) = (CON(1)+CON(13))/2.0
000055      000    C     ZERO OUT ALL SOURCE LOCATIONS AND EXTRA LOCATIONS
```

```
000056  000        DO 35 I = 1,NND
000057  000        LE = IE+I
000058  000        X(LE) = 0.0
000059  000        Q(I) = 0.0
000060  000     35 CONTINUE
000061  000  C     SHIFT THE ARITHMETIC TEMPERATURES INTO THE EXTRA LOCATIONS
000062  004        IF(NND.EQ.NNT) GO TO 45
000063  004        DO 31 I=L1,NNT
000064  004        X(IE+I) = T(I)
000065  004     31 CONTINUE
000066  000        IF(NNA.LE.0) GO TO 45
000067  000        DO 40 I = L1,NNC
000068  000        Q(I) = 0.0
000069  000     40 CONTINUE
000070  000     45 KON(12) = 0
000071  000        CALL VARBLI
000072  000        IF(KON(12).NE.0) GO TO 10
000073  000        J1 = 0
000074  000        J2 = 1
000075  000        TCGM = 0.0
000076  000        CKM = 1.E+8
000077  000  C     CALCULATE Q SUM AND G SUM
000078  000        DO 85 I = 1,NND
000079  000        LE = IE+I
000080  000        INCLUDE VARC,LIST
000081  000        INCLUDE VARD,LIST
000082  000     50 J1 = J1+1
000083  000        LG = FLD(5,16,NSO1(J1))
000084  000  C     CHECK FOR LAST CONDUCTOR
000085  000        IF(LG.EQ.0) GO TO 85
000086  000        LTA = FLD(22,14,NSO1(J1))
000087  000        INCLUDE VARG,LIST
000088  000  C     CHECK FOR RADIATION CONDUCTOR
000089  000        IF(FLD(3,1,NSO1(J1)).EQ.0) GO TO 55
000090  000        T1 = T(I)+460.0
000091  000        T2 = T(LTA)+460.0
000092  000        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000093  005         GV= GV *CON(50)
000094  000        GO TO 60
000095  000     55 GV = G(LG)
000096  000  C     OBTAIN THE Q RATE THRU THE CONDUCTOR
000097  000     60 QDOT = GV*(X(LTA)-T(I))
000098  000        Q(I) = Q(I)+QDOT
000099  000  C     SAVE SUMMATION OF CONDUCTORS
000100  000        X(LE) = X(LE)+GV
000101  000  C     CHECK FOR ADJOINING DIFFUSION NODE
000102  000        IF(LTA.GT.NND.OR.FLD(21,1,NSO1(J1)).EQ.1) GO TO 65
000103  000  C     SAVE SUMMATION OF CONDUCTORS FOR ADJOINING NODE
000104  000        LEA = IE+LTA
000105  000        X(LEA) = X(LEA)+GV
000106  000        Q(LTA) = Q(LTA)-QDOT
000107  000  C     CHECK FOR LAST CONDUCTOR
000108  000     65 IF(NSO1(J1).GT.0) GO TO 50
000109  000     85 CONTINUE
000110  000  C     OBTAIN NEW DIFFUSION TEMPERATURES, DTMPCC AND CSGMIN
000111  000        DO 100 I = 1,NND
000112  000        LE = IE+I
```

```
000113  000  C     CALCULATE C/SK MINIMUM
000114  004        IF(.NOT.X(LE).GT.0.0) GO TO 90
000115  000        T1 = C(I)/X(LE)
000116  000        IF(T1.GE.CKM) GO TO 90
000117  000        CKM = T1
000118  000        KON(35) = I
000119  000  C     COMPUTE NEW TEMPERATURES USING CALCULATED SOURCE TERMS
000120  000     90 T1 = TSTEP*Q(I)/C(I)
000121  000  C     CALCULATE THE ABSOLUTE VALUE TEMPERATURE CHANGE
000122  000        T2 = ABS(T1)
000123  000  C     SAVE THE LARGEST TEMPERATURE CHANGE
000124  000        IF(TCGM.GE.T2) GO TO 95
000125  000        TCGM = T2
000126  000        KON(36) = I
000127  000  C     STORE THE TEMPERATURES
000128  000     95 X(LE) = T(I)
000129  000        T(I) = T(I)+T1
000130  000    100 CONTINUE
000131  000        CON(17) = CKM
000132  000        DELTA = CKM/CON(4)
000133  001        KOP = CON(7)
000134  003        IF(NSYS .NE. 0) CALL FLUID(1,0,0,0.,KOP)
000135  000  C     CHECK FOR FIRST PASS
000136  000        IF(PASS.GT.0.0) GO TO 115
000137  000  C     UNDO THE TEMPERATURE CALCULATIONS
000138  004    105 DO 110 I=1,NNT
000139  000        LE = IF+I
000140  000        T(I) = X(LE)
000141  000    110 CONTINUE
000142  000        IF(PASS.GT.0.0) GO TO 15
000143  000        PASS = 1.0
000144  000        CON(1) = TPRINT
000145  000        CON(2) = 0.0
000146  000        TSTEP = DELTA*0.95
000147  000        GO TO 195
000148  000  C     IS THE TIME STEP USED LESS THAN THE TIME STEP CALCULATED
000149  000    115 IF(TSTEP.LE.DELTA) GO TO 130
000150  000  C     COMPUTE THE TIME STEP
000151  000        TSTEP = DELTA*0.95
000152  000        GO TO 105
000153  000    120 TSTEP = 0.95*TSTEP*CON(6)/TCGM
000154  000        GO TO 105
000155  000    125 TSTEP = 0.95*TSTEP*CON(11)/TCGM
000156  000        GO TO 105
000157  000  C     SEE IF THE TEMPERATURE CHANGE WAS TOO LARGE
000158  000    130 IF(TCGM.GT.CON(6)) GO TO 120
000159  000  C     STORE THE MAXIMUM DIFFUSION TEMPERATURE CHANGE
000160  000        CON(15) = TCGM
000161  000  C     CHECK TO SEE IF THERE ARE ANY ARITHMETIC NODES
000162  000        IF(NNA.LE.0) GO TO 185
000163  000  C     COMPUTE ARITHMETIC TEMPERATURES BY SUCCESSIVE POINT OVER-RELAX
000164  000        DN = CON(9)
000165  000        DB = 1.0-DN
000166  000        LAX = KON(5)
000167  000        DO 170 I = 1,LAX
000168  000        JJ1 = J1
000169  000        JJ2 = J2
```

```
000170  000          TCGM = 0.0
000171  000          KON(20) = I
000172  000          DO 165 L = L1,NNC
000173  000          SUMC = 0.0
000174  000          SUMCV = 0.0
000175  000          IF(I.GT.1) GO TO 6000
000176  000          INCLUDE VRG2,LIST
000177  000      135 JJ1 = JJ1+1
000178  000          LG = FLD(5,16,NSQ1(JJ1))
000179  000          LTA = FLD(22,14,NSQ1(JJ1))
000180  000          IF(I.GT.1) GO TO 4000
000181  000          INCLUDE VRG2,LIST
000182  000    C     CHECK FOR RADIATION CONDUCTOR
000183  000          IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 140
000184  000          T1 = T(L)+460.0
000185  000          T2 = T(LTA)+460.0
000186  000          GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000187  005           GV= GV *CON(50)
000188  000          GO TO 145
000189  000      140 GV = G(LG)
000190  000      145 SUMC = SUMC+GV
000191  000          SUMCV = SUMCV+GV*T(LTA)
000192  000    C     CHECK FOR LAST CONDUCTOR
000193  000          IF(NSQ1(JJ1).GT.0) GO TO 135
000194  000          T2 = DD+T(L)+DN*(SUMCV+Q(L))/SUMC
000195  000    C     OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000196  000          T1 = ABS(T(L)-T2)
000197  000    C     STORE THE NEW TEMPERATURE
000198  000          T(L) = T2
000199  000    C     SAVE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000200  000          IF(TCGM.GE.T1) GO TO 165
000201  000          TCGM = T1
000202  000          KON(37) = L
000203  000      165 CONTINUE
000204  000    C     SEE IF RELAXATION CRITERIA WAS MET
000205  000          IF(TCGM.LE.CON(19)) GO TO 175
000206  000      170 CONTINUE
000207  000    C     STORE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000208  000      175 CON(30) = TCGM
000209  000    C     COMPUTE THE ARITHMETIC TEMPERATURE CHANGE
000210  000          TCGM = 0.0
000211  000          DO 180 I = L1,NNC
000212  000          LE = 1E+I
000213  000          T1 = ABS(T(I)-X(LE))
000214  000          IF(T1.LT.TCGM) GO TO 180
000215  000          TCGM = T1
000216  000          KON(38) = I
000217  000      180 CONTINUE
000218  000    C     SEE IF ATMPCA WAS SATISFIED
000219  000          IF(TCGM.GT.CON(11)) GO TO 125
000220  000          CON(16) = TCGM
000221  000      185 KON(12) = 0
000222  000          CALL VARBL2
000223  000    C     CHECK THE BACKUP SWITCH
000224  000          IF(KON(12).NE.0) GO TO 105
000225  000    C     ADVANCE TIME
000226  000          CON(13) = CON(1)
```

```
000227  000          TSUM = TSUM+TSTEP
000228  000          TSTEP = DELTA*0.95
000229  000    C     CHECK FOR TIME TO PRINT
000230  000          IF(TSUM.GE.CON(18)) GO TO 190
000231  000    C     CHECK FOR PRINT EVERY ITERATION
000232  000          IF(KON(7).EQ.0) GO TO 10
000233  000          CALL OUTCAL
000234  000          GO TO 10
000235  000    C     TRY TO EVEN THE OUTPUT INTERVALS
000236  000      190 TPRINT = TPRINT+TSUM
000237  000      195 CALL OUTCAL
000238  000    C     IS TIME GREATER THAN END COMPUTE TIME
000239  000          IF(CON(1)*1.000001.LT.CON(3)) GO TO 5
000240  000          NTH = IE
000241  000          NOIM = NLA
000242  000          RETURN
000243  000      995 WRITE(6,885)
000244  000          GO TO 1000
000245  000      997 WRITE(6,887)
000246  000          GO TO 1000
000247  000      998 WRITE(6,888) I
000248  000          GO TO 1000
000249  000      999 WRITE(6,889)
000250  000     1000 CALL OUTCAL
000251  000          CALL EXIT
000252  000      885 FORMAT(46H CNFRWD REQUIRES SHORT PSEUDO-COMPUTE SEQUENCE)
000253  000      887 FORMAT(20H TIME STEP TOO SMALL)
000254  000      888 FORMAT(I8,20H LOCATIONS AVAILABLE)
000255  000      889 FORMAT(19H NO OUTPUT INTERVAL)
000256  000          END

END ELT.

*HDG,P CNFWBK
```

```
@ELT,L CNFWBK
ELT077 RL1870 02/28-03:19:02-(8,)
000001      000          SUBROUTINE CNFWBK
000002      000    C     IMPLICIT FORWARD-BACKWARD DIFFERENCING EXECUTION SUBROUTINE
000003      000    C     THE LONG PSEUDO-COMPUTE SEQUENCE IS REQUIRED, SINDA  FORTRAN V
000004      000    C     ALL NODES RECEIVE A SUCCESSIVE POINT ITERATION
000005      000    C     RELAXATION CRITERIA MUST BE SPECIFIED
000006      000    C     OVER-RELAXATION IS ALLOWED, THE DAMPENING FACTORS ARE ADDRESSABLE
000007      001          LOGICAL FLOW
000008      001          COMMON /FDIMNS/ NTYPE, NSYS
000009      000          INCLUDE COMM,LIST
000010      000          INCLUDE DEFF,LIST
000011      000          IF(KON(5).LE.0) GO TO 999
000012      000          IF(CON(6).LE.0.) CON(6) = 1.E+8
000013      000          IF(CON(8).LE.0.) CON(8) = 1.E+8
000014      000          IF(CON(9).LE.0.) CON(9) = 1.0
000015      000          IF(CON(10).LE.0.) CON(10) = 1.0
000016      000          IF(CON(11).LE.0.) CON(11) = 1.E+8
000017      000          IF(CON(3).LE.CON(13)) GO TO 990
000018      000          IF(CON(18).LE.0.) GO TO 998
000019      000          IF(NNA.GT.0.AND.CON(19).LE.0.) GO TO 997
000020      000          IF(CON(22).LE.0.) GO TO 996
000021      000          IF(NND.GT.0.AND.CON(26).LE.0.) GO TO 995
000022      000          IF(KON(31).NE.1) GO TO 991
000023      006           IF(CON(50) .LE. 0) CON(50)= 1.
000024      001          TZERO = -460.
000025      000          PASS = -1.0
000026      000          NN = NND+1
000027      000          NLA = NDIM
000028      001          NNC = NND + NNA
000029      001          NSP = NND
000030      001          IF (NSYS .NE. 0) NSP = NNT
000031      001          IE1 = NTH
000032      001          IE2 = IE1 + NNT
000033      001          IE3 = IE2 + NSP
000034      001          J = NND + NSP + NNT
000035      000          NTH = NTH+J
000036      000          NDIM =  NDIM-J
000037      000    C     CHECK FOR EXTRA LOCATIONS FOR CALCULATED NODES
000038      000          IF(NDIM.LT.0) GO TO 994
000039      001          FLOW = .FALSE.
000040      001          IF (NSYS .EQ. 0) GO TO 4
000041      001          FLOW = .TRUE.
000042      001          DO 3 I=1,NNC
000043      001          NX(IE2+I) = 0
000044      001        3 CONTINUE
000045      001        4 TPRINT = CON(13)
000046      000    C     INITALIZE TIME SUM BETWEEN OUTPUT INTERVALS
000047      000        5 TSUM = 0.0
000048      000    C     DOES OLD TIME PLUS THE OUTPUT INTERVAL EXCEED THE STOP TIME
000049      000          IF(CON(13)+CON(18).GT.CON(3)) CON(18) = CON(3)-CON(13)
000050      000    C     DONT EXCEED IT
000051      000       10 TSTEPN = CON(22)
000052      000          IF(TSTEPN.LE.CON(8)) GO TO 20
000053      000       15 TSTEPN = CON(8)
000054      000          GO TO 35
000055      000    C     DOES THE TIME SUM PLUS THE TIME STEP EXCEED OUTPUT INTERVAL
```

```
000056  000      20 IF(TSUM+TSTEPN-CON(18)) 30,35,25
000057  000  C      DONT EXCEED IT
000058  000      25 TSTEPN = CON(18)-TSUM
000059  000         GO TO 35
000060  000  C      DOES TIME SUM PLUS TWO TIME STEPS EXCEED OUTPUT INTERVAL
000061  000      30 IF(TSUM+2.0*TSTEPN.LE.CON(18)) GO TO 35
000062  000  C      APPROACH THE OUTPUT INTERVAL GRADUALLY
000063  000         TSTEPN = (CON(18)-TSUM)/2.0
000064  000  C      STORE DELTA TIME STEP IN THE CONSTANTS
000065  000      35 CON(2) = TSTEPN
000066  000  C      CALCULATE THE NEW TIME
000067  000         IF(PASS.GT.0.) GO TO 40
000068  000         CON(1) = TPRINT
000069  000         CON(2) = 0.0
000070  000         GO TO 45
000071  000      40 CON(1) = TPRINT+TSUM+TSTEPN
000072  000  C      COMPUTE THE MEAN TIME BETWEEN ITERATIONS
000073  000      45 CON(19) = (CON(1)+CON(13))/2.0
000074  000         LAX = KON(5)
000075  000         DN = CON(10)
000076  000         DD = 1.0-DN
000077  000         AN = CON(9)
000078  000         AA = 1.0-AN
000079  000         TSTEP = TSTEPN/2.0
000080  000  C      DO THE RELAXATION LOOP
000081  000         DO 240 K1 = 1,LAX
000082  000         KON(20) = K1
000083  000         J1 = 0
000084  000         RLXA = 0.0
000085  000         RLXD = 0.0
000086  001         KOP = CON(7)
000087  001         IF (K1 .GT. 1) GO TO 106
000088  000         J2 = 1
000089  000  C      ZERO OUT ALL SOURCE LOCATIONS AND SHIFT TEMPERATURES
000090  000         DO 50 I = 1,NNC
000091  000      50 Q(I) = 0.0
000092  000         DO 55 I = 1,NNT
000093  000         IE1 = IE1+I
000094  000      55 X(IE1) = T(I)
000095  004         IF(FLOW) CALL FLUID(2,IF1,IE2,0.,KOP)
000096  000         KON(12) = 0
000097  000         CALL VARB()
000098  000  C      CHECK THE GROUP SWITCH
000099  000         IF(KON(12).NE.0) GO TO 15
000100  000  C      CHECK FOR FIRST PASS
000101  000         IF(PASS.GE.0.) GO TO 60
000102  000         CALL OUTCAL
000103  000         PASS = 1.0
000104  000         GO TO 10
000105  000      60 RC = 1.E+8
000106  000         JJ = 0
000107  000  C      CALCULATE FIRST PASS TEMPERATURES AND CSGMIN
000108  000         DO 105 I = 1,NND
000109  000         INCLUDE VARC,LIST
000110  000  C      FOLD DELTAT INTO THE CAPACITANCES
000111  000         C(I) = C(I)/TSTEP
000112  000         R1 = 0.0
```

```
000113  000          S = 0.0
000114  000          G2 = 0.0
000115  000          INCLUDE VARG,LIST
000116  000          Q(I) = 2.0*Q(I)+C(I)*(T(I)+460.0)
000117  000          QSUM = Q(I)
000118  000          GSUM = C(I)
000119  001          IF ( .NOT. FLOW) GO TO 70
000120  001          LMP = NX(IE2+I)
000121  001          IF (LMP .EQ. 0) GO TO 70
000122  005          HA = X(IE2+LMP)
000123  005          Q(I) = Q(I)+HA*(T(LMP)-T(I))
000124  005          QSUM = QSUM+HA*(T(LMP)-TZERO)
000125  005          GSUM = GSUM+HA
000126  000       70 J1 = J1+1
000127  000          LG = FLD(5,16,NSQ1(J1))
000128  001          IF (LG  EQ. 0) GO TO 80
000129  000          LTA = FLD(22,14,NSQ1(J1))
000130  000          LTAE = LTA+IE1
000131  000          INCLUDE VARG,LIST
000132  000          T1 = T(I)+460.0
000133  000          T2 = T(LTA)+460.0
000134  000    C     CHECK FOR RADIATION CONDUCTOR
000135  000          IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 75
000136  006           R1= R1 +G(LG)*CON(50)
000137  006           QSUM= QSUM + G(LG)*CON(50)*T2**4
000138  006           G2= G2 + G(LG)*(T1*T1+T2*T2)*(T1+T2)*CON(50)
000139  006           Q(I)= Q(I)+ G(LG)*CON(50)*((X(LTAE)+460.)**4-T1**4)
000140  000          GO TO 80
000141  000       75 GV = G(LG)
000142  000          Q(I) = Q(I)+GV*(X(LTAE)-T(I))
000143  000          G2 = G2+GV
000144  000          GSUM = GSUM+GV
000145  000          QSUM = QSUM+GV*T2
000146  000    C     CHECK FOR LAST CONDUCTOR
000147  000       80 IF(NSQ1(J1).GT.0) GO TO 70
000148  000    C     DAMPEN RADIATION ON THIS NODE IF PRESENT
000149  000          IF(R1.LE.0.) GO TO 100
000150  000          R2 = R1*T1**4
000151  000          T2 = (QSUM-R2)/GSUM
000152  000          R1 = R1*T2**4
000153  000          S = (R1+R2)/2.0
000154  000    C     OBTAIN THE NEW TEMPERATURE
000155  000      100 T(I) = (DN*((QSUM-S)/GSUM)+DD*T1)-460.0
000156  000          R1 = C(I)/G2
000157  000          IF(R1.GE.RC) GO TO 105
000158  001          IF ( .NOT. G2 .GT. 0.0) GO TO 105
000159  000          RC = R1
000160  000          KON(35) = I
000161  000      105 CONTINUE
000162  000    C     CONVERT TEMPERATURES TO RANKINE
000163  000          DO 65 I = 1,NNT
000164  000          LE1 = IE1+I
000165  000          T(I) = T(I)+460.
000166  000       65 X(LE1) = X(LE1)+460.
000167  000          CON(17) = RC*TSTEP
000168  000          GO TO 225
000169  000    C     NOW RELAX THE NETWORK BY SUCCESSIVE POINT AND EXTRAPOLATION
```

```
000170  004     106 IF(FLOW) CALL FLUID(2,IE1,IE2,TZERO,KOP)
000171  000     110 JJ = JJ+1
000172  000         DO 165 I = 1,NND
000173  000         R1 = 0.0
000174  000         S = 0.0
000175  000         QSUM = Q(I)
000176  000         GSUM = C(I)
000177  001         IF ( .NOT. FLOW) GO TO 115
000178  001         LMP = NX(IE2+I)
000179  001         IF (LMP .EQ. 0) GO TO 115
000180  001         QSUM = QSUM + X(IE2+LMP)*T(LMP)
000181  001         GSUM = GSUM + X(IE2+LMP)
000182  000     115 J1 = J1+1
000183  000         LG = FLD(5,16,NSQ1(J1))
000184  001         IF (LG .EQ. 0) GO TO 125
000185  000         LTA = FLD(22,14,NSQ1(J1))
000186  000     C   CHECK FOR RADIATION CONDUCTOR
000187  000         IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 120
000188  006          R1= R1+G(LG)*CON(50)
000189  006          QSUM= QSUM +G(LG)*T(LTA)**4 *CON(50)
000190  000         GO TO 125
000191  000     120 GSUM = GSUM+G(LG)
000192  000         QSUM = QSUM+G(LG)*T(LTA)
000193  000     C   CHECK FOR LAST CONDUCTOR
000194  000     125 IF(NSQ1(J1).GT.0) GO TO 115
000195  000     C   DAMPEN RADIATION ON THIS NODE IF PRESENT
000196  000         IF(R1.LE.0.) GO TO 145
000197  000         R2 = R1*T(I)**4
000198  000         T2 = (QSUM-R2)/GSUM
000199  000         R1 = R1*T2**4
000200  000         S = (R1+R2)/2.0
000201  000     C   OBTAIN THE NEW TEMPERATURE
000202  000     145 T2 = DN*((QSUM-S)/GSUM)+OD*T(I)
000203  000     C   OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000204  000         T1 = ABS(T(I)-T2)
000205  000     C   STORE THE NEW AND OLD TEMPERATURES
000206  000         GO TO (160,155,150), JJ
000207  000     150 LE2 = IE2+I
000208  000         LE3 = IE3+I
000209  000         R1 = T2-T(I)
000210  000         X(LE2) = T(I)
000211  000         X(LE3) = R1/(R1-X(LE3))
000212  000         GO TO 160
000213  000     155 LE3 = IE3+I
000214  000         X(LE3) = T2-T(I)
000215  000     160 T(I) = T2
000216  000         IF(RLXD.GE.T1) GO TO 165
000217  000         RLXD = T1
000218  000         KK1 = I
000219  000     165 CONTINUE
000220  000         GO TO (180,180,170), JJ
000221  000     C   PERFORM LINEAR EXTRAPOLATION ON THE ERROR FUNCTION CURVE
000222  000     170 JJ = 0
000223  000         DO 175 I = 1,NND
000224  000         LE3 = IE3+I
000225  000         LE2 = IE2+I
000226  006          GV= GV *CON(50)
```

```
000227  000  C     SEE IF THE EXTRAPOLATION IS ALLOWABLE
000228  007        IF(X(LE3).GE.0.) GO TO 173
000229  000  C     LIMIT THE EXTRAPOLATION
000230  000        IF(X(LE3).LT.-10.) X(LE3) = -10.
000231  000        T(I) = X(LE3)*X(LE2)+(1.0-X(LE3))*T(I)
000232  007     173 NX(LE2) = 0
000233  000     175 CONTINUE
000234  000     180 IF(NNA.LE.0) GO TO 220
000235  000         JJ1 = J1
000236  000         JJ2 = J2
000237  000         DO 230 I = 1,NNT
000238  000     230 T(I) = T(I)-460.0
000239  000         DO 215 I = NN,NPS
000240  000         L = I
000241  000         IF(K1.GT.2) GO TO 6000
000242  000         INCLUDE VRG2,LIST
000243  000         GSUM = 0.0
000244  000         QSUM = Q(I)
000245  000     185 JJ1 = JJ1+1
000246  000         LG = FLD(5,16,NSQ1(JJ1))
000247  000         LTA = FLD(22,14,NSQ1(JJ1))
000248  000         IF(K1.GT.2) GO TO 4000
000249  000         INCLUDE VRG2,LIST
000250  000         T1 = T(I)+460.0
000251  000         T2 = T(LTA)+460.0
000252  000  C      CHECK FOR RADIATION CONDUCTOR
000253  000         IF(FLD(3,1,NSQ1(JJ1)).EQ.0) GO TO 190
000254  000         GV = G(LG)*(T1+T1+T2+T2)*(T1+T2)
000255  000         GO TO 195
000256  000     190 GV = G(LG)
000257  000     195 GSUM = GSUM+GV
000258  000         QSUM = QSUM+GV*T2
000259  000  C      CHECK FOR LAST CONDUCTOR
000260  000         IF(NSQ1(JJ1).GT.0) GO TO 185
000261  000  C      CALCULATE THE NEW TEMPERATURE
000262  000         T2 = AN*QSUM/GSUM+AA*T1
000263  000         T1 = ABS(T2-T1)
000264  000         T(I) = T2-460.0
000265  000         IF(RLXA.GE.T1) GO TO 215
000266  000         RLXA = T1
000267  000         KK2 = I
000268  000     215 CONTINUE
000269  000         DO 235 I = 1,NNT
000270  000     235 T(I) = T(I)+460.0
000271  000  C      SEE IF THE ARITHMETIC RELAXATION CRITERIA WAS MET
000272  000         IF(RLXA.GT.CON(19)) GO TO 225
000273  000  C      SEE IF THE DIFFUSION RELAXATION CRITERIA WAS MET
000274  000     220 IF(RLXB.LE.CON(26)) GO TO 245
000275  000     225 IF(KON(7).EQ.0) GO TO 240
000276  000         CALL OUTCAL
000277  000     240 CONTINUE
000278  000         IF(KON(28).GE.65) CALL TOPLIN
000279  000         WRITE(6,882)
000280  000         KON(28) = KON(28)+2
000281  000  C      SEE IF THE TEMPERATURE CHANGES WERE TOO LARGE
000282  000     245 TCGD = 0.0
000283  000         TCGA = 0.0
```

```
000284  000          DO 250 I = 1,NND
000285  000          LE = IE1+I
000286  000          C(I) = C(I)*TSTEP
000287  000          T1 = ABS(T(I)-X(LE))
000288  000          IF(TCGD.GT.T1) GO TO 250
000289  000          TCGD = T1
000290  000          KON(36) = I
000291  000      250 CONTINUE
000292  000          IF(TCGD.LE.CON(6)) GO TO 265
000293  000          TSTEPN = 0.95*TSTEPN*CON(6)/TCGD
000294  000      255 DO 260 I = 1,NNT
000295  000          LE = IE1+I
000296  000      260 T(I) = X(LE)-460.0
000297  000          GO TO 30
000298  000      265 IF(NNA.LE.0) GO TO 275
000299  000          DO 270 I = NN,NNC
000300  000          LE = IE1+I
000301  000          T1 = ABS(T(I)-X(LE))
000302  000          IF(TCGA.GT.T1) GO TO 270
000303  000          TCGA = T1
000304  000          KON(38) = I
000305  000      270 CONTINUE
000306  000          IF(TCGA.LE.CON(11)) GO TO 275
000307  000          TSTEPN = 0.95*TSTEPN*CON(11)/TCGA
000308  000          GO TO 255
000309  000  C       CONVERT TEMPERATURES BACK TO FAHRENHEIT
000310  000      275 DO 280 I = 1,NNT
000311  000      280 T(I) = T(I)-460.0
000312  000  C       STORE THE TEMPERATURE AND RELAXATION CHANGES
000313  000          CON(15) = TCGD
000314  000          CON(16) = TCGA
000315  000          CON(27) = RLXD
000316  000          IF(RLXA.GT.RLXD) GO TO 285
000317  000          KK2 = KK1
000318  000          RLXA = RLXD
000319  000      285 KON(37) = KK2
000320  000          CON(30) = RLXA
000321  000          KON(12) = 0
000322  000          CALL VARBL2
000323  000  C       CHECK THE BACKUP SWITCH
000324  000          IF(KON(12).NE.0) GO TO 255
000325  000  C       ADVANCE TIME
000326  000          CON(13) = CON(1)
000327  000          TSUM = TSUM+TSTEPN
000328  000  C       CHECK FOR TIME TO PRINT
000329  000          IF(TSUM.GE.CON(18)) GO TO 290
000330  000  C       CHECK FOR PRINT EVERY ITERATION
000331  000          IF(KON(7).NE.0) CALL OUTCAL
000332  000          GO TO 10
000333  000  C       TRY TO EVEN THE OUTPUT INTERVALS
000334  000      290 TPRINT = TPRINT+TSUM
000335  000          CALL OUTCAL
000336  000  C       IS TIME GREATER THAN END COMPUTE TIME
000337  000          IF(CON(11+1.000001.LT.CON(3)) GO TO 5
000338  000          NTH = IE1
000339  000          NDIM = NLA
000340  000          RETURN
```

```
000341  000      990 WRITE(6,880)
000342  000          GO TO 1000
000343  000      991 WRITE(6,881)
000344  000          GO TO 1000
000345  000      994 WRITE(6,884) NDIM
000346  000          GO TO 1000
000347  000      995 WRITE(6,885)
000348  000          GO TO 1000
000349  000      996 WRITE(6,886)
000350  000          GO TO 1000
000351  000      997 WRITE(6,887)
000352  000          GO TO 1000
000353  000      998 WRITE(6,888)
000354  000          GO TO 1000
000355  000      999 WRITE(6,889)
000356  000     1000 CALL OUTCAL
000357  000          CALL EXIT
000358  000      880 FORMAT(29H TRANSIENT TIME NOT SPECIFIED)
000359  000      881 FORMAT(45H CNFWBK REQUIRES LONG PSEUDO-COMPUTE SEQUENCE)
000360  000      882 FORMAT(28H RELAXATION CRITERIA NOT MET)
000361  000      884 FORMAT(I8,20H LOCATIONS AVAILABLE)
000362  000      885 FORMAT(10H NO DRLXCA)
000363  000      886 FORMAT(10H NO DTIME)
000364  000      887 FORMAT(10H NO ARLXCA)
000365  000      888 FORMAT(19H NO OUTPUT INTERVAL)
000366  000      889 FORMAT(9H NO NLOOP)
000367  000          END

END ELT.


*HOG,P CODERD/PB
```

```
@ELT,L CODERD/PB
ELTOT7 RL1870 02/28-03:19:05-(0,1)
000001       000       SUBROUTINE CODERD                                                  CDR   1
000002       000   C                                                                      CDR   2
000003       000   C       SUBROUTINE CODERD (CODE READ) READS THE TITLE BLOCK AND THE    CDR   3
000004       000   C       NEXT FOUR BLOCK HEADER CARDS.  IE. BCD 3NODE DATA,             CDR   4
000005       000   C       BCD 3CONDUCTOR DATA, BCD 3CONSTANTS DATA, AND                  CDR   5
000006       000   C       BCD 3ARRAY DATA.                                               CDR   6
000007       000   C                                                                      CDR   7
000008       000           COMMON/JPS/JPSTOT,FACTOR
000009       000           COMMON /BUCKET/ B(1)                                           CDR   8
000010       000           COMMON /LOGIC/ LNODE,LCOND,LCONST,LARRAY,LPRINT,KBRNCH         CDR   9
000011       000          1   ,IFIXC(50),KTPRNT,AYPRNT,GENERL,LQ,LONG2                   VERS-005
000012       000           COMMON /TAPE/  NIN, NOUT, INTERN, L830, L84P, LUT1,LUT2,LUT3,LUT4 CDR  11
000013       000           COMMON /DATA/ NND,NNA,NNB,NNT,NGL,NGR,NGT,NUC,NEC1,NEC2,NCT,LENA, CDR  12
000014       000          1     ERDATA,PROGRM,ENDRUN,LSEQ1,LSEQ2,LONG                    CDR  13
000015       000           COMMON /PLOGIC/  PARINT, PARFIN, PNODE, PCOND, PCONST, PARRAY, CDR  14
000016       000          1                 PTITLE, PCHGID                                CDR  15
000017       000           COMMON /POINT/ LOC(20),LEN(20),LENBKT,TITLE(20)                CDR  16
000018       000           INTEGER  ALPH, THERM, PCSLNG, CODE, END, REMARK, TEMPB, CONDB, CDR  17
000019       000          1         CONSTB, ARRYB, ENDDAT, PRINT, HINIT, GENRLP, BLANK, TITLE,CDR 18
000020       000          2         FINE, PCHGID                                          CDR  19
000021       000           INTEGER  COL1,COMMNT,PCSSHT,BLOCK                              CDR  20
000022       000           INTEGER QB,ENDPRM                                              CDR  21
000023       000           LOGICAL LNODE,LLOGIC,LEND,NOREAD                               CDR  22
000024       000           LOGICAL  LCOND, LCONST, LARRAY, LPRINT, GENERL, LONG           CDR  23
000025       000           LOGICAL   PARINT, PARFIN, PNODE, PCOND, PCONST, PARRAY, PTITLE CDR  24
000026       000           LOGICAL KTPRNT,AYPRNT,LQ,LONG2                                 VERS-005
000027       000           DIMENSION ALPH(14), IB(1), BLOCK(4), LLOGIC(1)                 CDR  26
000028       000           DIMENSION FIXC(1)                                              VERS-005
000029       000           EQUIVALENCE (IFIXC,FIXC)                                       VERS-005
000030       000           EQUIVALENCE (B,IB), (LLOGIC,LNODE)                             CDR  27
000031       000           DATA  END /6HEND   /,BLANK/6H      /                           CDR  28
000032       000           DATA  THERM/6HTHERMA/ ,  PCSLNG/6HL SPCS/                      CDR  29
000033       000           DATA  REMARK/6HREM   /, PRINT/6H *    /                        CDR  30
000034       000           DATA  TEMPB /6HNODE D/,CONDB/6HCONDUC/, CONSTB/6HCONSTA/,ARRYB CDR  31
000035       000          1      /6HARRAY /, GENRLP/6HGENERA/, HINIT/6HINITIA/, ENDDAT   CDR  32
000036       000          2      /6HEND OF/, FINE/6HFINAL /                              CDR  33
000037       000           DATA COMMNT /1HC/, PCSSHT /6HL SPCS/, ITWO /1H2/               CDR  34
000038       000           DATA (BLOCK(I),I=1,4) /6HNODE D, 6HCONDUC, 6HCONSTA, 6HARRAY / CDR  35
000039       000           DATA QB/6HSOURCE/                                              CDR  36
000040       000           DATA ENDPRM/6HEND PA/                                          CDR  37
000041       000           COMMON/SRDCOM/COL1,COL2Y,ALPH,CODE,N                                V 6
000042       000   C                                                                      CDR  38
000043       000   C       INITIALIZATION                                                 CDR  39
000044       000   C                                                                      CDR  40
000045       000           KFLOW=0                                                        CDR  41
000046       000           ILAST=0                                                        CDR  42
000047       000           PTITLE=.FALSE.                                                 CDR  43
000048       000           PNODE=.FALSE.                                                  CDR  44
000049       000           PCOND=.FALSE.                                                  CDR  45
000050       000           PCONST=.FALSE.                                                 CDR  46
000051       000           PARRAY=.FALSE.                                                 CDR  47
000052       000           LPRINT=.FALSE.                                                 CDR  48
000053       000           KTPRNT=.FALSE.                                                 CDR  49
000054       000           AYPRNT=.FALSE.                                                 CDR  50
000055       000           PARINT=.FALSE.
```

```
000056  000        PARFIN=.FALSE.                                          CDR  51
000057  000        LONG2=.FALSE.                                           VERS-005
000058  000        LQ=.FALSE.                                              CDR  52
000059  000  C                                                             CDR  53
000060  000  C        READ DCD 3THERMAL/GENERAL CARD                       CDR  54
000061  000  C                                                             CDR  55
000062  000     10 CONTINUE                                                CDR  56
000063  000        CALL SREADC(1)                                               V 6
000064  000        IF (COL1.NE.COMMNT) GO TO 20                            CDR  58
000065  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR  59
000066  000        GO TO 10                                                CDR  60
000067  000     20 CONTINUE                                                CDR  61
000068  000        IF (ALPH(3).EQ.ENDDAT) GO TO 520                        CDR  62
000069  000        IF (ALPH(3).EQ.ENDPRM) GO TO 10                         CDR  63
000070  000        WRITE (NOUT,620)                                        CDR  64
000071  000        WRITE (NOUT,660) ALPH                                   CDR  65
000072  000  C                                                             CDR  66
000073  000  C        DEBUG PRINT IF * IN COLUMN 80                        CDR  67
000074  000  C                                                             CDR  68
000075  000        IF (ALPH(19).EQ.PRINT) LPRINT=.TRUE.                    CDR  69
000076  000        IF (ALPH(3).NE.THERM) GO TO 60                          CDR  70
000077  000  C                                                             CDR  71
000078  000  C        THERMAL PROBLEM - CHECK FOR LONG OR SHORT PSEUDO COMPUTE SEQ.  CDR  72
000079  000  C                                                             CDR  73
000080  000        IF (ALPH(4).NE.PCSLNG) GO TO 30                         CDR  74
000081  000        LONG=.TRUE.                                             CDR  75
000082  000        IF (ALPH(5).EQ.ITWO) LONG2=.TRUE.                       VERS-005
000083  000        GO TO 80                                                CDR  76
000084  000     30 IF (ALPH(4).NE.PCSSHT) GO TO 500                        CDR  77
000085  000        GO TO 80                                                CDR  78
000086  000  C                                                             CDR  79
000087  000  C        CHECK FOR INITIAL PARAMETER RUN                      CDR  80
000088  000  C                                                             CDR  81
000089  000     40 CONTINUE                                                CDR  82
000090  000        IF (ALPH(3).NE.HINIT) GO TO 50                          CDR  83
000091  000        PARINT=.TRUE.                                           CDR  84
000092  000        PCNGID=HINIT                                            CDR  85
000093  000        WRITE (LB3D) (HINIT,I=1,50)                             VERS   3
000094  000        CALL INCORE (0)                                         CDR  87
000095  000        GO TO 80                                                CDR  88
000096  000  C                                                             CDR  89
000097  000  C        FINAL PARAMETER RUN                                  CDR  90
000098  000  C                                                             CDR  91
000099  000     50 IF (ALPH(3).NE.FINE) GO TO 510                          CDR  92
000100  000        PARFIN=.TRUE.                                           CDR  93
000101  000        PCNGID=FINE                                             CDR  94
000102  000        WRITE (LB3D) (FINE,I=1,50)                              VERS   3
000103  000        CALL INCORE (0)                                         CDR  96
000104  000        GO TO 80                                                CDR  97
000105  000  C                                                             CDR  98
000106  000  C        CHECK FOR GENERAL PROBLEM                            CDR  99
000107  000  C                                                             CDR 100
000108  000     60 CONTINUE                                                CDR 101
000109  000        IF (ALPH(3).NE.GENRLP) GO TO 40                         CDR 102
000110  000        GENERL=.TRUE.                                           CDR 103
000111  000        DO 70 I=1,10                                            CDR 104
000112  000        LOC(I)=0                                                CDR 105
```

```
000113  000      70 CONTINUE                                                    CDR 106
000114  000  C                                                                  CDR 107
000115  000  C        SET UP TITLE                                              CDR 108
000116  000  C                                                                  CDR 109
000117  000      80 CONTINUE                                                    CDR 110
000118  000         M=0                                                         CDR 111
000119  000         J1=1                                                        CDR 112
000120  000      90 CALL SREADC(2)                                              V 6
000121  000         IF (COL1.NE.COMMNT) GO TO 100                               CDR 114
000122  000         WRITE (NOUT,640) BLANK,ALPH,COL1                            CDR 115
000123  000         GO TO 90                                                    CDR 116
000124  000     100 CALL SREADC(3)                                              V 6
000125  000         IF (CODE.EQ.END) GO TO 120                                  CDR 118
000126  000         PTITLE=.TRUE.                                               CDR 119
000127  000         WRITE (NOUT,630) CODE,N,(ALPH(I),I=1,N)                     CDR 120
000128  000         M=M+N                                                       CDR 121
000129  000         IF (ILAST.GT.20) GO TO 90                                   CDR 122
000130  000         J=J1                                                        CDR 123
000131  000         K=0                                                         CDR 124
000132  000         DO 110 I=J,M                                                CDR 125
000133  000         ILAST=I                                                     CDR 126
000134  000         IF (I.GT.20) GO TO 90                                       CDR 127
000135  000         K=K+1                                                       CDR 128
000136  000         TITLE(I)=ALPH(K)                                            CDR 129
000137  000     110 CONTINUE                                                    CDR 130
000138  000         J1=M+1                                                      CDR 131
000139  000         GO TO 90                                                    CDR 132
000140  000     120 WRITE (NOUT,650) CODE                                       CDR 133
000141  000         IF ((ILAST.EQ.0).AND.(PARINT.OR.PARFIN)) GO TO 140          CDR 134
000142  000         IF (ILAST.GE.20) GO TO 140                                  CDR 135
000143  000         ILAST=ILAST+1                                               CDR 136
000144  000  C                                                                  CDR 137
000145  000  C        FILL OUT TITLE WITH BLANKS                                CDR 138
000146  000  C                                                                  CDR 139
000147  000         DO 130 I=ILAST,20                                           CDR 140
000148  000         TITLE(I)=BLANK                                              CDR 141
000149  000     130 CONTINUE                                                    CDR 142
000150  000  C                                                                  CDR 143
000151  000  C        WRITE TITLE ON TAPES                                      CDR 144
000152  000  C                                                                  CDR 145
000153  000     140 CONTINUE                                                    CDR 146
000154  000         CALL WRTDTA (0)                                             CDR 147
000155  000         IF (PARINT.OR.PARFIN) GO TO 530                             CDR 148
000156  000         CALL WRTPMT (0)                                             CDR 149
000157  000  C                                                                  CDR 150
000158  000  C        ZERO ARRAY OF FIXED CONSTANTS FOR CALLS TO DATARD          CDR 151
000159  000  C                                                                  CDR 152
000160  000         DO 150 I=1,50                                               CDR 153
000161  000         IFIXC(I)=0                                                  CDR 154
000162  000     150 CONTINUE                                                    CDR 155
000163  000         IF (.NOT.GENERL) GO TO 160                                  CDR 156
000164  000         IFIXC(31)=2                                                 CDR 157
000165  000         GO TO 260                                                   CDR 158
000166  000     160 CONTINUE                                                    CDR 159
000167  000         IF (LONG) IFIXC(31)=1                                       CDR 160
000168  000         FIXC(50)=1.
000169  000  C                                                                  CDR 161
```

```
000170  000  C        READ (BCD 3NODE DATA) BLOCK                          CDR 162
000171  000  C                                                             CDR 163
000172  000    170 CONTINUE                                                CDR 164
000173  000        CALL SREADC(1)                                            V 6
000174  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR 166
000175  000        IF (COL1.EQ.COMMNT) GO TO 170                           CDR 167
000176  000        IF (ALPH(1).EQ.REMARK) GO TO 170                        CDR 168
000177  000        IF (ALPH(3).NE.TEMPB) GO TO 510                         CDR 169
000178  000    180 CONTINUE                                                CDR 170
000179  000        CALL SREADC(1)                                            V 6
000180  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR 172
000181  000        IF (COL1.EQ.COMMNT) GO TO 180                           CDR 173
000182  000        IF (ALPH(1).EQ.END) GO TO 230                           CDR 174
000183  000        KBRANCH=1                                               CDR 175
000184  000        LNODE=.TRUE.                                            CDR 176
000185  000        LCOND=.FALSE.                                           CDR 177
000186  000        LCONST=.FALSE.                                          CDR 178
000187  000        LARRAY=.FALSE.                                          CDR 179
000188  000        NINC=LENBKT/6                                           CDR 180
000189  000        LOC(1)=1                                                CDR 181
000190  000        LEN(1)=0                                                CDR 182
000191  000        DO 190 I=2,6                                            CDR 183
000192  000        FACTOR=1.0
000193  000        LOC(I)=LOC(I-1)+NINC                                    CDR 184
000194  000        LEN(I)=0                                                CDR 185
000195  000    190 CONTINUE                                                CDR 186
000196  000        CALL DATARD                                             CDR 187
000197  000        CALL SQUEEZ (1,5)                                       CDR 188
000198  000        CALL WRTDTA (1)                                         CDR 189
000199  000        CALL WRTPNT (1)                                         CDR 190
000200  000        IF (.NOT.LPRINT) GO TO 230                              CDR 191
000201  000        WRITE (NOUT,720) NND,NNA,NNB,NNT                        CDR 192
000202  000        WRITE (NOUT,700) (I,LOC(I),LEN(I),I=1,5)                CDR 193
000203  000        M=LOC(5)+LEN(5)-1                                       CDR 194
000204  000        WRITE (NOUT,710) (I,IB(I),B(I),B(I),B(I),I=1,M)         CDR 195
000205  000        GO TO 230                                               CDR 196
000206  000  C                                                             CDR 197
000207  000  C        READ (BCD 3SOURCE DATA) BLOCK IF ANY                 CDR 198
000208  000  C                                                             CDR 199
000209  000    200 CONTINUE                                                CDR 200
000210  000        IF (ALPH(3).NE.OB) GO TO 510                            CDR 201
000211  000    210 CONTINUE                                                CDR 202
000212  000        CALL SREADC(1)                                            V 6
000213  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR 204
000214  000        IF (COL1.EQ.COMMNT) GO TO 210                           CDR 205
000215  000        IF (ALPH(1).EQ.END) GO TO 230                           CDR 206
000216  000        LNODE=.FALSE.                                           CDR 207
000217  000        LO=.TRUE.                                               CDR 208
000218  000        LEN(2)=0                                                CDR 209
000219  000        LEN(3)=0                                                CDR 210
000220  000        CALL DATARD                                             CDR 211
000221  000        CALL SQUEEZ (1,5)                                       CDR 212
000222  000        IF (.NOT.LPRINT) GO TO 220                              CDR 213
000223  000        WRITE (NOUT,700) (I,LOC(I),LEN(I),I=1,4)                CDR 214
000224  000        M1=LOC(2)                                               CDR 215
000225  000        M=LOC(3)+LEN(3)-1                                       CDR 216
000226  000        WRITE (NOUT,710) (I,B(I),B(I),B(I),I=M1,M)              CDR 217
```

```
000227  000     220 CONTINUE                                                  CDR 218
000228  000  C                                                                CDR 219
000229  000  C        READ (BCD 3CONDUCTOR DATA) BLOCK                        CDR 220
000230  000  C                                                                CDR 221
000231  000     230 CONTINUE                                                  CDR 222
000232  000         JPSTOT=0                                                  VERS 7
000233  000         END FILE 27
000234  000         REWIND 27
000235  000         CALL SREADC(1)                                             V 6
000236  000         WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                    CDR 224
000237  000         IF (COL1.EQ.COMMNT) GO TO 230                             CDR 225
000238  000         IF (ALPH(1).EQ.REMARK) GO TO 230                          CDR 226
000239  000         IF (ALPH(3).NE.CONDB) GO TO 200                           CDR 227
000240  000     240 CONTINUE                                                  CDR 228
000241  000         CALL SREADC(1)                                             V 6
000242  000         WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                    CDR 230
000243  000         IF (COL1.EQ.COMMNT) GO TO 240                             CDR 231
000244  000         IF (ALPH(1).EQ.END) GO TO 260                             CDR 232
000245  000         KBRNCH=2                                                  CDR 233
000246  000         LNODE=.FALSE.                                             CDR 234
000247  000         LCOND=.TRUE.                                              CDR 235
000248  000         LCONST=.FALSE.                                            CDR 236
000249  000         LARRAY=.FALSE.                                            CDR 237
000250  000         NNEW=LENBRT-(LOC(5)+LEN(5))+1                             CDR 238
000251  000         NINC=NNEW/5                                               CDR 239
000252  000         LOC(6)=LOC(5)+LEN(5)                                      CDR 240
000253  000         LEN(6)=0                                                  CDR 241
000254  000         DO 250 I=7,10                                             CDR 242
000255  000         LOC(I)=LOC(I-1)+NINC                                      CDR 243
000256  000         LEN(I)=0                                                  CDR 244
000257  000     250 CONTINUE                                                  CDR 245
000258  000         FACTOR=1.
000259  000         CALL DATARD                                               CDR 246
000260  000         JJIST=LOC(6)                                              VERS 7
000261  000         JJEND=LOC(6)+LEN(6)-1                                     VERS 7
000262  000         WRITE (27) (B(JJ),JJ=JJIST,JJEND)                         VERS 7
000263  000         CALL WRTPNT(2)                                            VERS 7
000264  000         CALL WRTBTA(2)                                            VERS 7
000265  000         READ (27) (B(JJ),JJ=JJIST,JJEND)                          VERS 7
000266  000         CALL SQUEEZ(6,10)                                         VERS 7
000267  000         IF (.NOT.LPRINT) GO TO 280                                CDR 250
000268  000         WRITE (NOUT,730) NGL,NGR,NGT                              CDR 251
000269  000         WRITE (NOUT,700) (I,LOC(I),LEN(I),I=6,10)                 CDR 252
000270  000         M1=LOC(6)                                                 CDR 253
000271  000         M=LOC(10)+LEN(10)-1                                       CDR 254
000272  000         WRITE (NOUT,710) (I,(B(I),B(I),B(I)),I=M1,M)              CDR 255
000273  000  C                                                                CDR 256
000274  000  C        READ (BCD 3CONSTANTS DATA) BLOCK                        CDR 257
000275  000  C                                                                CDR 258
000276  000     260 CONTINUE                                                  CDR 259
000277  000         CALL SREADC(1)                                             V 6
000278  000         WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                    CDR 261
000279  000         IF (COL1.EQ.COMMNT) GO TO 260                             CDR 262
000280  000         IF (ALPH(1).EQ.REMARK) GO TO 260                          CDR 263
000281  000         DATA IFLOW / 6HFLOW B /
000282  000         IF (ALPH(3) .NE. IFLOW) GO TO 265
000283  000         KFLOW = 1
```

```
000284  000          CALL FLOW1                                                  CDR 264
000285  000          GO TO 260                                                   CDR 265
000286  000      265 CONTINUE                                                    CDR 266
000287  000          IF (ALPH(3).NE.CONSTB) GO TO 510                            CDR 267
000288  000          IF (ALPH(14).EQ.PRINT) KTPRNT=.TRUE.                        CDR 268
000289  000      270 CONTINUE                                                    CDR 269
000290  000          CALL SREADC(1)                                               V 6
000291  000          WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                      CDR 271
000292  000          IF (COL1.EQ.COMMNT) GO TO 270                               CDR 272
000293  000          KBRNCH=3                                                    CDR 273
000294  000          LNODE=.FALSE.                                               CDR 274
000295  000          LCOND=.FALSE.                                               CDR 275
000296  000          LCONST=.TRUE.                                               CDR 276
000297  000          LARRAY=.FALSE.                                              CDR 277
000298  000          NNEW=LENBKT-(LOC(10)+LEN(10))+1                             CDR 278
000299  000          NINC=NNEW/2                                                 CDR 279
000300  000          LOC(11)=LOC(10)+LEN(10)                                     CDR 280
000301  000          IF (GENERL) LOC(11)=1                                       CDR 281
000302  000          LEN(11)=0                                                   CDR 282
000303  000          LOC(12)=LOC(11)+NINC                                        CDR 283
000304  000          LEN(12)=0                                                   CDR 284
000305  000          CALL DATARD                                                 CDR 285
000306  000          CALL SQUEEZ (11,12)                                         CDR 286
000307  000          CALL WRTDTA (3)                                             CDR 287
000308  000          CALL WRTPMT (3)                                             CDR 288
000309  000          IF (.NOT.LPRINT) GO TO 280                                  CDR 289
000310  000          WRITE (NOUT,740) NUC,NEC1,NEC2,NCT                          CDR 290
000311  000          WRITE (NOUT,750) (I,IFIXC(I),IFIXC(I),IFIXC(I),I=1,50)      CDR 291
000312  000          WRITE (NOUT,700) (I,LOC(I),LEN(I),I=11,12)                  CDR 292
000313  000          M1=LOC(11)                                                  CDR 293
000314  000          M=LOC(12)+LEN(12)-1                                         CDR 294
000315  000          WRITE (NOUT,710) (I,IB(I),B(I),B(I),I=M1,M)                 CDR 295
000316  000  C                                                                   CDR 296
000317  000  C           READ (BCD 3ARRAY DATA) BLOCK                            CDR 297
000318  000  C                                                                   CDR 298
000319  000      280 CONTINUE                                                    CDR 299
000320  000          CALL SREADC(1)                                               V 6
000321  000          WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                      CDR 301
000322  000          IF (COL1.EQ.COMMNT) GO TO 280                               CDR 302
000323  000          IF (ALPH(1).EQ.REMARK) GO TO 280                            CDR 303
000324  000          IF (ALPH(3).NE.ARRYB) GO TO 510                             CDR 304
000325  000          IF (ALPH(14).EQ.PRINT) AYPRNT=.TRUE.                        CDR 305
000326  000      290 CONTINUE                                                    CDR 306
000327  000          CALL SREADC(1)                                               V 6
000328  000          WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                      CDR 308
000329  000          IF (COL1.EQ.COMMNT) GO TO 290                               CDR 309
000330  000          IF (ALPH(1).EQ.END) GO TO 300                               CDR 310
000331  000          KBRNCH=4                                                    CDR 311
000332  000          LNODE=.FALSE.                                               CDR 312
000333  000          LCOND=.FALSE.                                               CDR 313
000334  000          LCONST=.FALSE.                                              CDR 314
000335  000          LARRAY=.TRUE.                                               CDR 315
000336  000          LOC(13)=LOC(12)+LEN(12)                                     CDR 316
000337  000          LEN(13)=0                                                   CDR 317
000338  000          LOC(14)=LOC(13)+200                                         CDR 318
000339  000          LEN(14)=0                                                   CDR 319
000340  000          LOC(15)=LOC(14)+200                                         CDR 320
```

```
000341  000        LEN(15)=0                                          CDR 317
000342  000        CALL DATARD                                        CDR 318
000343  000        CALL IMBED                                          V 9
000344  000    300 CONTINUE                                           CDR 319
000345  000        CALL SQUEEZ (13,15)                                CDR 320
000346  000        CALL WRTOTA (4)                                    CDR 321
000347  000        CALL WRTPMT (4)                                    CDR 322
000348  000        IF (.NOT.LPRINT) GO TO 310                         CDR 323
000349  000        WRITE (NOUT,760) LENA                              CDR 324
000350  000        WRITE (NOUT,700) (I,LOC(I),LEN(I),I=13,15)         CDR 325
000351  000        M1=LOC(13)                                         CDR 326
000352  000        M=LOC(15)+LEN(15)-1                                CDR 327
000353  000        WRITE (NOUT,710) (I,IB(I),B(I),B(I),I=M1,M)        CDR 328
000354  000  C                                                        CDR 329
000355  000  C        NORMAL RETURN                                     CDR 330
000356  000  C                                                        CDR 331
000357  000    310 CONTINUE                                           CDR 332
000358  000        IF(KFLOW) 312,314,312
000359  000    312 CALL FLOW2
000360  000        GO TO 316
000361  000    314 J = 0
000362  000        WRITE(LB30) (J,I=1,6)
000363  000    316 IF (GENERL) GO TO 320
000364  000  C        SET LOC AND LEN FOR CALL TO PSEUDO                CDR 334
000365  000        NNEW=LENBKT-(LOC(15)+LEN(15))+1                    CDR 335
000366  000        NINC=NNEW/2                                        CDR 336
000367  000        LOC(16)=LOC(15)+LEN(15)                            CDR 337
000368  000        LEN(16)=0                                          CDR 338
000369  000        LOC(17)=LOC(16)+NINC                               CDR 339
000370  000        LEN(17)=0                                          CDR 340
000371  000  C        CONVERT ARRAYS AND CONSTANTS TO FORTRAN          CDR 341
000372  000    320 CONTINUE                                           CDR 342
000373  000        M=4                                                CDR 343
000374  000        J=1                                                CDR 344
000375  000    330 IF (LEN(M).EQ.0) GO TO 400                         CDR 345
000376  000        K=LOC(M)                                           CDR 346
000377  000        KEND=K+LEN(M)                                      CDR 347
000378  000    340 ITYPE=FLD(0,6,IB(K))                               CDR 348
000379  000        FLD(0,5,IADDR)=ITYPE                               CDR 349
000380  000        ITYPE=MOD(ITYPE,4)                                 CDR 350
000381  000        IEND=1                                             CDR 351
000382  000        IF ((ITYPE.EQ.0).OR.(ITYPE.EQ.2)) IEND=2           CDR 352
000383  000        DO 390 I=1,IEND                                    CDR 353
000384  000        LITA=FLD(6,1,IB(K))                                CDR 354
000385  000        IANUM=FLD(7,14,IB(K))                              CDR 355
000386  000        IF (LITA.EQ.1) GO TO 350                           CDR 356
000387  000        CALL RELACT (2,IANUM,J,M)                          CDR 357
000388  000        GO TO 360                                          CDR 358
000389  000    350 IANUM=IANUM+NUC+1                                  CDR 359
000390  000        IF (M.EQ.9) IANUM=IANUM+NEC1                       CDR 360
000391  000    360 FLD(5,1,IADDR)=LITA                                CDR 361
000392  000        FLD(6,16,IADDR)=IANUM                              CDR 362
000393  000        LITK=FLD(21,1,IB(K))                               CDR 363
000394  000        KNUM=FLD(22,14,IB(K))                              CDR 364
000395  000        IF (LITK.EQ.1) GO TO 370                           CDR 365
000396  000        CALL RELACT (3,KNUM,J,M)                           CDR 366
000397  000        GO TO 380                                          CDR 367
```

```
000398  000     370 KNUM=KNUM+NUC+1                                          CDR 368
000399  000         IF (M.EQ.9) KNUM=KNUM+NEC1                               CDR 369
000400  000         IF (KNUM.GT.8191) CALL ERRMES (40,KNUM,0,0)              CDR 370
000401  000     380 FLD(22,1,IADDR)=LITK                                     CDR 371
000402  000         FLD(23,13,IADDR)=KNUM                                    CDR 372
000403  000         IB(K)=IADDR                                              CDR 373
000404  000         IADDR=0                                                  CDR 374
000405  000         K=K+1                                                    CDR 375
000406  000     390 CONTINUE                                                 CDR 376
000407  000         J=J+1                                                    CDR 377
000408  000         IF ((ITYPE.EQ.0).OR.(ITYPE.EQ.3)) K=K+1                  CDR 378
000409  000         IF (K.LT.KEND) GO TO 340                                 CDR 379
000410  000     400 IF (M.EQ.9) GO TO 410                                    CDR 380
000411  000         M=9                                                      CDR 381
000412  000         GO TO 330                                                CDR 382
000413  000     410 CONTINUE                                                 CDR 383
000414  000         IF (.NOT.LQ) GO TO 490                                   CDR 384
000415  000         NCC=NUC+NEC1+NEC2+1                                      CDR 385
000416  000         J=-1                                                     CDR 386
000417  000         K=LOC(2)                                                 CDR 387
000418  000         KEND=K+LEN(2)                                            CDR 388
000419  000         IADDR=0                                                  CDR 389
000420  000     420 ITYPE=FLD(0,6,IB(K))                                     CDR 390
000421  000         K=K+1                                                    CDR 391
000422  000         IEND=1                                                   CDR 392
000423  000         IF (ITYPE.GT.3) IEND=2                                   CDR 393
000424  000         DO 480 I=1,IEND                                          CDR 394
000425  000         IF (ITYPE.EQ.1) GO TO 450                                CDR 395
000426  000         LITA=FLD(6,1,IB(K))                                      CDR 396
000427  000         IANUM=FLD(7,19,IB(K))                                    CDR 397
000428  000         IF (LITA.EQ.1) GO TO 430                                 CDR 398
000429  000         CALL RELACT (2,IANUM,J,2)                                CDR 399
000430  000         GO TO 440                                                CDR 400
000431  000     430 IANUM=IANUM+NCC                                          CDR 401
000432  000     440 FLD(5,1,IADDR)=LITA                                      CDR 402
000433  000         FLD(6,16,IADDR)=IANUM                                    CDR 403
000434  000     450 LITK=FLD(21,1,IB(K))                                     CDR 404
000435  000         KNUM=FLD(22,14,IB(K))                                    CDR 405
000436  000         IF (LITK.EQ.1) GO TO 460                                 CDR 406
000437  000         CALL RELACT (3,KNUM,J,2)                                 CDR 407
000438  000         GO TO 470                                                CDR 408
000439  000     460 KNUM=KNUM+NCC                                            CDR 409
000440  000         IF (KNUM.GT.8191) CALL ERRMES (40,KNUM,0,0)              CDR 410
000441  000     470 FLD(22,1,IADDR)=LITK                                     CDR 411
000442  000         FLD(23,13,IADDR)=KNUM                                    CDR 412
000443  000         IB(K)=IADDR                                              CDR 413
000444  000         IADDR=0                                                  CDR 414
000445  000         K=K+1                                                    CDR 415
000446  000     480 CONTINUE                                                 CDR 416
000447  000         J=J-1                                                    CDR 417
000448  000         IF (K.LT.KEND) GO TO 420                                 CDR 418
000449  000     490 CONTINUE                                                 CDR 419
000450  000         LNODE=.FALSE.                                            CDR 420
000451  000         LCOND=.FALSE.                                            CDR 421
000452  000         LCONST=.FALSE.                                           CDR 422
000453  000         LARRAY=.FALSE.                                           CDR 423
000454  000         RETURN                                                   CDR 424
```

```
000455  000  C
000456  000  C        ERROR RETURN                                          CDR 425
000457  000  C                                                              CDR 426
000458  000    500 WRITE (NOUT,690)                                        CDR 427
000459  000        ERDATA=2.0                                              CDR 428
000460  000        GO TO 520                                               CDR 429
000461  000    510 WRITE (NOUT,680)                                        CDR 430
000462  000        ERDATA=2.0                                              CDR 431
000463  000    520 CONTINUE                                                CDR 432
000464  000        ENDRUN=1.0                                              CDR 433
000465  000        RETURN                                                  CDR 434
000466  000  C                                                             CDR 435
000467  000  C        PARAMETER RUNS                                        CDR 436
000468  000  C                                                             CDR 437
000469  000    530 CONTINUE                                                CDR 438
000470  000        LEND=.FALSE.                                            CDR 439
000471  000        NOREAD=.FALSE.                                          CDR 440
000472  000        IST=1                                                   CDR 441
000473  000        IF (GENERL) IST=3                                       CDR 442
000474  000        DO 610 I=IST,4                                          CDR 443
000475  000        IF (LEND.OR.NOREAD) GO TO 580                           CDR 444
000476  000    540 CALL SREADC(1)                                          CDR 445
                                                                           V 6
000477  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR 447
000478  000        IF (COL1.EQ.COMMNT) GO TO 540                           CDR 448
000479  000        IF (ALPH(1).EQ.REMARK) GO TO 540                        CDR 449
000480  000        IF (ALPH(3).NE.ENDPRM) GO TO 550                        CDR 450
000481  000        LEND=.TRUE.                                             CDR 451
000482  000        GO TO 580                                               CDR 452
000483  000    550 IBHC=ALPH(3)                                            CDR 453
000484  000        DO 560 J=1,4                                            CDR 454
000485  000        IF (IBHC.EQ.BLOCK(J)) GO TO 570                         CDR 455
000486  000    560 CONTINUE                                                CDR 456
000487  000        GO TO 510                                               CDR 457
000488  000    570 CALL SREADC(1)                                          
                                                                           V 6
000489  000        WRITE (NOUT,670) BLANK,COL27,ALPH,COL1                  CDR 459
000490  000        IF (COL1.EQ.COMMNT) GO TO 570                           CDR 460
000491  000        IF (ALPH(3).EQ.REMARK) GO TO 570                        CDR 461
000492  000    580 CALL INCORE (I)                                         CDR 462
000493  000        IF (ALPH(1).EQ.END) GO TO 600                           CDR 463
000494  000        NOREAD=.TRUE.                                           CDR 464
000495  000        IF (IBHC.NE.BLOCK(I)) GO TO 600                         CDR 465
000496  000        NOREAD=.FALSE.                                          CDR 466
000497  000        KBRNCH=I                                                CDR 467
000498  000        DO 590 J=1,4                                            CDR 468
000499  000        LLOGIC(J)=.FALSE.                                       CDR 469
000500  000    590 CONTINUE                                                CDR 470
000501  000        LLOGIC(I)=.TRUE.                                        CDR 471
000502  000        CALL DATARD                                             CDR 472
000503  000    600 CALL WRTDTA (I)                                         CDR 473
000504  000    610 CONTINUE                                                CDR 474
000505  000        GO TO 490                                               CDR 475
000506  000  C                                                             CDR 476
000507  000  C                                                             CDR 477
000508  000    620 FORMAT (1H1//)                                          CDR 478
000509  000    630 FORMAT (7X,A9,I1,11A6,A2)                               CDR 479
000510  000    640 FORMAT (A1,13A6,2A1)                                    CDR 480
000511  000    650 FORMAT (7X,A6)                                          CDR 481
```

```
000512  000     660 FORMAT (7X,A4,A1,11A6,A2)                                    CDR 482
000513  000     670 FORMAT (A1,A6,A4,A1,11A6,A2,A1)                              CDR 483
000514  000     680 FORMAT (6H * * *,82H DATA  BLOCKS  IN  IMPROPER  ORDER  OR  ILLEGACDR 484
000515  000        1H  BLOCK  DESIGNATION  ENCOUNTERED .)                        CDR 485
000516  000     690 FORMAT (6H * * *,90H THE PSEUDO COMPUTE SEQUENCE INDICATOR MUST BECDR 486
000517  000        1 EITHER SPCS OR LPCS, AND START IN COLUMN 21)                CDR 487
000518  000     700 FORMAT (19H ARRAYS LOC AND LEN,/(3I10))                      CDR 488
000519  000     710 FORMAT (12H DATA BUCKET,/(I10,I20,E20.5,5X,O20))             CDR 489
000520  000     720 FORMAT (/4H NND,I6,4H NNA,I6,4H NNB,I6,4H NNT,I6)            CDR 490
000521  000     730 FORMAT (/4H NGL,I6,4H NGR,I6,4H NGT,I6)                      CDR 491
000522  000     740 FORMAT (/4H NUC,I6,5H NEC1,I6,5H NEC2,I6,4H NCT,I6)          CDR 492
000523  000     750 FORMAT (/22H FIXED CONSTANTS ARRAY,/(I3,I20,E20.5,8X,O12))   CDR 493
000524  000     760 FORMAT (/5H LENA,I6)                                         CDR 494
000525  000         END                                                          CDR 495-
```

END ELT.

*HDG,P COMBIN

```
4ELT,L COMBIN
ELTOT7 RLIB70 02/28-03:19:09-(0,)
000001       000          SUBROUTINE COMBIN(NTAPE, KT, INC, IUNIT)
000002       000    C
000003       000          DIMENSION NBUFR(27), ALPHA(15), XSTART(7), XSTOP(7)
000004       000    C
000005       000          COMMON /XVARY/ DATA(1)
000006       000    C
000007       000          DATA ALPHA /1HA,1HB,1HC,1HD,1H5,1H6,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
000008       000         1            1HL,1HM/
000009       000    C
000010       000    C
000011       000          WRITE(6,3)
000012       000        3 FORMAT(1H1,10X,30HOUTPUT FROM COMBINE SUBROUTINE//)
000013       000          IF(IUNIT .EQ. 0) IUNIT = 7
000014       000          REWIND KT
000015       000          IF(NTAPE.GT.0) GO TO 7
000016       000          NTAPE = -NTAPE
000017       000          READ(5,6) (XSTART(I),XSTOP(I),I=1,NTAPE)
000018       000        6 FORMAT(14F5.3)
000019       000        7 IUNIT1=IUNIT-1
000020       000          DO 39 L=1,NTAPE
000021       000          M=0
000022       000          N=0
000023       000          I=L+IUNIT1
000024       000          REWIND I
000025       000       15 READ(I) (NBUFR(J),J=1,26), NSL, (DATA(J),J=1,NSL)
000026       000          NBUFR(27) = NSL
000027       000          IF(L-1),15
000028       000        9 READ(I)TIME,(DATA(K),K=1,NTOTAL)
000029       000          IF(TIME-XSTART(L))9
000030       000          IF(TIME-XTIME),21,30
000031       000          WRITE(6,12)
000032       000       12 FORMAT(1H0,10X,34HTAPES ARE NOT IN THE CORRECT ORDER)
000033       000          CALL EXIT
000034       000          WRITE(KT) NBUFR, (DATA(J),J=1,NSL)
000035       000          NTOTAL = 0
000036       000          DO 18 J=17,27
000037       000       18 NTOTAL=NTOTAL+NBUFR(J)
000038       000       21 READ(I)TIME,(DATA(K),K=1,NTOTAL)
000039       000          IF(TIME)24
000040       000          IF(TIME-XSTART(L))21
000041       000          IF(XSTOP(L))27,27
000042       000          IF(TIME-XSTOP(L))27
000043       000          TIME=-TIME
000044       000       24 IF(L-NTAPE)33,30,33
000045       000       27 M=M-1
000046       000          IF(M),,21
000047       000          M=INC
000048       000          XTIME=TIME
000049       000       30 WRITE(KT) TIME,(DATA(K),K=1,NTOTAL)
000050       000          IF(N.EQ.0) WRITE(6,31) TIME,L
000051       000       31 FORMAT(13X F10.5, 26H HAS BEEN LOADED FROM TAPE I2)
000052       000          N=1
000053       000          IF(TIME),21,21
000054       000          END FILE KT
000055       000          REWIND KT
```

```
000056      000       33 REWIND I
000057      000          WRITE(6,36)L,XTIME
000058      000       36 FORMAT(13X 4HTAPE I2, 10H ENDING AT F10.5,     28H HAS BEEN LOADED
000059      000         1 ON NEW TAPE/)
000060      000       39 CONTINUE
000061      000          IF(K.LE.15)WRITE(6,42)NTAPE,ALPHA(KT)
000062      000       42 FORMAT(//10X9HDATA FROM I2,33H TAPES HAS BEEN COMBINED ON UNIT A2)
000063      000          RETURN
000064      000          END

END ELT.


*HOG,P CRVINT
```

```
*ELT,L SICOMM
ELTOTT RLIB70 02/28-03:21:29-(0,)
000001      000    COMM   PROC
000002      000           COMMON /TITLE/H(1) /TEMP/T(1) /CAP/C(1) /SOURCE/Q(1) /COND/G(1)
000003      000           COMMON /PC1/NSQ1(1) /PC2/NSQ2(1) /KONST/K(1) /ARRAY/A(1)
000004      000           COMMON /FIXCON/KON(1) /XSPACE/NDIM,NTH,X(1)
000005      000           COMMON /DIMENS/ NND,NNA,NNT,NGT,NCT,NAT,LSQ1,LSQ2
000006      000           DIMENSION CON(1),XK(1),NX(1)
000007      000           EQUIVALENCE (KON(1),CON(1)),(K(1),XK(1)),(X(1),NX(1))
000008      000    END

END ELT.


*HDG,P SIDEFF
```

```
*ELT,L CRVINT
ELTOTT RL1870 02/28-03:19:10-(0,)
000001      000          SUBROUTINE CRVINT(A,B)
000002      000    C
000003      000          DIMENSION A(1), B(1)
000004      000    C
000005      000          EQUIVALENCE (D,N)
000006      000    C
000007      000    C
000008      000          D = A(1)
000009      000          ICA = N
000010      000          D = B(1)
000011      000          ICB = N
000012      000          IF(MOD(ICA,2) .NE. 0) GO TO 100
000013      000          IF(ICA .NE. ICB) GO TO 100
000014      000          IF(ICA .LT. 4) GO TO 100
000015      000          B(2) = A(2)
000016      000          B(3) = 0.0
000017      000          DO 50 I=3,ICA,2
000018      000          B(I+1) = A(I+1)
000019      000          B(I+2) = B(I) + 0.5*(A(I+2)+A(I))*(A(I+1)-A(I-1))
000020      000       50 CONTINUE
000021      000          RETURN
000022      000    C
000023      000      100 WRITE(6,101) ICA, ICB
000024      000      101 FORMAT(1H0 120(1H*) // 46H INCORRECT ARRAY LENGTH INPUT TO CRVINT,
000025      000         1 ICA = I5, 7H, ICB = I5 // 1X 120(1H*))
000026      000          CALL WLKBCK
000027      000          CALL EXIT
000028      000          END

END ELT.


*HDG,* CSGDMP
```

```
*ELT,L CSGDMP
ELTOT7 RLIB70 02/28-03:19:11-(3,)
000001      000          SUBROUTINE CSGDMP
000002      000    C     THIS SUBROUTINE LISTS EACH DIFFUSION NODE BY ACTUAL NODE NUMBER
000003      000    C     GIVING ITS CAPACITANCE AND CSG PRODUCT. EACH CONNECTED CONDUCTOR
000004      000    C     IS LISTED BY ACTUAL CONDUCTOR NUMBER GIVING ITS VALUE AND TYPE
000005      000    C     AND THE ACTUAL NODE NUMBER AND TYPE OF THE ADJOINING NODE.
000006      000    C     EITHER PSEUDO COMPUTE SEQUENCE IS ALLOWED.   SINDA  FORTRAN V
000007      000          INCLUDE COMM,LIST
000008      000          COMMON /POINTN/ LNODE,LCOND,LCONS,LARRY,ICOMP
000009      000          INCLUDE DEFF,LIST
000010      003           IF(CON(50) .LE. 0) CON(50)= 1.
000011      000          IJK = 1
000012      000          GO TO 2
000013      000          ENTRY RCDUMP
000014      000          IJK = 2
000015      000        2 CALL NONLIN
000016      000          IF(LNODE.EQ.0) CALL NNREAD(1)
000017      000          IF(LCOND.EQ.0) CALL NNREAD(2)
000018      000          IF(IJK.EQ.1) CALL OUTCAL
000019      000          NNC = NND+NNA
000020      000    C     ZERO OUT ALL SOURCE LOCATIONS
000021      000          DO 5 I = 1,NND
000022      000        5 Q(I) = 0.0
000023      000          IF(KON(29).EQ.0.OR.KON(28).GE.63) CALL TOPLIN
000024      000          NNN = KON(31)+1
000025      000          GO TO (10,15), NNN
000026      000       10 WRITE(6,998) NNC
000027      000          GO TO 20
000028      000       15 WRITE(6,997) NNC
000029      000       20 KON(28) = KON(28)+3
000030      000          J1 = 0
000031      000          IF(NND.EQ.0) GO TO 152
000032      000    C     CALCULATE G SUM
000033      000          DO 45 I = 1,NND
000034      000       25 J1 = J1+1
000035      000          LG = FLD(5,16,NSO1(J1))
000036      000          IF(LG.EQ.0) GO TO 45
000037      000          LTA = FLD(22,14,NSO1(J1))
000038      000    C     CHECK FOR RADIATION CONDUCTOR
000039      000          IF(FLD(3,1,NSO1(J1)).EQ.0) GO TO 30
000040      000          T1 = T(I)+460.
000041      000          T2 = T(LTA)+460.
000042      000          GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000043      003           GV= GV *CON(50)
000044      000          GO TO 35
000045      000       30 GV = G(LG)
000046      000       35 Q(I) = Q(I)+GV
000047      000          IF(NNN.EQ.2.OR.LTA.GT.NND.OR.FLD(21,1,NSO1(J1)).EQ.1) GO TO 40
000048      000          Q(LTA) = Q(LTA)+GV
000049      000       40 IF(NSO1(J1).GT.0) GO TO 25
000050      000       45 CONTINUE
000051      000    C     CALCULATE C/SK MINIMUM AND MAXIMUM AND NODES ON WHICH THEY OCCUR
000052      000          RCMN = 1.E+20
000053      000          RCMX = 0.0
000054      000          DO 55 I = 1,NND
000055      000          Q(I) = C(I)/Q(I)
```

```
000056  000        IF(Q(I).GT.RCMN) GO TO 50
000057  001        IF(.NOT. Q(I) .GT. 0.0) GO TO 50
000058  000        RCMN = Q(I)
000059  000        N = I
000060  000     50 IF(Q(I).LT.RCMX) GO TO 55
000061  000        RCMX = Q(I)
000062  000        M = I
000063  000     55 CONTINUE
000064  000        WRITE(6,996) X(N+LNODE),RCMN,X(M+LNODE),RCMX
000065  000        WRITE(6,977)
000066  000        KON(28) = KON(28)+2
000067  000        IF(KON(28).GE.65) CALL TOPLIN
000068  000        WRITE(6,995)
000069  000        WRITE(6,977)
000070  000        KON(28) = KON(28)+2
000071  000        J1 = 0
000072  000        DO 150 I = 1,NND
000073  000        WRITE(6,994) X(I+LNODE),C(I),Q(I)
000074  000        KON(28) = KON(28)+1
000075  000     90 J1 = J1+1
000076  000        LG = FLD(5,16,NSQ1(J1))
000077  000        IF(LG.EQ.0) GO TO 129
000078  000        LTA = FLD(22,14,NSQ1(J1))
000079  000        NNN = 1
000080  000        IF(FLD(3,1,NSQ1(J1)).EQ.1) NNN = 5
000081  000        IF(LTA.LE.NND) GO TO 92
000082  000        NNN = NNN+1
000083  000        IF(LTA.LE.NNC) GO TO 95
000084  000        NNN = NNN+1
000085  000        GO TO 95
000086  000     92 IF(FLD(21,1,NSQ1(J1)).EQ.1) NNN = NNN+3
000087  000     95 GO TO (100,105,110,112,115,120,125,127), NNN
000088  000    100 WRITE(6,993) X(LG+LCOND),G(LG),X(LTA+LNODE)
000089  000        GO TO 130
000090  000    105 WRITE(6,992) X(LG+LCOND),G(LG),X(LTA+LNODE)
000091  000        GO TO 130
000092  000    110 WRITE(6,991) X(LG+LCOND),G(LG),X(LTA+LNODE)
000093  000        GO TO 130
000094  000    112 WRITE(6,986) X(LG+LCOND),G(LG),X(LTA+LNODE)
000095  000        GO TO 130
000096  000    115 WRITE(6,990) X(LG+LCOND),G(LG),X(LTA+LNODE)
000097  000        GO TO 130
000098  000    120 WRITE(6,989) X(LG+LCOND),G(LG),X(LTA+LNODE)
000099  000        GO TO 130
000100  000    125 WRITE(6,988) X(LG+LCOND),G(LG),X(LTA+LNODE)
000101  000        GO TO 130
000102  000    127 WRITE(6,985) X(LG+LCOND),G(LG),X(LTA+LNODE)
000103  000        GO TO 130
000104  000    129 WRITE(6,987)
000105  000    130 KON(28) = KON(28)+1
000106  000        IF(KON(28).LT.65) GO TO 135
000107  000        CALL TOPLIN
000108  000        WRITE(6,995)
000109  000        WRITE(6,977)
000110  000        KON(28) = KON(28)+2
000111  000    135 IF(NSQ1(J1).GT.0) GO TO 90
000112  000    150 CONTINUE
```

```
000113  000        IF(NNA.LE.0) GO TO 999
000114  000    152 WRITE(6,977)
000115  000        KON(28) = KON(28)+1
000116  000        IF(KON(28).GE.63) CALL TOPLIN
000117  000        WRITE(6,984)
000118  000        WRITE(6,977)
000119  000        KON(28) = KON(28)+2
000120  000        NN = NND+1
000121  000        DO 210 I = NN,NNC
000122  000    155 J1 = J1+1
000123  000        LG = FLD(5,16,NSQ1(J1))
000124  000        LTA = FLD(22,14,NSQ1(J1))
000125  000        NNN = 1
000126  000        IF(FLD(3,1,NSQ1(J1)).EQ.1) NNN = 4
000127  000        IF(LTA.LE.NND) GO TO 158
000128  000        NNN = NNN+1
000129  000        IF(LTA.LE.NNC) GO TO 158
000130  000        NNN = NNN+1
000131  003         IF(NNN .GE. 4) GLG= G(LG)*CON(50)
000132  000    158 GO TO (160,165,170,175,180,185), NNN
000133  000    160 WRITE(6,983) X(I+LNODE),X(LTA+LNODE),X(LG+LCOND),G(LG)
000134  000        GO TO 190
000135  000    165 WRITE(6,982) X(I+LNODE),X(LTA+LNODE),X(LG+LCOND),G(LG)
000136  000        GO TO 190
000137  000    170 WRITE(6,981) X(I+LNODE),X(LTA+LNODE),X(LG+LCOND),G(LG)
000138  000        GO TO 190
000139  003    175  WRITE(6,980) X(I+LNODE),X(LTA+LNODE),X(LG+LCOND),GLG
000140  000        GO TO 190
000141  003    180  WRITE(6,979) X(I+LNODE),X(LTA+L ODE),X(LG+LCOND),GLG
000142  000        GO TO 190
000143  003    185  WRITE(6,978) X(I+LNODE),X(LTA+LNODE),X(LG+LCOND),GLG
000144  000    190 KON(28) = KON(28)+1
000145  000        IF(KON(28).LT.65) GO TO 195
000146  000        CALL TOPLIN
000147  000        WRITE(6,984)
000148  000        WRITE(6,977)
000149  000        KON(28) = KON(28)+2
000150  000    195 IF(NSQ1(J1).GT.0) GO TO 155
000151  000    210 CONTINUE
000152  000    959 RETURN
000153  000    998 FORMAT(/,3H A I4,56H NODE PROBLEM USING SPCS, **** ALL NUMBERS ARE
000154  000       $ ACTUAL ****,/
000155  000    997 FORMAT(/,3H A I4,50H NODE PROBLEM USING LPCS, **** ALL NUMBERS ARE A
000156  000       $ ACTUAL ****,/)
000157  000    996 FORMAT(6H NODE I6,19H HAS THE CSGMIN OF 1PE12.5,7H, NODE I6,
000158  000       *       19H HAS THE CSGMAX OF 1PE12.5)
000159  000    995 FORMAT(59H NODE    C-VALUE    CSG-VALUE COND  TYPE G-VALUE TO NODE T
000160  000       *YPE)
000161  000    994 FORMAT(I6,2(1PE10.3))
000162  000    993 FORMAT(26X,I6,5H  LIN1PE10.3,I6,5H DIFF)
000163  000    992 FORMAT(26X,I6,5H  LIN1PE10.3,I6,5H ARTH)
000164  000    991 FORMAT(26X,I6,5H  LIN1PE10.3,I6,5H BOUN)
000165  000    990 FORMAT(26X,I6,5H  RAD1PE10.3,I6,5H DIFF)
000166  000    989 FORMAT(26X,I6,5H  RAD1PE10.3,I6,5H ARTH)
000167  000    988 FORMAT(26X,I6,5H  RAD1PE10.3,I6,5H BOUN)
000168  000    987 FORMAT(26X,29H THIS NODE HAS BEEN PROCESSED)
000169  000    986 FORMAT(26X,I6,5H  LIN1PE10.3,I6,24H DIFF, ONE WAY CONDUCTOR)
```

```
000170      000      985 FORMAT(26X,I6,5H  RAD1PE10.3,I6,24H DIFF, ONE WAY CONDUCTOR)
000171      000      984 FORMAT(40H ARITHMETIC NODE PSEUDO-COMPUTE SEQUENCE)
000172      000      983 FORMAT(6H NODE I6,14H TO DIFF NODE I6,15H THRU LIN CO-D I6,15H,CON
000173      000         *D VALUE IS 1PE11.5)
000174      000      982 FORMAT(6H NODE I6,14H TO ARTH NODE I6,15H THRU LIN COND I6,15H,CON
000175      000         *D VALUE IS 1PE11.5)
000176      000      981 FORMAT(6H NODE I6,14H TO BOUN NODE I6,15H THRU LIN COND I6,15H,CON
000177      000         *D VALUE IS 1PE11.5)
000178      000      980 FORMAT(6H NODE I6,14H TO DIFF NODE I6,15H THRU RAD COND I6,15H,CON
000179      000         *D VALUE IS 1PE11.5)
000180      000      979 FORMAT(6H NODE I6,14H TO ARTH NODE I6,15H THRU RAD COND I6,15H,CON
000181      000         *D VALUE IS 1PE11.5)
000182      000      978 FORMAT(6H NODE I6,14H TO BOUN NODE I6,15H THRU RAD COND I6,15H,CON
000183      000         *D VALUE IS 1PE11.5)
000184      000      977 FORMAT(1H )
000185      000          END

END ELT.


#HDG,P CYCLE
```

CYCLE DATE 022875 PAGE 1

```
@ELT,L CYCLE
ELTOT7 RLIB70 02/28-03:19:15-(0,)
000001      000          SUBROUTINE CYCLE(TIME,A,NAME)
000002      000    C
000003      000          DIMENSION A(1)
000004      000    C
000005      000          EQUIVALENCE (A1,N)
000006      000    C
000007      000    C
000008      000          A1 = A(1)
000009      000          IF(MOD(N,2) .NE. 0) GO TO 910
000010      000          ASSIGN 900 TO II
000011      000       10 IF(A(2) - TIME) 30,II,20
000012      000       20 DELTA = A(2) - A(N)
000013      000          GO TO 50
000014      000       30 IF(TIME .LT. A(N)) GO TO II
000015      000          DELTA = A(N) - A(2)
000016      000       50 IF(DELTA) 60,900,60
000017      000       60 DO 100 I=2,N,2
000018      000          A(I) = A(I) + DELTA
000019      000      100 CONTINUE
000020      000          ASSIGN 105 TO II
000021      000          GO TO 10
000022      000      105 CALL LINECK(2)
000023      000          WRITE(6,110) NAME
000024      000      110 FORMAT(7H0ARRAY  A6, 16H HAS BEEN CYCLED )
000025      000          CALL LINECK(2)
000026      000          CALL GENOUT(A(2),1,N,'0CYCLED ARRAY')
000027      000      900 RETURN
000028      000      910 WRITE(6,920) NAME
000029      000      920 FORMAT(30H0WRONG ARRAY LENGTH FOR ARRAY  A6, 20H IN SUBROUTINE CYC
000030      000         1LE )
000031      000          CALL GENOUT(A(2),1,N,'0ARRAY TO BE CYCLED ')
000032      000          CALL WLKBCK
000033      000          CALL EXIT
000034      000          END

END ELT.


@HDG,P ONSUM1
```

```
@ELT,L SIDEFF
ELT077 RLIB70 02/28-03:21:30-(0,)
000001      000   DEFF     PROC
000002      000   C************  CONTROL CONSTANT DEFINITIONS AND NAMES  ***************
000003      000   C     CONTROL CONSTANT 1 CONTAINS THE NEW PROBLEM TIME              (TIMEN)
000004      000   C     CONTROL CONSTANT 2 CONTAINS THE TIME STEP USED                (DTIMEU)
000005      000   C     CONTROL CONSTANT 3 CONTAINS THE PROBLEM STOP TIME             (TIMEND)
000006      000   C     CONTROL CONSTANT 4 CONTAINS THE TIME STEP FACTOR,EXPLICIT     (CSGFAC)
000007      000   C     CC5 IS THE INPUT NUMBER OF ITERATION DO LOOPS, INTEGER        (NLOOP)
000008      000   C     CC6 CONTAINS THE DIFFUSION TEMPERATURE CHANGE ALLOWED         (DTMPCA)
000009      000   C     CC7 CONTAINS THE OUTPUT EACH ITERATION SWITCH                 (OPEITR)
000010      000   C     CC8 CONTAINS THE MAXIMUM ALLOWED TIME STEP                    (DTIMEH)
000011      000   C     CC9 CONTAINS THE NEW ARITHMETIC TEMP. DAMPING FACTOR          (DAMPA)
000012      000   C     CC10 CONTAINS THE NEW DIFFUSION TEMP. DAMPING FACTOR          (DAMPB)
000013      000   C     CC11 CONTAINS THE MAXIMUM ALLOWED ARITHMETIC TEMP. CHANGE     (ATMPCA)
000014      000   C     CC12 CONTAINS THE BACKUP SWITCH CHECKED AFTER VARIABLES       (BACKUP)
000015      000   C     CC13 CONTAINS THE PRESENT TIME OR PROBLEM START TIME          (TIMED)
000016      000   C     CC14 CONTAINS THE MEAN TIME BETWEEN AN ITERATION              (TIMEM)
000017      000   C     CC15 CONTAINS THE DIFFUSION TEMPERATURE CHANGE CALCULATED     (DTMPCC)
000018      000   C     CC16 CONTAINS ARITHMETIC TEMPERATURE CHANGE CALCULATED        (ATMPCC)
000019      000   C     CONTROL CONSTANT 17 IS RESERVED FOR THE C/SG MINIMUM          (CSGMIN)
000020      000   C     CONTROL CONSTANT 18 CONTAINS THE OUTPUT INTERVAL              (OUTPUT)
000021      000   C     CC19 CONTAINS THE ARITHMETIC RELAXATION CRITERIA ALLOWED      (ARLXCA)
000022      000   C     CC20 CONTAINS THE NUMBER OF RELAXATION LOOPS USED,INTEGER     (LOOPCT)
000023      000   C     CC21 CONTAINS THE MINIMUM ALLOWED TIME STEP                   (DTIMEL)
000024      000   C     CC22 IS FOR THE INPUT TIME STEP IMPLICIT                      (DTIMEI)
000025      000   C     CC23 CONTAINS THE C/SG MAXIMUM                                (CSGMAX)
000026      000   C     CC24 CONTAINS THE C/SG RANGE ALLOWED                          (CSGRAL)
000027      000   C     CC25 CONTAINS THE C/SG RANGE CALCULATED                       (CSGRCL)
000028      000   C     CC26 CONTAINS THE DIFFUSION RELAXATION CRITERIA ALLOWED       (DRLXCA)
000029      000   C     CC27 CONTAINS THE DIFFUSION RELAXATION CHANGE CALCULATED      (DRLXCC)
000030      000   C     CC28 CONTAINS THE LINE COUNTER, INTEGER                       (LINECT)
000031      000   C     CC29 CONTAINS THE PAGE COUNTER, INTEGER                       (PAGECT)
000032      000   C     CC30 CONTAINS ARITHMETIC RELAXATION CHANGE CALCULATED         (ARLXCC)
000033      000   C     CC31 IS INDICATOR, 0=THERMAL SPCS,1=THERMAL LPCS,2=GENERAL    (LSPCS)
000034      000   C     CC32 CONTAINS THE ENERGY BALANCE OF THE SYSTEM, IN - OUT       (ENGBAL)
000035      000   C     CC33 CONTAINS THE DESIRED ENERGY BALANCE, USER INPUT           (BALENG)
000036      000   C     CC34 CONTAINS THE NOCOPY SWITCH FOR MATRIX USERS              (NOCOPY)
000037      000   C     CC35 CONTAINS RELATIVE NODE NUMBER OF CSGMIN
000038      000   C     CC36 CONTAINS RELATIVE NODE NUMBER OF DTMPCC
000039      000   C     CC37 CONTAINS RELATIVE NODE NUMBER OF ARLXCC
000040      000   C     CC38 CONTAINS RELATIVE NODE NUMBER OF ATMPCC
000041      000   C     CC39-40-41-42-43 CONTAIN DUMMY INTEGER CONSTANTS   (I-J-K-L-MTEST)
000042      000   C     CC44-45-46-47-48 CONTAIN DUMMY FLOATING CONSTANTS  (R-S-T-U-VTEST)
000043      000   C     CC49 IS THE QUASI-LINEARIZATION INTERVAL FOR CINDSM            (LAXFAC)
000044      000   C     CC50 IS THE STEFAN-BOLTZMANN CONSTANT                          (SIGMA)
000045      000    END

END ELT.


@HDG,P SIDMCC
```

SIDMCC DATE 022875 PAGE 1

```
*ELT,T SIDMCC
ELTOTT RLIB70 02/28-03:21:32-(0,)
000001      000    DMCC     PROC
000002      000         IF(FLD(1,1,NSQ1(J1+1)).EQ.0) GO TO 400
000003      000         NTYPE = FLD(0,5,NSQ2(J2))
000004      000         GO TO (399,398,398,398,399,398,398,398,399), NTYPE
000005      000     398 J2 = J2+1
000006      000     399 J2 = J2+1
000007      000     400 CONTINUE
000008      000    END

END ELT.


*HDG,P SIDMCC
```

```
SIDMQQ                                                                          DATE 022875        PAGE     1

#ELT,L SIDMQQ
ELTOT7 RLIB70 02/28-03:21:34-(0,)
000001       000    DMQQ     PROC
000002       000           IF(FLD(4,1,NSQ1(J1+1)).EQ.0) GO TO 700
000003       000           NTYPE = FLD(0,5,NSQ2(J2))
000004       000           J2 = J2+1
000005       000           GO TO (700,700,700,699,699,699,699,699,699,699,699,699),NTYPE     VERS   5
000006       000       699 J2 = J2+1
000007       000       700 CONTINUE
000008       000     END

END ELT.


#HDG,P SIDUMC
```

```
*ELT,L DNSUM2
ELTDT7 RL1870 02/28-03:19:17-(1,)
000001      000          SUBROUTINE DNSUM2(NFRM,LOC2,J2,SUM1,SUM2,N2)
000002      000    C
000003      000          COMMON /FLODAT/ NFLOW(1)
000004      000    C
000005      000    C
000006      000          JJ2 = J2
000007      000      200 JJ2 = JJ2 - 4
000008      000          IF(JJ2 .LT. 4) GO TO 700
000009      000          L = LOC2 + JJ2
000010      000          IF(NFLOW(L+2) .NE. NFRM) GO TO 200
000011      000          LOCD = NFLOW(L+3)
000012      000          IF(LOCD) 220,200,400
000013      000    C
000014      000      220 LOC3 = -LOCD
000015      000          J3   = NFLOW(LOC3) + 1
000016      000          CALL DNSUM3(NFRM,LOC3,J3,SUM1,SUM2,N3)
000017      000          GO TO 200
000018      000    C
000019      000      400 CALL FLOSUM(NFLOW(L),LOCD,SUM1,SUM2)
000020      000          GO TO 200
000021      000    C
000022      000      700 N2 = NFLOW(L+1)
000023      000          RETURN
000024      000          END

END ELT.


*HDG,P DUM
```

```
@ELT,L DUM
ELTDT7 RLI870 02/28-03:19:19-(0,)
000001       000          SUBROUTINE DUM
000002       000    C
000003       000          COMMON /PC1/ NS01(1)  /PC2/ NS02(1)
000004       000    C
000005       000    C
000006       000          ENTRY CDUM(J1,J2)
000007       000          INCLUDE DMCC,LIST
000008       000          RETURN
000009       000    C
000010       000    C
000011       000          ENTRY QDUM(J1,J2)
000012       000          INCLUDE DMQQ,LIST
000013       000          RETURN
000014       000    C
000015       000    C
000016       000          ENTRY GDUM(J1,J2)
000017       000          INCLUDE DMGG,LIST
000018       000          RETURN
000019       000    C
000020       000          END

END ELT.


@HDG,P FLBAL
```

SIDUMC DATE 022875 PAGE 1

```
#ELT,L SIDUMC
ELTOT7 RL1870 02/28-03:21:35-(0,)
000001      000    DUMC   PROC
000002      000           IF(FLD(1,1,NSQ1(J1+1)).EQ.0) GO TO 2000
000003      000           NTYPE = FLD(0,5,NSQ2(J2))
000004      000           GO TO (1999,1998,1998,1998,1999,1998,1998,1998,1999), NTYPE
000005      000    1998 J2 = J2+1
000006      000    1999 J2 = J2+1
000007      000    2000 CONTINUE
000008      000    END

END ELT.


#RDG,P SIDUM
```

```
@ELT,L FLBAL
ELTOT7 RLIB70 02/28-03:19:20-(6,)
000001      002          SUBROUTINE FLBAL(NPRN,L14,WIN,NPI,NPO,FROF,DWMX,DP)
000002      002    C
000003      002          LOGICAL COP, LPR
000004      002    C
000005      002          DIMENSION RDATA(1), EXT(1)
000006      002    C
000007      003          COMMON /POINTN /LNODE,LCOND,LCONS,LARRY,ICOMP,LTB,LP,LV
000008      002          COMMON /FLOBAT/ NDATA (1)
000009      002          COMMON /WDOT  / W     (1)
000010      002          COMMON /PRESS / P     (1)
000011      002          COMMON /FLOWG / GF    (1)
000012      002          COMMON /VALVP / VP    (1)
000013      002          COMMON /WDOTI / WI    (1)
000014      002          COMMON /FDATA / COP
000015      002          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000016      002    C
000017      002          EQUIVALENCE (RDATA,NDATA), (EXT,NEXT)
000018      002    C
000019      002    C
000020      002          L20=NDATA(L14)-3
000021      002          L22 = NDATA(L14+2)
000022      002          L23 = NDATA(L14+3)
000023      002          L25 = NTH + 1
000024      006          NTH = NTH + NPRN + 1
000025      006          LFSV = L25 + NPRN
000026      006          NFSV=0
000027      006          ASSIGN 624  TO KK
000028      006          ASSIGN 1098 TO LL
000029      002    C
000030      002    C
000031      002            IF(L23 .LT. 1) GO TO 602
000032      002            L40 = NDATA(L23)
000033      002            IF(L40 .LT. 1) GO TO 6..
000034      002    C
000035      006            K = LFSV
000036      002      530   DO 600 J=1,L40
000037      002            L41=NDATA(L23+J)
000038      005            NV   = NDATA(L41+1)
000039      002            NTS1 = NDATA(L41+2)
000040      002            NTS2 = NDATA(L41+3)
000041      002            E    = RDATA(L41+7)
000042      006            IF(E) 536,533,536
000043      006    C
000044      006      533   NEXT(K+1) = NV
000045      006            NEXT(K+4) = NTS1
000046      006            NEXT(K+5) = NTS2
000047      006            K = K + 5
000048      006            GO TO 600
000049      006    C
000050      006      536   IF(.NOT.COP) GO TO 540
000051      002            RVS1 = 0.0
000052      002            RVS2 = 0.0
000053      002      540   IF(NTS1 .LT. 1) GO TO 560
000054      002            RVS1 = W(NTS1)*E/VP(NV)/VP(NV)
000055      002            IF(GF(NTS1)) 550,545,550
```

```
000056      002      545 GF(NTS1) = 1.0/RVS1
000057      002          GO TO 560
000058      002      550 GF(NTS1) = 1.0/(1.0/GF(NTS1)+RVS1)
000059      002      560 IF(NTS2 .LT. 1) GO TO 570
000060      002          XS2 = 1.0 - VP(NV)
000061      002          RVS2 = W(NTS2)*E/XS2/XS2
000062      002          IF(GF(NTS2)) 565,563,565
000063      002      563 GF(NTS2) = 1.0/RVS2
000064      002          GO TO 570
000065      002      565 GF(NTS2) = 1.0/(1.0/GF(NTS2)+RVS2)
000066      002      C
000067      002      570 IF(.NOT. COP) GO TO 600
000068      002          CALL LINECK(3)
000069      002          IF(NTS1 .GT. 0) NTS1 = NEXT(LTB+NTS1)
000070      002          IF(NTS2 .GT. 0) NTS2 = NEXT(LTB+NTS2)
000071      002          XS1 = VP(NV)
000072      003          NV = NEXT(LV+NV)
000073      002          WRITE(6,580) NV, E, XS1, NTS1, RVS1, NTS2, RVS2
000074      002      580 FORMAT(/               7X 7HNV    = I10  , 8X 7HE     = G13.8,
000075      002         1  5X 7HXS1   = G13.8, 5X 7HNTS1  = I10  , 8X 7HRVS1  = G13.8/
000076      002         2  7X 7HNTS2  = I10  , 8X 7HRVS2  = G13.8 )
000077      002      C
000078      002      C
000079      002      600 CONTINUE
000080      002      C
000081      002      C
000082      006          NFSV = (K-LFSV)/5
000083      006          IF(NFSV .LT. 1) GO TO 602
000084      006          ASSIGN 623  TO KK
000085      006          ASSIGN 1085 TO LL
000086      006          L = LFSV
000087      006          DO 6018 I=1,NFSV
000088      006          NTS1 = NEXT(L+4)
000089      006          NTS2 = NEXT(L+5)
000090      006          DO 6011 J=4,L20,4
000091      006          K = L14 + J
000092      006          IF(NTS1 .EQ. NDATA(K) .OR. NTS2 .EQ. NDATA(K)) GO TO 6013
000093      006       6011 CONTINUE
000094      006      C ERROR
000095      006       6013 NPU = NDATA(K+1)
000096      006          NEXT(L+3) = NPU
000097      006          DO 6014 J=4,L20,4
000098      006          K = L14 + J
000099      006          IF(NPU .EQ. NDATA(K+2)) GO TO 6016
000100      006       6014 CONTINUE
000101      006      C ERROR
000102      006       6016 NEXT(L+2) = NDATA(K)
000103      006          L = L + 5
000104      006       6018 CONTINUE
000105      006      C
000106      006        602 MPRN = NPRN + 2*NFSV
000107      006          M = MPRN*(MPRN+1)/2
000108      006          NTH = NTH + 5*NFSV
000109      002          L26 = NTH + 1
000110      002          NTH = NTH + M + 3
000111      002          L27 = NTH + 1
000112      006          NTH = NTH + MPRN + 1
```

```
000113  002        L28 = NTH + 1
000114  006        NTH = NTH + MPRN + 1
000115  002        IF(NDIM .GE. NTH-L25+1) GO TO 610
000116  002  C
000117  002        NEED = NTH - NDIM -L25 + 1
000118  002        CALL TOPLIN
000119  002        WRITE(6,605) NEED, NDATA(L14+1)
000120  002    605 FORMAT(83H0* * *  INSUFFICIENT DYNAMIC STORAGE AVAILABLE FOR FLOW
000121  002       1BALANCING SUBROUTINE  * * * // 8X 5HSHORT I5, 23H LOCATIONS FOR NE
000122  002       2TWORK A6)
000123  002        CALL WLKBCK
000124  002        CALL EXIT
000125  002  C.
000126  002    610 DO 620 J=L26,NTH
000127  002        NEXT(J) = 0
000128  002    620 CONTINUE
000129  002  C
000130  002        NEXT(L26) = M + 2
000131  006        NEXT(L26+1) = MPRN
000132  006        NEXT(L26+2) = MPRN
000133  006        NEXT(L27)   = MPRN
000134  002  C
000135  006        NEXT(L28) = MPRN
000136  006        DO 622 J=1,MPRN
000137  002        NEXT(L28+J) = NEXT(L25+J)
000138  002    622 CONTINUE
000139  002  C
000140  002  C ASSEMBLE COEFFICIENT MATRIX
000141  002  C
000142  002        DO 625 J=4,L20,4
000143  002        K = L14 + J
000144  002        NTB = NDATA(K)
000145  006        GO TO KK (623,624)
000146  006  C
000147  006    623 L = LFSV + 4
000148  006        NFRM = NPRN + 1
000149  006        DO 6235 I=1,NFSV
000150  006        IF(NTB .EQ. NEXT(L)) GO TO 6241
000151  006        NFRM = NFRM + 1
000152  006        IF(NTB .EQ. NEXT(L+1)) GO TO 6241
000153  006        NFRM = NFRM + 1
000154  006        L = L + 5
000155  006   6235 CONTINUE
000156  006    624 NFRM = NDATA(K+1)
000157  006   6241 NTO  = NDATA(K+2)
000158  002        NR   = MINO (NFRM,NTO)
000159  002        NC   = MAXO (NFRM,NTO)
000160  002        NRNR = (NR+1)*NR/2 + 2
000161  002        NRNC = (NC-1)*NC/2 + NR + 2
000162  002        NCNC = (NC+1)*NC/2 + 2
000163  002        EXT(L26+NRNR) = EXT(L26+NRNR) + GF(NTB)
000164  002        EXT(L26+NRNC) = EXT(L26+NRNC) - GF(NTB)
000165  002        EXT(L26+NCNC) = EXT(L26+NCNC) + GF(NTB)
000166  002    625 CONTINUE
000167  002        IF(.NOT. COP) GO TO 630
000168  002        CALL LINECK(2)
000169  002        CALL GENOUT(1,1,0,'0MATRIX BEFORE REDUCTION')
```

```
000170  002        CALL LINECK(2)
000171  002        CALL GENOUT(EXT(L28+1),1,NEXT(L28)  ,'0PRESSURE NODES (RELATIVE)')
000172  002        CALL LINECK(2)
000173  006        CALL GENOUT (EXT(L26+3),1,NEXT(L26)-2,'0COEFFICIENT MATRIX')
000174  002        CALL LINECK(2)
000175  006        CALL GENOUT(EXT(L27+1),1,MPRN  ,'0RIGHT HAND SIDE')
000176  002  C
000177  002  C IMPOSED FLOW RATES INTO RHS
000178  002  C
000179  002    630 DO 680 J=1,NPRN
000180  002        NPR = NEXT(L25+J)
000181  002        EXT(L27+J) = WI(NPR)
000182  002    680 CONTINUE
000183  006        IF(NFSV .LT. 1) GO TO 690
000184  006        K = L27 + NPRN
000185  006        L = LFSV
000186  006        DO 685 I=1,NFSV
000187  006        NV  = NEXT(L+1)
000188  006        NTU = NEXT(L+2)
000189  006        NPU = NEXT(L+3)
000190  006        EXT(K+1) = VP(NV)*W(NTU)
000191  006        EXT(K+2) = (1.0-VP(NV))*W(NTU)
000192  006        K = K + 1
000193  006        CALL CMPRSS(P(NPU),NPU,EXT(L26+3),EXT(L27+1),NEXT(L28),$860)
000194  006        L = L + 5
000195  006    685 CONTINUE
000196  002  C
000197  002  C INLET FLOW RATE INTO RHS
000198  002  C
000199  002    690 N = NPRN
000200  002    700 IF(NPI .LT. 1) GO TO 780
000201  002        NIFNR = NPI
000202  002        CALL PRN(NEXT(L25),N,NIFNR)
000203  002        IF(N .LE. NPRN) GO TO 760
000204  002  C
000205  002        NIFN = NPI
000206  002    720 CALL TOPLIN
000207  006        IF (LP.EQ.0) CALL NNREAD(5)
000208  002        IF(NIFN .GT. 0) NIFN = NEXT(LP +NIFN)
000209  002        WRITE(6,740) NIFN, NDATA(L14+1)
000210  002    740 FORMAT(69H0* * *  ERROR IN LOCATING PRESSURE NODE WITH IMPOSED FLO
000211  002       1W RATE  * * * // 0X 4HNODE I5, 27H WAS NOT FOUND FOR NETWORK A6)
000212  002        CALL WLKBCK
000213  002        CALL EXIT
000214  002  C
000215  002    760 EXT(L27+NIFNR) = WIN
000216  002  C
000217  002  C SPECIFIED PRESSURES INTO COEFFICIENT MATRIX AND RHS
000218  002  C
000219  002    780 IF(L22 .LT. 1) GO TO 840
000220  002        IF(NDATA(L22) - 1) 840,800,810
000221  002    800 IF(NPO .GT. 0) GO TO 810
000222  002        NSPRN = NDATA(L22+1)
000223  002        LPR = .FALSE.
000224  002        RPR = P(NSPRN)
000225  002        P(NSPRN) = 0.0
000226  002        CALL CMPRSS(0.0,NSPRN,EXT(L26+3),EXT(L27+1),NEXT(L28),$860)
```

```
000227  002          GO TO 920
000228  002      810 L60 = NDATA(L22)
000229  002          DO 820 J=1,L60
000230  002          NSPRN  = NDATA(L22+J)
000231  002          CALL CMPRSS(P(NSPRN),NSPRN,EXT(L26+3),EXT(L27+1),
000232  002         1 NEXT(L28),$860)
000233  002      820 CONTINUE
000234  002          LPR = .FALSE.
000235  002          RPR = 0.
000236  002          IF(NPO) 920,920,845
000237  002      C
000238  002      840 IF(NPO .LT. 1) GO TO 920
000239  002          LPR = .TRUE.
000240  002          RPR = P(NPO)
000241  002          P(NPO) = 0.0
000242  002      845 CALL CMPRSS(0.0,NPO,EXT(L26+3),EXT(L27+1),NEXT(L28),$850)
000243  002          GO TO 920
000244  002      C
000245  002      850 NSPRN = NPO
000246  002      860 CALL TOPLIN
000247  002          IF(NSPRN .GT. 0) NSPRN = NEXT(LP+NSPRN)
000248  002          WRITE(6,880) NSPRN, NDATA(L14+1)
000249  002      880 FORMAT(70H0* * *  ERROR IN LOCATING PRESSURE NODE WITH PRESSURE SP
000250  002         1ECIFIED  * * * // 8X 4HNODE I5, 27H WAS NOT FOUND FOR NETWORK A6)
000251  002          CALL WLKBCK
000252  002          CALL EXIT
000253  002      C
000254  002      C OUTLET PRESSURE INTO COEFFICIENT MATRIX AND RHS
000255  002      C
000256  002      C
000257  002      C SOLVE FOR PRESSURES
000258  002      C
000259  002      920 MPRN = NEXT(L28)
000260  002          NEXT(L26) = MPRN*(MPRN+1)/2 + 2
000261  002          NEXT(L26+1) = MPRN
000262  002          NEXT(L26+2) = MPRN
000263  002          IF(.NOT. COP) GO TO 930
000264  002          CALL LINECK(2)
000265  002          CALL GENOUT(1,1,0,'0MATRIX AFTER  REDUCTION')
000266  002          CALL LINECK(2)
000267  002          CALL GENOUT(EXT(L26+1),1,NEXT(L28)  ,'0PRESSURE NODES (RELATIVE)')
000268  002          CALL LINECK(2)
000269  006          CALL GENOUT (EXT(L26+3),1,NEXT(L26)-2,'0COEFFICIENT MATRIX')
000270  002          CALL LINECK(2)
000271  006          CALL GENOUT(EXT(L27+1),1,MPRN  ,'0RIGHT HAND SIDE')
000272  002          CALL LINECK(2)
000273  002      930 CALL SYMSOL(EXT(L26+3),MPRN,EXT(L27+1),$1020)
000274  002          IF(COP) CALL GENOUT(EXT(L27+1),1,NEXT(L28),'0PRESSURES')
000275  002          GO TO 1060
000276  002      C
000277  002     1020 CALL TOPLIN
000278  002          WRITE(6,1040) NDATA(L14+1)
000279  002     1040 FORMAT(61H0* * *  ERROR IN SOLVING PRESSURE/FLOW EQUATIONS FOR NET
000280  002         1WORK A6, 7H  * * *)
000281  002          CALL WLKBCK
000282  002          NTH = L25 - 1
000283  002          CALL OUTCAL
```

```
000284  002        CALL EXIT
000285  002  C
000286  002  C
000287  002  C UPDATE PRESSURES
000288  002  C
000289  006   1060 MPRN = MPRN - 2*NFSV
000290  006        DO 1080 J=1,MPRN
000291  002        NPR = NEXT(L28+J)
000292  002        P(NPR) = EXT(L27+J)
000293  002   1080 CONTINUE
000294  002  C
000295  002  C CALCULATE NEW FLOW RATES
000296  002  C
000297  002        DO 1120 J=4,L20,4
000298  002        K    = L14 + J
000299  002        NTB  = NDATA(K)
000300  002        NFRM = NDATA(K+1)
000301  002        NTO  = NDATA(K+2)
000302  002        NFRM = NEXT(L25+NFRM)
000303  002        NTO  = NEXT(L25+NTO )
000304  006        PNFRM = P(NFRM)
000305  006        GO TO LL (1085,1098)
000306  006   1085 L = LFSV + 4
000307  006        NPU = MPRN + 1
000308  006        DO 1090 I=1,NFSV
000309  006        IF(NTB .EQ. NEXT(L)) GO TO 1095
000310  006        NPU = NPU + 1
000311  006        IF(NTB .EQ. NEXT(L+1)) GO TO 1095
000312  006        NPU = NPU + 1
000313  006        L = L + 5
000314  006   1090 CONTINUE
000315  006        GO TO 1098
000316  006   1095 PNFRM = EXT(L27+NPU)
000317  006   1098 NFRM = NEXT(LP+NFRM)
000318  006        PNTO = P(NTO)
000319  006        NTO  = NEXT(LP+NTO )
000320  006        TEMP = GF(NTB)*(PNFRM-PNTO)
000321  002        WNTB = W(NTB)
000322  002        TEMP = WNTB + FROF*(TEMP-WNTB)
000323  002        W(NTB) = TEMP
000324  002        DWMX = AMAX1(ABS(TEMP/WNTB-1.0),DWMX)
000325  002  C
000326  002        IF(.NOT. COP) GO TO 1120
000327  002        CALL LINECK(3)
000328  004         GFNTB= GF(NTB)
000329  002        NTB = NEXT(LTB+NTB)
000330  009        WRITE(6,1100) NTB,NFRM,NTO,GFNTB,PNFRM,PNTO,WNTB,TEMP
000331  002   1100 FORMAT(/                  7X 7HNTB    = I10   , 8X 7HNFRM  = I10   ,
000332  002       1  8X 7HNTO    = I10   , 8X 7HGF     = G13.8/32X 7HP(NFRM) G13.8,
000333  002       2  5X 7HP(NTO)= G13.8, 5X 7HWOLD  = G13.8, 5X 7HWNEW  = G13.8)
000334  002   1120 CONTINUE
000335  006  C
000336  006        IF(NFSV .LT. 1) GO TO 1129
000337  006        DO 1127 J=4,L20,4
000338  006        K = L14 + J
000339  006        L = LFSV
000340  006        NTB = NDATA(K)
```

```
006          NPU = MPRN + 1
006          DO 1124 I=1,NFSV
006          IF(NTB .EQ. NEXT(L+4) .OR. NTB .EQ. NEXT(L+5)) GO TO 1126
006          NPU = NPU + 2
006          L = L + 5
006     1124 CONTINUE
006          GO TO 1127
006     1126 NPR = NEXT(L+3)
006          P(NPR) = AMAX1(EXT(L27+NPU),EXT(L27+NPU+1))
006     1127 CONTINUE
006     C
006     1129 DP = 0.0
002          IF(NPI .GT. 0 .AND. NPO .GT. 0) DP = P(NPI) - P(NPO)
002          IF(LPR) GO TO 1130
002          IF(L22 .LT. 1) GO TO 1160
002          IF(NDATA(L22) .NE. 1) GO TO 1160
002          P(NSPRN) = RPR
002          GO TO 1135
002     1130 P(NPO) = RPR
002     1135 DO 1140 J=1,MPRN
002          NPR = NEXT(L28+J)
002          P(NPR) = P(NPR) + RPR
002     1140 CONTINUE
002     1160 NTH = L25 - 1
002          RETURN
002          END
```

END ELT.

@HDG,P FLOCOM

```
*ELT,L FLOCOM
ELTOT7 RLIBTO 02/28-03:19:23-(1,)
000001      000          SUBROUTINE FLOCOM
000002      000    C
000003      000          LOGICAL LCOPY
000004      000    C
000005      000          DIMENSION LABEL(10), NAME(10), NSIZE(10)
000006      000    C
000007      000          COMMON /CROBLK/ LSTART, LECARD, LCOPY
000008      000          COMMON /FDIMNS/ NTYPE , NSYS  , NTB  , NP, NV, NFD
000009      000          COMMON /SRDCOM/ ZBY(236), NRNSFR(22)
000010      000    C
000011      000          DATA LABEL / 6HFLOBAT, 6HSYSDAT, 6HTYPDAT, 6HWDOT  , 6HPRESS ,
000012      000         1             6HFLOWG , 6HVALVP , 6HWDOTI , 6HFLOWR , 6HDELTAP /
000013      000    C
000014      000          DATA NAME  / 6HFLOW  , 6HSYSTEM, 6HTYPE  , 6HW     , 6HP     ,
000015      000         1             6HGF    , 6HVP    , 6HWI    , 6HAFR   , 6HDP    /
000016      000    C
000017      000    C
000018      000          NSIZE( 1) = MAX0(1,NFD  )
000019      000          NSIZE( 2) = MAX0(1,NSYS )
000020      000          NSIZE( 3) = MAX0(1,NTYPE)
000021      000          NSIZE( 4) = MAX0(1,NTB  )
000022      000          NSIZE( 5) = MAX0(1,NP   )
000023      000          NSIZE( 6) = NSIZE(4)
000024      000          NSIZE( 7) = MAX0(1,NV   )
000025      000          NSIZE( 8) = NSIZE(5)
000026      000          NSIZE( 9) = NSIZE(4)
000027      000          NSIZE(10) = NSIZE(4)
000028      000          NC = 15
000029      000          DO 500 I=1,10
000030      000          CALL CLRVEC
000031      000          IF(I .LT. 2 .OR. I .GT. 3) GO TO 200
000032      000          ENCODE(100,NRNSFR) LABEL(I), NAME(I), NC, NSIZE(I)
000033      000      100 FORMAT(6X 8HCOMMON / A6, 2H/  A6, 1H( I2, 1H, I6, 1H) 45X )
000034      000          NC = 10
000035      000          GO TO 400
000036      000      200 ENCODE(300,NRNSFR) LABEL(I), NAME(I), NSIZE(I)
000037      000      300 FORMAT(6X 8HCOMMON / A6, 2H/  A6, 1H( I6, 1H) 48X )
000038      000      400 CALL BLKCRD
000039      000      500 CONTINUE
000040      000          CALL CLRVEC
000041      000          ENCODE(600,NRNSFR)
000042      000      600 FORMAT(6X 45HCOMMON /FDIMNS/ NTYPE, NSYS, NTB, NP, NV, NFD 33X )
000043      000          CALL BLKCRD
000044      000          RETURN
000045      000          END

END ELT.


*NDG,P FLOP
```

```
*ELT,L FLOP
ELTOT7 RL1870 02/28-03:19:24-(1,)
000001      000          SUBROUTINE FLOP(N,L,R,I,J)
000002      000    C
000003      000          LOGICAL L, ERR
000004      000    C
000005      000          DIMENSION SYSTEM(15,1), RDATA(1)
000006      000    C
000007      000          COMMON /SYSDAT/ NSYSTM(15,1)
000008      000          COMMON /ARRAY / NDATA (1)
000009      000          EQUIVALENCE (SYSTEM(1,1),NSYSTM(1,1)), (RDATA(1),NDATA(1))
000010      000    C
000011      000          DATA MAX1 / 99999/
000012      000          DATA NOUT / 6 /
000013      000    C
000014      000    C
000015      000          N = NSYSTM(I,J)
000016      000          L = .TRUE.
000017      000          IF(N .GT. 0 .AND. N .LT. MAX1) GO TO 20
000018      000          R = SYSTEM(I,J)
000019      000          GO TO 40
000020      000       20 IC = NDATA(N)
000021      000          IF(MOD(IC,2) .EQ. 0) GO TO 30
000022      000          WRITE(NOUT,25) N, IC
000023      000       25 FORMAT(70H0* * *  WRONG ARRAY LENGTH FOR FLUID PROPERTY CURVE STOR
000024      000         1ED AT LOCATION  I6,  6H, IC =  I6, 7H  * * * /)
000025      000          ERR = .TRUE.
000026      000          GO TO 50
000027      000       30 IF(IC .NE. 2) GO TO 50
000028      000          R = RDATA(N+2)
000029      000       40 L = .FALSE.
000030      000       50 RETURN
000031      000          END

END ELT.


*HDG,P FLOPLT
```

```
*ELT,L FLOPLT
ELTOT7 RL1870 02/28-03:19:25-(5,)
000001      001          DIMENSION NX(1), MVF(26), NVF(11), NFC(3)
000002      001          DIMENSION YLO(75),YHI(75),ORD(1),BUFR(4000),XY(33000)
000003      001          INTEGER  TITLEA(12),TITLEB(20),TITLEC(20),TITLES(9,75)                GEPLT003
000004      001         1 ,ITYTLS(9,75),BCDX(4),BCDY(4,11),ITEM(75),ITYPE(75)                  GEPLT004
000005      001         2 ,IGS(76),KEYA(11),KEYB(12),BLANK                                     GEPLT005
000006      001         3 ,TMESCL(3)
000007      001          DIMENSION  LOC(76),ABS(1)                                             GEPLT006
000008      001          DIMENSION ITMAVG(50), AVG(150), AVGLOC(100), HDR(12)
000009      001    C                                                                           GEPLT007
000010      001          COMMON  NPTS,TPG,BUFR                                                 GEPLT008
000011      001          COMMON /XYARY/ XY
000012      002          DIMENSION NSP(11)
000013      001          EQUIVALENCE (BUFR(1),ABS(1)),(BUFR(2001),ORD(1)),                     GEPLT009
000014      001         1            (TITLES(1,1),ITYTLS(1,1))                                 GEPLT010
000015      001          EQUIVALENCE (NX,XY)
000016      001    C                                                                           GEPLT011
000017      001    C               ARRAY DEFINITIONS                                           GEPLT012
000018      001    C                                                                           GEPLT013
000019      001    C ABS     - ABSCISSA VALUES FOR THE CURRENT FRAME                           GEPLT014
000020      001    C BCDX    - ITEM NAMES AND DIMENSION INFORMATION ON THE ABCISSA             GEPLT015
000021      001    C BCDY    - ITEM NAMES AND DIMENSION INFORMATION ON THE ORDINATE            GEPLT016
000022      001    C BUFR    - BUFFER FOR READING HISTORY TAPE RECORDS                         GEPLT017
000023      001    C IGS     - ARRAY FOR STORING THE ITEM TYPE INDICES                         GEPLT018
000024      001    C ...M.   - THE ITEM NUMBERS TO BE PLOTTED                                  GEPLT019
000025      001    C ITYPE   - THE ITEM TYPES FOR THE RESPECTIVE ITEM NUMBERS                  GEPLT020
000026      001    C KEYA    - ITEM TYPE CODE ARRAY                                            GEPLT021
000027      001    C KEYB    - INDEX TO ITEM TYPE IN BUFR ARRAY                                GEPLT022
000028      001    C LOC     - INDEX TO ITEM ON EACH TIME RECORD (ERROR CODE IF NEGATIVE)      GEPLT023
000029      001    C ORD     - ORDINATE VALUES FOR THE CURRENT FRAME                           GEPLT024
000030      001    C TITLEA  - GENERAL TITLE FOR EACH FRAME                                    GEPLT025
000031      001    C TITLEB  - TITLE OF 1-ST AND 2-ND ITEMS ON THE CURRENT FRAME               GEPLT026
000032      001    C TITLEC  - TITLE OF 3-RD AND 4-TH ITEMS ON THE CURRENT FRAME               GEPLT027
000033      001    C TITLES  - THE ITEM PLOTTING SYMBOLS AND DESCRIPTIONS                      GEPLT028
000034      001    C XY      - ARRAY FOR ITEMS TO BE PLOTTED (INCLUDING TIME)                  GEPLT029
000035      001    C YHI     - THE MAXIMUM ORDINATE VALUES                                     GEPLT030
000036      001    C YLO     - THE MINIMUM ORDINATE VALUES                                     GEPLT031
000037      001    C                                                                           GEPLT032
000038      001    C               WORD DEFINITIONS                                            GEPLT033
000039      001    C                                                                           GEPLT034
000040      001    C ITEMS   - THE NUMBER OF ITEMS PER TIME RECORD FOR PLOTTING - MAX = 75     GEPLT035
000041      001    C NGRDS   - THE NUMBER OF GRIDS REQUIRED TO SPAN THE RANGE (TZ - TA)        GEPLT036
000042      001    C NSIZE   - THE NUMBER OF WORDS ALLOTTED TO THE XY ARRAY                    GEPLT037
000043      001    C NTOTL   - NUMBER OF WORDS USED IN THE XY ARRAY                            GEPLT038
000044      001    C NTYMS   - THE NUMBER OF POINTS TO BE PLOTTED ( = NSIZE/ITEMS)             GEPLT039
000045      001    C NWRDS   - THE NUMBER OF ITEMS PER TIME RECORD ON THE HISTORY TAPE         GEPLT040
000046      001    C                                                                           GEPLT041
000047      001    C               INITIALIZATION                                              GEPLT042
000048      001    C                                                                           GEPLT043
000049      001          DATA (BCDX(I),I=1,4) /24H     TIME - (******)     /
000050      001          DATA (TMESCL(I),I=1,3) /18H( SEC ( MIN (HOURS/
000051      001          DATA NVF/'(38X,6HLOADED,F11.5,22H ******    LOOKING FOR,F11.5,6H**
000052      001         1*****)'/
000053      001          DATA NFC/' SEC.  MIN.  HRS. '/
000054      001          DATA MVF/'(1H1,48X,23HP L O T   P R O G R A M//24X,8HTITLE  -,2X,
000055      001         112A6///28X,5HFROM ,F10.3,10H****** TO  , F10.3,  13H*******, WITH ,
```

```
000056  001       2F10.3, 15H****** PER GRID//////  )'/
000057  001  C                                                          GEPLT045
000058  001        DATA ((BCDY(I,J),I=1,4),J=1,11) /
000059  001       1 24H                         ,
000060  001       2 24H     PRESSURE DROP       ,
000061  001       3 24H        PRESSURE         ,
000062  001       4 24H  VALVE POSITION DATA    ,
000063  001       5 24H                         ,
000064  001       6 24H                         ,
000065  001       7 24H                         ,
000066  001       8 24H       FLOW RATE         ,
000067  001       9 24H      TEMPERATURE        ,
000068  001       X 24H      TEMPERATURE        ,
000069  001       1 24H      TEMPERATURE         /
000070  001  C                                                          GEPLT052
000071  001  C                                                ITEM TYPE CODESGEPLT053
000072  001        DATA KEYA / 2HAA, 2HDP, 2HPR, 2HVP, 2HBB, 2HCC,
000073  001       1            2HBT, 2HFR, 2HFT, 2HTT, 2HST      /          GEPLT055
000074  001  C                                                          GEPLT056
000075  001        DATA BLANK /6H      /                                 GEPLT057
000076  001  C                                                          GEPLT050
000077  001        DATA 01, 02 /0007700000000, 0050005050505 /
000078  001  C                                                          GEPLT108
000079  001        INTEGER  PSYM(4)                                      GEPLT109
000080  001        DATA  (PSYM(I),I=1,4)/24H[1   [2   [3   [4   /       GEPLT110
000081  001        EXTERNAL  TABL1V                                      GEPLT111
000082  001        CALL CHSIZV(2,2)                                      GEPLT112
000083  001        CALL BITSTV(12,10,TABL1V)                             GEPLT113
000084  001  C                                                          GEPLT114
000085  001        CALL RESET                                            GEPLT115
000086  001        NCASE = 1                                             GEPLT116
000087  001        NSIZE=33000
000088  001        GO TO 60                                              GEPLT118
000089  001     20 CALL CLOCK(ETIME)                                     GEPLT119
000090  001        WRITE (6,40) ETIME                                    GEPLT120
000091  001     40 FORMAT(////11X,'COMPUTER TIME = ',F10.5,'   MINUTES') GEPLT121
000092  001  C                                READ THE CASE CARDS AND PRINT THE HEADINGGEPLT122
000093  001     60 READ(5,80,END=160) TITLEA, TA, TZ, TPG, ITMX, MPNT, NTP, KT, INC,
000094  001       1 IUNIT, ASTAT, ASTOP
000095  001     80 FORMAT(12A6/3F10.0, 6I5, 2F10.0)
000096  001        IF(KT .LT. 1) KT = 23
000097  001  C                                        CHECK FOR COMBINE OPTIONGEPLT126
000098  001        IF (NTP.EQ.0) GO TO 100                               GEPLT127
000099  001        CALL COMBIN(NTP,KT,INC,IUNIT)                         GEPLT128
000100  001        CALL CLOCK(ETIME)                                     GEPLT129
000101  001        WRITE (6,40) ETIME                                    GEPLT130
000102  001    100 IHSTRY=KT                                             GEPLT131
000103  001  C
000104  001  C                                      CHECK FOR BLANK - END OF JOBGEPLT138
000105  001        IF (TPG) 160,160,180                                  GEPLT139
000106  001    160 CALL EXIT                                             GEPLT140
000107  001  C                                                          GEPLT141
000108  001    180 IF(ITMX .LT. 1 .OR. ITMX .GT. 3) GO TO 200
000109  001        BCDX(3) = TMESCL(ITMX)
000110  001        NVP( 5) = NFC(ITMX)
000111  001        NVP(10) = NFC(ITMX)
000112  001        NVP(15) = NFC(ITMX)
```

```
000113  001        MVF(19) = NFC(ITMX)
000114  001        MVF(23) = NFC(ITMX)
000115  001     200 WRITE(6,MVF) TITLEA, TA, TZ, TPG
000116  001   C                                                              GEPLT154
000117  001   C                        READ AND PRINT THE HISTORY TAPE HEADER LABELGEPLT155
000118  002     240 READ (IHSTRY) HDR,(LOC(I),I = 13,16),NSP(1),NW,NPR,NV,
00119   004        1 (NSP(I),I=5,10), NSL,
000120  004        2 (NX(NSIZE-2*NW-NPR-NV-NSL+I),I=1,NW),
000121  004        3 (NX(NSIZE-NW-NPR-NV-NSL+I),I=1,NPR),
000122  004        4 (NX(NSIZE-NW-NV-NSL+I),I=1,NV),
000123  004        5 (NX(NSIZE-NSL+I),I=1,NSL)
000124  002         NSP(2) = NW
000125  002         NSP(3) = NPR
000126  002         NSP(4) = NV
000127  002         NSP(11) = NSL
000128  001         LOC1 = NSIZE - 2*NW- NPR - NV - NSL
000129  001         DO 250 I=1,NW
000130  004         NX(NSIZE-NW-NSL+I) = NX(LOC1+I)
000131  001     250 CONTINUE
000132  001         NSIZE = LOC1
000133  001   C                          INDEX AND COUNT THE ITEMS ON THE HISTORY TAPEGEPLT157
000134  001         KEYB(1) = 1                                                GEPLT158
000135  001         DO 260 I=2,12                                              GEPLT159
000136  001     260 KEYB(I) = KEYB(I-1) + NSP(I-1)
000137  001         NWRDS = KEYB(12)                                           GEPLT161
000138  001         IF (NCASE.NE.1) GO TO 300                                  GEPLT162
000139  004         WRITE(6,280)HDR,(NSP(I),KEYA(I),I=1,11)
000140  001     280 FORMAT(52X,25HTHE HISTORY TAPE LABEL IS//29X,12A6///18X,22HTHE ITEGEPLT164
000141  001        1M COUNTS ARE - , 6(I6,A2) / 40X 5(I6,A2)/////)             GEPLT165
000142  001   C                                                              GEPLT166
000143  001     300 IFINIS = 0                                               GEPLT167
000144  001         JFINIS = 0                                               GEPLT168
000145  001   C                                    READ THE ITEMS TO BE PLOTTEDGEPLT172
000146  001         ITEMS = 76                                               GEPLT173
000147  001         I = 1                                                    GEPLT174
000148  001         J = 0                                                    GEPLT175
000149  001         NOAVG = 0
000150  001         KSW = 0
000151  001     320 READ(5,340) ITEM(I),ITYPE(I),IREL,KAVG,
000152  004        1 (TITLES(JJ,I),JJ=2,9),YLO(I),YHI(I)
000153  001     340 FORMAT(I5,A2,I1,I2,8A6,2X2F10.0)
000154  001   C                                  TEST FOR END OF JOB  -  BLANK CARDGEPLT179
000155  001         IF (ITEM(I) .EQ. 0) GO TO 20                             GEPLT180
000156  001         IF (ITEM(I) .EQ. 0) GO TO 360                            GEPLT181
000157  001   C
000158  001         K = I + 1
000159  004         DO 341 JJ=1,11
000160  004         IF(ITYPE(I).EQ.KEYA(JJ)) GO TO 342
000161  001     341 CONTINUE
000162  004         WRITE(6,440) I,ITEM(I),ITYPE(I)
000163  001         LOC(K) = -1
000164  001         GO TO 320
000165  001     342 IACT = IABS(ITEM(I))
000166  004         NS = NSP(JJ)
000167  004         LOC2 = LOC1+KEYB(JJ)-1
000168  001         DO 343 L=1,NS
000169  001         IF(NX(LOC2+L) .EQ. IACT) GO TO 344
```

```
000170  001    343 CONTINUE
000171  004        WRITE(6,500) I,ITEM(I),ITYPE(I)
000172  001        LOC(K) = -2
000173  001        GO TO 320
000174  004    344 LOC(K) = KEYB(JJ)+L
000175  005        IGS(I) = JJ
000176  004        WRITE(6,510) I,ITEM(I),ITYPE(I),LOC(K)
000177  001  C                                  CHECK FOR NEW GRID SET SPECIFIED BY USERGEPLT182
000178  001        IF (ITEM(I) .LT. 0) J = 0                                          GEPLT183
000179  001        IF (ITEM(I) .LT. 0) KSW = 0
000180  001  C                                  PUT BCD PLOTTING SYMBOL INTO TITLES ARRAYGEPLT184
000181  001        J = J+1                                                            GEPLT185
000182  001        ITYTLS(1,I) = BLANK                                                GEPLT186
000183  001        FLD(30,6,ITYTLS(1,I)) = J + KSW + 48
000184  001        IF(NAVG .EQ. 0 .OR. NOAVG .GE. 50) GO TO 345
000185  001        NOAVG = NOAVG + 1
000186  001        ITMAVG(NOAVG) = I
000187  001        IF(NAVG .LT. 10) GO TO 345
000188  001        ITMAVG(NOAVG) =-I
000189  001        FLD(0,30,ITYTLS(1,I)) = 6H YES
000190  001        KSW = KSW + 1
000191  001  C                  BUMP ITEM COUNTER AND CHECK FOR MAXIMUM NUMBER OF ITEMSGEPLT188
000192  001    345 I = I + 1
000193  001        IF(I+1 .LT. ITEMS) GO TO 320
000194  001    360 ITEMS = I                                                          GEPLT191
000195  001        DO 370 L=1,NOAVG
000196  001        AVG(L) = 0.
000197  001    370 CONTINUE
000198  001  C                                        SET FIRST ITEM FOR NEW GRID SETGEPLT192
000199  001        ITEM(1) = -IABS(ITEM(1))                                           GEPLT193
000200  001  C                                    FIND THE TYPE CODE IN THE KEYA ARRAYGEPLT194
000201  001    380 LOC(1) = 1                                                         GEPLT195
000202  004    440 FORMAT(41X15,5X4HITEM18,A2,20H  TYPE CODE IN ERROR )
000203  004    500 FORMAT(41X15,5X4HITEM18,A2,17H  IS OUT OF RANGE )
000204  004    510 FORMAT(41X15,5X4HITEM18,A2, 4H  AT I6)
000205  001  C                                                                       GEPLT222
000206  001  C                                START LOADING THE DATA FROM THE HISTORY TAPEGEPLT223
000207  001  C                                                                       GEPLT224
000208  001  C                                     COMPUTE THE MAXIMUM NUMBER OF RECORDSGEPLT225
000209  001    560 NTYMS = NSIZE/ITEMS                                              GEPLT226
000210  001        WRITE (6,580)                                                    GEPLT227
000211  001    580 FORMAT(1H1,44X,40HPOSITIONING AND READING THE HISTORY TAPE/)     GEPLT228
000212  001  C                                              POSITION THE HISTORY TAPEGEPLT229
000213  001        NTPTS = 0
000214  001        I = 1                                                            GEPLT230
000215  001        J = 1                                                            GEPLT231
000216  001    600 READ (IHSTRY) (BUFR(L),L=1,NWRDS)                                GEPLT232
000217  001  C                                          CHECK FOR END OF DATA FILEGEPLT233
000218  001        IF (BUFR(1).LT.0.0) GO TO 780                                    GEPLT234
000219  001  C                                       CHECK FOR REQUESTED START TIMEGEPLT235
000220  001        IF (BUFR(1).LT.TA) GO TO 620                                     GEPLT236
000221  001        GO TO 660                                                        GEPLT237
000222  001    620 IF(MPNT .EQ. 1) WRITE(6,NVF) BUFR(1), TA
000223  001        GO TO 600
000224  001    660 IF(MPNT .EQ. 1) WRITE(6,NVF) BUFR(1), TZ
000225  001  C                                       CHECK FOR REQUESTED FINAL TIMEGEPLT242
000226  001        IF (BUFR(1) .GT. TZ) IFINIS = 1                                  GEPLT243
```

```
000227  001  C                                  PICK UP THE ITEM/ITYPE ARRAY QUANTITIESGEPLT244
000228  001       DO 740 L=1,ITEMS                                                    GEPLT245
000229  001       M = LOC(L)                                                          GEPLT246
000230  001  C                                            CHECK FOR ERROR ITEMGEPLT247
000231  001       IF (M .LT. 0) GO TO 740                                             GEPLT248
000232  001     680 XY(J) = BUFR(M)                                                   GEPLT255
000233  001  C                                         BUMP THE XY ARRAY SUBSCRIPTGEPLT262
000234  001     740 J = J+1                                                         GEPLT263
000235  001       IF(NOAVG .EQ. 0) GO TO 752
000236  001       IF(BUFR(1) .LT. ASTRT-.0005      BUFR(1) .GT. ASTOP+.0005)
000237  001      X   GO TO 752
000238  001       NTPTS = NTPTS + 1
000239  001       STOP = BUFR(1)
000240  001       DO 749 L=1,NOAVG
000241  001       MM = IABS(ITMAVG(L))
000242  001       M = LOC(MM+1)
000243  001       IF(M .LT. 0) GO TO 749
000244  001       M = MM + J - ITEMS
000245  001       AVG(L) = AVG(L) + XY(M)
000246  001       IF(NTPTS .GT. 1) GO TO 743
000247  001       ISTART = I - 1
000248  001       START = BUFR(1)
000249  001       AVGLOC(L     ) = BUFR(1)
000250  001       AVGLOC(L+ 50) = BUFR(1)
000251  001       AVG   (L+ 50) = XY(M)
000252  001       AVG   (L+100) = XY(M)
000253  001       GO TO 749
000254  001     743 IF(XY(M) .LE. AVG(L+ 50)) GO TO 746
000255  001       AVGLOC(L     ) = BUFR(1)
000256  001       AVG   (L+ 50) = XY(M)
000257  001       GO TO 749
000258  001     746 IF(XY(M) .GE. AVG(L+100)) GO TO 749
000259  001       AVGLOC(L+ 50) = BUFR(1)
000260  001       AVG   (L+100) = XY(M)
000261  001     749 CONTINUE
000262  001     752 LJ = J - ITEMS + 1
000263  001       LJI = J - 1                                                         GEPLT265
000264  001       IF(MPNT .EQ. 1) WRITE(6,760) (XY(L),L=LJ,LJI)
000265  001     760 FORMAT (10F11.3)                                                  GEPLT267
000266  001       IF (IFINIS .EQ. 1) GO TO 800                                        GEPLT268
000267  001       I = I+1                                                             GEPLT269
000268  001  C                                          CHECK FOR MAXIMUM NUMBER OF POINTSGEPLT270
000269  001       IF (I .LE. NTYMS) GO TO 600                                         GEPLT271
000270  001     780 NTYMS = I-1                                                       GEPLT272
000271  001       GO TO 820                                                           GEPLT273
000272  001     800 NTYMS = I                                                         GEPLT274
000273  001  C                    COMPUTE THE NUMBER OF WORDS USED IN THE XY ARRAYGEPLT275
000274  001     820 NTOTL = J-1                                                       GEPLT276
000275  001       REWIND IHSTRY                                                       GEPLT277
000276  001       WRITE(6,840) ITEMS, I, NTOTL                                        GEPLT278
000277  001     840 FORMAT (1H0,I10,42H  DATA VALUES HAVE BEEN STORED FOR EACH OF,I6, GEPLT279
000278  001        1 13H  TIME POINTS/1X,I10,30H  DATA VALUES HAVE BEEN STORED)      GEPLT280
000279  001  C
000280  001       IF(NTPTS .EQ. 0) GO TO 852
000281  001       WRITE(6,843) HDR, NTPTS, ASTRT, ASTOP, START, STOP
000282  001     843 FORMAT(1H1 12A6/
000283  001        X 'OTHE NUMERICAL AVERAGES FOR THE FOLLOWING ITEMS WERE REQUESTED'
```

```
000284  001       X ' FOR THE' I4,' TIME POINTS'/' BEGINNING WITH'F7.3,
000285  001       X ' HRS., AND ENDING WITH' F7.3,' HRS '/' ACTUAL TIMES -' F7.3,
000286  001       X ' HRS., AND            ' F7.3,' HRS.'///  ITEM  TYPE     DESCRIPT .
000287  001       XION' 41X 'AVERAGE'7X'MAX VALUE'5X'TIME'7X'MIN VALUE'5X'TIME'/)
000288  001        TPTS = NTPTS
000289  001        DO 849 L=1,NOAVG
000290  001        M = IABS(ITMAVG(L))
000291  001        MM = LOC(M+1)
000292  001        IF(MM .LT. 0) GO TO 849
000293  001        AVG(L) = AVG(L)/TPTS
000294  004        WRITE(6,846)ITEM(M),ITYPE(M),(TITLES(J,M),J=2,9),AVG(L),
000295  001       X AVG(L+50), AVGLOC(L), AVG(L+100), AVGLOC(L+50)
000296  001     846 FORMAT(1X15, 4XA2, 5X8A6, 1XF10.2, 2(6XF10.2, 2XF7.3))
000297  001     849 CONTINUE
000298  001     852 CONTINUE
000299  001   C                                  FIND THE MAXIMUM AND MINIMUM ORDINATESGEPLT404
000300  001     920 I = 1                                                          GEPLT405
000301  001     940 J = I                                                          GEPLT406
000302  001   C                                        PICK UP THE INPUT VALUESGEPLT407
000303  001        YB = YLO(I)                                                     GEPLT408
000304  001        YT = YHI(I)                                                     GEPLT409
000305  001        LYS = -1                                                        GEPLT410
000306  001        IF (YT-YB) 960,960,980                                          GEPLT411
000307  001     960 YB = 1.E10                                                     GEPLT412
000308  001         YT = -1.E10                                                    GEPLT413
000309  001     980 K = I+1                                                        GEPLT414
000310  001   C                                           CHECK FOR ERROR ITEMSGEPLT415
000311  001        IF (LOC(K) .LT. 0) GO TO 1020                                   GEPLT416
000312  001        IF ((YHI(I)-YLO(I)).GT.0.01) LYS = 1                            GEPLT417
000313  001   C                                      COMPARE WITH THE TAPE VALUESGEPLT418
000314  001        DO 1000 L=K,NTOTL,ITEMS                                         GEPLT419
000315  001        YB = AMIN1(XY(L),YB)                                            GEPLT420
000316  001     1000 YT = AMAX1(XY(L),YT)                                          GEPLT421
000317  001   C                                         CHECK FOR LAST ITEMSGEPLT422
000318  001     1020 I = I+1                                                       GEPLT423
000319  001        IF (I .LT. ITEMS) GO TO 1040                                    GEPLT424
000320  001        JFINIS = 1                                                      GEPLT425
000321  001        GO TO 1060                                                      GEPLT426
000322  001   C                                       CHECK FOR NEW GRID SETGEPLT427
000323  001     1040 IF (ITEM(I) .GT. 0) GO TO 980                                 GEPLT428
000324  001     1060 YLO(J) = YB
000325  001        YHI(J) = YT                                                     GEPLT435
000326  001        IF (JFINIS .EQ. 0) GO TO 940                                    GEPLT436
000327  001   C                                    PRINT THE ITEMS TO BE PLOTTEDGEPLT437
000328  001        WRITE (6,1100)                                                  GEPLT438
000329  001     1100 FORMAT(1H1 14X 'ITEM TYPE' 6X 'AVG  PLOTTING SYMBOL AND'
000330  001         X ' DESCRIPTION' 28X 'Y-MIN      Y-MAX     STATUS'/)
000331  001        JJ = ITEMS - 1
000332  001        DO 1200 I=1,JJ
000333  004        WRITE(6,1140)I,ITEM(I),ITYPE(I),(TITLES(J,I),J=1,9),YLO(I),
000334  001       1                  YHI(I), LOC(I+1)                            GEPLT446
000335  001     1140 FORMAT(4X15,5X15,2XA2,7XA6,1X8A6,6X1P2E11.3,I7)
000336  001        FLO(6,6,ITYTLS(1,I)) = FLO(30,6,ITYTLS(1,I))
000337  001        ITYTLS(1,I) = OR(AND(ITYTLS(1,I),01),02)
000338  001     1200 CONTINUE
000339  001   C                                                                   GEPLT453
000340  001   C                                                                   GEPLT454
```

```
000341  001  C                                                    START THE PLOTTINGGEPLT455
000342  001   1220 WRITE (6,1240)                                                  GEPLT456
000343  001   1240 FORMAT(1H1,58X,14HSTARTING PLOTS/)                             GEPLT457
000344  001  C                                          COMPUTE THE NUMBER OF GRIDS REQUIREDGEPLT458
000345  001       NGRDS = .9999 + (TZ-TA)/TPG                                     GEPLT459
000346  001  C                                        INITIALIZE THE ABSCISSA LIMITSGEPLT460
000347  001       IR = 1                                                          GEPLT461
000348  001       ABSR = TA                                                       GEPLT462
000349  001  C                                             CENTER THE CASE TITLEGEPLT463
000350  001       NCA = NBLANK (TITLEA,12)                                        GEPLT464
000351  001       NPA = 590 - 6*NCA                                               GEPLT465
000352  001       NPX = 276
000353  001       NCB = 54
000354  001  C                                           START THE GRID SET LOOPGEPLT466
000355  001       DO 1620 I=1,NGRDS                                               GEPLT467
000356  001       KFINIS = 0                                                      GEPLT468
000357  001  C                                         SET THE LEFT-HAND LIMITSGEPLT469
000358  001       IL = IR                                                         GEPLT470
000359  001       ABSL = ABSR                                                     GEPLT471
000360  001       ABSR = ABSL+TPG                                                 GEPLT472
000361  001       ISW = 1
000362  001       NAVG = 0
000363  001       CALL DXDYV(1,ABSL,ABSR,DELX,NEX,LABX,NUMX,30.0,IERX)
000364  001       ZBSL = INT((ABSL-DELX)/DELX)*DELX
000365  001       ZBSR = INT((ABSR+DELX)/DELX)*DELX
000366  001       IF(ZBSL+DELX .LE. ABSL) ZBSL = ZBSL + DELX
000367  001       IF(ZBSR-DELX .GE. ABSR) ZBSR = ZBSR - DELX
000368  001       IF(ABSR .GE. 10.) NUMX=NUMX-1
000369  001  C                                            LOAD THE ABSCISSA VALUESGEPLT473
000370  001  C                                               SET THE ITEM COUNTERGEPLT484
000371  001       J = 1                                                           GEPLT485
000372  001  C                                             SET THE CURVE COUNTERGEPLT486
000373  001   1300 K = 1                                                         GEPLT487
000374  001  C                                          CLEAR THE SUBTITLE ARRAYSGEPLT488
000375  001       DO 1320 L=1,20                                                  GEPLT489
000376  001       TITLEB(L) = BLANK                                               GEPLT490
000377  001   1320 TITLEC(L) = BLANK                                             GEPLT491
000378  001       CALL FILMAV(0)                                                  GEPLT492
000379  001       JC = J                                                          GEPLT494
000380  001   1340 DO 1360 L=1,9                                                 GEPLT495
000381  001   1360 TITLEB(L) = TITLES(L,JC)                                      GEPLT496
000382  001       NC = FLD(6,6,ITYTLS(1,JC)) - 48
000383  001       IF (NC.GT.4) GO TO 1380  * TOO MANY CURVES                      GEPLT498
000384  001       IF (LOC(JC+1).LT.0) GO TO 1380  * ERROR ITEM                    GEPLT499
000385  001       FLD(0,30,TITLEB(1)) = FLD(0,30,PSYM(NC))
000386  001       NPY = 1005 - NC*18                                              GEPLT503
000387  001       NMAR = NC
000388  001  C                                                 WRITE THE SUBTITLESGEPLT504
000389  001       CALL WRITE2V(NPX,NPY,1023,90,1,NCB,1,TITLEB,NL)                 GEPLT505
000390  001   1380 JC = JC + 1                                                   GEPLT506
000391  001       IF (ITEM(JC).GT.0) GO TO 1340                                   GEPLT507
000392  001       NPY = 1024 -(NPY-9)                                             GEPLT508
000393  001       NMAR = (NMAR + 1)*18
000394  001       IF(NOAVG .GT. 0) NMAR = NMAR + 18
000395  001  C                                                    DRAW THE GRIDGEPLT509
000396  001  C        SUBROUTINE DXDYV CALCULATES CERTAIN ARGUMENTS FOR GRIDV, SUCH AS
000397  001  C        THE INCREMENTS FOR LINE SPACING DELX AND DELY.  THE FOLLOWING
```

```
000398  001  C        PROCEDURE ADJUSTS THE  MAX AND MIN LIMITS OF THE GRID TO
000399  001  C        INSURE THAT THEY ARE INTEGRAL MULTIPLES OF THE INCREMENTS
000400  001         CALL DXDYV(2,YLO(J),YHI(J),DELY,NEY,LABY,NUMY,30.0,IERY)
000401  001         YLOJ = INT((YLO(J) - DELY)/DELY)*DELY
000402  001         YHIJ = INT((YHI(J) + DELY)/DELY)*DELY
000403  001         IF(YLOJ + DELY .LE. YLO(J)) YLOJ = YLOJ + DELY
000404  001         IF(YHIJ - DELY .GE. YHI(J)) YHIJ = YHIJ - DELY
000405  001         YLO(J) = YLOJ
000406  001         YHI(J) = YHIJ
000407  001         CALL SETMIV(140,20,50,NMAR)
000408  001         CALL SETCIV(12,18)
000409  001         IF(LABY.EQ.10) LABY = 5
000410  001         LABY = -LABY
000411  001         CALL GRID1V(2,ZBSL,ZBSR,YLO(J),YHI(J),DELX,DELY,NEX,NEY,LABX,
000412  001        1      LABY,NUMX,NUMY)
000413  001  C                                                        LABEL THE AXESGEPLT511
000414  001         L = IGS(J)                                                    GEPLT512
000415  001         CALL RITE2V(456,9,1023,90,1,24,1,BCDX,NL)                     GEPLT513
000416  001         CALL RITE2V(92,300,1023,180,1,24,1,BCDY(1,L),NL)              GEPLT514
000417  001  C                                                  WRITE THE CASE TITLEGEPLT515
000418  001         CALL RITE2V(NPA,1005,1023,90,1,NCA,1,TITLEA,NL)               GEPLT516
000419  001  C                                               CHECK FOR TOO MANY CURVESGEPLT517
000420  001   1400 IF(K.GT.9) GO TO 1440
000421  001  C                                                  CHECK FOR ERROR ITEMGEPLT519
000422  001         IF(LOC(J+1).LT.0) GO TO 1480
000423  001  C                                        LOAD THE RESPECTIVE ORDINATE VALUESGEPLT521
000424  001         IF(ISW .EQ. 0) GO TO 1409
000425  001         JJ = 0
000426  001         DO 1403 KK=IL,NTYMS
000427  001         L = ITEMS( KK-1) + 1
000428  001         IR=KK
000429  001         IF(XY(L) .L.. ABSR) GO TO 1406
000430  001         JJ = JJ + 1
000431  001   1403 ABS(JJ) = XY(L)
000432  001   1406 NPTS = JJ
000433  001         ISW = 0
000434  001   1409 DO 1420 L=1,NPTS
000435  001         M = ITEMS*(IL+L-2)+J+1                                        GEPLT523
000436  001   1420 ORD(L) = XY(M)                                                 GEPLT524
000437  001  C                        BRANCH TO THE APPROPRIATE PLOTIV SUBROUTINE CALLGEPLT525
000438  001         CALL GOPLOT(K)                                                GEPLT526
000439  001         GO TO 1520
000440  001  C                                                     TOO MANY CURVESGEPLT528
000441  001   1440 WRITE (6,1460) (TITLES(L,J),L=1,9)                           GEPLT529
000442  001   1460 FORMAT(15X,8HSKIPPING,5X,9A6,5X,28HTOO MANY CURVES ON THIS GRID)  GEPLT530
000443  001         GO TO 1550
000444  001  C                                                        ERROR ITEMGEPLT532
000445  001   1480 WRITE (6,1500) (TITLES(L,J),L=1,9)                           GEPLT533
000446  001   1500 FORMAT(15X,8HSKIPPING,5X,9A6,5X,21HTHIS ITEM IS IN ERROR)    GEPLT534
000447  001         GO TO 1550
000448  001   1520 WRITE (6,1540) (TITLES(L,J),L=1,9)                           GEPLT536
000449  001   1540 FORMAT(15X,8HPLOTTING,5X,9A6)                                GEPLT537
000450  001  C                                     BUMP THE ITEM AND CURVE COUNTERSGEPLT538
000451  001   1550 IF(NOAVG .EQ. 0) GO TO 1575
000452  001         IF(J .NE. IABS(ITMAVG(NAVG+1))) GO TO 1575
000453  001         NAVG = NAVG + 1
000454  001         IF(NOAVG .EQ. NAVG) NOAVG = 0
```

```
000455      001          IF(ITMAVG(NAVG) .GT. 0) GO TO 1575
000456      001          IF(K+1.GT. 4) GO TO 1575
000457      001          IF(LOC(J+1) .LT. 0) GO TO 1575
000458      001          NC = FLD(6,6,ITYTLS(1,J)) - 47
000459      001          K = K +1
000460      001          TITLEB(1) = PSYM(K)
000461      001          TITLEB(2) = 6HAVERAG
000462      001          TITLEB(3) = 6HE
000463      001          DO 1553 L=4,20
000464      001     1553 TITLEB(L) = 6H
000465      001          NPY =1005 - 18*NC
000466      001          CALL RITE2V(NPX,NPY,1023,90,1,NCB,1,TITLEB,NL)
000467      001          ISW = 1
000468      001          DO 1556 L=1,NTPTS
000469      001          ABS(L) = ABS(ISTART+L)
000470      001     1556 ORD(L) = AVG(NAVG)
000471      001          NPTS = NTPTS
000472      001          CALL GOPLOT(K)
000473      001     1560 CONTINUE
000474      001     1575 J = J + 1
000475      001          K = K+1                                                GEPLT540
000476      001     C                                  CHECK FOR END OF ITEMSGEPLT541
000477      001          IF (J .LT. ITEMS) GO TO 1580                           GEPLT542
000478      001          KFINIS = 1                                             GEPLT543
000479      001          GO TO 1600                                             GEPLT544
000480      001     C                                     CHECK FOR NEW GRIDGEPLT545
000481      001     1580 IF (ITEM(J) .GT. 0) GO TO 1400                         GEPLT546
000482      001     1600 IF (KFINIS.EQ.0) GO TO 1300                            GEPLT547
000483      001     C                               REFERENCE THE NEW GRID SETGEPLT548
000484      001          IL = IR                                                GEPLT549
000485      001          ABSL = ABSR                                            GEPLT550
000486      001     1620 CONTINUE                                               GEPLT551
000487      001          I08 = 0                                                GEPLT552
000488      001     1640 CALL FILMAV(0)                                         GEPLT553
000489      001          NCASE = NCASE + 1                                      GEPLT554
000490      001          GO TO 240                                              GEPLT555
000491      001          END                                                    GEPLT556

END ELT.


*MOD,P FLORES
```

```
*ELT,L FLORES
ELTOT7 RLIB70 02/28-03:19:29-(2,)
000001      000          SUBROUTINE FLORES(L30,NTB)
000002      000    C
000003      000          LOGICAL LRO, LMU, COP
000004      000    C
000005      000          DIMENSION RDATA(1)
000006      000    C
000007      000          COMMON /ARRAY / TABLE (1)
000008      000          COMMON /TEMP  / T     (1)
000009      000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000010      000          COMMON /POINTR/ LNODE, LCOND, LCONS, LARRY, ICOMP, LTB, LPR
000011      000          COMMON /FLODAT/ NDATA (1)
000012      000          COMMON /TYPDAT/ TYPE  (10,1)
000013      000          COMMON /WDOT  / W     (1)
000014      000          COMMON /FLOWG / GF    (1)
000015      000          COMMON /FLOWR / AFR   (1)
000016      000          COMMON /FDATA / COP, LFO, NRO, RO, LMU, NMU, XMU, GC2
000017      000    C
000018      000    C
000019      000          EQUIVALENCE (RDATA(1),NDATA(1)), (HL,NHL), (MFF,FFM)
000020      000    C
000021      000          DATA MAXI /65000/
000022      000    C
000023      000    C
000024      000          WNTB = ABS(W(NTB))
000025      000          W4   = 4.0*WNTB
000026      000          RSUM = AFR(NTB)
000027      000          IC   = NDATA(L30)
000028      000    C
000029      000    C FLUID LUMP LOOP
000030      000    C
000031      000          DO 200 I=1,IC,3
000032      000          K = L30 + I
000033      000          NFL   = NDATA(K)
000034      001          IF (NFL .LT. 1) GO TO 200
000035      000          ITYPE = NDATA(K+1)
000036      000          NTL   = NDATA(K+2)
000037      000          CSA   = TYPE( 1,ITYPE)
000038      000          WP    = TYPE( 2,ITYPE)
000039      000          HL    = TYPE( 5,ITYPE)
000040      000          FFM   = TYPE( 6,ITYPE)
000041      000          FFC   = TYPE( 7,ITYPE)
000042      000          FLLD  = TYPE(10,ITYPE)
000043      000    C
000044      000          IF(LRO) CALL DIDEG1(T(NFL),TABLE(NRO), RO)
000045      000          IF(LMU) CALL DIDEG1(T(NFL),TABLE(NMU),XMU)
000046      000          RE = W4/XMU/WP
000047      000          IF(NHL .GT. 0 .AND. NHL .LT. MAXI) CALL DIDEG1(RE,TABLE(NHL),HL)
000048      000          IF(RE .GT. 2000.0) GO TO 100
000049      000          WMU = XMU
000050      000          IF(LMU) CALL DIDEG1(T(NTL),TABLE(NMU),WMU)
000051      000          FF = 64.0/RE*SQRT(WMU/XMU)
000052      001          GO TO 160
000053      000      100 IF(MFF .EQ. 0) GO TO 120
000054      000          CALL DIDEG1(RE,TABLE(MFF),FF)
000055      000          GO TO 160
```

```
000056  000     120 IF(RE .LT. 4000.0) GO TO 140
000057  000         FF = 0.316/SQRT(SQRT(RE))
000058  000         GO TO 160
000059  000     140 FF = 0.2086082052 + RE*(-0.1868265324E-3 + RE*(0.6236703785E-7
000060  000        1   + RE*(-0.65545818E-11)))
000061  000     160 R = (FF*FFC*FLLD +HL)*WNTB/GC2/CSA/CSA/RO
000062  000         RSUM = RSUM + R
000063  000     C
000064  000         IF(.NOT. COP) GO TO 200
000065  000         CALL LINECK(5)
000066  000         WRITE(6,180) NEXT(LNODE+NFL), T(NFL), ITYPE, NEXT(LNODE+NTL),
000067  000        1 T(NTL), WP, CSA, FLLD, MFF, FFC, HL, RO, XMU, RE, FF, R
000068  000     180 FORMAT(/              7X 7HNFL    = I10   , 8X 7HT(NFL)= G13.8,
000069  000        1  5X 7HITYPE = I10   , 8X 7HNTL    = I10   , 8X 7HT(NTL)= G13.8/
000070  000        2  7X 7HWP    = G13.8, 5X 7HCSA    = G13.8, 5X 7HFLL    = G13.8,
000071  000        3  5X 7HFF    = I10   , 8X 7HFFC    = G13.8/ 7X 7HHL     = G13.8,
000072  000        4  5X 7HRO    = G13.8, 5X 7HMU     = G13.8, 5X 7HRE     = G13.8,
000073  000        5  5X 7HFF    = G13.8/ 7X 7HR      = G13.8 )
000074  000     200 CONTINUE
000075  000         GF(NTB) = 1.0/RSUM
000076  000     C
000077  000         IF(.NOT. COP) GO TO 300
000078  000         CALL LINECK (2)
000079  000         WRITE(6,220) GF(NTB)
000080  000     220 FORMAT(/ 7X 7HG(NTB)= G13.8)
000081  000     C
000082  000     300 RETURN
000083  000     C
000084  000         END

END ELT.


*HOG.P FLOSOL
```

```
*ELT,L FLOSOL
ELTOT7 AL1870 02/28-03:19:31-(2,)
000001      000          SUBROUTINE FLOSOL
000002      000     C
000003      000          LOGICAL COP, LRO, LMU, LPASS, LPUMP
000004      000     C
000005      000          DIMENSION RDATA(1), EXT(1), RFLOW(1), NSYSTM(15,1)
000006      000     C
000007      000          COMMON /ARRAY / NDATA (1)
000008      000          COMMON /TEMP  / T     (1)
000009      000          COMMON /FIXCON/ CON   (1)
000010      000          COMMON /XSPACE/ NDIM, NTB, NEXT(1)
000011      000          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRAY, ICOMP, LTB, LPR
000012      000          COMMON /FLODAT/ NFLOW (1)
000013      000          COMMON /SYSDAT/ SYSTEM(15,1)
000014      000          COMMON /PRESS / P     (1)
000015      000          COMMON /VALVP / VP    (1)
000016      000          COMMON /WDOTI / WI    (1)
000017      000          COMMON /FDIMNS/ NTYPE,NSYS,NTB,NPR,NV
000018      000          COMMON /FDATA / COP, LRO, NRO, RO, LMU, NMU, XMU, GC2
000019      000          COMMON /FDATA / TOL, MXPASS, FROF
000020      000     C
000021      000          EQUIVALENCE (RFLOW(1),NFLOW(1)), (TSEN,NSEN), (TSET,NSET)
000022      000          EQUIVALENCE (RDATA(1),NDATA(1)), (EXT(1),NEXT(1))
000023      000          EQUIVALENCE (CON(1),TIMEN), (CON(2),DTIMEU), (CON(28),LC)
000024      000          EQUIVALENCE (SYSTEM(1,1),NSYSTM(1,1))
000025      000     C
000026      000          DATA MAXI /65000/
000027      000     C
000028      000     C
000029      000     C VALVES
000030      000     C
000031      000          IF (NV .LT. 1) GO TO 200
000032      000          L41 = 1
000033      000          DO 195 J=1,NV
000034      000          NVLV  = NFLOW(L41+1)
000035      000          MODE  = NFLOW(L41+4)
000036      000          IF(MODE .EQ. 0) GO TO 190
000037      000          XMIN1 = RFLOW(L41+5)
000038      000          XMAX1 = RFLOW(L41+6)
000039      000          NSEN = NFLOW(L41+8)
000040      000          IC    = NFLOW(L41)
000041      000          IF(IC .EQ. 10) GO TO 160
000042      000          NSET = NFLOW(L41+9)
000043      000          IF(NSEN .GT. 0  .AND.  NSEN .LT. 10000) TSEN = T(NSEN)
000044      000          IF(NSET .GT. 0  .AND.  NSET .LT. 10000) TSET = T(NSET)
000045      000          IF(IC .NE. 12) GO TO 125
000046      000     C
000047      000     C RATE LIMITED
000048      000     C
000049      000          TDB = RFLOW(L41+10)
000050      000          IF(ABS(TSEN-TSET) - TDB) 190,190,110
000051      000      110 IF(TSEN .GT. TSET + TDB) GO TO 120
000052      000          XDOT = AMAX1(RFLOW(L41+11)*(TSEN-TSET-TDB),-RFLOW(L41+12))
000053      000          VP(NVLV) = AMAX1(VP(NVLV)+XDOT*DTIMEU,XMIN1)
000054      000          GO TO 190
000055      002      120 XDOT = AMIN1(RFLOW(L41+11)*(TSEN-TSET+TDB), RFLOW(L41+12))
```

```
000056  000          VP(NVLV) = AMIN1(VP(NVLV)+XDOT*DTIMEU,XMAX1)
000057  000          GO TO 190
000058  000    C
000059  000    C POLYNOMIAL
000060  000    C
000061  000      125 DT = TSEN - TSET
000062  000          XSS = RFLOW(L41+10) + DT*(RFLOW(L41+11) + DT*(RFLOW(L41+12)
000063  000         1                    + DT*(RFLOW(L41+13) + DT*(RFLOW(L41+14)
000064  000         2                    + DT*(RFLOW(L41+15))))))
000065  000          IF(XSS - XMIN1) 135,150,140
000066  000      135 XSS = RFLOW(L41+5)
000067  000          GO TO 150
000068  000      140 IF(XSS - XMAX1) 150,150,145
000069  000      145 XSS = RFLOW(L41+6)
000070  000      150 VP(NVLV) = XSS + (VP(NVLV)-XSS)*EXP(-DTIMEU/RFLOW(L41+6))
000071  000          GO TO 190
000072  000    C
000073  000    C SWITCHING
000074  000    C
000075  000      160 IF((IFIX((2.0*Y(NSEN)-RFLOW(L41+9)-RFLOW(L41+10))/(RFLOW(L41+9)
000076  000         1  - RFLOW(L41+10)))) 170,190,180
000077  000      170 VP(NVLV) = XMIN1
000078  000          GO TO 190
000079  000      180 VP(NVLV) = XMAX1
000080  000    C
000081  000      190 L41 = L41 + RFLOW(L41) + 1
000082  000      195 CONTINUE
000083  000    C
000084  000    C SYSTEM LOOP
000085  000    C
000086  000      200 DO 1000 I=1,NSYS
000087  000    C
000088  000    C CHECK MPASS OPTION
000089  000    C
000090  000          NSYSTM(14,I) = NSYSTM(14,I) - 1
000091  000          IF(NSYSTM(14,I) .GT. 0) GO TO 1000
000092  000          NSYSTM(14,I) = NSYSTM(6,I)
000093  000          NRO = NSYSTM(2,I)
000094  000          LRO = .TRUE.
000095  000          IF(NRO .GT. 1  .AND.  NRO .LT. MAX1) GO TO 220
000096  000          LRO = .FALSE.
000097  000          RO  = SYSTEM(2,I)
000098  000      220 NMU = NSYSTM(3,I)
000099  000          LMU = .TRUE.
000100  000          IF(NMU .GT. 1  .AND.  NMU .LT. MAXI) GO TO 230
000101  000          LMU =  FALSE.
000102  000          XMU = SYSTEM(3,I)
000103  000      230 GC2 = SYSTEM(5,I)*2.0
000104  000          TOL    = SYSTEM(7,I)
000105  000          MXPASS = NSYSTM(8,I)
000106  000          PROF   = SYSTEM(9,I)
000107  000          COP = .FALSE.
000108  000          IF (NSYSTM(10,I) .EQ. 0) GO TO 232
000109  000          COP = .TRUE.
000110  000          IF (LNODE .EQ. 0) CALL NNREAD(1)
000111  000          IF (LTB .EQ. 0) CALL NNREAD(5)
000112  000    C
```

```
000113      000   C CHECK PUMP OPTION
000114      000   C
000115      000     232 LOCP = NSYSTM(11,I)
000116      000         IF(LOCP .LT. 1) GO TO 240
000117      000         IF(NFLOW(LOCP) - 2) 240,235,260
000118      000     235 NPI   = NFLOW(LOCP+1)
000119      000         NPUMP = NFLOW(LOCP+2)
000120      000         CALL DIDEGI(TIMEN,NDATA(NPUMP),WI(NPI))
000121      000     240 LPUMP = .FALSE.
000122      000         GO TO 300
000123      000     260 LPUMP = .TRUE.
000124      000         NPI = NFLOW(LOCP+1)
000125      000         NPO = NFLOW(LOCP+2)
000126      000         IF(NFLOW(LOCP) .GT. 3) GO TO 280
000127      000         KPUMP = 1
000128      000         NPUMP = NFLOW(LOCP+3)
000129      000         NP = NDATA(NPUMP)
000130      000         WMX = RDATA(NPUMP+NP-1)
000131      000         DPMX = RDATA(NPUMP+2)
000132      000         GO TO 290
000133      000     280 KPUMP = 2
000134      000         A0 = RFLOW(LOCP+3)
000135      000         A1 = RFLOW(LOCP+4)
000136      000         A2 = RFLOW(LOCP+5)
000137      000         A3 = RFLOW(LOCP+6)
000138      000         A4 = RFLOW(LOCP+7)
000139      000   C
000140      000   C SYSTEM SOLUTION
000141      000   C
000142      000     290 LPASS = .FALSE.
000143      000     300 LOCNET = NSYSTM(12,1)
000144      000         NAME = NFLOW(LOCNET+1)
000145      000         DO 960 KPASS=1,20
000146      000   C
000147      000         IF(.NOT. COP) GO TO 640
000148      000         CALL TOPLIN
000149      000         WRITE(6,620) KPASS, NAME
000150      000     620 FORMAT(70H0* * *  CHECKOUT PRINT FOR PRESSURE/FLOW COMPUTATION SUB
000151      000        1ROUTINE  * * * // 8X 7HKPASS = I3, 5H FOR  9A6)
000152      000         LC = LC + 4
000153      000         IF(LPASS) GO TO 640
000154      000         WRITE(6,330) TOL, MXPASS, PROF
000155      000     330 FORMAT(1H0 18X 5HTOL = G10.5, 9H MXPASS = I5, 7H PROF = G10.5)
000156      000         LC = LC + 2
000157      000   C
000158      000     640 CALL NTSOL(LOCNET)
000159      000   C
000160      000         IF(.NOT. LPUMP) GO TO 1000
000161      000   C
000162      000   C PUMP
000163      000   C
000164      000         WS = WI(NPI)
000165      000         TEST = 0.001*WS
000166      000         PTOL=TOL*WS
000167      000         DPS = P(NPI) - P(NPO)
000168      000         WK  = WS
000169      000         DPK = DPS
```

```
000170   000          GO TO (660,800), KPUMP
000171   000    C
000172   000    C TABULATED PUMP CURVE
000173   000    C
000174   000      660 IF(LPASS) GO TO 665
000175   000          C = DPS/WS
000176   000          D = 0.0
000177   000          GO TO 670
000178   000      665 C = (DPS-DPL)/(WS-WL)
000179   000          D = DPL - WL*C
000180   000      670 IF(.NOT. COP) GO TO 690
000181   000          CALL LINECK(2)
000182   000          WRITE(6,680)
000183   000      680 FORMAT(1HO 7X 39HCHECKOUT PRINT FOR TABULATED PUMP CURVE )
000184   000      690 DO 740 J=1,100
000185   000          WA = AMIN1(WS,WMX)
000186   000          DPB = AMIN1(DPS,DPMX)
000187   000          CALL DIDEG1(WA,NDATA(NPUMP),DPA)
000188   000          CALL REVPOL(DPB,NDATA(NPUMP),WB)
000189   000          IF(.NOT. COP) GO TO 710
000190   000          CALL LINECK(3)
000191   000          WRITE(6,705) J, WS, WA, WB, DPS, DPA, DPB
000192   000      705 FORMAT(/              7X 7HJ     = I10  , 8X 7HWS    = G13.8,
000193   000         1  5X 7HWA    = G13.8, 5X 7HWB    = G13.8/32X 7HDPS   = G13.8,
000194   000         2  5X 7HDPA   = G13.8, 5X 7HDPB   = G13.8)
000195   000      710 A= (DPB-DPA)/(WB-WA)
000196   000          B = DPA - WA*A
000197   000          WNEW = (D-B)/(A-C)
000198   000          IF(ABS(WNEW-WS) - TEST) 940,940,720
000199   000      720 WS = WNEW
000200   000          DPS = A*WS + B
000201   000      740 CONTINUE
000202   000          CALL TOPLIN
000203   000          WRITE(6,760) NAME
000204   000      760 FORMAT(79H0* * *  SUBROUTINE FLOSOL FAILED TO CONVERGE TO A SOLUTI
000205   000         ION FOR FLOW RATE  * * * // 8X 4HFOR  A4)
000206   000          WRITE(6,770)
000207   000      770 FORMAT(//8X 52HSYSTEM TOTAL FLOW RATE IS SUPPLIED BY AN INPUT CURV
000208   000         1E)
000209   000          GO TO 900
000210   000    C
000211   000    C POLYNOMIAL PUMP CURVE
000212   000    C
000213   000      800 CHECK = 0.001*DPS
000214   000          A00 = A0
000215   000          A11 = A1 - DPS/WS
000216   000          WNEW = WS
000217   000          IF(.NOT. LPASS) GO TO 820
000218   000          TEMP = (DPS-DPL)/(WS-WL)
000219   000          A00  = A0 - DPL + TEMP*WL
000220   000          A11  = A1 - TEMP
000221   000      820 DO 860 J=1,100
000222   000          FWNEW = A00 + WNEW*(A11 + WNEW*(A2 + WNEW*(A3 + WNEW*A4)))
000223   000          IF(ABS(FWNEW) - CHECK) 940,940,840
000224   000      840 FP = A11 + WNEW*(2.0*A2 + WNEW*(3.0*A3 + WNEW*4.0*A4))
000225   000          WNEW = WNEW - FWNEW/FP
000226   000      860 CONTINUE
```

```
000227      000   C
000228      000         CALL TOPLIN
000229      000         WRITE(6,760) NAME
000230      000         WRITE(6,880)
000231      000     880 FORMAT(//8X 89HSYSTEM TOTAL PRESSURE DROP IS SUPPLIED BY A FOURTH
000232      000        1ORDER POLYNOMIAL FUNCTION OF FLOW RATE )
000233      000     900 CALL WLKBCK
000234      000         CALL OUTCAL
000235      000         CALL EXIT
000236      000   C
000237      000     940 IF(ABS(WK-WNEW)-PTOL)1000,1000,950
000238      000     950 WI(NPI) = WNEW
000239      001          WI(NPO)= -WNEW
000240      000         LPASS = .TRUE.
000241      000         WL = WK
000242      000         DPL = DPK
000243      000     960 CONTINUE
000244      000   C
000245      000         CALL TOPLIN
000246      000         WRITE(6,980) NAME
000247      000     980 FORMAT(116H0* * *  SUBROUTINE FLOSOL FAILED TO CONVERGE TO A SOLUT
000248      000        1ION TO TRUE SYSTEM CHARACTERISTICS AND TRUE PUMP CURVE  * * * //
000249      000        2 8X 4HFOR  9A6)
000250      000         CALL WLKBCK
000251      000         CALL OUTCAL
000252      000         CALL EXIT
000253      000   C
000254      000    1000 CONTINUE
000255      000         RETURN
000256      000         END

END ELT.


*HOG,P FLOSUM
```

FLOSUM DATE 022675 PAGE 1

```
*ELT,L FLOSUM
ELTOTT RLIB70 02/28-03:19:33-(1,)
000001      000          SUBROUTINE FLOSUM(NTB,LOCD,SUM1,SUM2)
000002      000    C
000003      000          COMMON /FLODAT/ NFLOW(1)
000004      000          COMMON /WDOT  / W    (1)
000005      000          COMMON /TEMP  / T    (1)
000006      000    C
000007      000    C
000008      000          IC   = NFLOW(LOCD)
000009      000          LMP  = IABS(NFLOW(LOCD+IC-2))
000010      000          SUM1 = SUM1 + W(NTB)*TPOL(5,T(LMP))
000011      000          SUM2 = SUM2 + W(NTB)
000012      000          RETURN
000013      000          END

END ELT.


*HDG,P FLOTMP
```

```
#ELT,L FLOTMP
ELTOTT RL1870 02/28-03:19:34-(2,)
000001      002          SUBROUTINE FLOTMP(TMPTIM)
000002      000    C
000003      002          INTEGER HEADER(12)
000004      000    C
000005      002           COMMON /WDOT/ W(1)
000006      002           COMMON /VALVP/ VP(1)
000007      002           COMMON /PRESS/ P(1)
000008      002           COMMON /DELTAP/ DP(1)
000009      002           COMMON /FDIMNS/ NTYPE,NSYS,NTB,NP,NV
000010      000          COMMON /FIXCON/ CON(1)
000011      000          COMMON /TEMP   / T(1)
000012      000          COMMON /DIMENS/ NND, NNA, NTL
000013      000    C
000014      000          DATA IUT / 24 /
000015      000    C
000016      000    C
000017      002           READ(IUT) HEADER,(LL,I=1,5),NW,NPR,NVP,LL,LL,LL,NW,LL,LL,NSL
000018      002          IF(NP .NE. NPR) GO TO 10
000019      002          IF(NV .NE. NVP) GO TO 10
000020      002          IF(NTB .NE. NW ) GO TO 10
000021      000          IF(NTL   .EQ. NSL) GO TO 20
000022      000       10 CALL TOPLIN
000023      002          WRITE(6,15) HEADER, NP, NV, NTB, NTL, NPR, NVP, NW, NSL
000024      000       15 FORMAT(82H0* * *  ITEM COUNTS FROM HISTORY TAPE DO NOT MATCH ITEM
000025      000         1COUNTS FOR THIS RUN  * * *// 8X 29HTHE HISTORY TAPE LABEL IS -
000026      000         2 12A6 // 8X 43HTHE ITEM COUNTS FOR THIS RUN ARE  - - - - -  I5,
000027      000         3 3HNPR, I5, 3HNVP, I5, 3HNW , I5, 3HNSL /
000028      000         4          8X 43HTHE ITEM COUNTS FROM THE HISTORY TAPE ARE -  I5,
000029      000         5 3HNPR, I5, 3HNVP, I5, 3HNW , I5, 3HNSL /)
000030      000          CALL WLKBCK
000031      000          CALL EXIT
000032      000    C
000033      002       20  READ(IUT) XTIME,(DP(I),I=1,NW),(P(I),I=1,NPR),
000034      002         *  (VP(I),I=1,NVP),(W(I),I=1,NW),(T(I),I=1,NSL)
000035      000          IF(XTIME .LT. 0.0   ) GO TO 30
000036      000          IF(XTIME .LT. TMPTIM) GO TO 20
000037      000          GO TO 50
000038      000       30 XTIME = -XTIME
000039      000          WRITE(6,40)
000040      000       40 FORMAT(80H0HISTORY TAPE READ TIME IS GREATER THAN THE LAST TIME PO
000041      000         1INT ON THE HISTORY TAPE )
000042      000       50 WRITE(6,60) XTIME
000043      000       60 FORMAT(62H0INITIAL TEMPERATURES AND VALVE POSITIONS INPUT FROM U-T
000044      000         1APE AT  G12.5 )
000045      000          RETURN
000046      000          END

END ELT.


#HOG,P FLOWIN
```

```
*ELT,L FLOWIN
ELT077 RL1870 02/28-03:19:36-(2,)
000001      000          SUBROUTINE FLOWIN(L,NFLOW,JSW)
000002      000    C
000003      000          LOGICAL ERR
000004      000    C
000005      000          DIMENSION NFLOW(1)
000006      000    C
000007      000          COMMON /TAPE  / NIN, NOUT
000008      000          COMMON /CARD  / KRD, KOL, MXKOL
000009      000          COMMON /CIMAGE/ KARD(1)
000010      001            COMMON /FLOERR/ERR
000011      000    C
000012      000          L = 0
000013      000       15 CALL SKPB(JSW)
000014      000          GO TO(50,250), JSW
000015      000       50 I = 1
000016      000          CALL SUBN(NFLOW(1),ISW)
000017      000          GO TO(65,55,220,240,220), ISW
000018      000       55 WRITE(NOUT,60)
000019      000       60 FORMAT(46H0* * *  NP1 MUST BE INPUT AS AN INTEGER  * * * /)
000020      000          GO TO 240
000021      000       65 KOL = KOL + 1
000022      000          CALL SKPB(JSW)
000023      000          GO TO(70,250), JSW
000024      000       70 I = 2
000025      000          IF(KARD(KOL) .EQ. 1HA) GO TO 100
000026      000    C
000027      000          CALL SUBN(NFLOW(2),ISW)
000028      000          GO TO(85,200,220,240,220), ISW
000029      000       85 KOL = KOL + 1
000030      000          CALL SKPB(JSW)
000031      000          GO TO(90,250), JSW
000032      000       90 I = 3
000033      000          IF(KARD(KOL) .NE. 1HA) GO TO 115
000034      000      100 KOL = KOL + 1
000035      000          CALL SUBN(NFLOW(I),ISW)
000036      000          GO TO(200,105,220,240,220), ISW
000037      000      105 WRITE(NOUT,110)
000038      000      110 FORMAT(59H0* * *  FLOATING POINT NUMBER INPUT FOR ARRAY NUMBER  *
000039      000         1* * /)
000040      000          GO TO 240
000041      000      115 DO 116 I=4,7
000042      000          NFLOW(I) = 0
000043      000      116 CONTINUE
000044      000          DO 150 I=3,7
000045      000          CALL SUBN(NFLOW(I),ISW)
000046      000          GO TO(120,145,130,240,130), ISW
000047      000      120 IF(NFLOW(I) .EQ. 0) GO TO 145
000048      000          J = I - 3
000049      000          WRITE(NOUT,125) J
000050      000      125 FORMAT(9H0* * *  C  I1, 48H MUST BE INPUT AS A FLOATING POINT NUMB
000051      000         1ER  * * * /)
000052      000          GO TO 240
000053      000      130 IF(I .GT. 3) GO TO 160
000054      000          WRITE(NOUT,135)
000055      000      135 FORMAT(48H0* * *  POLYNOMIAL CONSTANTS NOT SUPPLIED  * * * /)
```

```
000056      001          GO TO 240
000057      000      145 KOL = KOL + 1
000058      000      150 CONTINUE
000059      000      160 L = 7
000060      000          GO TO 210
000061      000      200 L = 1
000062      000      210 IF(ISW .EQ. 5) GO TO 245
000063      000          CALL SKPTE(JSW)
000064      000          RETURN
000065      000      220 WRITE(NOUT,225)
000066      000      225 FORMAT(48H0* * *  END FOUND WITHIN FLOW SOURCE DATA  * * * /)
000067      001      240  ERR =.TRUE.
000068      002      230 GO TO(235,250), JSW
000069      002      235  CALL SKPTE(JSW)
000070      000          GO TO(15,250), JSW
000071      000      245 JSW = 2
000072      000      250 RETURN
000073      000          END

END ELT.


*HDG,P FLOW1
```

```
#ELT,L FLOW1
ELTOT7 RLI870 02/28-03:19:37-(1,)
000001      000          SUBROUTINE FLOW1
000002      000    C
000003      000          LOGICAL ERR, TEST1, TEST2, TEST3
000004      000    C
000005      000          DIMENSION KBLOCK(6)
000006      000    C
000007      000          COMMON / TAPE / NIN,NOUT,INTERN
000008      000          COMMON /CARD  / KRD, KOL, MXKOL
000009      000          COMMON /CIMAGE/ KARD(80)
000010      000          COMMON /BUCKET/ IB(1)
000011      000          COMMON /POINT / LOC(20), LEN(20), LENBKT
000012      000          COMMON /FDIMNS/ NTYPE,NSYS, NTB, NP, NV, NFD
000013      000           COMMON /FLOERR/ERR
000014      000    C
000015      000          DATA  KBLOCK / 4HNETW, 4HSUBN, 4HFLUI,
000016      000         1               4HVALV, 4HFLOW, 4HEND  /
000017      000    C
000018      000    C
000019      000          NTYPE = 0
000020      000          NSYS  = 0
000021      000          NTB   = 0
000022      000          NP    = 0
000023      000          NV    = 0
000024      000          NFD   = 0
000025      000          TEST1 = .FALSE.
000026      000          TEST2 = .FALSE.
000027      000          TEST3 = .FALSE.
000028      000          IBLOCK = 4H
000029      000          NSP = 0
000030      000          DO 10 I=1,20
000031      000          IF(LEN(I) .EQ. 0) GO TO 10
000032      000          NSP = LOC(I) + LEN(I)
000033      000       10 CONTINUE
000034      000          KSP = NSP
000035      000       25 CALL CARDIN(JSW)
000036      000       30 ENCODE(35,IBLOCK) (KARD(I),I=13,16)
000037      000       35 FORMAT(4A1)
000038      000          DO 40 I=1,6
000039      000          IF(KBLOCK(I) .EQ. IBLOCK) GO TO 50
000040      000       40 CONTINUE
000041      000          WRITE(NOUT,45)
000042      000       45 FORMAT(39H0* * *  INVALID BLOCK DESIGNATOR  * * * /)
000043      000          ERR = .TRUE.
000044      000          RETURN
000045      000       50 CONTINUE
000046      000          GO TO(100,105,300,400,500,600), I
000047      000      100 J = 20
000048      000          GO TO 110
000049      000      105 J = 23
000050      000      110 NAME = 6H
000051      000          DO 115 KOL=J,MXKOL
000052      000          IF(KARD(KOL) .NE. 1H ) GO TO 125
000053      000      115 CONTINUE
000054      000          GO TO 130
000055      000      125 J = MIN0(MXKOL,KOL+3)
```

```
000056  000        ENCODE(35,NAME) (KARD(K),K=KOL,J)
000057  000        KOL = J
000058  000    130 CALL CARDIN (JSW)
000059  000        GO TO (150,200),I
000060  000    150 CONTINUE
000061  000        TEST1 = .TRUE.
000062  000        CALL SYSPAR(IB(NSP+3),M,IB(NSP+20),JSW)
000063  000        L = NSP + 1
000064  000        IB(NSP+1) = 16
000065  000        IB(NSP+2) = 1
000066  000        IB(NSP+14)= NAME
000067  000        NSP = NSP + 17
000068  000        IB(NSP+1) = M + 1
000069  000        IB(NSP+2) = 2
000070  000        NSP = NSP + M + 2
000071  000        GO TO(195,25), JSW
000072  000    195 NSYS = NSYS + 1
000073  000    200 CONTINUE
000074  000    C
000075  000        K = 4
000076  000    210 M = NSP + K
000077  000        CALL TUBIN(IB(M),JSW)
000078  000        GO TO(215,230), JSW
000079  000    215 NTB = NTB + 1
000080  000        CALL LUMPIN(L,IB(M+4),JSW)
000081  000        IB(M+3) = L
000082  000        K = K + L + 4
000083  000        GO TO(210,230), JSW
000084  000    230 IB(NSP+1) = K - 2
000085  000        IB(NSP+2) = 4
000086  000        IB(NSP+3) = NAME
000087  000        NSP = NSP + K - 1
000088  000        GO TO 25
000089  000    C
000090  000    C FLUID LUMP DATA
000091  000    C
000092  000    300 CONTINUE
000093  000        CALL CARDIN (JSW)
000094  000        KODE = 6
000095  000        K = 3
000096  000        TEST2 = .TRUE.
000097  000    310 M = NSP + K
000098  000        CALL FLTYP(IB(M),JSW)
000099  000        GO TO(315,530), JSW
000100  000    315 CALL FLUMP(L,IB(M+11),JSW)
000101  000        IF(L) 325,325,320
000102  000    320 IB(M+10) = L
000103  000        NTYPE = NTYPE + 1
000104  000        K = K + L + 11
000105  000    325 CONTINUE
000106  000    326 GO TO(310,530), JSW
000107  000    C
000108  000    C VALVES
000109  000    C
000110  000    400 CONTINUE
000111  000        CALL CARDIN (JSW)
000112  000        KODE = 5
```

```
000113  000          K = 3
000114  000      410 M = NSP + K
000115  000          CALL VLVIN(L,IB(M+2),IX,JSW)
000116  000          IF(L) 425,425,420
000117  000      420 IB(M) = L + 1
000118  000          IB(M+1) = IX
000119  000          K = K + L + 2
000120  000          NV = NV + 1
000121  000      425 GO TO(410,530), JSW
000122  000    C
000123  000    C FLOW SOURCE
000124  000    C
000125  000      500 CONTINUE
000126  000          CALL CARDIN (JSW)
000127  000          KODE = 3
000128  000          TEST3 = .TRUE.
000129  000      505 K = 3
000130  000      510 M = NSP + K
000131  000          CALL FLOWIN(L,IB(M+1),JSW)
000132  000          IF (L-1) 520,520,515
000133  000      515 IB(M) = L
000134  000          K = K + L + 1
000135  000      520 GO TO(510,530), JSW
000136  000      530 IB(NSP+1) = K - 2
000137  000          IB(NSP+2) = KODE
000138  000          NSP = NSP + K - 1
000139  000          GO TO 25
000140  000    C
000141  000    C
000142  000    C
000143  000      600 CONTINUE
000144  000          IF(TEST1) GO TO 610
000145  000          WRITE(NOUT,605)
000146  000      605 FORMAT(45H0* * *  SYSTEM PARAMETERS NOT SUPPLIED  * * * /)
000147  000          ERR = .TRUE.
000148  000    C
000149  000      610 IF(TEST2) GO TO 620
000150  000          WRITE(NOUT,615)
000151  000      615 FORMAT(43H0* * *  FLUID LUMP DATA NOT SUPPLIED  * * * /)
000152  000          ERR = .TRUE.
000153  000    C
000154  000      620 IF(TEST3) GO TO 630
000155  000          WRITE(NOUT,625)
000156  000      625 FORMAT(44H0* * *  FLOW SOURCE DATA NOT SUPPLIED  * * * /)
000157  000          ERR = .TRUE.
000158  000      630 CONTINUE
000159  000          NLOCS = NSP - KSP
000160  000          NFLOW = INTERN
000161  000          REWIND NFLOW
000162  000          WRITE(NFLOW) NLOCS, (IB(I+KSP),I=1,NLOCS)
000163  000          REWIND NFLOW
000164  000          RETURN
000165  000          END

END ELT.
```

@HDG,P FLOW2

```
#ELT,L FLOW2
ELTOT7 RL1B70 02/28-03:19:39-(11,)
000001      007          SUBROUTINE FLOW2
000002      007    C
000003      007          LOGICAL ERR
000004      007    C
000005      007          COMMON / TAPE / NIN,NOUT,INTERN,LB3D,LB4D,LUT1,LUT2,LUT3
000006      007          COMMON /BUCKET/ IB(1)
000007      007          COMMON / POINT / LLOC(20),LLEN(20),LENBKT
000008      007          COMMON /FDIMNS/ NTYPE, NSYS, NTB, NP, NV, NFD
000009      007           COMMON /FLOERR/ERR
000010      007           COMMON/DATA/ DUM(12),ERDATA
000011      007    C
000012      007          DATA MAXI / 999999 /
000013      007    C
000014      007          NFLOW = INTERN
000015      007          NSP = 0
000016      007          DO 10 I=1,20
000017      007          IF (LLEN(I) . EQ . 0) GO TO 10
000018      007          NSP = LLOC(I) + LLEN(I)
000019      007       10 CONTINUE
000020      007          LTB = NSP
000021      007          LV  = LTB + NTB
000022      007          LVX = LV  + NV
000023      007          LPR = LVX + NV
000024      007          NSP = NSP + 3*NTB + 2*NV
000025      007          REWIND NFLOW
000026      007          READ(NFLOW) NLOC, (IB(NSP+I),I=1,NLOC)
000027      007          KSP = NSP + NLOC
000028      007          REWIND NFLOW
000029      007    C
000030      007    C FLUID TYPE DATA BLOCK
000031      007    C
000032      007          J = NSP
000033      007       25 IF(IB(J+2) .EQ. 6) GO TO 30
000034      007          J = J + IB(J+1) + 1
000035      007          IF(J-KSP) 25,80,80
000036      007       30 IC = IB(J+1)
000037      007          K = 2
000038      007       40 IF(K .GT. IC) GO TO 75
000039      007          L = J + K
000040      007          ISW1 = 1
000041      007          ISW2 = 1
000042      007          ISW3 = 1
000043      007          IF(IB(L+5).GT.0.AND.IB(L+5).LT.MAXI) CALL ATOR(1,IB(L+5),0,0,ISW1)
000044      007          IF(IB(L+6).GT.0.AND.IB(L+6).LT.MAXI) CALL ATOR(1,IB(L+6),0,0,ISW2)
000045      007          IF(IB(L+8).GT.0.AND.IB(L+8).LT.MAXI) CALL ATOR(1,IB(L+9),0,0,ISW3)
000046      007          ISW = ISW1 + ISW2 + ISW3
000047      007          IF(ISW .NE. 3) ERR = .TRUE.
000048      007          WRITE(NFLOW) (IB(L+I),I=1,10)
000049      007          K = K + IB(L+11) + 11
000050      007          GO TO 40
000051      007       75 LTYP = J
000052      007    C
000053      007    C LOAD ACTUAL TUBE NUMBERS AND ACTUAL PRESSURE NODE NUMBERS FOR SORTING
000054      007    C
000055      007       80 NT = 0
```

```
000056  007          NPR = 0
000057  007          J   = NSP
000058  007      100 IF(IB(J+2) .NE. 5) GO TO 110
000059  007          K = 3
000060  007          IC  = IB(J+1)
000061  007      105 IF(K .GT. IC) GO TO 110
000062  007          L = J + K
000063  007          NT = NT + 1
000064  007          IB(LTB+NT) = IB(L+1)
000065  007          IB(LPR+NPR+1) = IB(L+2)
000066  007          IB(LPR+NPR+2) = IB(L+3)
000067  007          NPR = NPR + 2
000068  007          K = K + IB(L+4) + 4
000069  007          GO TO 105
000070  007      110 J = J + IB(J+1) + 1
000071  007          IF(J .LT. KSP) GO TO 100
000072  007      C
000073  007      C SORT AND CHECK ACTUAL TUBE NUMBER LIST
000074  007      C
000075  007      135 IF(NT .LT. 2) GO TO 160
000076  007          CALL SRTLST(NT,IB(LTB+1))
000077  007          CALL GENOUT (IB(LTB+1),1,NT,'0TUBE NUMBER LIST')
000078  007          DO 150 I=2,NT
000079  007          IF(IB(LTB+I).GT.IB(LTB+I-1)) GO TO 150
000080  007          WRITE(NOUT,140) IB(LTB+I)
000081  007      140 FORMAT(33H0* * *  MORE THAN ONE TUBE NUMBER I6, 7H  * * * /)
000082  007          ERR = .TRUE.
000083  007      150 CONTINUE
000084  007      160 K = LPR + 1
000085  007          IF(NPR .LT. 2) GO TO 175
000086  007          CALL SRTLST(NPR,IB(LPR+1))
000087  007          DO 170 I=2,NPR
000088  007          IF(IB(LPR+I) .EQ. IB(K)) GO TO 170
000089  007          K = K + 1
000090  007          IB(K) = IB(LPR+I)
000091  007      170 CONTINUE
000092  007          CALL GENOUT (IB(LPR+1),1,K-LPR,'0PRESSURE NODE LIST')
000093  007      175 NP = K - LPR
000094  007      C
000095  007          LOC = 1
000096  007      C
000097  007      C VALVES
000098  007      C
000099  007          IF(NV .LT. 1) GO TO 250
000100  007          NVLV = 0
000101  007          J = NSP
000102  007      200 IF(IB(J+2) .NE. 5) GO TO 210
000103  007          K = 2
000104  007          IC = IB(J+1)
000105  007      205 IF(K .GT. IC) GO TO 210
000106  007          L = J + K
000107  007          NVLV = NVLV + 1
000108  007      C STORE X1 AND VALVE NUMBER
000109  007          IB(LVX+NVLV) = IB(L+2)
000110  007          IB(LV +NVLV) = IB(L+3)
000111  007          K = K + IB(L+1) + 1
000112  007          GO TO 205
```

```
000113      007      210 J = J + IB(J+1) + 1
000114      007           IF(J .LT. KSP) GO TO 200
000115      007      C SORT AND CHECK VALVE NUMBER LIST
000116      007           IF(NVLV .LT. 2) GO TO 214
000117      007           JC = NV-1
000118      007           DO 213 I = 1,JC
000119      007           ASSIGN 214 TO XX
000120      007           KC = NV-I
000121      007           DO 212 K = 1,KC
000122      007           IF (IB(LV+K+1)-IB(LV+K))211,212,212
000123      007      211 KEEP = IB(LV+K+1)
000124      007           IB(LV+K+1) = IB(LV+K)
000125      007           IB(LV+K) = KEEP
000126      007           KEEP = IB(LVX+K+1)
000127      007           IB(LVX+K+1) = IB(LVX+K)
000128      007           IB(LVX+K) = KEEP
000129      007           ASSIGN 213 TO XX
000130      007      212 CONTINUE
000131      007           GO TO XX
000132      007      213 CONTINUE
000133      007      214 CALL GENOUT(IB(LV+1),1,NV,'OVALVE NUMBER LIST')
000134      007           CALL GENOUT (IB(LVX+1),1,NV,'OINITIAL VALVE POSITIONS')
000135      007           IF(NVLV .LT. 2) GO TO 225
000136      007           DO 220 I=2,NVLV
000137      007           IF(IB(LV+I) .GT. IB(LV+I-1)) GO TO 220
000138      007           WRITE(NOUT,215) IB(LV+I)
000139      007      215 FORMAT(34H0* * *  MORE THAN ONE VALVE NUMBER I6, 7H  * * * /)
000140      007           ERR = .TRUE.
000141      007      220 CONTINUE
000142      007      C RELATIVIZE TUBE NUMBERS, PRESSURE NODES AND TEMPERATURE NODES
000143      007      225 J = NSP
000144      007      230 IF(IB(J+2) .NE. 5) GO TO 240
000145      007           K = 2
000146      007           IC = IB(J+1)
000147      007      235 IF(K .GT. IC) GO TO 240
000148      007           L = J + K
000149      007           JC = IB(L+1) - 1
000150      007           ISW2 = 1
000151      007           ISW3 = 1
000152      007            NVALVE=IB(L+3)
000153      007           CALL ATOR(3,IB(L+3),LV+1,NTB,ISW1)
000154      007           IF(IB(L+4) .GT. 0) CALL ATOR(4,IB(L+4),LTB+1,NTB,ISW2)
000155      007           IF(IB(L+5) .GT. 0) CALL ATOR(4,IB(L+5),LTB+1,NTB,ISW3)
000156      007           ISW4 = 1
000157      007           ISW5 = 1
000158      009           IF(IB(L+10).GT.0. .AND. IB(L+10).LT.MAXI)
000159      009          *  CALL ATOR(2,IB(L+10),C,0,ISW4)
000160      009           IF(IB(L+11).GT.0. .AND. IB(L+11).LT.MAXI)
000161      009          *  CALL ATOR(2,IB(L+11),0,0,ISW5)
000162      007           ISW = ISW1 + ISW2 + ISW3 + ISW4 + ISW5
000163      007           IF(ISW .NE. 5) ERR = .TRUE.
000164      007            IF(ISW .NE. 5) WRITE(NOUT,236) NVALVE
000165      007      236  FORMAT('0* * *  ABOVE MESSAGE(S) REFER TO VALVE',I6,'  * * *'/)
000166      007      C WRITE VALVE DATA
000167      007           WRITE (NFLOW) JC,(IB(L+I+2),I=1,JC)
000168      007           IB(L+2) = LOC
000169      007           K = K + IB(L+1) + 1
```

```
000170  007          LOC = LOC + JC + 1
000171  007          GO TO 235
000172  007      240 J = J + IB(J+I) + 1
000173  007          IF(J .LT. KSP) GO TO 230
000174  007  C
000175  007  C RELATIVIZE TUBE NUMBERS AND PRESSURE NODES
000176  007  C EXPAND FLUID LUMP AND TUBE LUMP PAIRS AND ADD TYPE NUMBER FOR FLUID
000177  007  C
000178  007      250 J = NSP
000179  007          NET = 0
000180  007          NR1 = 0
000181  007      260 IF (IB(J+2) . NE . 4) GO TO 440
000182  007          NET = NET + 1
000183  007          K = 3
000184  007          IC = IB(J+1)
000185  007      270 IF(K .GT. IC) GO TO 440
000186  007          L = J + K
000187  007          NTUBE = IB(L+1)
000188  007  C RELATIVIZE TUBE NUMBER AND PRESSURE NODES
000189  007          CALL ATOR(4,IB(L+1),LTB+1,NTB,ISW1)
000190  007          CALL ATOR(5,IB(L+2),LPR+1,NP ,ISW2)
000191  007          CALL ATOR(5,IB(L+3),LPR+1,NP ,ISW3)
000192  007          ISW = ISW1 + ISW2 + ISW3 - 3
000193  007          L = L + 4
000194  007          JC = IB(L)
000195  007  C
000196  007          IF(JC .LT. 2) GO TO 430
000197  007          N = 0
000198  007          DO 400 I=1,JC,2
000199  007          M = KSP + N
000200  007          N = N + 3
000201  007          NFL = IABS(IB(L+I))
000202  007          NFLR = NFL
000203  007          CALL ATOR(2,NFLR,0,0,ISW1)
000204  007          IB(M+1) = ISIGN(NFLR,IB(L+I))
000205  007  C LOCATE FLUID TYPE DATA
000206  007          NTYP = 0
000207  007          JJ = LTYP
000208  007          KC = IB(JJ+1)
000209  007          KK = 2
000210  007      280 IF(KK .GT. KC) GO TO 340
000211  007          LL = JJ + KK + 1
000212  007          NTYP = NTYP + 1
000213  007          MM = 1
000214  007          LC = IB(LL)
000215  007      290 IF(MM .GT. LC) GO TO 320
000216  007  C CHECK FOR MULTIPLE INPUT IN TYPE DATA
000217  007          IF(IB(LL+MM) .LT. 0) GO TO 300
000218  007          IF(IB(LL+MM) .EQ. NFL) GO TO 360
000219  007          MM = MM + 1
000220  007          GO TO 290
000221  007      300 IST = -IB(LL+MM)
000222  007          IND = IB(LL+MM+1)
000223  007          INC = IB(LL+MM+2)
000224  007          DO 310 II=IST,IND,INC
000225  007          IF (II . EQ . NFL) GO TO 360
000226  007      310 CONTINUE
```

```
000227  007          MM = MM + 3
000228  007          GO TO 290
000229  007      320 KK = KK + LC + 1
000230  007          GO TO 280
000231  007      340 WRITE(NOUT,350) NFL
000232  007      350 FORMAT(42H0* * *  TYPE DATA NOT FOUND FOR FLUID LUMP I6,
000233  007         1 7H  * * * /)
000234  007          ISW = 1
000235  007          NTYP=0
000236  007      360 IB(M+2) = NTYP
000237  007          NTL = IABS(IB(L+I+1))
000238  007          CALL ATOR(2,NTL,0,0,ISW2)
000239  007          IB(M+3) = ISIGN(NTL,IB(L+I+1))
000240  007          ISW = ISW + ISW1 + ISW2 - 2
000241  007      400 CONTINUE
000242  007          IF(ISW .EQ. 0) GO TO 420
000243  007          ERR = .TRUE.
000244  007          WRITE(NOUT,410) NTUBE
000245  007      410  FORMAT('0* * *  ABOVE MESSAGE(S) REFER TO TUBE',I6,'  * * *',/)
000246  007      420 WRITE(NFLOW) N, (IB(KSP+I),I=1,N)
000247  007          IB(L+1) = LOC
000248  007          LOC = LOC + N + 1
000249  007          NR1 = NR1+1
000250  007      430 K = K + JC + 4
000251  007          GO TO 270
000252  007      440 J = J + IB(J+1) + 1
000253  007          IF(J .LT. KSP) GO TO 260
000254  007      C
000255  007      C RELATIVIZE SPECIFIED PRESSURE NODES
000256  007      C
000257  007           J=NSP
000258  007      455 IF(IB(J+2) .NE. 2) GO TO 470
000259  007          ISW = 0
000260  007          IC = IB(J+1)
000261  007          K = J + 2
000262  007           IC2=IC/2
000263  007           DO 460 I=1,IC2
000264  007           CALL ATOR(5,IB(K+I),LPR+1,NP,ISW1)
000265  007           ISW = ISW+ ISW1
000266  007      460 CONTINUE
000267  007           IF(ISW .EQ. IC2) GO TO 470
000268  007          WRITE(NOUT,465)
000269  007      465 FORMAT (48H0* * *  ERROR IN SPECIFIED PRESSURE NODES  * * *)
000270  007          ERR = .TRUE.
000271  007      470 J = J + IB(J+1) + 1
000272  007          IF(J .LT. KSP) GO TO 455
000273  007      C
000274  007      C SQUEEZE BUCKET HERE IF REQUIRED
000275  007      C
000276  007          NET2 = 2*NET
000277  007          LNET = KSP
000278  007          LNAME = LNET + NET
000279  007          KSP = KSP + NET2
000280  007          KNET = 0
000281  007      C
000282  007      C LOAD NETWORK CONNECTIONS DATA
000283  007      C
```

```
000284  007        DO 1440 IPASS=1,NET2
000285  007    700 J = NSP
000286  007    710 IF(IB(J+2) .EQ. 4) GO TO 715
000287  007    711 J = J + IB(J+1) + 1
000288  007        IF(J-LNET) 710,1440,1440
000289  007    715 K = 3
000290  007        IC = IB(J+1)
000291  007    720 IF(K .GT. IC) GO TO 780
000292  007        L = J + K
000293  007        IF(IB(L+4) .NE. 1) GO TO 760
000294  007  C SUBNETWORK REFERENCE
000295  007        IF(IPASS .LE. NET) GO TO 735
000296  007        WRITE(NOUT,730) IB(L+5), IB(L+1), IB(J+14)
000297  007    730 FORMAT(19H0* * *  SUBNETWORK A4, 19H REFERENCED IN TUBE I6,
000298  007       1  26H OF NETWORK OR SUBNETWORK A4, 7H  * * * /)
000299  007        GO TO 760
000300  007    735 IF(KNET .LT. 1) GO TO 711
000301  007        NAME = IB(L+5)
000302  007        DO 740 I=1,KNET
000303  007        IF(IB(LNAME+I) .NE. NAME) GO TO 740
000304  007        IB(L+4) = -1
000305  007        IB(L+5) = IB(LNET+I)
000306  007        GO TO 760
000307  007    740 CONTINUE
000308  007        GO TO 711
000309  007    760 K = K + IABS(IB(L+4)) + 4
000310  007        GO TO 720
000311  007  C VALVE LOCATIONS
000312  007    780 NNV = 0
000313  007        LOCNV = 0
000314  007        IF(NVLV .LT. 1) GO TO 940
000315  007        JJ = NSP
000316  007    800 IF(IB(JJ+2) .NE. 5) GO TO 920
000317  007        JC = IB(JJ+1)
000318  007        KK = 2
000319  007    820 IF(KK .GT. JC) GO TO 920
000320  007        LL = JJ + KK
000321  007        K = 3
000322  007    840 IF(K .GT. IC) GO TO 900
000323  007        L = J + K
000324  007  C           NTS1
000325  007        IF(IB(LL+4) .EQ.  IB(L+1)) GO TO 860
000326  007  C           NTS2
000327  007        IF(IB(LL+5) .NE.  IB(L+1)) GO TO 880
000328  007    860 NNV = NNV + 1
000329  007        IF(NNV .EQ. 1) NNV = 2
000330  007        IB(KSP+NNV) = IB(LL+2)
000331  007        GO TO 900
000332  007    880 K = K + IABS(IB(L+4)) + 4
000333  007        GO TO 840
000334  007    900 KK = KK + IB(LL+1) + 1
000335  007        GO TO 820
000336  007    920 JJ = JJ + IB(JJ+1) + 1
000337  007        IF (JJ . LT . KSP) GO TO 800
000338  007    930 IF(NNV .LT. 1) GO TO 940
000339  007        IB(KSP+1) = NNV - 1
000340  007        LOCNV = LOC
```

```
000341  007  C SPECIFIED PRESSURES
000342  007    940 NSPR = 0
000343  007        LOCSPR = 0
000344  007        JJ = NSP
000345  007    960 IF(IB(JJ+2) .NE. 2) GO TO 1080
000346  007        JC = IB(JJ+1)
000347  007        KK = 3
000348  007    980 IF(KK .GT. JC) GO TO 1080
000349  007        LL = JJ + KK
000350  007        K = 3
000351  007   1000 IF(K .GT. IC) GO TO 1060
000352  007        L = J + K
000353  007        IF(IB(L+2) .EQ. IB(LL)) GO TO 1020
000354  007        IF(IB(L+3) .NE. IB(LL)) GO TO 1040
000355  007   1020 NSPR = NSPR + 1
000356  007        IF(NSPR .EQ. 1) NSPR = 2
000357  007        IB(KSP+NNV+NSPR) = IB(LL)
000358  007        GO TO 1060
000359  007   1040 K = K + IABS(IB(L+4)) + 4
000360  007        GO TO 1000
000361  007   1060 KK = KK + IB(LL+1) + 1
000362  007        GO TO 980
000363  007   1080 JJ = JJ + IB(JJ+1) + 1
000364  007        IF(JJ .LT. KSP) GO TO 960
000365  007        IF(NSPR .LT. 2) GO TO 1100
000366  007        LOCSPR = LOC + NNV
000367  007        IB(KSP+NNV+1) = NSPR - 1
000368  007  C LOAD NETWORK FOR SORT
000369  007   1100 LOCNET = KSP + NNV + NSPR
000370  007        IB(LOCNET+2) = IB(J+3)
000371  007        IB(LOCNET+3) = LOCSPR
000372  007        IB(LOCNET+4) = LOCNV
000373  007        K = 3
000374  007        NN = LOCNET + 4
000375  007   1120 IF(K .GT. IC) GO TO 1140
000376  007        L = J + K
000377  007        IB(NN+1) = IB(L+1)
000378  007        IB(NN+2) = IB(L+2)
000379  007        IB(NN+3) = IB(L+3)
000380  007        IB(NN+4) = ISIGN(IB(L+5),IB(L+4))
000381  007          IF(IB(L+4) .EQ. 0) IB(NN+4)=0
000382  007        NN = NN + 4
000383  007        K = K + IABS(IB(L+4)) + 4
000384  007        GO TO 1120
000385  007   1140 NNET = NN - LOCNET - 1
000386  007        IB(LOCNET+1) = NNET
000387  007  C SORT NETWORK
000388  007        NTBCHK = NNET/4-1
000389  007        IF(NTBCHK.LT.1) GO TO 1400
000390  007        NP1 = NTBCHK+1
000391  007        NM1 = NTBCHK-1
000392  007        NN = LOCNET + 4
000393  007        DO 1260 I=1,NTBCHK
000394  007        DO 1240 NPASS = 1,NP1
000395  007        NFRM = IB(NN+2)
000396  007        MM = NN + 4
000397  007        DO 1180 K=1,NTBCHK
```

```
000398      007          IF (IB(MM+3) .EQ. NFRM) GO TO 1200
000399      007          MM = MM + 4
000400      007     1180 CONTINUE
000401      007          NN = NN + 4
000402      007          IF(I.GT.NM1) GO TO 1260
000403      007          N1 = NN
000404      007          N2 = NN+4
000405      007          DO 1195 K = I,NM1
000406      007          IF(IB(N2+2).NE.NFRM) GO TO 1190
000407      007          DO 1185 M = 1,4
000408      007          KEEP = IB(N1+M)
000409      007          IB(N1+M) = IB(N2+M)
000410      007          IB(N2+M) = KEEP
000411      007     1185 CONTINUE
000412      007          N1 = N1+4
000413      007     1190 N2 = N2+4
000414      007     1195 CONTINUE
000415      007          GO TO 1260
000416      007     1200 DO 1220 K=1,4
000417      007          KEEP = IB(MM+K)
000418      007          IB(MM+K) = IB(NN+K)
000419      007          IB(NN+K) = KEEP
000420      007     1220 CONTINUE
000421      007     1240 CONTINUE
000422      007          WRITE(NOUT,1250) IB(J+3)
000423      007     1250 FORMAT(54H0* * *  ERROR IN SORTING TUBE CONNECTIONS FOR NETWORK
000424      007         *A4, 7H  * * */ 8X,25HCHECK PARALLEL FLOW PATHS/)
000425      007          ERR = .TRUE.
000426      007     1260 CONTINUE
000427      007     1400 NLOCS = NNV + NSPR + IB(LOCNET+1) + 1
000428      007          WRITE(NFLOW) NLOCS, (IB(KSP+I),I=1,NLOCS)
000429      007          KNET = KNET + 1
000430      007          IB(LNET+KNET) = LOC + NNV + NSPR
000431      007          LOC = LOC + NLOCS
000432      007          IB(LNAME+KNET) = IB(J+3)
000433      007          IB(J+2) = -IB(J+2)
000434      007     1440 CONTINUE
000435      007     C
000436      007     C RELATIVIZE PRESSURE NODES IN FLOW SOURCE DATA
000437      007     C
000438      007          LIFR = KSP
000439      007          KSP = KSP + NP
000440      007          J = NSP
000441      007          JC = 0
000442      007          DO 1445 I=1,NP
000443      007          IB(LIFR+I) = 0
000444      007     1445 CONTINUE
000445      007     1450 IF(IB(J+2) .NE. 3) GO TO 1490
000446      007          IC = IB(J+1)
000447      007          K = 3
000448      007     1460 IF(K .GT. IC) GO TO 1490
000449      007          L = J + K
000450      007          JC = IB(L)
000451      007          CALL ATOR(5,IB(L+1),LPR+1,NP,ISW1)
000452      007          ISW2 = 1
000453      007          IF(JC .NE. 2) GO TO 1470
000454      007          IF(IABS(ID(L+2)) .LT. MAXI) GO TO 1465
```

```
000455  007          NPI = IB(L+1)
000456  007          IB(LIFR+NPI) = IB(L+2)
000457  007          GO TO 1466
000458  007    1465  CALL ATOR(1,IB(L+2),0,0,ISW2)
000459  007    1466  ISW = ISW1 + ISW2 - 2
000460  007          GO TO 1480
000461  007    1470  CALL ATOR(5,IB(L+2),LPR+1,NP,ISW2)
000462  007          ISW = ISW1 + ISW2 - 2
000463  007           NPI =IB(L+1)
000464  007           NPO= IB(L+2)
000465  007          IF (JC .NE. 3) GO TO 1475
000466  007          CALL ATOR(1,IB(L+3),0,0,ISW3)
000467  007          ISW = ISW + ISW3 - 1
000468  010          LARRAY = LLOC(15) + IB(L+3) - 1
000469  011          ICA = ((IB(LARRAY)-2)/4)*2 +1
000470  007          IB(LIFR+NPI) = IB(LARRAY+ICA)
000471  007           IB(LIFR+NPO)=-IB(LARRAY+ICA)
000472  007          GO TO 1480
000473  007    1475  IB(LIFR+NPI) = IB(L+3)
000474  007          IB(LIFR+NPO) =-IB(L+3)
000475  007    1480  IF(ISW .NE. 0) ERR = .TRUE.
000476  007    1485  K = K + JC + 1
000477  007          GO TO 1460
000478  007    1490  J = J + IABS(IB(J+1)) + 1
000479  007          IF(J .LT. LNET) GO TO 1450
000480  007    C
000481  007    C COMPLETE SYSTEM PARAMETERS AND FLOW SOURCE DATA
000482  007    C
000483  007          J = NSP
000484  007          N = 0
000485  007    1500  IF(IB(J+2) .NE. 1) GO TO 1760
000486  007          L = J + 2
000487  007          NAME = IB(L+12)
000488  007          ISW = 0
000489  007          DO 1520 I=1,4
000490  007          ISW1 = 1
000491  007          IF(IB(L+I) .LT. MAXI) CALL ATOR(1,IB(L+I),0,0,ISW1)
000492  007          ISW = ISW + ISW1
000493  007    1520  CONTINUE
000494  007          ISW1 = 1
000495  007          IF(IB(L+13) .LT. MAXI) CALL ATOR(1,IB(L+13),0,0,ISW1)
000496  007          ISW  = ISW + ISW1
000497  007          IF(ISW .EQ. 5) GO TO 1550
000498  007          WRITE(NOUT,1530) NAME
000499  007    1530   FORMAT('0* * *  ABOVE MESSAGE(S) REFER TO NETWORK',A4,7H  * * *)
000500  007          ERR = .TRUE.
000501  007    C LOCATE NETWORK CONNECTIONS DATA FOR THIS SYSTEM
000502  007    1550  DO 1570 I=1,KNET
000503  007          IF(IB(LNAME+I) .EQ. NAME) GO TO 1580
000504  007    1570  CONTINUE
000505  007          WRITE(NOUT,1575) NAME
000506  007    1575  FORMAT(42H0* * *  NETWORK DATA NOT FOUND FOR SYSTEM A4,7H  * * *)
000507  007          ERR = .TRUE.
000508  007          GO TO 1585
000509  007    1580  IB(L+12) = IB(LNET+I)
000510  007    C FLOW SOURCE DATA FOR THIS NETWORK
000511  007    1585  LOCP = 0
```

```
000512  007        IF(JC .EQ. 0) GO TO 1740
000513  007        JJ = J + IB(J+1) + 1
000514  007        JJ = JJ + IB(JJ+1) + 1
000515  007        IF(IABS(IB(JJ+2)) .NE. 4) GO TO 1740
000516  007        KC = IB(JJ+1)
000517  007        IF(KC .LT. 2) GO TO 1740
000518  007        JJJ = NSP
000519  007   1600 IF(IB(JJJ+2) .NE. 3) GO TO 1680
000520  007        KKK = 2
000521  007        LL = IB(JJJ+1)
000522  007   1620 IF(KKK .GT. LC) GO TO 1680
000523  007        LLL = JJJ + KKK
000524  007        MC = IB(LLL+1)
000525  007        IF (IABS(IB(LLL+3)) . GT . MAXI) GO TO 1670
000526  007   1635 NPI = IB(LLL+2)
000527  007        KK = 3
000528  007   1640 IF(KK .GT. KC) GO TO 1670
000529  007        LL = JJ + KK
000530  007        IF(IB(LL+2) .EQ. NPI) GO TO 1700
000531  007        IF(IB(LL+3) .EQ. NPI) GO TO 1700
000532  007        KK = KK + IABS(IB(LL+4)) + 4
000533  007        GO TO 1640
000534  007   1670 KKK = KKK + MC + 1
000535  007        GO TO 1620
000536  007   1680 JJJ = JJJ + IB(JJJ+1) + 1
000537  007        IF(JJJ .LT. LNET) GO TO 1600
000538  007         GO TO 1740
000539  007   1700 LOCP = LOC + N
000540  007        LL = KSP + N + 1
000541  007        LLL = LLL + 1
000542  007        N = N + MC + 1
000543  007        IB(LL) = MC
000544  007        DO 1720 I=1,MC
000545  007        IB(LL+I) = IB(LLL+I)
000546  007   1720 CONTINUE
000547  007   1740 IB(L+1) = LOCP
000548  007   1760 J = J + IB(J+1) + 1
000549  007        IF(J .LT. LNET) GO TO 1500
000550  007        NR2 = 0
000551  007        IF (N .EQ. 0) GO TO 1770
000552  007        WRITE (NFLOW) N,(IB(KSP+I),I=1,N)
000553  007        NR2 = NR2+1
000554  007        LOC = LOC+N
000555  007   1770 NFO = LOC
000556  007        N = 0
000557  007   C
000558  007   C COLLECT SYSTEM PARAMETERS
000559  007   C
000560  007        J = NSP
000561  007   1800 IF(IB(J+2) .NE. 1) GO TO 1840
000562  007        L = J + 2
000563  007        LL = KSP + N
000564  007        N = N + 15
000565  007        DO 1820 I=1,15
000566  007        IB(LL+I) = IB(L+I)
000567  007   1820 CONTINUE
000568  007   1840 J = J + IB(J+1) + 1
```

```
000569      007          IF(J .LT. LNET) GO TO 1800
000570      007          WRITE(NFLOW) N,(IB(KSP+I),I=1,N)
000571      007    C
000572      007          WRITE(NFLOW) NP, (IB(LIFR+I), I=1,NP)
000573      007           DO 1842 I=1,NP
000574      007           IB(LIFR+I)=0
000575      007     1842  CONTINUE
000576      007           J=NSP
000577      007     1845  IF(IB(J+2) .NE. 2) GO TO 1849
000578      007           IC = IB(J+1)
000579      007           K = J+2
000580      007           DO 1847 I=1,IC,2
000581      007           NSPR=IB(K+I)
000582      007           IB(LIFR+NSPR)=IB(K+I+1)
000583      007     1847  CONTINUE
000584      007     1849  J= J+IB(J+1) +1
000585      007           IF(J .LE. LNET) GO TO 1845
000586      007           WRITE(NFLOW) NP,(IB(LIFR+I),I=1,NP)
000587      007    C
000588      007          IF (NV .GT. 0) WRITE (NFLOW) NV,(IB(LVX+I),I=1,NV)
000589      007    C
000590      007          WRITE(LUT1) (IB(LTB+I),I=1,NTB)
000591      007          WRITE(LUT1) (IB(LPR+I),I=1,NP )
000592      007          IF(NV .GT. 0) WRITE(LUT1) (IB(LV+I),I=1,NV)
000593      007           WRITE(LB3D) NTYPE,NSYS,NTB,NP,NV,NFD
000594      007          REWIND NFLOW
000595      007          LTYP = LTB
000596      007          N = 0
000597      007          L = LTYP
000598      007          IF(NTYPE .LT. 1) GO TO 1870
000599      007          DO 1860 J=1,NTYPE
000600      007          READ(NFLOW) (IB(L+I),I=1,10)
000601      007          L = L + 10
000602      007     1860 CONTINUE
000603      007     1870 LFLO = L
000604      007          IF(NV .LT. 1) GO TO 1900
000605      007          DO 1880 J=1,NV
000606      007          READ(NFLOW) N, (IB(L+I+1),I=1,N)
000607      007          IB(L+1) = N
000608      007          L = L + N + 1
000609      007     1880 CONTINUE
000610      007     1900 IF(NR1 .LT. 1) GO TO 1940
000611      007          DO 1920 J=1,NR1
000612      007          READ(NFLOW) N, (IB(L+I+1),I=1,N)
000613      007          IB(L+1) = N
000614      007          L = L + N + 1
000615      007     1920 CONTINUE
000616      007     1940 DO 1960 J=1,NET
000617      007          READ(NFLOW) N, (IB(L+I),I=1,N)
000618      007          L = L + N
000619      007     1960 CONTINUE
000620      007          IF(NR2 .LT. 1)GO TO 2000
000621      007          READ(NFLOW) N, (IB(L+I),I=1,N)
000622      007          L = L + N
000623      007     2000  WRITE(LB3D) (IB(LFLO+I),I=1,NFD)
000624      007           L=LFLO
000625      007          READ(NFLOW) N, (IB(L+I),I=1,N)
```

```
000626      007          WRITE(LB3D) (IB(L+I),I=1,N)
000627      007          LIFR=L
000628      007         READ(NFLOW) N, (IB(L+I),I=1,N)
000629      007         L = L + N
000630      007          LPR=L
000631      007          READ(NFLOW) N,(IB(L+I),I=1,N)
000632      007          L=L+N
000633      007         LVX = L
000634      007         N = 0
000635      007         IF(NV .GT. 0) READ(NFLOW) N, (IB(L+I),I=1,N)
000636      007         L = L + N
000637      007         REWIND NFLOW
000638      007   C
000639      007         N = LFLD - LTYP
000640      007         WRITE(LB3D) (IB(LTYP+I),I=1,N)
000641      007         X = 200.
000642      007         WRITE(LB3D) (X,I=1,NTB)
000643      007         IF(NV .GT. 0) WRITE(LB3D) (IB(LVX+I),I=1,NV)
000644      007          IF(ERR) ERDATA=2.0
000645      007          WRITE(LB3D) (IB(LPR+I),I=1,NP)
000646      007         WRITE(LB3D) (IB(LIFR+I),I=1,NP)
000647      007   C
000648      007         RETURN
000649      007         END

END ELT.


*ROG,P FLPRNT
```

```
@ELT,L FLPRNT
ELTOTT RLI870 02/28-03:19:44-(0,)
       000          SUBROUTINE FLPRNT(DATA,HEAD)
       000    C
       000          DIMENSION DATA(1), HEAD(9)
       000    C
       000          COMMON /FIXCON/ L(1)
       000    C
       000          EQUIVALENCE (NNT,D)
       000    C
       000    C
       000          D = DATA(1)
       000          IF(L(29) .EQ. 0 .OR. L(28) .GE. 60) CALL TOPLIN
       000          WRITE(6,101) HEAD
       000          L(28) = L(28) + 2
       000          NS = 1
       000          NF = 10
       000        5 IF(NF .GT. NNT) GO TO 20
       000       10 WRITE(6,100) (DATA(I+1),I=NS,NF), NF
       000          L(28) = L(28) + 1
       000          IF(L(28) .GE. 60) CALL TOPLIN
       000          IF(NF .EQ. NNT) RETURN
       000          NS = NF + 1
       000          NF = NF + 10
       000          GO TO 5
       000       20 WRITE(6,100) (DATA(I+1),I=NS,NNT)
       000          IF(L(28) .GE. 60) CALL TOPLIN
       000          RETURN
       000      100 FORMAT(1X 5G12.5, 5X 5G12.5, I5)
       000      101 FORMAT(1HO 21A6, A5)
       000          END

END ELT.

@HDG,P FLTYP
```

```
#ELT,L FLTYP
ELTOTT RLIB70 02/28-03:19:46-(1,)
000001      000          SUBROUTINE FLTYP(TYPDAT,JSW)
000002      000    C
000003      000          DIMENSION TYPDAT(1)
000004      000    C
000005      000          COMMON /TAPE  / NIN, NOUT
000006      000          COMMON /CARD  / KRD, KOL, MXKOL
000007      000          COMMON /CIMAGE/ KARD(80)
000008      000    C
000009      000       15 TYPDAT(6) = 0.0
000010      000          TYPDAT(7) = 1.0
000011      000          TYPDAT(8) = 1.0
000012      000          TYPDAT(9) = 1.0
000013      000    C
000014      000       25 DO 200 I=1,9
000015      000          CALL SKPB(JSW)
000016      000          GO TO(60,220), JSW
000017      000       60 K = KOL
000018      000          GO TO(110,110,110,110,90,90,110,70,80), I
000019      000    C F1
000020      000       70 CALL SUBN(TYPDAT(I),ISW)
000021      000          KSW = 1
000022      000          GO TO(195,75,150,160,250), ISW
000023      000       75 KSW = 2
000024      000          GO TO 195
000025      000    C F2
000026      000       80 GO TO(85,110), KSW
000027      000       85 IF(KARD(KOL) .EQ. 1HA) GO TO 95
000028      000          WRITE(NOUT,87)
000029      000       87 FORMAT(36H0* * *  ARRAY INPUT NOT FOUND  * * * /)
000030      000          GO TO 160
000031      000       90 IF(KARD(KOL) .NE. 1HA) GO TO 110
000032      000       95 KOL = KOL + 1
000033      000          CALL SUBN(TYPDAT(I),ISW)
000034      000          GO TO(195,96,150,160,250), ISW
000035      000       96 WRITE(NOUT,100) K
000036      000      100 FORMAT(62H0* * *  FLOATING POINT NUMBER INPUT FOR ARRAY NUMBER IN
000037      000         1COLUMN  I3, 7H  * * * /)
000038      000          GO TO 160
000039      000    C REAL NUMBERS
000040      000      110 CALL SUBN(TYPDAT(I),ISW)
000041      000          IF(.NOT. ABS(TYPDAT(I)) .GT. 0.0) GO TO 195
000042      000          GO TO(115,195,150,160,250), ISW
000043      000      115 WRITE(NOUT,120) K
000044      000      120 FORMAT(31H0* * *  INTEGER INPUT IN COLUMN  I3, 39H, FLOATING POINT
000045      000         1 NUMBER EXPECTED  * * * /)
000046      000      150 WRITE(NOUT,155)
000047      000      155 FORMAT(47H0* * *  END FOUND WITHIN FLUID TYPE DATA  * * * /)
000048      000      160 CALL SKPTE(JSW)
000049      000          GO TO(15,220), JSW
000050      000      195 KEEP = KARD(KOL)
000051      000          KOL = KOL + 1
000052      000          IF(KEEP .NE. 1H=) GO TO 200
000053      000          IF(I .GT. 4) GO TO 210
000054      000          WRITE(NOUT,196)
000055      000      196 FORMAT(60H0* * *  AT LEAST FIVE ITEMS MUST BE INPUT FOR FLUID LUMP
```

```
000056      000         1 DATA  * * */)
000057      000           GO TO 160
000058      000       200 CONTINUE
000059      000       210 TYPDAT(10) = TYPDAT(2)*TYPDAT(3)/(4.0*TYPDAT(1))
000060      000       220 RETURN
000061      000       250 JSW = 2
000062      000           RETURN
000063      000           END

END ELT.


@HDG,P FLUID
```

```
*ELT,L FLUID
ELTQT7 RL1870 02/28-03:19:47-(2,)
000001      002          SUBROUTINE FLUID(ICODE,LLOC1,LLOC2,ZERO,KOP)
000002      000    C
000003      000          LOGICAL LCP, LRO, LMU, LKT, COP, ERR
000004      000    C
000005      000          COMMON /FIXCON/ KON(1)
000006      000          COMMON /TEMP  / T     (1)
000007      000          COMMON /FLODAT/ NFLOW (1)
000008      000          COMMON /TYPDAT/ TYPE  (10,1)
000009      000          COMMON /SYSDAT/ NSYSTM(15,1)
000010      000          COMMON /FDATA / COP, LCP, NCP, RCP, LRO, NRO, RRO
000011      000          COMMON /FDATA / RH , LMU, NMU, RMU, LKT, NKT, RKT
000012      002          COMMON /FDATA / KODE, NLOC1, NLOC2, TZERO
000013      000          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRAY, ICOMP, LTB, LPR
000014      000           COMMON /FDIMNS/ NTYPE,NSYS
000015      000    C
000016      000          DATA NOUT / 6 /
000017      000    C
000018      000          KODE = ICODE
000019      000          NLOC1 = LLOC1
000020      000          NLOC2 = LLOC2
000021      002          TZERO = ZERO
000022      000          SHU1 = 0
000023      000          STU1 = 0
000024      000          KP1 = 0
000025      000    C
000026      000    C SYSTEM LOOP
000027      000    C
000028      000          DO 9000 I=1,NSYS
000029      000          CALL FLOP(RH ,LCP,RCP,13,I)
000030      000          CALL FLOP(NCP,LCP,RCP,1,I)
000031      000          CALL FLOP(NRO,LRO,RRO,2,I)
000032      000          CALL FLOP(NMU,LMU,RMU,3,I)
000033      000          CALL FLOP(NKT,LKT,RKT,4,I)
000034      000    C SETUP FLOW PROPERTIES
000035      000          COP = .FALSE.
000036      000          IF (NSYSTM(10,I) .EQ. 0 .AND. KOP .EQ. 0) GO TO 50
000037      000          COP = .TRUE.
000038      000          IF (LNODE .EQ. 0) CALL NNREAD(1)
000039      000          IF (LTB .EQ. 0) CALL NNREAD(5)
000040      000          KON(28) = 100
000041      000       50 LOC1 = NSYSTM(12,I)
000042      000          IC1 = NFLOW(LOC1)
000043      000    C
000044      000    C NETWORK LOOP
000045      000    C
000046      000          DO 8500 J1=4,IC1,4
000047      000          LOCD = NFLOW(LOC1+J1+3)
000048      000          IF(LOCD) 100,8500,5100
000049      000      100 KP2 = KP1
000050      000          SHU2 = SHU1
000051      000          STU2 = STU1
000052      000          LOC2 = -LOCD
000053      000          IC2 = NFLOW(LOC2)
000054      000    C      *
000055      000    C SUBNETWORK LOOP
```

```
000056  000   C
000057  000         DO 5000 J2=4,IC2,4
000058  000         LOCD = NFLOW(LOC2+J2+3)
000059  000         IF(LOCD) 200,5000,2150
000060  000     200 KP3  = KP2
000061  000         SHU3 = SHU2
000062  000         STU3 = STU2
000063  000         LOC3 = -LOCD
000064  000         IC3  = NFLOW(LOC3)
000065  000   C
000066  000   C SUBNETWORK LOOP
000067  000   C
000068  000         DO 2100 J3=4,IC3,4
000069  000         LOCD = NFLOW(LOC3+J3+3)
000070  000         IF(LOCD) 300,2100,400
000071  000     300 WRITE(NOUT,320) NFLOW(LOC3+1)
000072  000     320 FORMAT(19H0* * *  SUBNETWORK  A4, 37H MUST NOT CONTAIN A SUBNETWOR
000073  000        1K  * * * /)
000074  000         ERR = .TRUE.
000075  000         GO TO 2100
000076  000     400 ASSIGN 600 TO XX
000077  000         NTB  = NFLOW(LOC3+J3)
000078  000         NFRM = NFLOW(LOC3+J3+1)
000079  000         IC   = NFLOW(LOCD)
000080  000         X    = 0.0
000081  000   C
000082  000   C TUBE LOOP
000083  000   C
000084  000         DO 2000 K=1,IC,3
000085  000         L = LOCD + K
000086  000         LMP    = IABS(NFLOW(L))
000087  000         NTYP   = NFLOW(L+1)
000088  000         NTBLMP = NFLOW(L+2)
000089  000         X      = X + TYPE(3,NTYP)
000090  000         IF(NTBLMP) 1950,1950,XX
000091  000     500 HU = TPOL(5,TU)
000092  000         GO TO 1900
000093  000     600 IF(NFRM .EQ. KP3) GO TO 1800
000094  000         SUM1 = 0.0
000095  000         SUM2 = 0.0
000096  000         CALL UPSUM3(NFRM,LOC1,J1,LOC2,J2,LOC3,J3,SUM1,SUM2)
000097  000         SHU3 = SUM1/SUM2
000098  000         CALL RPOL(NH,SHU3,STU3)
000099  000         KP3 = NFRM
000100  000    1800 HU = SHU3
000101  000         TU = STU3
000102  000    1900 CALL TMPEQ(NTB,L,TYPE(1,NTYP),HU,TU,X)
000103  000    1950 TU = T(LMP)
000104  000         ASSIGN 500 TO XX
000105  000    2000 CONTINUE
000106  000   C
000107  000    2100 CONTINUE
000108  000   C
000109  000         GO TO 5000
000110  000   C
000111  000    2150 ASSIGN 2300 TO YY
000112  000         NTB  = NFLOW(LOC2+J2)
```

```
000113  000        NFRM = NFLOW(LOC2+J2+1)
000114  000        IC   = NFLOW(LOCD)
000115  000        X    = 0.0
000116  000  C
000117  000  C TUBE LOOP
000118  000  C
000119  000        DO 4500 K=1,IC,3
000120  000        L = LOCD + K
000121  000        LMP    = IABS(NFLOW(L))
000122  000        NTYP   = NFLOW(L+1)
000123  000        NTBLMP = NFLOW(L+2)
000124  000        X      = X + TYPE(3,NTYP)
000125  000        IF(NTBLMP) 4400,4400,YY
000126  000   2200 HU = TPOL(5,TU)
000127  000        GO TO 4300
000128  000   2300 IF(NFRM .EQ. KP2) GO TO 4200
000129  000        SUM1 = 0.0
000130  000        SUM2 = 0.0
000131  000        CALL UPSUM2(NFRM,LOC1,J1,LOC2,J2,SUM1,SUM2)
000132  000        SHU2 = SUM1/SUM2
000133  000        CALL RPOL(NH,SHU2,STU2)
000134  000        KP2  = NFRM
000135  000   4200 HU = SHU2
000136  000        TU = STU2
000137  000   4300 CALL TMPEQ(NTB,L,TYPE(1,NTYP),HU,TU,X)
000138  000   4400 TU = T(LMP)
000139  000        ASSIGN 2200 TO YY
000140  000   4500 CONTINUE
000141  000  C
000142  000   5000 CONTINUE
000143  000  C
000144  000        GO TO 8500
000145  000  C
000146  000   5100 ASSIGN 5300 TO ZZ
000147  000        NTB  = NFLOW(LOC1+J1)
000148  000        NFRM = NFLOW(LOC1+J1+1)
000149  000        IC   = NFLOW(LOCD)
000150  000        X    = 0.0
000151  000  C
000152  000  C TUBE LOOP
000153  000  C
000154  000        DO 8000 K=1,IC,3
000155  000        L = LOCD + K
000156  000        LMP    = IABS(NFLOW(L))
000157  000        NTYP   = NFLOW(L+1)
000158  000        NTBLMP = NFLOW(L+2)
000159  000        X      = X + TYPE(3,NTYP)
000160  000        IF(NTBLMP) 7900,7900,ZZ
000161  000   5200 HU = TPOL(5,TU)
000162  000        GO TO 7800
000163  000   5300 IF(NFRM .EQ. KP1) GO TO 7700
000164  000        SUM1 = 0.0
000165  000        SUM2 = 0.0
000166  000        CALL UPSUM1(NFRM,LOC1,J1,SUM1,SUM2)
000167  000        SHU1 = SUM1/SUM2
000168  000        CALL RPOL(NH,SHU1,STU1)
000169  000        KP1  = NFRM
```

```
000170      000     7700 HU = SHU1
000171      000          TU = STU1
000172      000     7800 CALL THPEQ(NTB,L,TYPE(1,NTYP),HU,TU,X)
000173      000     7900 TU = T(LMP)
000174      000          ASSIGN 5200 TO ZZ
000175      000     8000 CONTINUE
000176      000    C
000177      000     8500 CONTINUE
000178      000    C
000179      000     9000 CONTINUE
000180      000    C
000181      000          RETURN
000182      000          END

END ELT.


*HOG,P FLUMP
```

```
@ELT,L FLUMP
ELTOTT RL1870 02/28-03:19:49-(1,)
000001      000          SUBROUTINE FLUMP(L,NODES,JSW)
000002      000    C
000003      000          LOGICAL ERR
000004      000    C
000005      000          DIMENSION NODES(1)
000006      000    C
000007      000          COMMON /TAPE  / NIN, NOUT
000008      000          COMMON /CARD  / KRD, KOL, MXKOL
000009      000          COMMON /CIMAGE/ KARD(80)
000010      001           COMMON /FLOERR/ERR
000011      000    C
000012      000          L = 0
000013      000       15 CALL SKPB(JSW)
000014      000          GO TO(50,120), JSW
000015      000       50 IF(KARD(KOL) .NE. 1H( ) GO TO 75
000016      000          KOL = KOL + 1
000017      000          K = KOL
000018      000          CALL SUBN(N1,ISW)
000019      000          NODES(L+1) = -N1
000020      000          GO TO(55,85,95,105,94), JSW
000021      000       55 KOL = KOL + 1
000022      000          K = KOL
000023      000          CALL SUBN(NODES(L+2),ISW)
000024      000          GO TO(60,85,95,105,94), ISW
000025      000       60 N3 = 1
000026      000          IF(KARD(KOL) .EQ. 1H)) GO TO 65
000027      000          KOL = KOL + 1
000028      000          K = KOL
000029      000          CALL SUBN(N3,ISW)
000030      000          GO TO(65,85,95,105,94), ISW
000031      000       65 NODES(L+3) = N3
000032      000          IF(IABS(NODES(L+1)).GT.NODES(L+2)) NODES(L+3)=-IABS(NODES(L+3))
000033      000          L = L + 3
000034      000          GO TO 81
000035      000       75 K = KOL
000036      000          CALL SUBN(NODES(L+1),ISW)
000037      000          GO TO(80,85,105,105,79), ISW
000038      000       79 IF(NODES(L+1) .NE. 0) L = L + 1
000039      000          GO TO 125
000040      000       80 L = L + 1
000041      000       81 J = KOL
000042      000          DO 82 KOL=J,MXKOL
000043      000          IF(KARD(KOL) .EQ. 1H,) GO TO 83
000044      000       82 CONTINUE
000045      000          CALL CARDIN(JSW)
000046      000          GO TO(15,120), JSW
000047      000       83 KOL = KOL + 1
000048      000          GO TO 15
000049      000       85 WRITE(NOUT,90)
000050      000       90 FORMAT(58H0* * *  FLOATING POINT NUMBER INPUT FOR NODE NUMBER  * *
000051      000         1 * /)
000052      000          GO TO 104
000053      000       94 JSW = 2
000054      000       95 WRITE(NOUT,100)
000055      000      100 FORMAT(60H0* * *  END FOUND BEFORE COMPLETION OF MULTIPLE INPUT  *
```

```
000056      000        1 * * /)
000057      000     104 ERR = .TRUE.
000058      000     105 GO TO(110,120), JSW
000059      000     110 CALL SKPTE(JSW)
000060      000     120 RETURN
000061      000     125 JSW = 2
000062      000         RETURN
000063      000         END

END ELT.


*HOG,P FLUX
```

```
@ELT,L FLUX
ELT077 RLI870 02/28-03:19:50-(0,)
000001      000          SUBROUTINE FLUX(NFLXTP,DATA,NCRV,DQTIME,QTIME)
000002      000    C
000003      000          DIMENSION DATA(1)
000004      000    C
000005      000          COMMON /FIXCON/ TIMEN
000006      000    C
000007      000          EQUIVALENCE (D,N)
000008      000    C
000009      000    C
000010      000          IF(QTIME .GE. TIMEN) RETURN
000011      000          IF(NFLXTP .GT. 0) READ(NFLXTP) FLXTIM
000012      000          NFLXTP = IABS(NFLXTP)
000013      000       10 READ(NFLXTP) (NP, ((DATA(I+2*J*NP+2*J-2*NP-1),I=1,NP,2),
000014      000         1                   (DATA(I+2*J*NP+2*J-2*NP-1),I=2,NP,2),J=1,NCRV)
000015      000          READ(NFLXTP) FLXTIM
000016      000          QTIME = FLXTIM + DQTIME
000017      000          IF(QTIME .LE. TIMEN) GO TO 10
000018      000          WRITE(6,20) QTIME
000019      000       20 FORMAT(22H FLUX TABLES ENDING AT  G11.5, 15H HAVE BEEN READ )
000020      000          LOC = 1
000021      000          D = DATA(LOC)
000022      000          IC = N
000023      000          DO 30 J=1,IC,2
000024      000          DATA(LOC+J) = DATA(LOC+J) + DQTIME
000025      000       30 CONTINUE
000026      000          LOC = LOC + IC + 1
000027      000       40 CONTINUE
000028      000          NFLXTP = -NFLXTP
000029      000          RETURN
000030      000          END

END ELT.


@HOG,P FWDBCK
```

```
@ELT,L FWDBCK
ELTOT7 RLIB70 02/28-03;19;51-(1,)
000001       000          SUBROUTINE FWDBCK                                                  FWDBCK
000002       000    C                                                                        FWDBCK
000003       000    C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++FWDBCK
000004       000    C                                                                        FWDBCK
000005       000    C         THIS SUBROUTINE USES IMPLICIT FORWARD-BACKWARD DIFFERENCING    FWDBCK
000006       000    C      TO PERFORM TRANSIENT THERMAL ANALYSIS. THE FOURIER EQUATION IS    FWDBCK
000007       000    C      FIRST WRITTEN IN -FORWARD- FINITE DIFFERENCE FORM AND THEN IN     FWDBCK
000008       000    C      -BACKWARD- FINITE DIFFERENCE FORM.  THESE TWO FORMS ARE ADDED     FWDBCK
000009       000    C      AND THE RESULTING EQUATION IS SOLVED.  TO ACCOUNT FOR             FWDBCK
000010       000    C      NON-LINEARITIES THE GENERAL QUARTIC FORMULA IS USED WHEN           FWDBCK
000011       000    C      NECESSARY.                                                        FWDBCX
000012       000    C                                                                        FWDBCK
000013       000    C+++++++++++++++++++                                                     FWDBCK
000014       000    C                                                                        FWDBCK
000015       000    C      THE DEFINITIONS OF THE CONTROL CONSTANTS ARE.....                 FWDBCK
000016       000    C                                                                        STDSTL
000017       000    C     CONTROL CONSTANT 1 CONTAINS THE NEW PROBLEM TIME          (TIMEN)
000018       000    C     CONTROL CONSTANT 2 CONTAINS THE TIME STEP USED            (DTIMEU)
000019       000    C     CONTROL CONSTANT 3 CONTAINS THE PROBLEM STOP TIME         (TIMEND)
000020       000    C     CONTROL CONSTANT 4 CONTAINS THE TIME STEP FACTOR,EXPLICIT (CSGFAC)
000021       000    C     CC5 IS THE INPUT NUMBER OF ITERATION DO LOOPS, INTEGER    (NLOOP)
000022       000    C     CC6 CONTAINS THE DIFFUSION TEMPERATURE CHANGE ALLOWED     (DTMPCA)
000023       000    C     CC7 CONTAINS THE OUTPUT EACH ITERATION SWITCH             (OPEITR)
000024       000    C     CC8 CONTAINS THE MAXIMUM ALLOWED TIME STEP                (DTIMEH)
000025       000    C     CC9 CONTAINS THE NEW ARITHMETIC TEMP. DAMPING FACTOR      (DAMPA)
000026       000    C     CC10 CONTAINS THE NEW DIFFUSION TEMP  DAMPING FACTOR      (DAMPD)
000027       000    C     CC11 CONTAINS THE MAXIMUM ALLOWED ARITHMETIC TEMP. CHANGE (ATMPCA)
000028       000    C     CC12 CONTAINS THE BACKUP SWITCH CHECKED AFTER VARIABLES    (BACKUP)
000029       000    C     CC13 CONTAINS THE PRESENT TIME OR PROBLEM START TIME       (TIMEO)
000030       000    C     CC14 CONTAINS THE MEAN TIME BETWEEN AN ITERATION           (TIMEM)
000031       000    C     CC15 CONTAINS THE DIFFUSION TEMPERATURE CHANGE CALCULATED (DTMPCC)
000032       000    C     CC16 CONTAINS ARITHMETIC TEMPERATURE CHANGE CALCULATED     (ATMPCC)
000033       000    C     CONTROL CONSTANT 17 IS RESERVED FOR THE C/SG MINIMUM       (CSGMIN)
000034       000    C     CONTROL CONSTANT 18 CONTAINS THE OUTPUT INTERVAL           (OUTPUT)
000035       000    C     CC19 CONTAINS THE ARITHMETIC RELAXATION CRITERIA ALLOWED   (ARLXCA)
000036       000    C     CC20 CONTAINS THE NUMBER OF RELAXATION LOOPS USED,INTEGER  (LOOPCT)
000037       000    C     CC21 CONTAINS THE MINIMUM ALLOWED TIME STEP                (DTIMEL)
000038       000    C     CC22 IS FOR THE INPUT TIME STEP IMPLICIT                   (DTIMEI)
000039       000    C     CC23 CONTAINS THE C/SG MAXIMUM                             (CSGMAX)
000040       000    C     CC24 CONTAINS THE C/SG RANGE ALLOWED                       (CSGRAL)
000041       000    C     CC25 CONTAINS THE C/SG RANGE CALCULATED                    (CSGRCL)
000042       000    C     CC26 CONTAINS THE DIFFUSION RELAXATION CRITERIA ALLOWED    (DRLXCA)
000043       000    C     CC27 CONTAINS THE DIFFUSION RELAXATION CHANGE CALCULATED   (DRLXCC)
000044       000    C     CC28 CONTAINS THE LINE COUNTER, INTEGER                    (LINECT)
000045       000    C     CC29 CONTAINS THE PAGE COUNTER, INTEGER                    (PAGECT)
000046       000    C     CC30 CONTAINS ARITHMETIC RELAXATION CHANGE CALLULATED      (ARLXCC)
000047       000    C     CC31 IS INDICATOR, 0=THERMAL SPCS,1=THERMAL LPCS,2=GENERAL (LSPCS)
000048       000    C     CC32 CONTAINS THE ENERGY BALANCE OF THE SYSTEM, IN - OUT   (ENGBAL)
000049       000    C     CC33 CONTAINS THE DESIRED ENERGY BALANCE, USER INPUT       (BALENG)
000050       000    C     CC34 CONTAINS THE NOCOPY SWITCH FOR MATRIX USERS           (NOCOPY)
000051       000    C     CC35 CONTAINS RELATIVE NODE NUMBER OF CSGMIN
000052       000    C     CC36 CONTAINS RELATIVE NODE NUMBER OF DTMPCC
000053       000    C     CC37 CONTAINS RELATIVE NODE NUMBER OF ARLXCC
000054       000    C     CC38 CONTAINS RELATIVE NODE NUMBER OF ATMPCC
000055       000    C     CC39 CONTAINS RELATIVE NODE NUMBER OF DRLXCC
```

```
000056  000  C     CC40-41-42-43  CONTAIN DUMMY INTEGER CONSTANTS       (J-K-L-MTEST)
000057  000  C     CC44-45-46-47-48 CONTAIN DUMMY FLOATING CONSTANTS  (R-S-T-U-VTEST)
000058  000  C     CC49 IS THE QUASI-LINEARIZATION INTERVAL FOR CINDSM       (LAXFAC)
000059  000  C     CC50 CONTAINS THE STEFAN-BOLTZMANN CONSTANT                (SIGMA)
000060  000  C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++STDSTL
000061  000  C                                                                    FWDBCK
000062  000  C                                                                    FWDBCK
000063  000  C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++FWDBCK
000064  000  C                                                                    FWDBCK
000065  000  C                                                                    STDSTL
000066  001        LOGICAL FLOW
000067  001        COMMON /FDIMNS/ NTYP, NSYS
000068  000        COMMON /TITLE/H(1) /TEMP/T(1) /CAP/C(1) /SOURCE/Q(1) /COND/G(1)
000069  000        COMMON /PC1/NSQ1(1) /PC2/NSQ2(1) /KONST/K(1) /ARRAY/A(1)
000070  000        COMMON /DIMENS/ NND,NNA,NNT,NGT,NCT,NAT,LSQ1,LSQ2
000071  000        DIMENSION  KON(50)
000072  000        DIMENSION CON(1),XK(1),NX(1)
000073  000        EQUIVALENCE (KON(1),CON(1)),(K(1),XK(1)),(X(1),NX(1))
000074  000         EQUIVALENCE (NT,LTA),(NG,LG)
000075  000         COMMON /XSPACE/    NOIM    , NTH     , X(1)                    FWDBCK
000076  000         COMMON /FIXCON/ TIMEN , DTIMEU  , TIMEND  , CSGFAC  , NLOOP   ,  FWDBCK
000077  000        1     DTMPCA  , OPEITR  , DTIMEH  , DAMPA   , DAMPD    , ATMPCA ,  FWDBCK
000078  000        2     BACKUP  , TIMEO   , TIMEM   , DTMPCC  , ATMPCC   , CSGMIN ,  FWDBCK
000079  000        3     OUTPUT  , ARLXCA  , LOOPCT  , DTIMEL  , DTIMEI   , EXTLIM  ,  FWDBCK
000080  000        4     PERIOD  , NPRD    , DRLXCA  , DRLXCC  , NLINE    , NPAGE   ,  FWDBCK
000081  000        5     ARLXCC  , LSPCS   , ENGBAL  , BALENG  , NOCOPY   , NCSGM   ,  FWDBCK
000082  000        6     NDTMPC  , NAARLXC , NATMPC  , NOARLXC ,  JTEST   ,  KTEST  ,
000083  000        7      LTEST  , MTEST   ,  RTEST  ,  STEST  ,  TTEST   ,  UTEST  ,
000084  000        8       VTEST , LAXFAC  ,  SIGMA
000085  000         EQUIVALENCE  (CON(1),TIMEN)
000086  000  C                                                                    FWDBCK
000087  000  C                                                                    FWDBCK
000088  000  C                                                                    FWDBCK
000089  000        DOUBLE PRECISION R16,R3,RSR2,CC,FF,GG,ZZ
000090  000        DATA IFIRST/0/
000091  000        NOUT =6
000092  000  C        INITIALIZING CONTROL CONSTANTS.                             FWDBCK
000093  000  C                                                                    FWDBCK
000094  001        TZERO = -460.
000095  000         IF(OUTPUT.LE.0.0)              GO TO 500                     FWDBCK
000096  000         IF(DTIMEH.LE.0.0)              DTIMEH = 1.0E+8
000097  000         IF(DTIMEI.LE.0.0)              GO TO 510                     FWDBCK
000098  000         IF(NND.LE.0)                   GO TO 560                     FWDBCK
000099  000         IF(DRLXCA.LE.0.0)              GO TO 520                     FWDBCK
000100  000         IF(DTMPCA.LE.0.0)              DTMPCA = 1.0E+8
000101  000         IF(NNA.LE.0)                   GO TO 400                     FWDBCK
000102  000         IF(ARLXCA.LE.0.0)              GO TO 530                     FWDBCK
000103  000         IF(ATMPCA.LE.0.0)              ATMPCA = 1.0E+8
000104  000    400 IF (CON(50) .LE. 0.) CON(50) =1.
000105  000         IF(NLOOP.LE.0)                 GO TO 540                     FWDBCK
000106  000         IF(LSPCS.NE.1)                 GO TO 550
000107  000         GO TO 700                                                    FWDBCK
000108  000    500  WRITE(NOUT,501)                                             FWDBCK
000109  000    501  FORMAT('0++ERROR++ OUTPUT MUST BE SPECIFIED')
000110  000         GO TO 600                                                    FWDBCK
000111  000    510  WRITE (NOUT,511)                                           FWDBCK
000112  000    511  FORMAT('0++ERROR++ DTIMEI MUST BE SPECIFIED')
```

```
000113  000        GO TO 600                                              FWDBCK
000114  000   520  WRITE (NOUT,521)                                       FWDBCK
000115  000   521  FORMAT('0++ERROR++ DRLXCA MUST BE SPECIFIED')
000116  000        GO TO 600                                              FWDBCK
000117  000   530  WRITE (NOUT,531)                                       FWDBCK
000118  000   531  FORMAT('0++ERROR++ ARLXCA MUST BE SPECIFIED')
000119  000        GO TO 600                                              FWDBCK
000120  000   540  WRITE (NOUT,541)                                       FWDBCK
000121  000   541  FORMAT('0++ERROR++ NLOOP MUST BE SPECIFIED')
000122  000        GO TO 600                                              FWDBCK
000123  000   550  WRITE (NOUT,551)                                       FWDBCK
000124  000   551  FORMAT('0++ERROR++ -FWDBCK- REQUIRES -LPCS-')
000125  000        GO TO 600                                              FWDBCK
000126  000   560  WRITE (NOUT,561)                                       FWDBCK
000127  000   561  FORMAT('0++ERROR++  -FWDBCK- REQUIRES AT LEAST ONE DIFFUSION'
000128  000       $        ' NODE')
000129  000   600  CALL OUTCAL                                            FWDBCK
000130  000       RETURN
000131  000   700  CALL TOPLIN                                           FWDBCK
000132  000        ITRAN       = 2                                        FWDBCK
000133  000        ITNTH       = NTH+1                                    FWDBCK
000134  000        NN          = NND+1                                    FWDBCK
000135  000        NNC         = NND+NNA                                  FWDBCK
000136  000        R16 = 1.0D0/16.0D0
000137  000        R3  = 1.0D0/3.0D0
000138  000        RSR2= 1.0D0/SQRT(2.0D0)
000139  001        FLOW = .FALSE.
000140  001        NSP = 0
000141  001        IXF = NTH
000142  001        IF(NSYS .LT. 1) GO TO 750
000143  001        FLOW = .TRUE.
000144  001        NSP = NNT
000145  001        DO 725 I=1,NNT
000146  001        NX(IXF+I) = 0
000147  001   725  CONTINUE
000148  000   C                                                           FWDBCK
000149  000   C++++++++++                                                 FWDBCK
000150  000   C                                                           FWDBCK
000151  000   C        SETTING UP DYNAMIC STORAGE (3*NNC)                 FWDBCK
000152  000   C                                                           FWDBCK
000153  001   750  IX1 = IXF + NSP
000154  001        IX2         = IX1+NNT                                  FWDBCK
000155  000        IX3         = IX2+NNC                                  FWDBCK
000156  000        IXL         = NDIM                                     FWDBCK
000157  000        J           = IX3+NNC                                  FWDBCK
000158  000        WRITE (NOUT,801)   J                                   FWDBCK
000159  000   801  FORMAT('0++NOTE++ -FWDBCK- REQUIRES'I6,
000160  000       $        ' WORDS OF DYNAMIC STORAGE'/)
000161  000        NLINE       = NLINE+3                                  FWDBCK
000162  000   900  NTH         = NTH+J                                    FWDBCK
000163  000        NDIM        = NDIM-J                                   FWDBCK
000164  000        IF(NDIM.GE.0)                  GO TO 1100              FWDBCK
000165  000        NDIM        = IABS(NDIM)                               FWDBCK
000166  000        WRITE (NOUT,1001)  NDIM                                FWDBCK
000167  000  1001  FORMAT('0++ERROR++ -FWDBCK- REQUIRES'I5,' MORE WORDS OF DYNAMIC'
000168  000       $        ' STORAGE')
000169  000        CALL OUTCAL                                            VER1+9
```

```
000170  000        RETURN
000171  000  C                                                              FWDBCK
000172  000  C++++++++++                                                    FWDBCK
000173  000  C        SET UP INITIAL VALUES, ZERO OUT SOURCE TERMS CALL -VARBL1-  FWDBCK
000174  000  C        AND CALL -OUTCAL- FOR START CONDITIONS                 FWDBCK
000175  000  C                                                              FWDBCK
000176  000  C                                                              FWDBCK
000177  000   1100  IDIM        = NDIM                                       FWDBCK
000178  000         ITH         = NTH                                        FWDBCK
000179  000         TIMEN       = TIMEO                                      FWDBCK
000180  000         DTIMEU      = 0.0                                        FWDBCK
000181  000         TIMEM       = TIMEO                                      FWDBCK
000182  000         SUMDTP      = 0.0                                        FWDBCK
000183  000         DO 1140   I = 1,NNC                                      FWDBCK
000184  000   1140  G(I)        = 0.0                                        FWDBCK
000185  000         CALL VARBL1                                              FWDBCK
000186  000         CALL OUTCAL                                              FWDBCK
000187  000  C                                                              FWDBCK
000188  000  C++++++++++                                                    FWDBCK
000189  000  C                                                              FWDBCK
000190  000  C        COMPUTING TIME INTERVAL AND OUTPUT INTERVAL            FWDBCK
000191  000  C                                                              FWDBCK
000192  000   1300  DTIMEU      = AMIN1(DTIMEI,DTIMEH)                       FWDBCK
000193  000         CHKSUM      = ( SUMDTP + DTIMEU )
000194  000        FLD(35,1,CHKSUM) =1
000195  000         IF(CHKSUM.LT.OUTPUT)          GO TO 1400                 FWDBCK
000196  000         DTIMEU      = OUTPUT-SUMDTP                              FWDBCK
000197  000         GO TO 1500                                               FWDBCK
000198  000   1400  IF(CHKSUM+DTIMEU.LT.OUTPUT)  GO TO 1500                 FWDBCK
000199  000         DTIMEU      = (OUTPUT-SUMDTP)/2.0                        FWDBCK
000200  000   1500  TIMEN       = TIMEO+DTIMEU                               FWDBCK
000201  000         IF(TIMEN.LT.TIMEND)           GO TO 1540                 FWDBCK
000202  000         TIMEN       = TIMEND                                     FWDBCK
000203  000         DTIMEU      = TIMEND-TIMEO                               FWDBCK
000204  000         SUMDTP      = OUTPUT-DTIMEU                              VER1+11
000205  000   1540  TIMEM       = (TIMEN+TIMEO)/2.0                          FWDBCK
000206  000         IF(DTIMEU.GT.DTIMEL)          GO TO 1600                 FWDBCK
000207  000         WRITE (NOUT,1541)  DTIMEU                                FWDBCK
000208  000   1541  FORMAT('0**ERROR** DTIMEU IS LESS THAN OR EQUAL TO DTIMEL,'
000209  000        $         '  DTIMEU ='G13.6)
000210  000         CALL OUTCAL                                              VER1+9
000211  000        RETURN
000212  000   1600  DTD2CC      = DTIMEU/2.0                                 FWDBCK
000213  000         CSGMIN      = 1.0E+8
000214  000         ASSIGN 1650 TO IPASS                                     FWDBCK
000215  000         ASSIGN 4001 TO ITER
000216  000          DIFMAX     = 50.0
000217  000         IF(ABS(EXTLIM).NE.0.0)          DIFMAX = ABS(EXTLIM)
000218  000  C                                                              FWDBCK
000219  000  C+++++++++                                                     FWDBCK
000220  000  C       RELAX THE DIFFUSION AND ARITHMETIC NODES               FWDBCK
000221  000  C                                                              FWDBCK
000222  000         DO 5001 LOOPCT = 1,NLOOP
000223  000        J1 = 0
000224  000         ORLXCC      = 0.0                                        FWDBCK
000225  000         ARLXCC      = 0.0                                        FWDBCK
000226  001        KOP = 0
```

```
000227  001        IF(OPEITR.NE.0.0) KOP = 1
000228  000         GO TO IPASS                                              FWDBCK
000229  000  C                                                               FWDBCK
000230  000  C        ON FIRST PASS ZERO OUT SOURCE TERMS AND CALL -VARBL1-   FWDBCK
000231  000  C        TO SET CONDITIONS FOR THIS INTERVAL                     FWDBCK
000232  000  C                                                               FWDBCK
000233  001   1650 ASSIGN 2550 TO IPASS
000234  000        J2 = 1
000235  000         BACKUP      = 0.0                                        FWDBCK
000236  000         DO 1700   I = 1,NNC                                      FWDBCK
000237  000   1700  Q(I)        = 0.0                                        FWDBCK
000238  000         CALL VARBL1                                              FWDBCK
000239  000         IF(BACKUP.NE.0.0)              GO TO 1300                FWDBCK
000240  000  C                                                               FWDBCK
000241  000  C        CONVERT TEMPERATURES TO RANKINE AND LOAD DYNAMIC STORAGE FWDBCK
000242  000  C                                                               FWDBCK
000243  000         DO 1800   I = 1,NNT                                      FWDBCK
000244  000   1800  T(I)        = T(I)+460.0                                 FWDBCK
000245  001         DO 1850   I = 1,NNT                                      FWDBCK
000246  000   1850  X(IX1+I)    = T(I)                                       FWDBCK
000247  001  C
000248  001        IF(FLOW) CALL FLUID(2,IX1,IXF,TZERO,KOP)
000249  000  C                                                               FWDBCK
000250  000  C        ON FIRST PASS SET UP THE CONSTANT TERMS FOR THE DIFFUSION FWDBCK
000251  000  C        NODES IN THE CAPACITANCE AND IN THE SOURCE ARRAYS.      FWDBCK
000252  000  C                                                               FWDBCK
000253  000         DO 2500   I = 1,NND                                      FWDBCK
000254  000         TI          = T(I)                                       FWDBCK
000255  000        T(I) = T(I) -460.
000256  000        IF(FLD(1,1,NSO1(J1+1)).EQ.0) GO TO 2000
000257  000        NTYPE = FLD(0,5,NSO2(J2))
000258  000        LA = FLD(5,17,NSO2(J2))
000259  000        LK = FLD(22,14,NSO2(J2))
000260  000        GO TO (1005,1010,1015,1020,1025,1030,1035,1040,1045), NTYPE
000261  000   1005 CALL DIDIWM(T(I),A(LA),XK(LK),C(I))
000262  000        GO TO 1999
000263  000   1010 CALL DIDIWM(T(I),A(LA),XK(LK),C1)
000264  000   1012 J2 = J2+1
000265  000        LA = FLD(5,17,NSO2(J2))
000266  000        LK = FLD(22,14,NSO2(J2))
000267  000        CALL DIDIWM(T(I),A(LA),XK(LK),C2)
000268  000        GO TO 1998
000269  000   1015 C1 = XK(LK)*XK(LA)
000270  000        GO TO 1012
000271  000   1020 CALL DIDIWM(T(I),A(LA),XK(LK),C1)
000272  000        J2 = J2+1
000273  000        LA = FLD(5,17,NSO2(J2))
000274  000        LK = FLD(22,14,NSO2(J2))
000275  000        C2 = XK(LK)*XK(LA)
000276  000        GO TO 1998
000277  000   1025 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C(I))
000278  000        GO TO 1999
000279  000   1030 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C1)
000280  000   1032 J2 = J2+1
000281  000        LA = FLD(5,17,NSO2(J2))
000282  000        LK = FLD(22,14,NSO2(J2))
000283  000        CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C2)
```

```
000284  000        GO TO 1998
000285  000   1035 C1 = XK(LK)*XK(LA)
000286  000        GO TO 1032
000287  000   1040 CALL FLYAWM(A(LA),T(I),A(LA+1),XK(LK),C1)
000288  000        J2 = J2+1
000289  000        LA = FLD(5,17,NSQ2(J2))
000290  000        LK = FLD(22,14,NSQ2(J2))
000291  000        C2 = XK(LK)*XK(LA)
000292  000        GO TO 1998
000293  000   1045 CALL D2D1WM(T(I),CON(14),A(LA),XK(LK),C(I))
000294  000        GO TO 1999
000295  000   1998 C(I) = C1+C2
000296  000   1999 J2 = J2+1
000297  000   2000 CONTINUE
000298  000         C(I)        = C(I)/DTD2CC                          FWDBCK
000299  000         CIP         = C(I)                                 FWDBCK
000300  000        IF(FLD(4,1,NSQ1(J1+1)).EQ.0) GO TO 5000
000301  000        NTYPE = FLD(0,5,NSQ2(J2))
000302  000        LA = FLD(5,17,NSQ2(J2))
000303  000        LK = FLD(22,14,NSQ2(J2))
000304  000        GO TO (4005,4010,4015,4020,4025,4030,4035,4040,4030,   VERS  5
000305  000       *       4050,4050,4050),NTYPE                          VERS  5
000306  000   4005 Q(I) = XK(LK)+Q(I)
000307  000        GO TO 4999
000308  000   4010 Q1 = 0.0
000309  000   4012 CALL D1D1WM(T(I),A(LA),XK(LK),Q2)
000310  000        GO TO 4998
000311  000   4015 Q1 = 0.0
000312  000   4017 CALL D1D1WM(CON(14),A(LA),XK(LK),Q2)
000313  000        GO TO 4998
000314  000   4020 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000315  000   4022 J2 = J2+1
000316  000        LA = FLD(5,17,NSQ2(J2))
000317  000        LK = FLD(22,14,NSQ2(J2))
000318  000        GO TO 4017
000319  000   4025 Q1 = XK(LK)*XK(LA)
000320  000        GO TO 4022
000321  000   4030 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000322  000        J2 = J2+1
000323  000        LA = FLD(5,17,NSQ2(J2))
000324  000        LK = FLD(22,14,NSQ2(J2))
000325  000        Q2 = XK(LK)*XK(LA)
000326  000        GO TO 4998
000327  000   4035 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000328  000   4037 J2 = J2+1
000329  000        LA = FLD(5,17,NSQ2(J2))
000330  000        LK = FLD(22,14,NSQ2(J2))
000331  000        GO TO 4012
000332  000   4040 Q1 = XK(LK)*XK(LA)
000333  000        GO TO 4037
000334  000   4050 J2=J2+1                                             VERS  5
000335  000        JPSLA=FLD(5,17,NSQ2(J2))                           VERS  5
000336  000        JPSLK=FLD(22,14,NSQ2(J2))                          VERS  5
000337  000        SPJTIM=CON(14)*XK(JPSLA)*XK(JPSLK)                 VERS  5
000338  000        CALL D11MCY(XK(JPSLK),SPJTIM,A(LA),XK(LK),Q1)      VERS  5
000339  000        Q2=0.0                                             VERS  5
000340  000        GO TO 4998                                         VERS  5
```

```
000391  000   4998 Q(I) = Q1+Q2+Q(I)
000392  000   4999 J2 = J2+1
000393  000   5000 CONTINUE
000394  000         QI          = +2.0*Q(I)+TI*CIP                          FWDBCK
000395  000         GSUM        = 0.0                                       FWDBCK
000396  000         GSUML       = 0.0                                       FWDBCK
000397  000         GSUMR       = 0.0                                       FWDBCK
000398  000         ASUM        = 0.0                                       FWDBCK
000399  001       IF(.NOT. FLOW) GO TO 70
000350  001       LMP = NX(IXF+I)
000351  001       IF(LMP .EQ. 0) GO TO 70
000352  001       HA = X(IXF+LMP)
000353  001       QSUM = HA*T(LMP)
000354  001       QI = QI + HA*X(IX1+LMP)
000355  001       GSUML = HA
000356  000    70 J1 = J1+1
000357  000       LG = FLD(5,16,NSQ1(J1))
000358  001       IF(LG.EQ.0) GO TO 81
000359  000       LTA = FLD(22,14,NSQ1(J1))
000360  000       T(LTA) = T(LTA) -460.
000361  000       LTAE = LTA+IX1
000362  000       IF(FLD(2,1,NSQ1(J1)),EQ.0) GO TO 3000
000363  000       NTYPE = FLD(0,5,NSQ2(J2))
000364  000       LA = FLD(5,17,NSQ2(J2))
000365  000       LK = FLD(22,14,NSQ2(J2))
000366  000       GOTO(2005,2010,2015,2020,2025,2030-2035,2040,2045,2050,2055,
000367  000      1     2060,2065,2070,2073,2070) , NTYPE                    VER 6
000368  000   2005 TM = (T(I)+T(LTA))/2.0
000369  000   2007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000370  000       GO TO 2999
000371  000   2010 TM = T(I)
000372  000       GO TO 2007
000373  000   2015 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000374  000   2017 J2 = J2+1
000375  000       LA = FLD(5,17,NSQ2(J2))
000376  000       LK = FLD(22,14,NSQ2(J2))
000377  000       CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000378  000       GO TO 2998
000379  000   2020 G1 = XK(LK)*XK(LA)
000380  000       GO TO 2017
000381  000   2025 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000382  000       J2 = J2+1
000383  000       LA = FLD(5,17,NSQ2(J2))
000384  000       LK = FLD(22,14,NSQ2(J2))
000385  000       G2 = XK(LK)*XK(LA)
000386  000       GO TO 2998
000387  000   2030 TM = (T(I)+T(LTA))/2.0
000388  000   2032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000389  000       GO TO 2999
000390  000   2035 TM = T(I)
000391  000       GO TO 2032
000392  000   2040 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),G1)
000393  000   2042 J2 = J2+1
000394  000       LA = FLD(5,17,NSQ2(J2))
000395  000       LK = FLD(22,14,NSQ2(J2))
000396  000       CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000397  000       GO TO 2998
```

```
000398  000    2045 G1 = XK(LK)*XK(LA)
000399  000         GO TO 2042
000400  000    2050 CALL PLYANM(A(LA),T(I),A(LA+1),XK(LK),G1)
000401  000         J2 = J2+1
000402  000         LA = FLD(5,17,NSQ2(J2))
000403  000         LK = FLD(22,14,NSQ2(J2))
000404  000         G2 = XK(LK)*XK(LA)
000405  000         GO TO 2998
000406  000    2055 TM = (T(I)+T(LTA))/2.0
000407  000         CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000408  000         GO TO 2999
000409  000    2060 TM = T(LTA)
000410  000         GO TO 2007
000411  000    2065 TM = T(LTA)
000412  000         GO TO 2032
000413  000    2070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                    VER 6
000414  000    2071 TM  = (T(I) + T(LTA)) / 2.0                            VER 6
000415  000          J2 =  J2 + 1                                          VER 6
000416  000         LA  = FLD(5,17,NSQ2( J2))                              VER 6
000417  000         LK  = FLD(22,14,NSQ2( J2))                             VER 6
000418  000         IF(NTYPE .EQ. 16)  GO TO 2075                          VER 6
000419  000         CALL D2D1WM( TM, G1, A(LA), XK(LK), G(LG))             VER 6
000420  000         GO TO 2999                                             VER 6
000421  000    2073 G1 = XK(LA) *  XK(LK)                                  VER 6
000422  000         GO TO 2071                                             VER 6
000423  000    2075 G(LG) = G1 * XK(LA)  *  XK(LK)                         VER 6
000424  000         GO  TO 2999                                            VER 6
000425  000    2998 G(LG) = 1./(1./G1+1./G2)
000426  000         IF(FLD(3,1,NSQ1(J1)).EQ.1) G(LG) = G1*G2
000427  000    2999 J2 = J2+1
000428  000    3000 CONTINUE
000429  000         T(LTA) = T(LTA) +460.
000430  000          TNT          = T(NT)                                   FWDBCK
000431  000          GNG          = G(NG)                                   FWDBCK
000432  000          TNTO         = TNT                                     FWDBCK
000433  000          IF(NT.LE.NNC)                    TNTO = X(IX1+NT)     FWDBCK
000434  000         IF (FLD(3,1,NSQ1(J1)).NE. 0) GO TO 2100
000435  000          GSUML        = GSUML+GNG                               FWDBCK
000436  000          GSUM         = GSUM+GNG                                FWDBCK
000437  000          QI           = QI+GNG*TNTO                             FWDBCK
000438  000          QSUM         = QSUM+GNG*TNT                            FWDBCK
000439  000          GO TO 80
000440  000    2100 GNG = GNG *SIGMA
000441  000         GSUMR = GSUMR +GNG
000442  000          GSUM         = GSUM+GNG*(TNT+TI)*((TNT+TI)**2-2.0*TNT*TI)   FWDBCK
000443  000          QI           = QI+GNG*TNTO**4                          FWDBCK
000444  000          QSUM         = QSUM+GNG*TNT**4                         FWDBCK
000445  000    C     CHECK FOR LAST CONDUCTOR
000446  000       80 IF(NSQ1(J1).GT.0) GO TO 70
000447  001       81 T(I) = T(I) + 460.0
000448  000          IF(GSUMR.EQ.0.0)                 GO TO 2300            FWDBCK
000449  000    C                                                            FWDBCK
000450  000    C              SET UP CONSTANTS FOR USE SUBSEQUENT ITERATIONS AND   FWDBCK
000451  000    C              THEN SOLVE FOR TEMPERATURES BY THE GENERAL QUARTIC   FWDBCK
000452  000    C              FORMULA                                       FWDBCK
000453  000    C                                                            FWDBCK
000454  000          Q(I)         = TI*(TI**3+GSUMR+GSUML)-QI               FWDBCK
```

```
000455  000          CC          = (CIP+GSUML)/GSUMR                          FWDBCK
000456  000          FF          = -CC**2*R16                                 FWDBCK
000457  000          GG          = (Q(I)-QSUM)/GSUMR*R3                       FWDBCK
000458  000          GG          = FF**2-GG**3                                FWDBCK
000459  000          IF(GG.GT.0.0)                    GO TO 2250              FWDBCK
000460  000          T2          = 0.0                                        FWDBCK
000461  000          GO TO 2420                                               FWDBCK
000462  000   2270   GG          = SQRT(GG)                                   FWDBCK
000463  000          ZZ          = SQRT((GG-FF)**R3-(GG+FF)**R3)              FWDBCK
000464  000          T2          = RSR2*(-ZZ+SQRT(-ZZ**2+CC*RSR2/ZZ))         FWDBCK
000465  000          GO TO 2420                                               FWDBCK
000466  000   2300   IF(GSUML.EQ.0.0)                GO TO 2399
000467  000   C                                                               FWDBCK
000468  000   C            SET UP CONSTANTS FOR USE SUBSEQUENT ITERATIONS AND  FWDBCK
000469  000   C            THEN SOLVE FOR TEMPERATURES BY A LINEAR EQUATION    FWDBCK
000470  000   C                                                               FWDBCK
000471  000          Q(I)        = -TI*GSUML+QI                               FWDBCK
000472  000          T2          = (Q(I)+QSUM)/(CIP+GSUML)                    FWDBCK
000473  000          GO TO 2420                                               FWDBCK
000474  000   2399 T2 = T(I)
000475  000        IF (IFIRST .NE. 0) GO TO 2420
000476  000   2400  WRITE(NOUT,2401) I                                        FWDBCK
000477  000   2401  FORMAT('0**ERROR**  THE SUM OF THE CONDUCTORS ATTACHED TO *
000478  000               ' RELATIVE NODE*I5,* IS EQUAL OR LESS THAN ZERO')
000479  000        IF (I.LE. NND) GO TO 2420
000480  000        GO TO 3750
000481  000   C                                                               FWDBCK
000482  000   C       COMPUTE TEMPERATURE CHANGE AND SET THE NEW TEMPERATURE   FWDBCK
000483  000   C                                                               FWDBCK
000484  000   2420   T1          = T2-TI                                      FWDBCK
000485  000          T(I)        = T2                                         FWDBCK
000486  000          IF(ABS(DRLXCC).GE.ABS(T1))       GO TO 2450              FWDBCK
000487  000          DRLXCC      = T1                                         FWDBCK
000488  000          NDRLXC      = I                                          FWDBCK
000489  000   C                                                               FWDBCK
000490  000   C       COMPUTE CSGMIN FOR INFORMATION ONLY                      FWDBCK
000491  000   C                                                               FWDBCK
000492  001   2450 IF(GSUM.EQ.0.0) GO TO 2500
000493  001        CSG = CIP * DTD2CC/GSUM
000494  000          IF(CSG.GE.CSGMIN)                GO TO 2500              FWDBCK
000495  000          CSGMIN      = CSG                                        FWDBCK
000496  000          NCSGM       = I                                          FWDBCK
000497  000   2500   CONTINUE                                                 FWDBCK
000498  000          GO TO 3350                                               FWDBCK
000499  000   C                                                               FWDBCK
000500  000   C       ON SECOND AND SUBSEQUENT ITERATIONS USE CONSTANTS CREATED FWDBCK
000501  000   C       ON FIRST ITERATION                                       FWDBCK
000502  000   C                                                               FWDBCK
000503  000   C                                                               FWDBCK
000504  000   C         MAKE AN ITERATION PASS ON THE DIFFUSION NODES          FWDBCK
000505  000   C                                                               FWDBCK
000506  001   2550 IF(FLOW) CALL FLUID(2,IX1,IXF,TZERO,KOP)
000507  001        CSG = CIP * DTD2CC/GSUM
000508  001        CSG = CIP * DTD2CC/GSUM
000509  000   2600   DO 3300   I = 1,NND                                       FWDBCK
000510  000          TI          = T(I)                                       FWDBCK
000511  000          CIP         = C(I)                                       FWDBCK
```

```
000512  000          GSUM        = 0.0                                    FWDBCK
000513  000          GSUML       = 0.0                                    FWDBCK
000514  000          GSUMR       = 0.0                                    FWDBCK
000515  000          QSUM        = 0.0                                    FWDBCK
000516  001
000517  001
000518  001          IF(.NOT. FLOW) GO TO 115
000519  001          LMP = NX(IXF+I)
000520  001          IF(LMP .EQ. 0) GO TO 115
000521  001          HA = X(IXF+LMP)
000522  001          QSUM = HA*T(LMP)
000523  001          GSUML = HA
000524  001          GSUM  = HA
000525  000      115 J1 = J1+1
000526  000          LG = FLD(5,16,NSQ1(J1))
000527  001          IF(LG.EQ.0) GO TO 126
000528  000          LTA = FLD(22,14,NSQ1(J1))
000529  000          TNT         = T(NT)                                  FWDBCK
000530  000          GNG         = G(NG)                                  FWDBCK
000531  000    C     CHECK FOR RADIATION CONDUCTOR
000532  000          IF(FLD(3,1,NSQ1(J1)).NE.0) GO TO 120
000533  000          GSUML       = GSUML+GNG                              FWDBCK
000534  000          GSUM        = GSUM+GNG                               FWDBCK
000535  000          QSUM        = QSUM+GNG*TNT                           FWDBCK
000536  000          GO TO 125
000537  000      120 GNG = GNG *SIGMA
000538  000          GSUMR = GSUMR +GNG
000539  000          GSUM        = GSUM+GNG*(TNT+TI)*((TNT+TI)**2-2.0*TNT*TI)  FWDBCK
000540  000          QSUM        = QSUM+GNG*TNT**4                        FWDBCK
000541  000    C     CHECK FOR LAST CONDUCTOR
000542  000      125 IF(NSQ1(J1).GT.0) GO TO 115
000543  001      126 IF(GSUMR.EQ.0.0) GO TO 3200
000544  000    C                                                          FWDBCK
000545  000    C           SOLVE TEMPERATURE  BY GENERAL QUARTIC FORMULA  FWDBCK
000546  000    C                                                          FWDBCK
000547  000          CC          = (CIP*GSUML)/GSUMR                      FWDBCK
000548  000          FF          = -CC**2*R16                             FWDBCK
000549  000          GG          = (Q(I)-QSUM)/GSUMR*R3                   FWDBCK
000550  000          GG          = FF**2-GG**3                            FWDBCK
000551  000          IF(GG.GT.0.0)                  GO TO 3150            FWDBCK
000552  000          T2          = 0.0                                    FWDBCK
000553  000          GO TO 3250                                           FWDBCK
000554  000     3150 GG          = SQRT(GG)                               FWDBCK
000555  000          ZZ          = SQRT((GG-FF)**R3-(GG+FF)**R3)          FWDBCK
000556  000          T2          = RSR2*(-ZZ+SQRT(-ZZ**2+CC*RSR2/ZZ))     FWDBCK
000557  000          GO TO 3250                                           FWDBCK
000558  000    C                                                          FWDBCK
000559  000    C           SOLVE TEMPERATURE BY LINEAR EQUATION           FWDBCK
000560  000    C                                                          FWDBCK
000561  000     3200 T2          = (Q(I)+QSUM)/(CIP+GSUML)                FWDBCK
000562  000    C                                                          FWDBCK
000563  000    C       COMPUTE TEMPERATURE CHANGE AND SET THE NEW TEMPERATURE  FWDBCK
000564  000    C                                                          FWDBCK
000565  000     3250 TI          = T2-TI                                  FWDBCK
000566  000          T(I)        = T2                                     FWDBCK
000567  000          IF(ABS(DRLXCC).GE.ABS(TI))     GO TO 3290            FWDBCK
000568  000          DRLXCC      = TI                                     FWDBCK
```

```
000569  000          NDRLXC      = I                                          FWDBCK
000570  000    C                                                              FWDBCK
000571  000    C       COMPUTE CSGMIN FOR INFORMATION ONLY                    FWDBCK
000572  000    C                                                              FWDBCK
000573  000     3280 CSG          = CIP*DTD2CC/GSUM                           FWDBCK
000574  000          IF(CSG.GE.CSGMIN)                GO TO 3300              FWDBCK
000575  000          CSGMIN      = CSG                                        FWDBCK
000576  000          NCSGM       = I                                          FWDBCK
000577  000     3300 CONTINUE                                                 FWDBCK
000578  000    C                                                              FWDBCK
000579  000    C       MAKE AN ITERATION PASS ON THE ARITHMETIC NODES         FWDBCK
000580  000    C                                                              FWDBCK
000581  000     3350 IF(NNA.EQ.0)                     GO TO 3900               FWDBCK
000582  000          JJ1 = J1
000583  000          JJ2 = J2
000584  000           DO 3800   I = NN,NNC                                    FWDBCK
000585  000           TI          = T(I)                                      FWDBCK
000586  000          IF (LOOPCT .GT. 1) GO TO 6000
000587  000          T(I) = T(I) -460.
000588  000          L = 1
000589  000          IF(FLD(4,1,NSO1(JJ1+1)).EQ.0) GO TO 6000
000590  000          NTYPE = FLD(0,5,NSO2(JJ2))
000591  000          LA = FLD(5,17,NSO2(JJ2))
000592  000          LK = FLD(22,14,NSO2(JJ2))
000593  000          GO TO (5005,5010,5015,5020,5025,5030,5035,5040,5030,    VERS  5
000594  000         *       5050,5050,5050),NTYPE                            VERS  5
000595  000     5005 Q(L) = XK(LK)+Q(L)
000596  000          GO TO 5999
000597  000     5010 Q1 = 0.0
000598  000     5012 CALL D1D1WM(T(L),A(LA),XK(LK),Q2)
000599  000          GO TO 5998
000600  000     5015 Q1 = 0.0
000601  000     5017 CALL D1D1WM(CON(14),A(LA),XK(LK),Q2)
000602  000          GO TO 5998
000603  000     5020 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000604  000     5022 JJ2 = JJ2+1
000605  000          LA = FLD(5,17,NSO2(JJ2))
000606  000          LK = FLD(22,14,NSO2(JJ2))
000607  000          GO TO 5017
000608  000     5025 Q1 = XK(LK)*XK(LA)
000609  000          GO TO 5022
000610  000     5030 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000611  000          JJ2 = JJ2+1
000612  000          LA = FLD(5,17,NSO2(JJ2))
000613  000          LK = FLD(22,14,NSO2(JJ2))
000614  000          Q2 = XK(LK)*XK(LA)
000615  000          GO TO 5998
000616  000     5035 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000617  000     5037 JJ2 = JJ2+1
000618  000          LA = FLD(5,17,NSO2(JJ2))
000619  000          LK = FLD(22,14,NSO2(JJ2))
000620  000          GO TO 5012
000621  000     5040 Q1 = XK(LK)*XK(LA)
000622  000          GO TO 5037
000623  000     5050 JJ2=JJ2+1                                               VERS  5
000624  000          JPSLA=FLD(5,17,NSO2(JJ2))                               VERS  5
000625  000          JPSLK=FLD(22,14,NSO2(JJ2))                              VERS  5
```

```
000626  000        SPJTIM=CON(14)+XK(JPSLA)*XK(JPSLK)                    VERS  5
000627  000        CALL D11MCY(XK(JPSLK),SPJTIM,A(LA),XK(LK),Q1)         VERS  5
000628  000        Q2=0.0                                                VERS  5
000629  000        GO TO 5998                                            VERS  5
000630  000   5998 Q(L) = Q1+Q2+Q(L)
000631  000   5999 JJ2 = JJ2+1
000632  000   6000 CONTINUE
000633  000         GSUML       = 0.0                                    FWDBCK
000634  000         GSUMR       = 0.0                                    FWDBCK
000635  000         QSUM        = Q(1)                                   FWDBCK
000636  000    185 JJ1 = JJ1+1
000637  000        LG = FLD(5,16,NSQ1(JJ1))
000638  000        LTA = FLD(22,14,NSQ1(JJ1))
000639  000        IF (LOOPCT .GT. 1) GO TO 4002
000640  000        T(LTA) = T(LTA) -460.
000641  000        IF(FLD(2,1,NSQ1(JJ1)).EQ.0) GO TO 4000
000642  000        NTYPE = FLD(0,5,NSQ2(JJ2))
000643  000        LA = FLD(5,17,NSQ2(JJ2))
000644  000        LK = FLD(22,14,NSQ2(JJ2))
000645  000        GOTO(3005,3010,3015,3020,3025,3030,3035,3040,3045,3050,3055,
000646  000       1     3060,3065,3070,3073,3070) , NTYPE                 VER 6
000647  000   3005 TM = (T(L)+T(LTA))/2.0
000648  000   3007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000649  000        GO TO 3999
000650  000   3010 TM = T(L)
000651  000        GO TO 3007
000652  000   3015 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000653  000   3017 JJ2 = JJ2+1
000654  000        LA = FLD(5,17,NSQ2(JJ2))
000655  000        LK = FLD(22,14,NSQ2(JJ2))
000656  000        CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000657  000        GO TO 3998
000658  000   3020 G1 = XK(LK)*XK(LA)
000659  000        GO TO 3017
000660  000   3025 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000661  000        JJ2 = JJ2+1
000662  000        LA = FLD(5,17,NSQ2(JJ2))
000663  000        LK = FLD(22,14,NSQ2(JJ2))
000664  000        G2 = XK(LK)*XK(LA)
000665  000        GO TO 3998
000666  000   3030 TM = (T(L)+T(LTA))/2.0
000667  000   3032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000668  000        GO TO 3999
000669  000   3035 TM = T(L)
000670  000        GO TO 3032
000671  000   3040 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000672  000   3042 JJ2 = JJ2+1
000673  000        LA = FLD(5,17,NSQ2(JJ2))
000674  000        LK = FLD(22,14,NSQ2(JJ2))
000675  000        CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000676  000        GO TO 3998
000677  000   3045 G1 = XK(LK)*XK(LA)
000678  000        GO TO 3042
000679  000   3050 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000680  000        JJ2 = JJ2+1
000681  000        LA = FLD(5,17,NSQ2(JJ2))
000682  000        LK = FLD(22,14,NSQ2(JJ2))
```

```
000683  000        G2 = XK(LK)*XK(LA)
000684  000        GO TO 3998
000685  000   3055 TM = (T(L)+T(LTA))/2.0
000686  000        CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000687  000        GO TO 3999
000688  000   3060 TM = T(LTA)
000689  000        GO TO 3007
000690  000   3065 TM = T(LTA)
000691  000        GO TO 3032
000692  000   3070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                    VER 6
000693  000   3071 TM  = (T(L) + T(LTA)) / 2.0                            VER 6
000694  000        JJ2 = JJ2 + 1                                          VER 6
000695  000        LA  = FLD(5,17,NSO2(JJ2))                              VER 6
000696  000        LK  = FLD(22,14,NSO2(JJ2))                             VER 6
000697  000        IF(NTYPE .EQ.16) GO TO 3075                            VER 6
000698  000        CALL D2D1WM(TM,  G1,  A(LA), XK(LK), G(LG))            VER 6
000699  000        GO TO  3999                                            VER 6
000700  000   3073 G1 = XK(LA) * XK(LK)                                   VER 6
000701  000        GO TO 3071                                             VER 6
000702  000   3075 G(LG) = G1 * XK(LA) * XK(LK)                           VER 6
000703  000        GO TO 3999                                             VER 6
000704  000   3998 G(LG) = 1./(1./G1+1./G2)
000705  000        IF(FLD(3,1,NSO1(JJ1)).EQ.1) G(LG) = G1*G2
000706  000   3999 JJ2 = JJ2+1
000707  000   4000 CONTINUE
000708  000        T(LTA) = T(LTA) +460.
000709  000   4002 TNT = T(NT)
000710  000         GNG          = G(NG)                                   FWDBCK
000711  000   C    CHECK FOR RADIATION CONDUCTOR
000712  000        IF(FLD(3,1,NSO1(JJ1)).NE.0) GO TO 190
000713  000         QSUM         = QSUM+GNG*TNT                           FWDBCK
000714  000         GSUML        = GSUML+GNG                              FWDBCK
000715  000         GO TO 3550                                            FWDBCK
000716  000    190 GNG = GNG *SIGMA
000717  000        QSUM = QSUM +GNG*TNT**4
000718  000         GSUMR        = GSUMR+GNG                              FWDBCK
000719  000   C    CHECK FOR LAST CONDUCTOR
000720  000    3550 IF(NSO1(JJ1).GT.0) GO TO 185
000721  000        IF (LOOPCT .EQ. 1) T(I) = T(I) +460.
000722  000         IF(GSUML.GT.0.0 .OR.GSUMR.GT.0.0)GO TO 3599
000723  000        T2 = T(I)
000724  000        IF (LOOPCT .GT. 1 .OR. IFIRST .GT. 0) GO TO 3750
000725  000        GO TO 2400
000726  000   C                                                          FWDBCK
000727  000   C       CHECK TO SEE IF MORE ENERGY IS BEING REMOVED THAN THE FWDBCK
000728  000   C       NODE CAN SUPPLY                                    FWDBCK
000729  000   C                                                          FWDBCK
000730  000    3599  IF(QSUM.GT.0.0)  GO TO 3600
000731  000         T2           = 0.0                                    FWDBCK
000732  000         GO TO 3750                                            FWDBCK
000733  000    3600  IF(GSUMR.LE.0.0)                GO TO 3700          FWDBCK
000734  000         IF(GSUML.LE.0.0)                 GO TO 3650          FWDBCK
000735  000   C                                                          FWDBCK
000736  000   C       SOLVE TEMPERATURE BY GENERAL QUARTIC FORMULA       FWDBCK
000737  000   C                                                          FWDBCK
000738  000        CC           = GSUML/GSUMR                            FWDBCK
000739  000        FF           = -CC**2*R16                             FWDBCK
```

```
000740  000          GG          = -QSUM/GSUMR*R3                          FWDBCK
000741  000          GG          = SQRT(FF**2-GG**3)                       FWDBCK
000742  000          ZZ          = SQRT((GG-FF)**R3-(GG+FF)**R3)           FWDBCK
000743  000          T2          = RSR2*(-ZZ+SQRT(-ZZ**2+CC*RSR2/ZZ))      FWDBCK
000744  000          GO TO 3750                                            FWDBCK
000745  000  C                                                             FWDBCK
000746  000  C       SOLVE TEMPERATURE BY QUARTIC EQUATION                 FWDBCK
000747  000  C                                                             FWDBCK
000748  000   3650   T2          = (QSUM/GSUMR)**0.25                      FWDBCK
000749  000          GO TO 3750                                            FWDBCK
000750  000  C                                                             FWDBCK
000751  000  C       SOLVE TEMPERATURE BY LINEAR EQUATION                  FWDBCK
000752  000  C                                                             FWDBCK
000753  000   3700   T2          = QSUM/GSUML                              FWDBCK
000754  000  C                                                             FWDBCK
000755  000  C       COMPUTE TEMPERATURE CHANGE AND SET THE NEW TEMPERATURE FWDBCK
000756  000  C                                                             FWDBCK
000757  000   3750   T1          = T2-T1                                   FWDBCK
000758  000          T(I)        = T2                                      FWDBCK
000759  000          IF(ABS(ARLXCC).GE.ABS(T1))       GO TO 3800           FWDBCK
000760  000          ARLXCC      = T1                                      FWDBCK
000761  000          NARLXC      = I                                       FWDBCK
000762  000   3800   CONTINUE                                              FWDBCK
000763  000  C                                                             FWDBCK
000764  000  C       CHECK TO SEE IF DIFFUSION AND ARITHMETIC RELAXIATION   FWDBCK
000765  000  C            CRITERIA WAS MET.                                FWDBCK
000766  000  C                                                             FWDBCK
000767  000   3900   IF(ABS(ARLXCC).LE.ARLXCA.AND.                         FWDBCK
000768  000            ABS(DRLXCC).LE.DRLXCA)       GO TO 5200            FWDBCK
000769  000  C                                                             FWDBCK
000770  000  C++++++++++                                                   FWDBCK
000771  000  C       IF THREE ITERATIONS HAVE BEEN PERFORMED,EXTRAPOLATE USING FWDBCK
000772  000  C       -AITKENS DEL SQUARED- METHOD.                         FWDBCK
000773  000  C                                                             FWDBCK
000774  000  C         X(E) = X(3)-(X(3)-X(2))**2/((X(3)-X(2)-(X(2)-X(1)) FWDBCK
000775  000  C                                                             FWDBCK
000776  000          GO TO ITER                                            FWDBCK
000777  000  C                                                             FWDBCK
000778  000  C        FIRST ITERATION                                      FWDBCK
000779  000  C                                                             FWDBCK
000780  000   4001   ASSIGN 4200 TO ITER
000781  000          DO 4100   I = 1,NNC                                   FWDBCK
000782  000   4100   X(IX2+I)    = T(I)                                    FWDBCK
000783  000          GO TO 5001
000784  000  C                                                             FWDBCK
000785  000  C           SECOND ITERATION                                  FWDBCK
000786  000  C                                                             FWDBCK
000787  000   4200   ASSIGN 4400 TO ITER                                   FWDBCK
000788  000          DO 4300   I = 1,NNC                                   FWDBCK
000789  000   4300   X(IX3+I)    = T(I)                                    FWDBCK
000790  000          GO TO 5001
000791  000  C                                                             FWDBCK
000792  000  C           THIRD ITERATION                                   FWDBCK
000793  000  C                                                             FWDBCK
000794  000   4400   ASSIGN 4001 TO ITER
000795  000          DO 4800   I = 1,NNC                                   FWDBCK
000796  000          T1          = X(IX2+I)                                FWDBCK
```

```
000797  000        T2          = X(IX3+I)                                    FWDBCK
000798  000        T3          = T(I)                                        FWDBCK
000799  000        IF(T1.LT.T2.AND.T2.LT.T3)       GO TO 4500               FWDBCK
000800  000        IF(T1.GT.T2.AND.T2.GT.T3)       GO TO 4500               FWDBCK
000801  000        GO TO 4800                                                FWDBCK
000802  000  4500  ANUM        = T3-T2                                       FWDBCK
000803  000        R1          = T2-T1                                       FWDBCK
000804  000        IF(ABS(ANUM).GE.ABS(R1))        GO TO 4800               FWDBCK
000805  000        ADEN        = ANUM-R1                                     FWDBCK
000806  000        IF(ABS(ADEN).LT.1.0E-10)        GO TO 4800
000807  000        TE          = T3-ANUM**2/ADEN                             FWDBCK
000808  000        TDIF        = TE-T3                                       FWDBCK
000809  000  C                                                               FWDBCK
000810  000  C       LIMIT THE EXTRAPOLATION TO + OR - TDIFMAX DEG., AND RESULTANT FWDBCK
000811  000  C       TEMPERATURES MUST BE POSITIVE                           FWDBCK
000812  000  C                                                               FWDBCK
000813  000        IF(ABS(TDIF).GT. DIFMAX)        TE = T3+SIGN(DIFMAX,TDIF)
000814  000        IF(TE.LT.0.0)                   TE = 0.0                  FWDBCK
000815  000        TDIF        = TE-T3                                       FWDBCK
000816  000        IF(I.GT.NND)                    GO TO 4600                FWDBCK
000817  000        IF(ABS(TDIF).LE.ABS(DRLXCC))    GO TO 4700                FWDBCK
000818  000        DRLXCC      = TDIF                                        FWDBCK
000819  000        NDRLXC      = I                                           FWDBCK
000820  000        GO TO 4700                                                FWDBCK
000821  000  4600  IF(ABS(TDIF).LE.ABS(ARLXCC))    GO TO 4700                FWDBCK
000822  000        ARLXCC      = TDIF                                        FWDBCK
000823  000        NARLXC      = I                                           FWDBCK
000824  000  4700  T(I)        = TE                                          FWDBCK
000825  000  4800  CONTINUE                                                  FWDBCK
000826  000  5001  CONTINUE
000827  000        NLINE       = NLINE+3                                     FWDBCK
000828  000        IF(NLINE.LE.60)                 GO TO 5100                FWDBCK
000829  000        CALL TOPLIN                                               FWDBCK
000830  000        NLINE       = NLINE+3                                     FWDBCK
000831  000  5100  WRITE (NOUT,5101)  TIMEO, TIMEN, NDRLXC, DRLXCC, NARLXC, ARLXCC  FWDBCK
000832  000  5101  FORMAT('0++CAUTION++ RELAXIATION CRITERIA NOT MET'
000833  000       .       ' TIMEO = 'G13.6,', TIMEN = 'G13.6/12X,
000834  000       .       ' DRLXCC('I5,' REL)='G13.6,'  ,ARLXCC('I5,
000835  000       .       ' REL)='G13.6)
000836  000  C                                                               FWDBCK
000837  000  C++++++++++                                                     FWDBCK
000838  000  C                                                               FWDBCK
000839  000  C       CHECKING TO SEE IF DELTA TEMPERATURE DIFFERENCE FOR A TIME FWDBCK
000840  000  C          INTERVAL IS WITHIN LIMITS. IF NOT CALCULATE A SMALLER FWDBCK
000841  000  C          TIME INTERVAL AND RE-DO THE TEMERATURE CALCULATIONS    FWDBCK
000842  000  C                                                               FWDBCK
000843  000  C          CHECK DIFFUSION NODE                                 FWDBCK
000844  000  C                                                               FWDBCK
000845  000   5200  DTMPCC      = 0.0                                         FWDBCK
000846  000        DO 5300   I = 1,NND                                       FWDBCK
000847  000        DTMP        = T(I)-X(IX1+1)                               FWDBCK
000848  000        IF(ABS(DTMP).LE.ABS(DTMPCC))    GO TO 5300                FWDBCK
000849  000        DTMPCC      = DTMP                                        FWDBCK
000850  000        NDTMPC      = I                                           FWDBCK
000851  000   5300  CONTINUE                                                 FWDBCK
000852  000        IF(ABS(DTMPCC).LE.DTMPCA)       GO TO 5500                FWDBCK
000853  000        DTIMEU      = 0.95*DTIMEU*DTMPCA/ABS(DTMPCC)              FWDBCK
```

```
000854  000        NLINE       = NLINE+2                                  FWDBCK
000855  000        IF(NLINE.GE.60)                 GO TO 5400             FWDBCK
000856  000        CALL TOPLIN                                            FWDBCK
000857  000        NLINE       = NLINE+2                                  FWDBCK
000858  000  5400  WRITE(NOUT,5401) TIMEO, DTIMEU, DTMPC, DTMPCC          FWDBCK
000859  000  5401  FORMAT('0++NOTE++ AT TIMEO='G13.6,' DTIMEU WAS REDUCED TO 'G13.6,
000860  000              ' BECAUSE DTMPCC('I5,'REL) ='G13.6,' .GT. DTMPCA')
000861  000        GO TO 6400                                             FWDBCK
000862  000  C                                                            FWDBCK
000863  000  C        CHECK ARITHMETIC NODES                              FWDBCK
000864  000  C                                                            FWDBCK
000865  000  5500  IF(NNA.LE.0)                   GO TO 7000             FWDBCK
000866  000        ATMPCC      = 0.0                                      FWDBCK
000867  000        DO 5600   I = NN,NNC                                   FWDBCK
000868  000        ATMP          = T(I)-X(IX1+I)                          FWDBCK
000869  000        IF(ABS(ATMP).LE.ABS(ATMPCC))    GO TO 5600             FWDBCK
000870  000        ATMPCC      = ATMP                                     FWDBCK
000871  000        NATMPC      = I                                        FWDBCK
000872  000  5600  CONTINUE                                               FWDBCK
000873  000        IF(ABS(ATMPCC).LE.ATMPCA)       GO TO 7000             FWDBCK
000874  000        DTIMEU      = 0.95*DTIMEU*ATMPCA/ABS(ATMPCC)
000875  000        NLINE       = NLINE+2                                  FWDBCK
000876  000        IF(NLINE.LE.60)                 GO TO 5700             FWDBCK
000877  000        CALL TOPLIN                                            FWDBCK
000878  000        NLINE       = NLINE+2                                  FWDBCK
000879  000  5700  WRITE(NOUT,5701) TIMEO, DTIMEU, NATMPC, ATMPCC         FWDBCK
000880  000  5701  FORMAT('0++NOTE++ AT TIMEO='G13.6,' DTIMEU WAS REDUCED TO 'G13.6,
000881  000              ' BECAUSE ATMPCC('I5,'REL) ='G13.6,' .GT. ATMPCA')
000882  000  C                                                            FWDBCK
000883  000  C        UNFOLD 2.0/DTIMEU FROM THE CAPACITANCE ARRAY.  WIPE OUT THE   FWDBCK
000884  000  C        CALCULATED TEMPERATURES AND RELOADD THE INITIAL TEMPERATURES  FWDBCK
000885  000  C        FOR THIS TIME STEP FROM DYNAMIC STORAGE.  CONVERT THE         FWDBCK
000886  000  C        TEMPEATURES BACK TO FARENHEIT.                                FWDBCK
000887  000  C                                                            FWDBCK
000888  000   6400  DO 6500   I = 1,NND                                   FWDBCK
000889  000   6500  C(I)        = C(I)*DT02CC                              FWDBCK
000890  000        DO 6650   I = 1,NNT                                    FWDBCK
000891  001   6650  T(I)        = X(IX1+I) -460.0                              FWDBCK
000892  000        GO TO 1500                                             FWDBCK
000893  000  C                                                            FWDBCK
000894  000  C                                                            FWDBCK
000895  000  C+++++++++                                                   FWDBCK
000896  000  C                                                            FWDBCK
000897  000  C      TIME STEP HAS BEEN SUCESSFULLY COMPLETED               FWDBCK
000898  000  C                                                            FWDBCK
000899  000  C        SET CAPACITANCE VALUE BACK TO NORMAL, PUT TEMPERATURES BACK  FWDBCK
000900  000  C        IN FARENHEIT, AND CALL -VARBL2-                       FWDBCK
000901  000  C                                                            FWDBCK
000902  000   7000  DO 7100   I = 1,NND                                   FWDBCK
000903  000         C(I)        = C(I)*DT02CC                              FWDBCK
000904  000   7100  CONTINUE                                               FWDBCK
000905  000        DO 7200   I = 1,NNT                                    FWDBCK
000906  000   7200  T(I)        = T(I)-460.0                               FWDBCK
000907  000  C                                                            FWDBCK
000908  000        BACKUP      = 0.0                                      FWDBCK
000909  000        CALL VARBL2                                            FWDBCK
000910  000        IF(BACKUP.EQ.0.0)               GO TO 7240             FWDBCK
```

```
000911  000  C                                                          FWDBCK
000912  000  C       BACKUP SWITCH HAS BEEN TURNED ON FROM -VARBL2-.  WIPE OUT   FWDBCK
000913  000  C       THE CALCULATED TEMPERATURES AND RELOAD THE INITIAL  FWDBCK
000914  000  C       TEMPERATURES FOR THE TIME STEP FROM DYNAMIC STORAGE.  FWDBCK
000915  000  C       CONVERT THE TEMPERATURES BACK TO FARENHEIT AND REDO  FWDBCK
000916  000  C       THE TIME STEP.                                      FWDBCK
000917  000  C                                                          FWDBCK
000918  001         DO 7220   I = 1,NNT                                 FWDBCK
000919  000   7220  T(I)        = X(IX1+I)-460.0                        FWDBCK
000920  000         GO TO 1300                                          FWDBCK
000921  000  C                                                          FWDBCK
000922  000  C       CHECK FOR TIME TO PRINT, INCREMENT TIME.            FWDBCK
000923  000  C                                                          FWDBCK
000924  000   7240  TIMEO         = TIMEN                               FWDBCK
000925  000         IFIRST =1
000926  000         SUMDTP        = SUMDTP+DTIMEU                       FWDBCK
000927  000         IF(QPEITR.NE.0.0)              CALL OUTCAL          FWDBCK
000928  000         IF(SUMDTP.LT.OUTPUT)           GO TO 7300           FWDBCK
000929  000         CALL OUTCAL                                         FWDBCK
000930  000        WRITE (6,8000) LOOPCT
000931  000   8000 FORMAT (/21H                LOOPCT=,I10/)
000932  000       NLINE = NLINE +3
000933  000         SUMDTP       = 0.0                                  FWDBCK
000934  000  C                                                          FWDBCK
000935  000  C       CHECK FOR TIME TO STOP                              FWDBCK
000936  000  C                                                          FWDBCK
000937  000   7300  IF(TIMEN+0.01*DTIMEU.LT.TIMEND) GO TO 1300          FWDBCK
000938  001         NTH           = IXF                                 FWDBCK
000939  000         NDIM          = IXL                                 FWDBCK
000940  000         RETURN                                              FWDBCK
000941  000         END                                                 FWDBCK

END ELT.


*HDG,P GENOUT
```

```
SIGDUM                                                          DATE 022875        PAGE     1

*ELT,L SIGDUM
ELTOTT RL1870 02/28-03:21:36-(0,)
000001       000    GDUM     PROC
000002       000           IF(FLD(2,1,NSQ1(K1)).EQ.0) GO TO 300
000003       000           NTYPE = FLD(0,5,NSQ2(K2))
000004       000           GO TO (299,299,298,298,298,299,299,298,298,298,299,299,299,       VER-6
000005       000          1       298,298,298 ), NTYPE                                       VER  6
000006     . 000      298 K2 = K2+1
000007       000      299 K2 = K2+1
000008       000      300 CONTINUE
000009       000     END

END ELT.


*NOG,P SIGDUM
```

```
*ELT,L GENOUT
ELTOT7 RLIB70 02/28-03:19:57-(2,)
000001      000          SUBROUTINE GENOUT(NDATA,ISTRT,ISTP,NAME)
000002      000    C
000003      000          DIMENSION FMT(12), NAME(22)
000004      000          DIMENSION NDATA(1)
000005      000          DATA MAXI /   1342177728/
000006      000          DATA MZERO /0777777777777/
000007      000          DATA FMT(1), FMT(12) / 6H(1X1P  , 6H I10)  /
000008      000          LOGICAL ONE, CKD
000009      000    C
000010      000          BASE = 6HE12.4,
000011      000          ASSIGN 32 TO MM
000012      000          CKD = .FALSE.
000013      000          GO TO 5
000014      000          ENTRY GENI(NDATA,ISTRT,ISTP,NAME)
000015      000          BASE = 6HI9, 3X
000016      000          ASSIGN 45 TO MM
000017      000          CKD = .TRUE.
000018      000          GO TO 5
000019      000          ENTRY GENR(NDATA,ISTRT,ISTP,NAME)
000020      000          BASE = 6HE12.4,
000021      000          ASSIGN 45 TO MM
000022      000          CKD = .TRUE.
000023      000        5 DO 6 J=1,21
000024      000          IF(NAME(J+1).EQ.MZERO)GO TO 7
000025      000        6 CONTINUE
000026      000        7 WRITE(6,10)(NAME(I),I=1,J)
000027      000       10 FORMAT(22A6)
000028      000          ONE = .FALSE.
000029      000          IF(ISTRT .EQ. 1 .AND. ISTP .EQ. 1) ONE = .TRUE.
000030      000       15 I=ISTRT
000031      000       20 IF(I .GT. ISTP) GO TO 70
000032      000          L=1
000033      000          DO 30 J=2,11
000034      000          FMT(J) = BASE
000035      000       30 CONTINUE
000036      000          IF(CKD .AND. ONE) GO TO 60
000037      000          IF(I .NE. ISTRT) GO TO 36
000038      000          M=ISTRT-10*(ISTRT/10)
000039      000          IF(M .EQ. 1) GO TO 36
000040      000          IF(M .EQ. 0) M = 10
000041      000          DO 35 J=2,M
000042      000          FMT(J)=6H 12X
000043      000       35 CONTINUE
000044      000          L=M
000045      000          J=I-M+10
000046      000          GO TO 38
000047      000       36 J=I+9
000048      000       38 IF(J .LE. ISTP) GO TO 39
000049      000          M = ISTP - J + 12
000050      000          J=ISTP
000051      000          DO 37 K=M,11
000052      000          FMT(K)=6H 12X
000053      000       37 CONTINUE
000054      000       39 GO TO MM, (32,45)
000055      000       32 DO 40 K=I,J
```

D-144

```
000056      000          L=L+1
000057      000          IF(IABS(NDATA(K)) .LE. MAXI)   FMT(L)=6HI9, 3X
000058      000       40 CONTINUE
000059      000       45 IF(ONE) GO TO 60
000060      000          WRITE(6,FMT)(NDATA(K),K=I,J),J
000061      000          I=J+1
000062      000          GO TO 20
000063      000       60 WRITE(6,FMT) NDATA(ISTRT)
000064      000       70 RETURN
000065      000          END

END ELT.


*HDG,P GENLNK/P8
```

```
@ELT,L GENLNK/PB
ELTOT7 RL1870 02/28-03:19:59-(3,)
000001      000          SUBROUTINE GENLNK                                          GNL   1
000002      000 ·  C                  GENERATES  LINK  ZERO  OF  USER'S  PROGRAM .   GNL   2
000003      000          COMMON /CRDBLK/ LSTART, LECARD, LCOPY, NW, KBLK(507),IMAGE(14)  GNL   3
000004      000          COMMON /TAPE/  NIN, NOUT, INTERN, LB3D, LB4P, LUT1         GNL   4
000005      000          COMMON /DATA/ NND,NNA,NNB,NNT,NGL,NGR,NGT,NUC,NEC1,NEC2,NCT,LENA,  GNL   5
000006      000         1     ERDATA,PROGRM,ENDRUN,LSEQ1,LSEQ2,LONG                 GNL   6
000007      000          COMMON /LOGIC/ DUMMY(58),GENERL                            GNL   7
000008      000          COMMON /PLOGIC/ PARINT, PARFIN, PNODE, PCOND, PCONST, PARRAY  GNL   8
000009      000          COMMON /PLOGIC/  PTITLE, PCHGID, LCONST, LARRAY            GNL   9
000010      000          DIMENSION  MAIN(9)                                         GNL  10
000011      000          DIMENSION  LOGICT(12), LOGICF(2)                           GNL  11
000012      000          LOGICAL  PNODE, PCOND, PCONST, PARRAY                      GNL  12
000013      000          LOGICAL  GENERL, LSTART, LCOPY                             GNL  13
000014      000          LOGICAL  LCONST, LARRAY                                    GNL  14
000015      000          DATA  (MAIN(I), I = 1,9) /6HCALL  ,6HINPUTT,6HCALL  ,6HEXECTN,  GNL  15
000016      000         1       6HGO TO ,6H1     ,6HEND   ,6H      ,6HINPUTG /     GNL  16
000017      000          DATA  (LOGICT(I), I = 1,12) / 6HLNODE ,6H = .TR,6HUE.   ,6HLCOND ,GNL  17
000018      000         1      6H = .TR,6HUE.   ,6HLCONST,6H = .TR,6HUE.   ,6HLARRAY,  GNL  18
000019      000         2      6H = .TR,6HUE.   /, (LOGICF(I), I = 1,2) /6H = .FA,6HLSE.  /GNL  19
000020      000          DATA LO /6HSINDA /                                         GNL  20
000021      000          COMMON/SRDCOM/ZDV(236),NRNSFR(22)
000022      000          LCONST=PCONST                                              GNL  21
000023      000          LARRAY=PARRAY                                              GNL  22
000024      000          LSTART=.TRUE.                                              GNL  23
000025      000          IMAGE(1)=LO                                                GNL  24
000026      000          CALL BLKCRD                                                GNL  25
000027      000          IF (GENERL) MAIN(2)=MAIN(9)                                GNL  26
000028      000          NNO=NND+NNA                                                GNL  27
000029      000          I1=NNT                                                     GNL  28
000030      000          I2=NND                                                     GNL  29
000031      003          I3=NNT                                                     GNL  30
000032      000          I4=NGT                                                     GNL  31
000033      000          I5=NCT                                                     GNL  32
000034      000          I6=LENA                                                    GNL  33
000035      000          I7=LSEQ1                                                   GNL  34
000036      000          I8=LSEQ2                                                   GNL  35
000037      000          IF (PNODE) GO TO 10                                        GNL  36
000038      000          I1=1                                                       GNL  37
000039      000          I2=1                                                       GNL  38
000040      000          I3=1                                                       GNL  39
000041      000          I7=1                                                       GNL  40
000042      000          LOGICT(2)=LOGICF(1)                                        GNL  41
000043      000          LOGICT(3)=LOGICF(2)                                        GNL  42
000044      000       10 CONTINUE                                                   GNL  43
000045      000          IF (PCOND) GO TO 20                                        GNL  44
000046      000          I4=1                                                       GNL  45
000047      000          I7=1                                                       GNL  46
000048      000          LOGICT(5)=LOGICF(1)                                        GNL  47
000049      000          LOGICT(6)=LOGICF(2)                                        GNL  48
000050      000       20 CONTINUE                                                   GNL  49
000051      000          IF (PCONST) GO TO 30                                       GNL  50
000052      000          I5=1                                                       GNL  51
000053      000          LOGICT(8)=LOGICF(1)                                        GNL  52
000054      000          LOGICT(9)=LOGICF(2)                                        GNL  53
000055      000       30 CONTINUE                                                   GNL  54
```

```
000056  000        IF (PARRAY) GO TO 40                                  GNL  55
000057  000        I6=1                                                  GNL  56
000058  000        LOGICT(11)=LOGICF(1)                                  GNL  57
000059  000        LOGICT(12)=LOGICF(2)                                  GNL  58
000060  000     40 CONTINUE                                              GNL  59
000061  000        IF (NNO.EQ.0) I2=1                                    GNL  60
000062  000        CALL CLRVEC
000063  000        ENCODE (100,NRNSFR) I1,I2,I3,I4,I5,I6
000064  000        LCOPY=.FALSE.                                         GNL  63
000065  000        CALL BLKCRD                                           GNL  64
000066  000        IF (LSEQ2.EQ.0) I8=1                                  GNL  65
000067  000        CALL CLRVEC
000068  000        ENCODE (110,NRNSFR) I7,I8
000069  000        CALL BLKCRD                                           GNL  67
000070  001        CALL FLOCOM
000071  000        DO 70 I=1,12,3                                        GNL  68
000072  000        NW=NW+1                                               GNL  69
000073  000        KBLK(NW)=MAIN(8)                                      GNL  70
000074  000        J=I+2                                                 GNL  71
000075  000        DO 50 K=I,J                                           GNL  72
000076  000        NW=NW+1                                               GNL  73
000077  000        KBLK(NW)=LOGICT(K)                                    GNL  74
000078  000     50 CONTINUE                                              GNL  75
000079  000        DO 60 K=1,10                                          GNL  76
000080  000        NW=NW+1                                               GNL  77
000081  000        KBLK(NW)=MAIN(8)                                      GNL  78
000082  000     60 CONTINUE                                              GNL  79
000083  000     70 CONTINUE                                              GNL  80
000084  000        DO 90 I=1,7,2                                         GNL  81
000085  000        NW=NW+1                                               GNL  82
000086  000        KBLK(NW)=MAIN(8)                                      GNL  83
000087  000        IF (I.EQ.1) KBLK(NW)=MAIN(6)                          GNL  84
000088  000        NW=NW+1                                               GNL  85
000089  000        KBLK(NW)=MAIN(I)                                      GNL  86
000090  000        NW=NW+1                                               GNL  87
000091  000        KBLK(NW)=MAIN(I+1)                                    GNL  88
000092  000        DO 80 J=1,11                                          GNL  89
000093  000        NW=NW+1                                               GNL  90
000094  000        KBLK(NW)=MAIN(8)                                      GNL  91
000095  000     80 CONTINUE                                              GNL  92
000096  000     90 CONTINUE                                              GNL  93
000097  000        RETURN                                                GNL  94
000098  000  C                                                           GNL  95
000099  000    100 FORMAT (6X,10HDIMENSION ,2HT(,I5,2H),,2HC(,I5,2H),,2HQ(,I5,2H),,2HGNL  96
000100  000       1G(,I5,2H),2HK(,I5,2H),,2HA(,I5,2H),14X)               GNL  97
000101  000    110 FORMAT (5X,6H1SEQ1(,I5,7H),SEQ2(,I5,1H),55H)          GNL  98
000102  000        END                                                   GNL  99
```

END ELT.

#HDG,P GFPRNT

GFPRNT DATE 022875 PAGE 1

```
*ELT,L GFPRNT
ELT077 RLIB70 02/28-03:20:01-(1,)
000001      000          SUBROUTINE GFPRNT
000002      000          COMMON / FLOWG / GF(1)
000003      000          COMMON / FDIMNS / NTYPE, NSYS, NTB
000004      000          CALL GENOUT (GF,1,NTB,'OFLOW CONDUCTORS')
000005      000          RETURN
000006      000          END

END ELT.


*HOG,P GOPLOT/A
```

```
*ELT,L GOPLOT/A
ELTOT7 RLIB70 02/28-03:20:02-(1,)
000001      000          SUBROUTINE  GOPLOT(NSYM)
000002      000          DIMENSION  ASYM(50),OSYM(50),BUFR(4000),ABS(1),ORD(1)
000003      000          COMMON  NPTS,TPG,BUFR
000004      000          EQUIVALENCE (BUFR(1),ABS(1)),(BUFR(2001),ORD(1))
000005      000          INTEGER  ISYM(2)
000006      000          DATA  ISYM/6H1234  ,6H      /
000007      000          FNP = NSYM*2 + 6
000008      000          DT = ABS(NPTS) - ABS(1)
000009      000          NP = FNP * DT/TPG + 1.5
000010      000          NPP = NP - 1
000011      000          K = NPTS / NPP
000012      000          ASYM(1) = ABS(1)
000013      000          OSYM(1) = ORD(1)
000014      000          ASYM(NP) = ABS(NPTS)
000015      000          OSYM(NP) = ORD(NPTS)
000016      000          KK = 0
000017      000          DO 10 I=2,NPP
000018      000          KK = KK + K
000019      001           IX=NXV(ABS(KK))
000020      001           IY=NYV(ORD(KK))
000021      001           CALL RITE2V(IX,IY,1023,90,1,1,NSYM,ISYM,NL)
000022      001       10  CONTINUE
000023      001           IXA=NXV(ABS(1))
000024      001           IYA=NYV(ORD(1))
000025      001           DO 15 I=2,NPTS
000026      001           IXB=NXV(ABS(I))
000027      001           IYB=NYV(ORD(I))
000028      001           CALL LINEV(IXA,IYA,IXB,IYB)
000029      001           IXA=IXB
000030      001           IYA=IYB
000031      001       15  CONTINUE
000032      000          RETURN
000033      000          END

END ELT.


*BDG,P HSTFLO
```

```
*ELT,L HSTFLO
ELTOT7 RLIBTO 02/28-03:20:04-(4,)
000001      004          SUBROUTINE HSTFLO (TINC)
000002      000    C
000003      000          COMMON /TEMP  / T(1)
000004      001          COMMON /UDOT  / W(1)
000005      001          COMMON /VALVP / VP(1)
000006      001          COMMON /PRESS / P(1)
000007      001          COMMON /DELTAP/ DP(1)
000008      001          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRAY, LCOMP, LTB, LPR, LVP
000009      000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000010      000          COMMON /FIXCON/ CON(1)
000011      000          COMMON /TITLE / HEADER(12)
000012      000          COMMON /DIMENS/ NNA, NND, NSL
000013      001          COMMON /FDIMNS/ NTYPE, NSYS, NTB, NP, NV
000014      000    C
000015      000          DATA KK / 0/
000016      000          DATA LL / 0/
000017      000          DATA LT /23/
000018      000    C
000019      000    C
000020      001          DT = CON(2)
000021      001          IF (KK .GT. 0) GO TO 10
000022      001          NW = MAXO(1,NTB)
000023      001          NPR = MAXO(1,NP)
000024      001          NVP = MAXO(1,NV)
000025      000          IF(LNODE .EQ. 0) CALL NNREAD(1)
000026      001          IF (LTB .EQ. 0) CALL NNREAD(5)
000027      001          WRITE (LT) HEADER,(LL,I=1,5),NW,NPR,NVP,LL,LL,LL NW,LL,LL,
000028      001         1NSL,(NEXT(LTB+I),I=1,NW),(NEXT(LPR+I),I=1,NPR),
000029      003         2(NEXT(LVP+I),I=1,NV),(NEXT(LNODE+I),I=1,NSL)
000030      000          TIME2 = 0.0
000031      000          TIME1 = CON(13) + CON(2)
000032      000          CALL HSTTP(TIME1)
000033      000          KK = 1
000034      000          GO TO 50
000035      000       10 TIME2 = TIME2 + DT
000036      000          IF(CON(1)+1.000001 .LT. CON(3)) GO TO 12
000037      000          GO TO 15
000038      000       12 IF(TIME2  .LT. TINC ) GO TO 50
000039      000          IF(CON(1) .LT. TIME1) GO TO 50
000040      000       15 TIME1 = CON(1)
000041      000          TIME2 = 0.0
000042      000          CALL HSTTP(CON(1))
000043      000          IF(CON(1)+1.000001 .LT. CON(3)) GO TO 50
000044      000       20 CALL HSTTP(-CON(1))
000045      000          KK = 0
000046      000          END FILE LT
000047      000       50 CONTINUE
000048      000          RETURN
000049      000    C
000050      000    C
000051      000          SUBROUTINE HSTTP(XTIME)
000052      000    C
000053      001          WRITE (LT) XTIME,(DP(I),I=1,NW),(P(I),I=1,NPR),
000054      001         1(VP(I),I=1,NVP),(W(I),I=1,NW),(T(I),I=1,NSL)
000055      000          END
```

END ELT.

*HOG,P HXCNT

```
@ELT,L HXCNT
ELTOTT RL1870 02/28-03:20:05-(0,)
000001      000          SUBROUTINE HXCNT (X1,X2,X3,X4,X5,X6,X7,X8,X9,ENTH1,ENTH2)
000002      000    C  ANAYSIS OF COUNTER FLOW HEAT EXCHANGERS
000003      000          DIMENSION CP(2), FR(2), WCP(2), TIN(2), TOUT(2), WCP(2)
000004      000          EQUIVALENCE (NUA,UA), (NCP,CP)
000005      000    C
000006      000          UA = X1
000007      000          FR(1) = X2
000008      000          FR(2) = X3
000009      000          CP(1) = X4
000010      000          CP(2) = X5
000011      000          TIN(1) = X6
000012      000          TIN(2) = X7
000013      000          TOUT(1) = X8
000014      000          TOUT(2) = X9
000015      000          DO 10 I=1,2
000016      000          IF(FR(1) .LT. 0.0) GO TO 100
000017      000       10 CONTINUE
000018      000          IF(NCP(1) .LT. 1 .OR. NCP(1) .GT. 65000) GO TO 3
000019      000          TAVG = 0.5*(TIN(1)+TOUT(1))
000020      000          CALL D1DEG1(TAVG,X4,CP(1))
000021      000        3 IF(NCP(2) .LT. 1 .OR. NCP(2) .GT. 65000) GO TO 6
000022      000          TAVG = 0.5*(TIN(2)-TOUT(2))
000023      000          CALL D1DEG1(TAVG,X5,CP(2))
000024      000        6 CONTINUE
000025      000          WCP(1) = FR(1)*CP(1)
000026      000          WCP(2) = FR(2)*CP(2)
000027      000          IF(IABS(NUA) .LE. 99999 .AND. IABS(NUA) .GT. 0)
000028      000         X CALL D2DEG1(FR(1),FR(2),X1,UA)
000029      000          IS = 1
000030      000          IL = 2
000031      000          IF(WCP(1) .LE. WCP(2)) GO TO 20
000032      000          IS = 2
000033      000          IL = 1
000034      000       20 WCPRAT = WCP(IS)/WCP(IL)
000035      000          IF(WCPRAT .GT. .001) GO TO 30
000036      000          EFF = 1.0
000037      000          GO TO 50
000038      000       30 IF(WCPRAT .LT. .999 .OR. WCPRAT .GT. 1.001) GO TO 40
000039      000          EFF = UA/(WCP(IS)+UA)
000040      000          GO TO 50
000041      000       40 E = EXP(-UA/WCP(IS) + UA/WCP(IL))
000042      000          EFF = (1.-E)/(1.-WCPRAT*E)
000043      000       50 TOUT(IS) = TIN(IS) - EFF*(TIN(IS)-TIN(IL))
000044      000          CALL D1DEG1(TIN(1),ENTH1,H11)
000045      000          CALL D1DEG1(TIN(2),ENTH2,H12)
000046      000          GO TO (75,80), IS
000047      000       75 CALL D1DEG1(TOUT(1),ENTH1,H01)
000048      000          H02 = H12 + (H11-H01)*FR(1)/FR(2)
000049      000          CALL REVPOL(H02,ENTH2,TOUT(2))
000050      000          GO TO 85
000051      000       80 CALL D1DEG1(TOUT(2),ENTH2,H02)
000052      000          H01 = H11 + (H12-H02)*FR(2)/FR(1)
000053      000          CALL REVPOL(H01,ENTH1,TOUT(1))
000054      000       85 X8 = TOUT(1)
000055      000          X9 = TOUT(2)
```

```
000056      000          RETURN
000057      000      100 WRITE(6,101) FR(I)
000058      000      101 FORMAT(1H0 131(1H*)///' THE NEGATIVE FLOW RATE'E15.8,' IS NOT ALLOW
000059      000         XED. EXECUTION TERMINATED IN SUBROUTINE HXCNT'//1X 131(1H*))
000060      000          CALL WLKBCK
000061      000          CALL EXIT
000062      000          END

END ELT.


*HDG,P HXCOND
```

```
*ELT,L HXCOND
ELTOT7 RLIB70 02/28-03:20:07-(0,)
000001       000          SUBROUTINE HXCOND(X1,WG,WC,NHG,NHC,TGIN,TCIN,PSIIN,P,XLAM,XMIMO,
000002       000,        1 PSIOUT,WL,TGOUT,TCOUT)
000003       000    C
000004       000          EQUIVALENCE (NEFF,EFF)
000005       000    C
000006       000          DATA MAXI / 65000 /
000007       000    C
000008       000    C
000009       000          IF(NHG .LT. 1 .OR. NHG .GT. MAXI) GO TO 100
000010       000          IF(NHC .LT. 1 .OR. NHC .GT. MAXI) GO TO 100
000011       000          IF(WG .LT. 0.0) GO TO 100
000012       000          IF(WC .LT. 0.0) GO TO 100
000013       000          EFF = X1
000014       000          IF(NEFF .GT. 1 .AND. NEFF .LT. MAXI)
000015       000         1  CALL D3DEG1(PSIIN,WG,WC,X1,EFF)
000016       000          TGOUT = TGIN - EFF*(TGIN-TCIN) + 459.69
000017       000          PBOUT = 0.1217*EXP(19.3*(TGOUT-500.0)/TGOUT)
000018       000          TGOUT = TGOUT - 459.69
000019       000          PSIOUT = XMIMO*PBOUT/(P-PBOUT)
000020       000          IF(PSIOUT .GE. PSIIN .OR. PSIOUT .LT. 0.0) PSIOUT = PSIIN.
000021       000          WL = WG*(PSIIN-PSIOUT)
000022       000          IF(TGOUT .GT. TGIN) WL = 0.0
000023       000          CALL D1DEG1(TCIN ,NHC,HCIN )
000024       000          CALL D1DEG1(TGIN ,NHG,HGIN )
000025       000          CALL D1DEG1(TGOUT,NHG,HGOUT)
000026       000          HCOUT = HCIN +(((HGIN-HGOUT)*WG) + WL*XLAM)/WC
000027       000          CALL REVPOL(HCOUT,NHC,TCOUT)
000028       000          WG = WG - WL
000029       000          RETURN
000030       000    C
000031       000      100 WRITE(6,101) NHG, NHC, WG, WC
000032       000      101 FORMAT(1H0 120(1H*) // 46H ERROR TERMINATION IN SUBROUTINE HXCOND,
000033       000         1 NHG = I10, 7H, NHC = I10, 6H, WG = G15.8, 6H, WC = G15.8 // 1X
000034       000         2 120(1H*))
000035       000          CALL WLKBCK
000036       000          CALL EXIT
000037       000          END

END ELT.


*HOG,P HXCROS
```

```
@ELT,L HXCROS
ELTOT7 RL1870 02/28-03:20:08-(0,)
000001      000          SUBROUTINE HXCROS(X1,X2,X3,X4,X5,X6,X7,X8,X9,KODE,ENTH1,ENTH2)
000002      000    C  ANALYSIS OF CROSS FLOW HEAT EXCHANGERS
000003      000          DIMENSION CP(2), FR(2), WCP(2), TIN(2), TOUT(2), WCP(2)
000004      000          EQUIVALENCE (NUA,UA), (WCP,CP)
000005      000    C
000006      000          UA = X1
000007      000          FR(1) = X2
000008      000          FR(2) = X3
000009      000          CP(1) = X4
000010      000          CP(2) = X5
000011      000          TIN(1) = X6
000012      000          TIN(2) = X7
000013      000          TOUT(1) = X8
000014      000          TOUT(2) = X9
000015      000          DO 10 I=1,2
000016      000          IF(FR(I) .LT. 0.0) GO TO 100
000017      000       10 CONTINUE
000018      000          IF(WCP(1) .LT. 1 .OR. WCP(1) .GT. 65000) GO TO 3
000019      000          TAVG = 0.5*(TIN(1)+TOUT(1))
000020      000          CALL D1DEGI(TAVG,X4,CP(1))
000021      000        3 IF(WCP(2) .LT. 1 .OR. WCP(2) .GT. 65000) GO TO 6
000022      000          TAVG = 0.5*(TIN(2)-TOUT(2))
000023      000          CALL D1DEGI(TAVG,X5,CP(2))
000024      000        6 CONTINUE
000025      000          WCP(1) = FR(1)*CP(1)
000026      000          WCP(2) = FR(2)*CP(2)
000027      000          IF(IABS(NUA) .LE. 99999 .AND. IABS(NUA) .GT. 0)
000028      000         X CALL D2DEGI(FR(1),FR(2),X1,UA)
000029      000          IS = 1
000030      000          IL = 2
000031      000          IF(WCP(1) .LE. WCP(2)) GO TO 20
000032      000          IS = 2
000033      000          IL = 1
000034      000       20 WCPRAT = WCP(IS)/WCP(IL)
000035      000          IF(WCPRAT .GT. .001) GO TO (30,40,50,60), KODE
000036      000          EFF = 1.0
000037      000          GO TO 70
000038      000       30 E = EXP(-UA**.78*WCP(IS)**.22/WCP(IL)) - 1.
000039      000          EFF = 1. - EXP(E*WCP(IL)*UA**.22/WCP(IS)**1.22)
000040      000          GO TO 70
000041      000       40 UAS = UA/WCP(IS)
000042      000          UAL = UA/WCP(IL)
000043      000          EFF = UAS/(UAS/(1.-EXP(-UAS)) + UAL/(1.-EXP(-UAL)) - 1.)
000044      000          GO TO 70
000045      000       50 EFF = (1.-EXP(-WCPRAT)*(1.-EXP(-UA/WCP(IS))))/WCPRAT
000046      000          GO TO 70
000047      000       60 EFF = 1. - EXP(-WCP(IL)/WCP(IS))*(1.-EXP(-UA/WCP(IL)))
000048      000       70 TOUT(IS) = TIN(IS) - EFF*(TIN(IS)-TIN(IL))
000049      000          CALL D1DEGI(TIN(1),ENTH1,HI1)
000050      000          CALL D1DEGI(TIN(2),ENTH2,HI2)
000051      000          GO TO (75,80), IS
000052      000       75 CALL D1DEGI(TOUT(1),ENTH1,HO1)
000053      000          HO2 = HI2 + (HI1-HO1)*FR(1)/FR(2)
000054      000          CALL REVPOL(HO2,ENTH2,TOUT(2))
000055      000          GO TO 85
```

```
000056      000       80 CALL D1DEG1(TOUT(2),ENTH2,H02)
000057      000          H01 = H11 + (H12-H02)*FR(2)/FR(1)
000058      000          CALL REVPOL(H01,ENTH1,TOUT(1))
000059      000       85 X8 = TOUT(1)
000060      000          X9 = TOUT(2)
000061      000          RETURN
000062      000      100 WRITE(6,101) FR(I)
000063      000      101 FORMAT(1HO 131(1H*)///' THE NEGATIVE FLOW RATE'E15.8,' IS NOT ALLOW
000064      000         XED. EXECUTION TERMINATED IN SUBROUTINE HXCROS'//1X 131(1H*))
000065      000          CALL WLKBCK
000066      000          CALL EXIT
000067      000          END

END ELT.


*HOG,P HXEFF
```

```
@ELT,L HXEFF
ELTOT7 RL1870 02/28-03:20:10-(0,)
000001       000          SUBROUTINE HXEFF (X1,X2,X3,X4,X5,X6,X7,X8,X9,ENTH1,ENTH2)
000002       000    C  ANALYSIS OF HEAT EXCHANGERS WITH EFFECTIVENESS GIVEN
000003       000          DIMENSION CP(2), FR(2), NCP(2), TIN(2), TOUT(2), WCP(2)
000004       000          EQUIVALENCE (NEFF,EFF), (NCP,CP)
000005       000    C
000006       000          EFF = X1
000007       000          FR(1) = X2
000008       000          FR(2) = X3
000009       000          CP(1) = X4
000010       000          CP(2) = X5
000011       000          TIN(1) = X6
000012       000          TIN(2) = X7
000013       000          TOUT(1) = X8
000014       000          TOUT(2) = X9
000015       000          DO 10 I=1,2
000016       000          IF(FR(I) .LT. 0.0) GO TO 100
000017       000       10 CONTINUE
000018       000          IF(NCP(1) .LT. 1 .OR. NCP(1) .GT. 65000) GO TO 3
000019       000          TAVG = 0.5*(TIN(1)+TOUT(1))
000020       000          CALL D1DEG1(TAVG,X4,CP(1))
000021       000        3 IF(NCP(2) .LT. 1 .OR. NCP(2) .GT. 65000) GO TO 6
000022       000          TAVG = 0.5*(TIN(2)+TOUT(2))
000023       000          CALL D1DEG1(TAVG,X5,CP(2))
000024       000        6 CONTINUE
000025       000          WCP(1) = FR(1)*CP(1)
000026       000          WCP(2) = FR(2)*CP(2)
000027       000          IF(IABS(NEFF) .LE. 99999 .AND. IABS(NEFF) .GT. 0)
000028       000         X CALL D2DEG1(FR(1),FR(2),X1,EFF)
000029       000          IS = 1
000030       000          IL = 2
000031       000          IF(WCP(1) .LE. WCP(2)) GO TO 20
000032       000          IS = 2
000033       000          IL = 1
000034       000       20 TOUT(IS) = TIN(IS) - EFF*(TIN(IS)-TIN(IL))
000035       000          CALL D1DEG1(TIN(1),ENTH1,HI1)
000036       000          CALL D1DEG1(TIN(2),ENTH2,HI2)
000037       000          GO TO (75,80), IS
000038       000       75 CALL D1DEG1(TOUT(1),ENTH1,HO1)
000039       000          HO2 = HI2 + (HI1-HO1)*FR(1)/FR(2)
000040       000          CALL REVPOL(HO2,ENTH2,TOUT(2))
000041       000          GO TO 85
000042       000       80 CALL D1DEG1(TOUT(2),ENTH2,HO2)
000043       000          HO1 = HI1 + (HI2-HO2)*FR(2)/FR(1)
000044       000          CALL REVPOL(HO1,ENTH1,TOUT(1))
000045       000       85 X8 = TOUT(1)
000046       000          X9 = TOUT(2)
000047       000          RETURN
000048       000      100 WRITE(6,101) FR(I)
000049       000      101 FORMAT(1HO 131(1H*)//' THE NEGATIVE FLOW RATE'E15.8,' IS NOT ALLOW
000050       000         XED. EXECUTION TERMINATED IN SUBROUTINE HXEFF'//1X 131(1H*))
000051       000          CALL WLKBCK
000052       000          CALL EXIT
000053       000          END

END ELT.
```

HXEFF

DATE 022875 PAGE 2

*HDG,P HXPAR

```
#ELT,L HXPAR
ELT077 RL1870 02/28-03:20:11-(0,)
000001      000          SUBROUTINE HXPAR (X1,X2,X3,X4,X5,X6,X7,X8,X9,ENTH1,ENTH2)
000002      000    C  ANALYSIS OF PARALLEL FLOW HEAT EXCHANGERS
000003      000          DIMENSION CP(2), FR(2), NCP(2), TIN(2), TOUT(2), WCP(2)
000004      000          EQUIVALENCE (NUA,UA), (NCP,CP)
000005      000    C
000006      000          UA = X1
000007      000          FR(1) = X2
000008      000          FR(2) = X3
000009      000          CP(1) = X4
000010      000          CP(2) = X5
000011      000          TIN(1) = X6
000012      000          TIN(2) = X7
000013      000          TOUT(1) = X8
000014      000          TOUT(2) = X9
000015      000          DO 10 I=1,2
000016      000          IF(FR(I) .LT. 0.0) GO TO 100
000017      000       10 CONTINUE
000018      000          IF(NCP(1) .LT. 1 .OR. NCP(1) .GT. 65000) GO TO 3
000019      000          TAVG = 0.5*(TIN(1)+TOUT(1))
000020      000          CALL D1DEG1(TAVG,X4,CP(1))
000021      000        3 IF(NCP(2) .LT. 1 .OR. NCP(2) .GT. 65000) GO TO 6
000022      000          TAVG = 0.5*(TIN(2)-TOUT(2))
000023      000          CALL D1DEG1(TAVG,X5,CP(2))
000024      000        6 CONTINUE
000025      000          WCP(1) = FR(1)*CP(1)
000026      000          WCP(2) = FR(2)*CP(2)
000027      000          IF(IABS(NUA) .LE. 99999 .AND. IABS(NUA) .GT. 0)
000028      000         X CALL D2DEG1(FR(1),FR(2),X1,UA)
000029      000          IS = 1
000030      000          IL = 2
000031      000          IF(WCP(1) .LE. WCP(2)) GO TO 20
000032      000          IS = 2
000033      000          IL = 1
000034      000       20 WCPRAT = WCP(IS)/WCP(IL)
000035      000          IF(WCPRAT .GT. .001) GO TO 30
000036      000          EFF = 1.0
000037      000          GO TO 50
000038      000       30 EFF = (1.-EXP(-UA/WCP(IS)-UA/WCP(IL)))/(1.+WCPRAT)
000039      000       50 TOUT(IS) = TIN(IS) - EFF*(TIN(IS)-TIN(IL))
000040      000          CALL D1DEG1(TIN(1),ENTH1,HI1)
000041      000          CALL D1DEG1(TIN(2),ENTH2,HI2)
000042      000          GO TO (75,80), IS
000043      000       75 CALL D1DEG1(TOUT(1),ENTH1,HO1)
000044      000          HO2 = HI2 + (HI1-HO1)*FR(1)/FR(2)
000045      000          CALL REVPOL(HO2,ENTH2,TOUT(2))
000046      000          GO TO 85
000047      000       80 CALL D1DEG1(TOUT(2),ENTH2,HO2)
000048      000          HO1 = HI1 + (HI2-HO2)*FR(2)/FR(1)
000049      000          CALL REVPOL(HO1,ENTH1,TOUT(1))
000050      000       85 X8 = TOUT(1)
000051      000          X9 = TOUT(2)
000052      000          RETURN
000053      000      100 WRITE(6,101) FR(I)
000054      000      101 FORMAT(1HO 131(1H*)///' THE NEGATIVE FLOW RATE'E15.8,' IS NOT ALLOW
000055      000         XED. EXECUTION TERMINATED IN SUBROUTINE HXPAR'//1X 131(1H*))
```

```
000056      000         CALL WLKBCK
000057      000         CALL EXIT
000058      000         END

END ELT.


*HOG,P IMBED
```

```
@ELT,L IMBED
ELTOT7 RL1670 02/28-03:20:13-(0,)
000001      000          SUBROUTINE IMBED
000002      000    C
000003      000          COMMON /BUCKET/ IB(1)
000004      000          COMMON /DATA  / DUM1(6), NGT, MUC, DUM2(4), ERDATA
000005      000          COMMON /POINT / LOC(20), LEN(20)
000006      000          DIMENSION KEYA(4)/2H*A,2H*K,2H*T,2H*G/
000007      000          LOGICAL CRDERR
000008      000    C
000009      000          L1 = LOC(13)
000010      000          L2 = LEN(13) + L1 -1
000011      000          MM = LOC(14)
000012      000          M2 = LOC(15) - 1
000013      000          DO 500 M=L1,L2
000014      000          M1 = M2 + 1
000015      000          M2 = M2 + IB(MM)
000016      000          MM = MM + 1
000017      000          KEY = 6H
000018      000          DO 400 I=M1,M2
000019      000          FLD(0,12,KEY) = FLD(0,12,IB(I))
000020      000          DO 10 K=1,4
000021      000          IF(KEY .EQ. KEYA(K)) GO TO 40
000022      000       10 CONTINUE
000023      000          GO TO 400
000024      000       40 NUM = IB(I)
000025      000          CALL CONVRT(12,30,NUM,CRDERR)
000026      000          IF (CRDERR) GO TO 380
000027      000    C
000028      000       85 GO TO (100,200,300,350), K
000029      000    C
000030      000    C ARRAYS
000031      000    C
000032      000      100 L= 1
000033      000          LL = LOC(14)
000034      000          IST = LOC(13)
000035      000          IEND = IST + LEN(13) - 1
000036      000          DO 140 JJ=IST,IEND
000037      000          IF(NUM .EQ. IB(JJ)) GO TO 390
000038      000          L = L + IB(LL)
000039      000          LL = LL + 1
000040      000      140 CONTINUE
000041      000          GO TO 380
000042      000    C
000043      000    C CONSTANTS
000044      000    C
000045      000      200 NLOC = LOC(11)
000046      000          NLEN = NUC
000047      000          GO TO 360
000048      000    C
000049      000    C TEMPERATURES
000050      000    C
000051      000      300 NLOC = LOC(1)
000052      000          NLEN = LEN(1)
000053      000          GO TO 360
000054      000    C
000055      000    C CONDUCTORS
```

```
000056      000    C
000057      000      350 NLOC = LOC(5)
000058      000          NLEN = NGT
000059      000    C
000060      000      360 CALL SEARCH(NUM,IB(NLOC),NLEN,L)
000061      000          IF(L) 380,380,390
000062      000      380 ERDATA = 1.0
000063      000          NN = I - M1 + 1
000064      000          WRITE(6,385) IB(I),NN,IB(M)
000065      000      385 FORMAT(8H * * *   A6, 23H REFERENCED AT LOCATION  I5,
000066      000         1 9H OF ARRAY  I5, 26H IS NOT IN THE LIST  * * *)
000067      000          GO TO 400
000068      000    C
000069      000      390 IB(I) = L
000070      000    C
000071      000      400 CONTINUE
000072      000    C
000073      000      500 CONTINUE
000074      000          RETURN
000075      000    C
000076      000          END

END ELT.


*MDG,P INPUTT/LF
```

```
*ELT,L INPUTT/LF
ELT07 RL1870 02/28-03:20:14-(3,)
000001      000          SUBROUTINE INPUTT
000002      000          COMMON /FDIMNS/ NTYP, NSYS, NTB, NP, NV, NFD
000003      000          COMMON /FLOBAT/ FLOW(1)
000004      000          COMMON /SYSDAT/ SYSTEM(15,1)
000005      000          COMMON /TYPDAT/ TYPE(10,1)
000006      000          COMMON /WDOT  / W(1)
000007      000          COMMON /PRESS / P(1)
000008      000          COMMON /FLOWG / GF(1)
000009      000          COMMON /VALVP / VP(1)
000010      000          COMMON /WDOT1 / WI(1)
000011      000          COMMON /FLOWR / AFR(1)
000012      000          COMMON /DELTAP/ DP(1)
000013      000          INCLUDE COMM,LIST
000014      000          DIMENSION DUMMY(1)
000015      000          EQUIVALENCE (DUMMY(1),NNO), (DUMMY(9),LENA)
000016      000          COMMON /LOGIC/ LNODE,LCOND,LCONST,LARRAY
000017      000          LOGICAL LNODE,LCOND,LCONST,LARRAY
000018      000          LOGICAL PTITLE
000019      000          INTEGER RECALL
000020      000          DATA RECALL /6HRECALL/
000021      000          DATA IEND/3HEND/, INIT/6HINITIA/, IFIN/5HFINAL/
000022      000          LB30=12
000023      000          NOUT=6
000024      000    10    CONTINUE
000025      000    C..PROBLEM IDENTIFICATION
000026      000          READ (LB30) IDENT
000027      000          IF (IDENT.EQ.IFIN) GO TO 505
000028      000          IF (IDENT.EQ.IEND) GO TO 150
000029      000    C..REGULAR RUN OR INITIAL PARAMETERS
000030      000    C        TITLE
000031      000          READ (LB30) (H(I),I=1,20)
000032      000    C        NODE INFO
000033      000          READ (LB30) NNO,NNA,NNT,(T(I),I=1,NNT)
000034      000          IF (NNO.EQ.0) GO TO 50
000035      000          READ (LB30) (C(I),I=1,NNO)
000036      000     50   CONTINUE
000037      000          NNO=NNO+NNA
000038      000          DO 60 I=1,NNT
000039      000          Q(I)=0.0
000040      000     60   CONTINUE
000041      000    C        CONDUCTOR INFO
000042      000          READ (LB30) NGT,(G(I),I=1,NGT)
000043      000    C        CONSTANTS INFO
000044      000          READ (LB30) NCT,(CON(I),I=1,50)
000045      000          IF (NCT.EQ.0) GO TO 120
000046      000          READ (LB30) (K(I),I=1,NCT)
000047      000    C        ARRAY INFO
000048      000     120  CONTINUE
000049      000          READ (LB30) NAT,LENA
000050      000          IF (LENA.EQ.0) GO TO 130
000051      000          READ (LB30) (A(I),I=1,LENA)
000052      000    C        PSEUDO COMPUTE SEQUENCE INFO
000053      000     130  IF (IDENT.EQ.INIT) GO TO 135
000054      000          READ(LB30) NTYP, NSYS, NTB, NP, NV, NFD
000055      000          IF(NSYS .LT. 1) GO TO 134
```

```
000056  000        READ(LB3D) (FLOW(I),I=1,NFD)
000057  000        READ(LB3D) ((SYSTEM(I,J),I=1,15),J=1,NSYS)
000058  000        READ(LB3D) ((TYPE(I,J),I=1,10),J=1,NTYP)
000059  000        READ(LB3D) (W(I),I=1,NTB)
000060  000        IF(NV .GT. 0) READ(LB3D) (VP(I),I=1,NV)
000061  001         READ(LB3D) (P(I),I=1,NP)
000062  002        READ(LB3D) (WI(I),I=1,NP)
000063  000    134 CONTINUE
000064  000        READ (LB3D) LSQ1,LSQ2,(NSQ1(I),I=1,LSQ1)
000065  000        IF (LSQ2.EQ.0) GO TO 140
000066  000        READ (LB3D) (NSQ2(I),I=1,LSQ2)
000067  000    135 IF (LSQ2.EQ.0) GO TO 140
000068  000        J1=0
000069  000        J2=1
000070  000        DO 200 I=1,NNO
000071  000        INCLUDE DMCC,LIST
000072  000        INCLUDE DMQQ,LIST
000073  000    185 J1=J1+1
000074  000        NTYPE=0
000075  000        INCLUDE DMGG,LIST
000076  000        IF ((NTYPE.EQ.2).OR.(NTYPE.EQ.7)) GO TO 190
000077  000        IF ((NTYPE.EQ.12).OR.(NTYPE.EQ.13)) GO TO 190
000078  000        GO TO 195
000079  000    190 LK=FLOC(22,14,NSQ2(J2-1))
000080  000        XK(LK) = ABS(XK(LK))
000081  000    195 IF (NSQ1(J1).GT.0) GO TO 185
000082  000    200 CONTINUE
000083  000    C
000084  000    140 CONTINUE
000085  000        IF (IDENT.EQ.RECALL) GO TO 10
000086  000        RETURN
000087  000    C..FINAL PARAMETER CHANGES
000088  000    505 CONTINUE
000089  000    C       TITLE
000090  000        READ (LB3D) PTITLE
000091  000        IF (.NOT.PTITLE) GO TO 510
000092  000        READ (LB3D) (H(I),I=1,20)
000093  000    C       NODE CHANGES
000094  000    510 CONTINUE
000095  000        READ (LB3D) JJ,KK
000096  000        IF (JJ.EQ.0) GO TO 520
000097  000        READ (LB3D) (NUM,T(NUM),I=1,JJ)
000098  000    520 IF (KK.EQ.0) GO TO 530
000099  000        READ (LB3D) (NUM,C(NUM),I=1,KK)
000100  000    C       CONDUCTOR CHANGES
000101  000    530 CONTINUE
000102  000        READ (LB3D) JJ
000103  000        IF (JJ.EQ.0) GO TO 540
000104  000        READ (LB3D) (NUM,G(NUM),I=1,JJ)
000105  000    C       CONSTANTS CHANGES
000106  000    540 CONTINUE
000107  000        READ (LB3D) JJ,KK
000108  000        IF (JJ.EQ.0) GO TO 550
000109  000        READ (LB3D) (NUM,KON(NUM),I=1,JJ)
000110  000    550 IF (KK.EQ.0) GO TO 560
000111  000        READ (LB3D) (NUM,K(NUM),I=1,KK)
000112  000    C       ARRAY CHANGES
```

```
000113      000    560  CONTINUE
000114      000         KON(28)=0
000115      000         KON(29)=0
000116      000         READ (LB30) JJ
000117      000         IF (JJ.EQ.0) GO TO 570
000118      000         READ (LB30) (NUM,A(NUM),I=1,JJ)
000119      000    C
000120      000    570  CONTINUE
000121      000         RETURN
000122      000    150  WRITE (NOUT,9000)
000123      000         CALL EXIT
000124      000    9000 FORMAT (/7X,'END OF DATA')
000125      000         END

END ELT.


*ROG,P LINECK
```

LINECK DATE 022875 PAGE 1

```
*ELT,L LINECK
ELTOT7 RLIB : 02/28-03:20:16-(0,)
000001      000          SUBROUTINE LINECK(I)
000002      000    C
000003      000          COMMON /FIXCON/ N(1)
000004      000    C
000005      000    C
000006      000          IF(N(28)+I .GT. 60 .OR. N(29) .EQ. 0) CALL TOPLIN
000007      000          N(28) = N(28) + I
000008      000          RETURN
000009      000          END

END ELT.


*HOG,P LUMPIN
```

```
*ELT,L LUMPIN
ELT077 RL1870 02/28-03:20:17-(1,)
000001      000          SUBROUTINE LUMPIN(L,LMP,JSW)
000002      000    C
000003      000          LOGICAL ERR
000004      000    C
000005      000          DIMENSION LMP(1), INTCK(12)
000006      000    C
000007      000          COMMON /TAPE  / NIN, NOUT
000008      000          COMMON /CARD  / KRD, KOL, MXKOL
000009      000          COMMON /CIMAGE/ KARD(80)
000010      001           COM: % /FLOERR/ERR
000011      000    C
000012      000          DATA INTCK / 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0,
000013      000         1             1H-, 1H+/
000014      000    C
000015      000          L = 0
000016      000      115 CALL SKPB(JSW)
000017      000          GO TO(150,440), JSW
000018      000      150 IF(KARD(KOL) .NE. 1H( ) GO TO 240
000019      000          KOL = KOL + 1
000020      000          CALL SUBN(NFLI,ISW)
000021      000          GO TO(155,375,385,430,450), ISW
000022      000      155 KOL = KOL + 1
000023      000          CALL SUBN(NTLI,ISW)
000024      000          GO TO(160,375,385,430,450), ISW
000025      000      160 KOL = KOL + 1
000026      000          CALL SUBN(NFLJ,ISW)
000027      000          GO TO(165,375,385,430,450),ISW
000028      000      165 KOL = KOL + 1
000029      000          CALL SUBN(NTLJ,ISW)
000030      000          GO TO(170,375,385,430,450), ISW
000031      000      170 I1 = 1
000032      000          I2 = 1
000033      000          IF(NFLJ .LT. NFLI) I1 = -1
000034      000          IF(NTLJ .LT. NTLI) I2 = -1
000035      000          IF(KARD(KOL) .EQ. 1H)) GO TO 180
000036      000          KOL = KOL + 1
000037      000          CALL SUBN(I1,ISW)
000038      000          GO TO(175,395,385,430,450), ISW
000039      000      175 KOL = KOL + 1
000040      000          CALL SUBN(I2,ISW)
000041      000          GO TO(185,395,385,430,450), ISW
000042      000      180 KOL = KOL + 1
000043      000      185 IF(MOD(NFLJ-NFLI,I1) .EQ. 0) GO TO 200
000044      000          WRITE(NOUT,190) NFLJ, NFLI, I1
000045      000      190 FORMAT(8H0* * *  I6, 2H - I6, 22H MUST BE A MULTIPLE OF I3,
000046      000         1 7H  * * * /)
000047      000          IF(MOD(NTLJ-NTLI,I2) .EQ. 0) GO TO 430
000048      000          GO TO 205
000049      000      200 IF(MOD(NTLJ-NTLI,I2) .EQ. 0) GO TO 210
000050      000      205 WRITE(NOUT,190) NTLJ, NTLI, I2
000051      000          GO TO 430
000052      000      210 IF((NTLJ-NTLI)/I2 .EQ. (NFLJ-NFLI)/I1) GO TO 220
000053      000          WRITE(NOUT,215)
000054      000      215 FORMAT(90H0* * *  THE NUMBER OF FLUID LUMPS GENERATED DOES NOT EQU
000055      000         1AL THE NUMBER OF TUBE LUMPS  * * * /)
```

```
000056  000          GO TO 430
000057  000      220 J = NTLI
000058  000          DO 225 I=NFLI,NFLJ,I1
000059  000          LMP(L+1) = I
000060  000          LMP(L+2) = J
000061  000          J = J + I2
000062  000          L = L + 2
000063  000      225 CONTINUE
000064  000          GO TO 320
000065  000      240 DO 250I=1,12
000066  000          IF(KARD(KOL) .EQ. INTCK(I)) GO TO 300
000067  000      250 CONTINUE
000068  000          IF (KARD(KOL).NE.1HE) GO TO 254
000069  000          IF (KARD(KOL+1).NE.1HN) GO TO 254
000070  000          IF (KARD(KOL+2).EQ.1HD) GO TO 435
000071  000  C
000072  000      254 K = KOL
000073  000      255 J = KOL + 1
000074  000          DO 260 KOL=J,MXKOL
000075  000          IF(KARD(KOL) .EQ. 1H,) GO TO 270
000076  000      260 CONTINUE
000077  000          M = MXKOL
000078  000          CALL CARDIN(JSW)
000079  000          GO TO 275
000080  000      270 M = KOL - 1
000081  000      275 M = MIN0(K+3,M)
000082  000          L = 1
000083  000          ENCODE(280,LMP(L)) (KARD(I),I=K,M)
000084  000      280 FORMAT(9A1)
000085  000          GO TO(435,440), JSW
000086  000  C
000087  000  C
000088  000  C
000089  000      300 CALL SUBN(LMP(L+1),ISW)
000090  000          GO TO(305,375,435,430,450), ISW
000091  000      305 KOL = KOL + 1
000092  000      310 CALL SUBN(LMP(L+2),ISW)
000093  000          GO TO(315,375,410,430,450),ISW
000094  000      315 L = L + 2
000095  000      320 J = KOL
000096  000          DO 325 KOL=J,MXKOL
000097  000          IF(KARD(KOL) .EQ. 1H,) GO TO 330
000098  000      325 CONTINUE
000099  000          CALL CARDIN(JSW)
000100  000          GO TO(115,440), JSW
000101  000      330 KOL = KOL + 1
000102  000          GO TO 115
000103  000  C
000104  000      375 WRITE(NOUT,380)
000105  000      380 FORMAT(58H0* * *  FLOATING POINT NUMBER INPUT FOR NODE NUMBER  * *
000106  000         .1 * /)
000107  000          GO TO 430
000108  000      385 WRITE(NOUT,390)
000109  000      390 FORMAT(60H0* * *  END FOUND BEFORE COMPLETION OF MULTIPLE INPUT  *
000110  000         1 * * /)
000111  000          GO TO 430
000112  000      395 WRITE(NOUT,400)
```

```
000113      000     400 FORMAT(56H0* * *  FLOATING POINT NUMBER INPUT FOR INCREMENT  * * *
000114      000        1 /)
000115      000          GO TO 430
000116      000     410 WRITE(NOUT,420)
000117      000     420 FORMAT(64H0* * *  FLUID LUMPS AND TUBE LUMPS MUST BE INPUT IN PAIR
000118      000        1S  * * * /)
000119      000     430 ERR = .TRUE.
000120      000     435 CALL SKPTE(JSW)
000121      000     440 RETURN
000122      000     450 JSW = 2
000123      000         RETURN
000124      000         END

END ELT.


@ADD,P MCOMB
```

```
*ELT,L MCOMB
ELTOT7 RLIB70 02/28-03:20:19-(0,)
000001      000          DIMENSION NBUFR(27), DATA(3000), ALPHA(15)
000002      000          DIMENSION XSTART(7),XSTOP(7)
000003      000          DIMENSION ADD(7)
000004      000          DATA XSTART, XSTOP, ADD/21*0./
000005      000          DATA ALPHA /1HA,1HB,1HC,1HD,1H5,1H6,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
000006      000         1           1HL,1HM/
000007      000          WRITE(6,3)
000008      000        3 FORMAT(1H1,10X,'OUTPUT FROM COMBIN ROUTINE'//)
000009      000          READ(5,120) NTAPE, IUNIT, KT, KODE2, INC
000010      000      120 FORMAT(6I5)
000011      000          IF(NTAPE .EQ. 0) GO TO 200
000012      000          IF( KT .EQ. 0) KT = 13
000013      000          IF(IUNIT .EQ. 0) IUNIT = 7
000014      000          REWIND KT
000015      000          KODE1 = 0
000016      000          IF(NTAPE .LT. 0) KODE1 = 1
000017      000          NTAPE = IABS(NTAPE)
000018      000          IF(KODE1 .NE. 0) READ(5,27) (XSTART(I), XSTOP(I), I=1,NTAPE)
000019      000          IF(KODE2 .NE. 0) READ(5,140) ADD
000020      000      140 FORMAT(7F10.0)
000021      000       27 FORMAT(14F5.3)
000022      000          DO 18 L = 1,NTAPE
000023      000          M=0
000024      000          I = L + IUNIT - 1
000025      000          REWIND I
000026      000          READ(I) (NBUFR(J),J=1,26), NSL, (DATA(J),J=1,NSL)
000027      000          NBUFR(27) = NSL
000028      000          IF (L .NE. 1) GO TO 8
000029      000          WRITE(KT) NBUFR, (DATA(J),J=1,NSL)
000030      000          NTOTAL = 0
000031      000          DO 6 J=17,27
000032      000          NTOTAL = NTOTAL + NBUFR(J)
000033      000        6 CONTINUE
000034      000        9 READ (I) TIME,(DATA(K),K=1,NTOTAL)
000035      000          TIME = TIME + ADD(L)
000036      000          IF (TIME .LT. 0.0 .AND. L .NE. NTAPE) GO TO 15
000037      000          IF (TIME .LT. 0.0) GO TO 10
000038      000          IF(TIME-XSTART(L))9,
000039      000          IF(XSTOP(L))33,33
000040      000          IF(TIME-XSTOP(L))33,
000041      000          TIME=-TIME
000042      000          IF(L-NTAPE)15,10,
000043      000       33 M=M-1
000044      000          IF(M .GT. 0) GO TO 9
000045      000          M = INC
000046      000          WTIME = TIME
000047      000       10 WRITE(KT) TIME, (DATA(K),K=1,NTOTAL)
000048      000          IF (TIME) 12,9,9
000049      000        8 READ (I) TIME,(DATA(K),K=1,NTOTAL)
000050      000          TIME = TIME + ADD(L)
000051      000          IF(TIME-XSTART(L))8,
000052      000          IF(TIME-XTIME)21,9,10
000053      000       21 WRITE (6,24)
000054      000       24 FORMAT (//10X34HTAPES ARE NOT IN THE CORRECT ORDER)
000055      000          CALL EXIT
```

```
000056      000       12 END FILE KT
000057      000          REWIND KT
000058      000       15 REWIND I
000059      000          XTIME = WTIME
000060      000          WRITE (6,20) L, XTIME
000061      000       20 FORMAT(13X 9HTAPE I3, 10H ENDING AT F10.5,     29H HAS BEEN LOADED
000062.     000         1 ON NEW TAPE./)
000063      000       18 CONTINUE
000064      000          IF(KT .GT. 15) GO TO 200
000065      000          WRITE(6,30) NTAPE, ALPHA(KT)
000066      000       30 FORMAT (1H010X9HDATA FROM12,38H PLOT TAPES HAS BEEN COMBINED ON UN
000067      000         1IT A2)
000068      000      200 STOP
000069      000          END

END ELT.


*BOG,P MFSD
```

```
@ELT,L MFSD
ELT077 RL1870 02/28-03:20:20-(0,)
000001       000          SUBROUTINE MFSD(A,N,$)
000002       000          DIMENSION A(1)
000003       000          DOUBLE PRECISION DPIV, DSUM
000004       000    C     INITIALIZE DIAGONAL-LOOP
000005       000          KPIV = 0
000006       000          DO 11 K=1,N
000007       000          KPIV = KPIV + K
000008       000          IND = KPIV
000009       000          LEND = K - 1
000010       000    C     START FACTORIZATION-LOOP OVER K-TH ROW
000011       000          DO 11 I=K,N
000012       000          DSUM = 0.D0
000013       000          IF(LEND) 2,4,2
000014       000    C     START INNER LOOP
000015       000        2 DO 3 L=1,LEND
000016       000          LANF = KPIV - L
000017       000          LIND = IND - L
000018       000        3 DSUM = DSUM + A(LANF)*A(LIND)
000019       000    C     TRANSFORM ELEMENT A(IND)
000020       000        4 DSUM = A(IND) - DSUM
000021       000          IF(I-K) 10,5,10
000022       000    C     TEST FOR NEGATIVE PIVOT ELEMENT AND FOR LOSS OF SIGNIFICANCE
000023       000        5 IF(DSUM) 12,12,9
000024       000    C     COMPUTE PIVOT ELEMENT
000025       000        9 DPIV = DSQRT(DSUM)
000026       000          A(KPIV) = DPIV
000027       000          DPIV = 1.D0/DPIV
000028       000          GO TO 11
000029       000    C     CALCULATE TERMS IN ROW
000030       000       10 A(IND) = DSUM*DPIV
000031       000       11 IND = IND + I
000032       000          RETURN
000033       000       12 RETURN 3
000034       000          END

END ELT.


@NDG,P NNREAD
```

```
*ELT,L NNREAD
ELT077 RL1070 02/28-03:20:21-(2,)
000001      000          SUBROUTINE NNREAD(N)
000002      000          INCLUDE COMM,LIST
000003      001          COMMON /FDIMNS/ NTYPE, NSYS, NTB, NP, NV
000004      001          COMMON /POINTN/ LNODE, LCOND, LCONS, LARY, ICOMP, LTB, LPR, LV
000005      001    C
000006      001          DATA LNODE, LCOND, LCONS, LARY, ICOMP, LTB, LPR, LV / 8*0 /
000007      001    C
000008      001    C
000009      000          LUT1 = 4
000010      000          REWIND LUT1
000011      000          L = NTH+NDIM
000012      000          J = L-NNT+1
000013      001          GO TO (5,10,15,20,30),N
000014      000        5 NDIM = NDIM-NNT
000015      000          IF(NDIM.LT.0) GO TO 98
000016      000          READ(LUT1) NNT,(X(I),I = J,L)
000017      000          DO 8 I=J,L
000018      000          FLO(1,2,X(I))=0
000019      000        8 CONTINUE
000020      000          LNODE = J-1
000021      000          RETURN
000022      000       10 NDIM = NDIM-NGT
000023      000          IF(NDIM.LT.0) GO TO 98
000024      000          READ(LUT1) NNT,(Z,I = J,L)
000025      000          J = L-NGT+1
000026      000          READ(LUT1) NGT,(X(I),I = J,L)
000027      000          LCOND = J-1
000028      000          RETURN
000029      000       15 NDIM = NDIM-NCT
000030      000          IF(NDIM.LT.0) GO TO 98
000031      000          IF (KON(31).EQ.2) GO TO 17
000032      000          READ(LUT1) NNT,(Z,I = J,L)
000033      000          J = L-NGT+1
000034      000          READ(LUT1) NGT,(Z,I = J,L)
000035      000       17 CONTINUE
000036      000          J = L-NCT+1
000037      000          READ (LUT1) Z,NCT,(X(I),I=J,L)
000038      000          LCONS = J-1
000039      000          RETURN
000040      000       20 NDIM = NDIM-2*NAT
000041      000          IF(NDIM.LT.0) GO TO 98
000042      000          IF (KON(31).EQ.2) GO TO 22
000043      000          READ(LUT1) NNT,(Z,I = J,L)
000044      000          J = L-NGT+1
000045      000          READ(LUT1) NGT,(Z,I = J,L)
000046      000       22 CONTINUE
000047      000          IF (NCT.EQ.0) GO TO 23
000048      000          J = L-NCT+1
000049      000          READ (LUT1) Z,NCT,(Z,I=J,L)
000050      000       23 CONTINUE
000051      000          J=L-2*NAT+1
000052      000          L=J+NAT-1
000053      000          READ(LUT1) NAT,(X(I),I = J,L)
000054      000          LARRY = J-1
000055      000          J=L+1
```

```
000056  000        L=J+NAT-1
000057  000        READ (LUTI) NAT,(X(I),I=J,L)
000058  000        L = 1
000059  000        DO 25 I = 1,NAT
000060  000        LOC=LARRY+NAT+I
000061  000        J = NX(LOC)
000062  000        NX(LOC) = L
000063  000        L = L+J
000064  000     25 CONTINUE
000065  000        RETURN
000066  001  C FLOW DATA
000067  001     30 NDIM = NDIM - NTB - NP - NV
000068  001        IF (NDIM .LT. 0) GO TO 98
000069  001        READ (LUTI) NNT,(Z,I=J,L)
000070  001        J = L - NGT + 1
000071  001        READ (LUTI) NGT,(Z,I=J,L)
000072  001        IF (NCT .EQ. 0) GO TO 33
000073  001        J = L - NCT + 1
000074  001        READ (LUTI) Z,NCT,(Z,I=J,L)
000075  001     33 J = L - NAT + 1
000076  001        READ (LUTI) NAT,(Z,I=J,L)
000077  001        READ (LUTI) NAT,(Z,I=J,L)
000078  001        J = L - NTB - NP - NV + 1
000079  001        L = J + NTB - 1
000080  001        READ (LUTI) (X(I),I=J,L)
000081  001        LTB = J - 1
000082  001        J = L + 1
000083  001        L = J + NP - 1
000084  001        READ (LUTI) (X(I),I=J,L)
000085  001        LPR = J - 1
000086  001        IF (NV .EQ. 0) RETURN
000087  001        J = L + 1
000088  001        L = J + NV - 1
000089  001        READ (LUTI) (X(I),I=J,L)
000090  001        LV = J - 1
000091  001        RETURN
000092  000     98 WRITE(6,99) NDIM
000093  000        CALL EXIT
000094  000     99 FORMAT(13H NNREAD SHORT,I5,23H DYNAMIC CORE LOCATIONS)
000095  000        END

END ELT.


@ADD,P NTSOL
```

```
@ELT,L NTSOL
ELTOT7 RL1870 02/28-03:20:23-(1,)
000001      000          SUBROUTINE NTSOL (L14)
000002      000    C
000003      000          LOGICAL COP, FIRST
000004      000    C
000005      000          DIMENSION ADATA(1)
000006      000    C
000007      000          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRY, ICOMP, LTB, LPR
000008      000          COMMON /FLODAT/ NDATA (1)
000009      000          COMMON /WDOT  / W     (1)
000010      000          COMMON /PRESS / P     (1)
000011      000          COMMON /FLOWG / GF    (1)
000012      000          COMMON /FLOWR / AFR   (1)
000013      000          COMMON /DELTAP/ DP    (1)
000014      000          COMMON /FDATA / COP, LRO, NRO, RO, LMU, NMU, XMU, GC2
000015      000          COMMON /FDATA / TOL, MXPASS, FRDF
000016      000    C
000017      000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000018      000    C
000019      000          EQUIVALENCE (ADATA,NDATA)
000020      000    C
000021      000    C
000022      000          L20=NDATA(L14)-3
000023      000          L25 = NTH + 1
000024      000          NEXT(L25) = NDIM
000025      000          NPAN = 0
000026      000          FIRST = .TRUE.
000027      000          EFRDF = 1.0
000028      000    C
000029      000    C PASS LOOP
000030      000    C
000031      000          DO 540 NPASS=1,MXPASS
000032      000          DWMX = 0.0
000033      000    C
000034      000          IF(.NOT. COP) GO TO 470
000035      000          IF(.NOT. FIRST) CALL TOPLIN
000036      000          CALL LINECK(4)
000037      000          WRITE(6,460) NPASS, NDATA(L14+1)
000038      000      460 FORMAT(///12H * * *  PASS I5, 13H FOR NETWORK A6, 7H  * * *)
000039      000    C
000040      000    C TUBE LOOP
000041      000    C
000042      000      470 DO 520 J=4,L20,4
000043      000          K    = L14 + J
000044      000          NTB  = NDATA(K)
000045      000          NFRM = NDATA(K+1)
000046      000          NTO  = NDATA(K+2)
000047      000          KDAT = NDATA(K+3)
000048      000    C
000049      000          IF(FIRST) GO TO 475
000050      000          NFRM = NEXT(L25+NFRM)
000051      000          NTO = NEXT(L25+NTO)
000052      000    C
000053      000      475 IF(.NOT. COP) GO TO 500
000054      000          CALL LINECK(3)
000055      000          WRITE(6,480) NEXT(LTB+NTB), NEXT(LPR+NFRM), NEXT(LPR+NTO),
```

```
000056  000       1 KDAT, W(NTB)
000057  000     480 FORMAT(//              7X 7HNTB   = I10  , 8X 7HNFRM  = I10  ,
000058  000       1  8X 7HNTO   = I10  , 8X 7HKDAT  = I10  , 8X 7HW(NTB)= G13.6 )
000059  000   C
000060  000     500 IF(KDAT) 505,501,510
000061  000     501 IF(AFR(NTB)) 502,503,502
000062  000     502 GF(NTB) = 1.0/AFR(NTB)
000063  001         GO TO 515
000064  000     503 GF(NTB) = 0.0
000065  001         GO TO 515
000066  000     505 NTH = NTH + NPRN + 1
000067  000         CALL NTSOL1(-KDAT,W(NTB),NFRM,NTO,DP1)
000068  000         NTH = L25 - 1
000069  000         GF(NTB) = W(NTB)/DP1
000070  000         IF(.NOT. COP) GO TO 517
000071  000         CALL LINECK(3)
000072  000         WRITE(6,506) NPASS,NDATA(L14+1)
000073  000     506 FORMAT(// 23H * * *  CONTINUING PASS I5, 13H FOR NETWORK A6,
000074  000        1 7H  * * *)
000075  000         GO TO 517
000076  000     510 CALL FLORES(KDAT,NTB)
000077  000   C
000078  000   C APPLY USER ADDED RESISTANCE TO FLOW CONDUCTOR
000079  000   C
000080  001     517 IF(AFR(NTB) .NE. 0.0) GF(NTB) = 1.0/(1.0/GF(NTB)+AFR(NTB))
000081  001     515 IF(.NOT. FIRST) GO TO 520
000082  000         CALL PRN(NEXT(L25),NPRN,NDATA(K+1))
000083  000         CALL PRN(NEXT(L25),NPRN,NDATA(K+2))
000084  000     520 CONTINUE
000085  000   C
000086  001         CALL FLBAL(NPRN,L14,  0,  0,  0, EFROF,DWMX,DPO )
000087  000   C
000088  000         IF(DWMX .GT. TOL) GO TO 530
000089  000         DO 525 J=4,L20,4
000090  000         K = L14 + J
000091  000         NFRM = NDATA(K+1)
000092  000         NTO = NDATA(K+2)
000093  000         NDATA(K+1) = NEXT(L25+NFRM)
000094  000         NDATA(K+2) = NEXT(L25+NTO)
000095  000   C
000096  000   C CALCULATE PRESSURE DROP IN TUBE
000097  000   C
000098  000         NTB  = NDATA(K)
000099  000         NFRM = NDATA(K+1)
000100  000         NTO  = NDATA(K+2)
000101  000         DP(NTB) = P(NFRM) - P(NTO)
000102  000     525 CONTINUE
000103  000         RETURN
000104  000     530 FIRST = .FALSE.
000105  000         EFROF = FROF
000106  000     540 CONTINUE
000107  000   C
000108  000         CALL TOPLIN
000109  000         WRITE(6,560) NDATA(L14+1), MXPASS, DWMX, TOL
000110  001     560 FORMAT(85H0* * *  SUBROUTINE NTSOL  FAILED TO CONVERGE TO A SOLUTI
000111  000        ION FOR PRESSURES FOR NETWORK  A6, 7H  * * * //
000112  000        2 8X 19HMAXIMUM PASSES    = I10   /
```

```
000113      000        3 8X 19HMAXIMUM CHANGE    - G13.8 /
000114      000        4 8X 19HMAXIMUM ALLOWABLE - G13.8  )
000115      000   C
000116      000         CALL WLKBCK
000117      000         CALL OUTCAL
000118      000         CALL GFRNT
000119      000         CALL EXIT
000120      000   C
000121      000         END

END ELT.


@ADD,P NTSOL1
```

```
@ELT,L NTSOL1
ELTOT7 RL1870 02/28-03:20:25-(2,)
000001      000          SUBROUTINE NTSOL1(L14 ,WIN,NPI,NPO,DP1)
000002      000    C
000003      000          LOGICAL COP, FIRST
000004      000    C
000005      000          DIMENSION RDATA(1)
000006      000    C
000007      000          COMMON /POINTW/ LNODE, LCOND, LCONS, LARRY, ICOMP, LTB, LPR
000008      000          COMMON /FLOAT/ NDATA (1)
000009      000          COMMON /WOOT  / W     (1)
000010      000          COMMON /PRESS / P     (1)
000011      000          COMMON /FLOWG / GF    (1)
000012      000          COMMON /FLOWR / AFR   (1)
000013      000          COMMON /DELTAP/ DP    (1)
000014      000          COMMON /FDATA / COP, LRQ, NRO, RO, LMU, NMU, XMU, GC2
000015      000          COMMON /FDATA / TOL, MXPASS, FROF
000016      000    C
000017      000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000018      000    C
000019      000          EQUIVALENCE (RDATA,NDATA)
000020      000    C
000021      000    C
000022      000          L20=NDATA(L14)-3
000023      000          L25 = NTH + 1
000024      000          NEXT(L25) = NDIM
000025      000          NPAN = 0
000026      000          FIRST = .TRUE.
000027      000          EFROF = 1.0
000028      000    C
000029      000    C PASS LOOP
000030      000    C
000031      000          DO 540 NPASS=1,MXPASS
000032      000          DWMX = 0.0
000033      000    C
000034      000          IF(.NOT. COP) GO TO 470
000035      000          IF(.NOT. FIRST) CALL TOPLIN
000036      000          CALL LINECK(4)
000037      000          WRITE(6,460) NPASS, NDATA(L14+1)
000038      000      460 FORMAT(///12H * * *  PASS I5, 13H FOR NETWORK A6, 7H * * *)
000039      000    C
000040      000    C TUBE LOOP
000041      000    C
000042      000      470 DO 520 J=4,L20,4
000043      000          K    = L14 + J
000044      000          NTB  = NDATA(K)
000045      000          NFRM = NDATA(K+1)
000046      000          NTO  = NDATA(K+2)
000047      000          KDAT = NDATA(K+3)
000048      000    C
000049      000          IF(FIRST) GO TO 475
000050      000          NFRM = NEXT(L25+NFRM)
000051      000          NTO = NEXT(L25+NTO)
000052      000    C
000053      000      475 IF(.NOT. COP) GO TO 500
000054      000          CALL LINECK(3)
000055      000          WRITE(6,480) NEXT(LTB+NTB), NEXT(LPR+NFRM), NEXT(LPR+NTO),
```

```
000056  000        1 KDAT, W(NTB)
000057  000     480 FORMAT(//              7X 7HNTB   = I10  , 8X 7HNFRM  = I10  ,
000058  000        1  8X 7HNTO   = I10  , 8X 7HKDAT  = I10  , 8X 7HW(NTB)= G13.8 )
000059  000     C
000060  000     500 IF(KDAT) 505,501,510
000061  000     501 IF(AFR(NTB)) 502,503,502
000062  000     502 GF(NTB) = 1.0/AFR(NTB)
000063  002         GO TO 515
000064  000     503 GF(NTB) = 0.0
000065  002         GO TO 515
000066  000     505 NTH = NTH + NPRN + 1
000067  000         CALL NTSOLN(-KDAT,W(NTB),NFRM,NTO,DPN)
000068  000         NTH = L25 - 1
000069  002         GF(NTB) = W(NTB)/DPN
000070  000         IF(.NOT. COP) GO TO 517
000071  000         CALL LINECK(3)
000072  000         WRITE(6,506) NPASS,NDATA(L14+1)
000073  000     506 FORMAT(// 23H * * *  CONTINUING PASS I5, 13H FOR NETWORK A6,
000074  000        1 7H  * * *)
000075  000         GO TO 517
000076  000     510 CALL FLORES(KDAT,NTB)
000077  000     C
000078  000     C APPLY USER ADDED RESISTANCE TO FLOW CONDUCTOR
000079  000     C
000080  002     517 IF(AFR(NTB) .NE. 0.0) GF(NTB) = 1.0/(1.0/GF(NTB)+AFR(NTB))
000081  002     515 IF(.NOT. FIRST) GO TO 520
000082  000         CALL PAN(NEXT(L25),NPRN,NDATA(K+1))
000083  000         CALL PAN(NEXT(L25),NPRN,NDATA(K+2))
000084  000     520 CONTINUE
000085  000     C
000086  001         CALL FLBAL (NPRN,L14,WIN,NPI,NPO,EPROF,DWMX,DPI)
000087  000     C
000088  000         IF(DWMX .GT. TOL) GO TO 530
000089  000         DO 525 J=4,L20,4
000090  000         K = L14 + J
000091  000         NFRM = NDATA(K+1)
000092  000         NTO = NDATA(K+2)
000093  000         NDATA(K+1) = NEXT(L25+NFRM)
000094  000         NDATA(K+2) = NEXT(L25+NTO)
000095  000     C
000096  000     C CALCULATE PRESSURE DROP IN TUBE
000097  000     C
000098  000         NTB  = NDATA(K)
000099  000         NFRM = NDATA(K+1)
000100  000         NTO  = NDATA(K+2)
000101  000         DP(NTB) = P(NFRM) - P(NTO)
000102  000     525 CONTINUE
000103  000         RETURN
000104  000     530 FIRST = .FALSE.
000105  000         EPROF = PROF
000106  000     540 CONTINUE
000107  000     C
000108  000         CALL TOPLIN
000109  000         WRITE(6,560) NDATA(L14+1), MXPASS, DWMX, TOL
000110  002     560 FORMAT(/5H0* * *  SUBROUTINE NTSOL1 FAILED TO CONVERGE TO A SOLUTI
000111  000        1ON FOR PRESSURES FOR NETWORK  A6, 7H  * * * //
000112  000        2 8X 19HMAXIMUM PASSES    = I10   /
```

```
000113      000        3 8X 19HMAXIMUM CHANGE    = G13.8 /
000114      000        4 8X 19HMAXIMUM ALLOWABLE = G13.8  )
000115      000   C
000116      000         CALL WLKBCK
000117      000         CALL OUTCAL
000118      000         CALL GFPRNT
000119      000         CALL EXIT
000120      000   C
000121      000         END
```

END ELT.

@ADD,P NTSOLN

```
@ELT,L NTSOLN
ELTOT7 RL1870 02/28-03:20:27-(2,)
000001       000          SUBROUTINE NTSOLN(L14,WIN,NPI,NPO,DPN)
000002       000    C
000003       000          LOGICAL COP, FIRST
000004       000    C
000005       000          DIMENSION RDATA(1)
000006       000    C
000007       000          COMMON /FLODAT/ NDATA (1)
000008       000          COMMON /WBOT  / W     (1)
000009       000          COMMON /PRESS / P     (1)
000010       000          COMMON /FLOWG / GF    (1)
000011       000          COMMON /FLOWR / AFR   (1)
000012       000          COMMON /DELTAP/ DP    (1)
000013       000          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRY, ICOMP, LTB, LPR
000014       000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000015       000    C
000016       000          COMMON /FDATA / COP, LRO, NRO, RO, LMU, NMU, XMU, GC2
000017       000          COMMON /FDATA / TOL, MXPASS, FRDF
000018       000    C
000019       000          EQUIVALENCE (RDATA,NDATA)
000020       000    C
000021       000    C
000022       000          L20=NDATA(L14)-3
000023       000          L25 = NTH + 1
000024       000          NEXT(L25) = NDIM
000025       000          NFRM = 0
000026       000          FIRST = .TRUE.
000027       000          EFRDF = 1.0
000028       000    C
000029       000    C PASS LOOP
000030       000    C
000031       000          DO 540 NPASS=1,MXPASS
000032       000          DWMX = 0.0
000033       000    C
000034       000          IF(.NOT. COP) GO TO 470
000035       000          IF(.NOT. FIRST) CALL TOPLIN
000036       000          CALL LINECK(4)
000037       000          WRITE(6,460) NPASS, NDATA(L14+1)
000038       000      460 FORMAT(///12H * * *  PASS I5, 13H FOR NETWORK A6, 7H  * * *)
000039       000    C
000040       000    C TUBE LOOP
000041       000    C
000042       000      470 DO 520 J=4,L20,4
000043       000          K    = L14 + J
000044       000          NTB  = NDATA(K)
000045       000          NFRM = NDATA(K+1)
000046       000          NTO  = NDATA(K+2)
000047       000          KDAT = NDATA(K+3)
000048       000    C
000049       000          IF(FIRST) GO TO 475
000050       000          NFRM = NEXT(L25+NFRM)
000051       000          NTO = NEXT(L25+NTO)
000052       000    C
000053       000      475 IF(.NOT. COP) GO TO 500
000054       000          CALL LINECK(3)
000055       000          WRITE(6,480) NEXT(LTB+NTO), NEXT(LPR+NFRM), NEXT(LPR+NTO),
```

```
000056  000        1 KDAT, W(NTB)
000057  000      480 FORMAT(//              7X 7HNTB   = I10  , 8X 7HNFRM  = I10  ,
000058  000         1  8X 7HNTO   = I10  , 8X 7HKDAT  = I10  , 8X 7HW(NTB)= G13.8 )
000059  000    C
000060  000      500 IF(KDAT) 505,501,510
000061  000      501 IF(AFR(NTB)) 502,503,502
000062  000      502 GF(NTB) = 1.0/AFR(NTB)
000063  002          GO TO 515
000064  000      503 GF(NTB) = 0.0
000065  002          GO TO 515
000066  000    C
000067  000      505 CALL TOPLIN
000068  000          WRITE(6,506) NDATA(L14+1)
000069  000      506 FORMAT(16H0* * *  NETWORK A6, 37H MUST NOT CONTAIN A SUBNETWORK  *
000070  000         1 * *)
000071  000          CALL WLKBCK
000072  000          CALL EXIT
000073  000    C
000074  000      510 CALL FLORES(KDAT,NTB)
000075  000    C
000076  000    C APPLY USER ADDED RESISTANCE TO FLOW CONDUCTOR
000077  000    C
000078  002      517 IF(AFR(NTB) .NE. 0.0) GF(NTB) = 1.0/(1.0/GF(NTB)+AFR(NTB))
000079  002      515 IF(.NOT. FIRST) GO TO 520
000080  000          CALL PRN(NEXT(L25),NPRN,NDATA(K+1))
000081  000          CALL PRN(NEXT(L25),NPRN,NDATA(K+2))
000082  000      520 CONTINUE
000083  000    C
000084  002          CALL FLBAL (NPRN,L14,WIN,NPI,NPO,EFROF,DWMX,DPN)
000085  000    C
000086  000          IF(DWMX .GT. TOL) GO TO 530
000087  000          DO 525 J=4,L20,4
000088  000          K = L14 + J
000089  000          NFRM = NDATA(K+1)
000090  000          NTO = NDATA(K+2)
000091  000          NDATA(K+1) = NEXT(L25+NFRM)
000092  000          NDATA(K+2) = NEXT(L25+NTO)
000093  000    C
000094  000    C CALCULATE PRESSURE DROP IN TUBE
000095  000    C
000096  000          NTB  = NDATA(K)
000097  000          NFRM = NDATA(K+1)
000098  000          NTO  = NDATA(K+2)
000099  000          DP(NTB) = P(NFRM) - P(NTO)
000100  000      525 CONTINUE
000101  000          RETURN
000102  000      530 FIRST = .FALSE.
000103  000          EFROF = FROF
000104  000      540 CONTINUE
000105  000    C
000106  000          CALL TOPLIN
000107  000          WRITE(6,560) NDATA(L14+1), MXPASS, DWMX, TOL
000108  002      560 FORMAT(85H0* * *  SUBROUTINE NTSOLN FAILED TO CONVERGE TO A SOLUTI
000109  000         1ON FOR PRESSURES FOR NETWORK  A6, 7H  * * * //
000110  000         2 8X 19HMAXIMUM PASSES     - I10   /
000111  000         3 8X 19HMAXIMUM CHANGE     - G13.8 /
000112  000         4 8X 19HMAXIMUM ALLOWABLE - G13.8  )
```

```
000113      000   C
000114      000         CALL WLKBCK
000115      000         CALL OUTCAL
000116      000         CALL GFPRNT
000117      000         CALL EXIT
000118      000   C
000119      000         END

END ELT.


*HDG,P POL
```

POL DATE 022875 PAGE 1

```
*ELT,L POL
ELTOT7 RL187J 02/28-03:20:29-(0,)
000001      000          FUNCTION POL(LOC,X)
000002      000    C
000003      000          COMMON /ARRAY/ NDATA(1)
000004      000    C
000005      000          CALL DIDEG1(X,NDATA(LOC),Y)
000006      000          POL = Y
000007      000          RETURN
000008      000          END

END ELT.


*HDG,P PRESUB/PB
```

```
@ELT,L PRESUB/PB
ELTOT7 RL1870 02/28-03:20:30-(2,)
000001      000          SUBROUTINE PRESUB (N)                                          PRS   1
000002      000          COMMON /TAPE/ NIN,NOUT                                         PRS   2
000003      000          COMMON /CRDBLK/ LSTART,LECARD,LCOPY,NW,ESLK(507),IMAGE(14)     PRS   3
000004      000          COMMON /DATA/ X(13),PROGRM,ENDRUN                              PRS   4
000005      000          COMMON /LOGIC/ DUMMY(58),GENERL                                PRS   5
000006      000          INTEGER REM                                                    PRS   6
000007      000          INTEGER COL1,COMMNT                                            PRS   7
000008      000          LOGICAL LSTART,LCOPY,LECARD                                    PRS   8
000009      000          DIMENSION NAMBLK(4), NAME(4)                                   PRS   9
000010      000          DATA REM/6H REM  /                                             PRS  10
000011      000          DATA (NAMBLK(I),I=1,4)/5HION   ,6HLES 1 ,6HLES 2 ,6H CALLS/   PRS  11
000012      000          DATA (NAME(I),I=1,4)/6HEXECTN,6HVARBL1,6HVARBL2,6HOUTCAL/      PRS  12
000013      000          DATA COMMNT /000000000010/                                     PRS  13
000014      000          DATA IC /1HC/                                                  PRS  14
000015      000          DATA BLANK /1H /                                               PRS  15
000016      000       10 CONTINUE                                                       PRS  16
000017      000          CALL SREADC(8)                                                    V 6
000018      000          COL1=FLD(0,6,IMAGE(1))                                         PRS  18
000019      000          IF (COL1.NE.COMMNT) GO TO 20                                   PRS  19
000020      000          FLD(0,6,IMAGE(1))=BLANK                                        PRS  20
000021      000          COL1=IC                                                        PRS  21
000022      000          WRITE (NOUT,50) (IMAGE(I),I=1,12),COL1                         PRS  22
000023      000          GO TO 10                                                       PRS  23
000024      000       20 CONTINUE                                                       PRS  24
000025      000          WRITE (NOUT,40) (IMAGE(I),I=1,12)                              PRS  25
000026      000          IF (IMAGE(2).EQ.REM) GO TO 10                                  PRS  26
000027      000          IF (IMAGE(4).NE.NAMBLK(N)) GO TO 30                            PRS  27
000028      000          LSTART=.TRUE.                                                  PRS  28
000029      000          LECARD=.TRUE.                                                  PRS  29
000030      000          IMAGE(1)=NAME(N)                                               PRS  30
000031      000          CALL BLKCRD                                                    PRS  31
000032      000          LCOPY=.FALSE.                                                  PRS  32
000033      001          CALL FLOCOM
000034      000          RETURN                                                         PRS  33
000035      000       30 CONTINUE                                                       PRS  34
000036      000          PROGRM=1.0                                                     PRS  35
000037      000          ENDRUN=1.0                                                     PRS  36
000038      000          WRITE (NOUT,60)                                                PRS  37
000039      000          RETURN                                                         PRS  38
000040      000    C                                                                    PRS  39
000041      000    C                                                                    PRS  40
000042      000       40 FORMAT (12A6)                                                  PRS  41
000043      000       50 FORMAT (12A6,8X,A1)                                            PRS  42
000044      000       60 FORMAT (6H * * *,87H EXECUTION  BLOCKS  IN  IMPROPER  ORDER  OR 1PRS  43
000045      000         1LLEGAL  BLOCK  DESIGNATION  ENCOUNTERED .)                     PRS  44
000046      000          END                                                            PRS  45-

END ELT.


@HOG,P PRINTW
```

```
*ELT,L PRINTW
ELTOTT RL1870 02/28-03:20:31-(2,)
000001       002          SUBROUTINE PRINTW(LOC,VAR,N,C)
000002       000    C
000003       000          DIMENSION VAR(1)
000004       000    C
000005       000          COMMON /FIXCON/ KON(27), LC
000006       000          COMMON /XSPACE/ NDIM, NTH, NX(1,
000007       000    C
000008       000          DATA NOUT / 6 /
000009       000    C
000010       000          WRITE(NOUT,100)
000011       000          J = 1
000012       000       50 L = J + 4
000013       000          IF(L .GT. N) L = N
000014       000          IF(LC .LT. 60) GO TO 200
000015       000          CALL TOPLIN
000016       000          WRITE(NOUT,100)
000017       000      100 FORMAT(1H )
000018       000          LC = LC + 1
000019       000      200 LC = LC + 1
000020       000          WRITE(NOUT,300) (C,NX(LOC+I),VAR(I),I=J,L)
000021       000      300 FORMAT(5(1X A3, I6, 1H= G10 5, 1X))
000022       002          IF(L.EQ.N) RETURN
000023       000          J = L + 1
000024       000          GO TO 50
000025       000          END

END ELT.


*HOG,P PRN
```

```
*ELT,L PRN
ELT,0T7 RL1870 02/28-03:20:32-(0,)
000001       000          SUBROUTINE PRN(LOC,N,NODE)
000002       000    C
000003       000          DIMENSION LOC(1)
000004       000    C
000005       000    C
000006       000          IF(N .LT. 1) GO TO 20
000007       000          DO 10 J=1,N
000008       000          IF(LOC(J+1) .EQ. NODE) GO TO 30
000009       000       10 CONTINUE
000010       000       20 N = N + 1
000011       000          IF(N .GT. LOC(1)) GO TO 40
000012       000          LOC(N+1) = NODE
000013       000          NODE = N
000014       000          RETURN
000015       000       30 NODE = J
000016       000          RETURN
000017       000    C
000018       000       40 NEED = N - LOC(1)
000019       000          CALL TOPLIN
000020       000          WRITE(6,50) NEED
000021       000       50 FORMAT(03H0* * *  INSUFFICIENT DYNAMIC STORAGE AVAILABLE FOR FLOW
000022       000         1BALANCING SUBROUTINE  * * * // 8X 5HSHORT I5, 10H LOCATIONS)
000023       000          CALL WEKBCK
000024       000          CALL EXIT
000025       000    C
000026       000          END

END ELT.


*HDG,P PSEUDO
```

```
@ELT,L PSEUDO
ELTOTT RLIB70 02/28-03:20:33-(3,1
000001      000          SUBROUTINE PSEUDO                                                    PSD   1
000002      000          COMMON /BUCKET/ IB(1)                                                PSD   2
000003      000          COMMON /DATA/ NND,NNA,DUM1(10),ERDATA,DUM2(2),LSEQ1,LSEQ2,LONG       PSD   3
000004      000          COMMON /POINT/ LOC(20),LEN(20)                                       PSD   4
000005      000          COMMON /LOGIC/ DUM(4),LPRINT,DUMMY(55),LONG2                         VERS-005
000006      000          COMMON /TAPE/ NIN,NOUT,INTERN,LB30                                   PSD   6
000007      000          DIMENSION B(1)                                                       PSD   7
000008      000          EQUIVALENCE (IB,B)                                                   PSD   8
000009      000          LOGICAL NLC,ONENUM,LPRINT,LONG,MATCH,NOCOND                          PSD   9
000010      000          LOGICAL ISSET,NLQ,LONG2                                              VERS-005
000011      000    C                                                                          PSD  11
000012      000          NLC=.FALSE.                                                          PSD  12
000013      000          NLQ=.FALSE.                                                          PSD  13
000014      000          NPASS=0                                                              PSD  14
000015      000          NC=0                                                                 PSD  15
000016      000          IEND=NND+NNA                                                         PSD  16
000017      000          DO 520 I=1,IEND                                                      PSD  17
000018      000    C        CHECK FOR NONLINEAR CAPACITANCE                                  PSD  18
000019      000          IF (I.GT.NND) GO TO 60                                               PSD  19
000020      000          M=LOC(1)+I-1                                                         PSD  20
000021      000          NLCID=FLD(1,1,IB(M))                                                 PSD  21
000022      000          IF (NLCID.EQ.0) GO TO 60                                             PSD  22
000023      000    C        PROCESS NONLINEAR CAPACITANCE                                    PSD  23
000024      000          NLC=.TRUE.                                                           PSD  24
000025      000          M=LOC(4)+NC                                                          PSD  25
000026      000          ITYPE=FLD(0,5,IB(M))                                                 PSD  26
000027      000          GO TO (10,20,30,40,10,20,30,40,10), ITYPE                            PSD  27
000028      000       10 JTYPE=ITYPE                                                          PSD  28
000029      000          CALL PCS2 (IB(M),IPCS2A,LITA)                                        PSD  29
000030      000          FLD(0,5,IPCS2A)=JTYPE                                                PSD  30
000031      000          NC=NC+1                                                              PSD  31
000032      000          ONENUM=.TRUE.                                                        PSD  32
000033      000          GO TO 50                                                             PSD  33
000034      000       20 CALL PCS2 (IB(M),IPCS2A,LITA)                                        PSD  34
000035      000          JTYPE=ITYPE                                                          PSD  35
000036      000          IF (LITA.EQ.1) JTYPE=ITYPE+1                                         PSD  36
000037      000          CALL PCS2 (IB(M+1),IPCS2B,LITA)                                      PSD  37
000038      000          IF (LITA.EQ.1) JTYPE=ITYPE+2                                         PSD  38
000039      000          FLD(0,5,IPCS2A)=JTYPE                                                PSD  39
000040      000          ONENUM=.FALSE.                                                       PSD  40
000041      000          NC=NC+2                                                              PSD  41
000042      000          GO TO 50                                                             PSD  42
000043      000       30 JTYPE=ITYPE-2                                                        PSD  43
000044      000          CALL PCS2 (IB(M),IPCS2A,LITA)                                        PSD  44
000045      000          FLD(0,5,IPCS2A)=JTYPE                                                PSD  45
000046      000          ONENUM=.TRUE.                                                        PSD  46
000047      000          NPASS=NPASS+1                                                        PSD  47
000048      000          IF (NPASS.LT.IB(M+1)) GO TO 50                                       PSD  48
000049      000          NPASS=0                                                              PSD  49
000050      000          NC=NC+2                                                              PSD  50
000051      000          GO TO 50                                                             PSD  51
000052      000       40 CALL PCS2 (IB(M),IPCS2A,LITA)                                        PSD  52
000053      000          JTYPE=ITYPE-2                                                        PSD  53
000054      000          IF (LITA.EQ.1) JTYPE=ITYPE-1                                         PSD  54
000055      000          CALL PCS2 (IB(M+1),IPCS2B,LITA)                                      PSD  55
```

```
000056  000          IF (LITA.EQ.1) JTYPE=ITYPE                          PSD  56
000057  000          FLD(0,5,IPCS2A)=JTYPE                               PSD  57
000058  000          ONENUM=.FALSE.                                      PSD  58
000059  000          NPASS=NPASS+1                                       PSD  59
000060  000          IF (NPASS.LT.IB(M+2)) GO TO 50                      PSD  60
000061  000          NPASS=0                                             PSD  61
000062  000          NC=NC+3                                             PSD  62
000063  000       50 M=LOC(17)+LEN(17)                                   PSD  63
000064  000          IF (M.GE.LOC(18)) CALL FINDRM (17,M)                PSD  64
000065  000          IB(M)=IPCS2A                                        PSD  65
000066  000          LEN(17)=LEN(17)+1                                   PSD  66
000067  000          IF (ONENUM) GO TO 60                                PSD  67
000068  000          M=LOC(17)+LEN(17)                                   PSD  68
000069  000          IF (M.GE.LOC(18)) CALL FINDRM (17,M)                PSD  69
000070  000          IB(M)=IPCS2B                                        PSD  70
000071  000          LEN(17)=LEN(17)+1                                   PSD  71
000072  000  C          CHECK FOR Q FROM SOURCE BLOCK                    PSD  72
000073  000       60 CONTINUE                                            PSD  73
000074  000          M=LOC(1)+I-1                                        PSD  74
000075  000          NLQID=FLD(2,1,IB(M))                                PSD  75
000076  000          IF (NLQID.EQ.0) GO TO 140                           PSD  76
000077  000  C          PROCESS IMPRESSED Q                              PSD  77
000078  000          NLQ=.TRUE.                                          PSD  78
000079  000          MST=LOC(2)                                          PSD  79
000080  000          MEND=MST+LEN(2)-1                                   PSD  80
000081  000          DO 70 M=MST,MEND                                    PSD  81
000082  000          ITYPE=FLD(0,6,IB(M))                                PSD  82
000083  000          IF (ITYPE.EQ.0) GO TO 70                            PSD  83
000084  000          NODNUM=FLD(6,15,IB(M))                              PSD  84
000085  000          IF (NODNUM.NE.1) GO TO 70                           PSD  85
000086  000          MM=M+1                                              PSD  86
000087  000          GO TO 80                                            PSD  87
000088  000       70 CONTINUE                                            PSD  88
000089  000       80 GO TO (90,100,100,110,120,125),ITYPE                VERS   5
000090  000       90 IPCS2A=0                                            PSD  90
000091  000          FLD(0,5,IPCS2A)=ITYPE                               PSD  91
000092  000          KNUM=FLD(23,13,IB(MM))                              PSD  92
000093  000          FLD(23,13,IPCS2A)=KNUM                              PSD  93
000094  000          ONENUM=.TRUE.                                       PSD  94
000095  000          GO TO 130                                           PSD  95
000096  000      100 CALL PCS2 (IB(MM),IPCS2A,LITA)                      PSD  96
000097  000          JTYPE=ITYPE                                         PSD  97
000098  000          FLD(0,5,IPCS2A)=JTYPE                               PSD  98
000099  000          ONENUM=.TRUE.                                       PSD  99
000100  000          GO TO 130                                           PSD 100
000101  000      110 CALL PCS2 (IB(MM),IPCS2A,LITA)                      PSD 101
000102  000          JTYPE=ITYPE                                         PSD 102
000103  000          IF (LITA.EQ.1) JTYPE=JTYPE+1                        PSD 103
000104  000          CALL PCS2 (IB(MM+1),IPCS2B,LITA)                    PSD 104
000105  000          IF (LITA.EQ.1) JTYPE=JTYPE+2                        PSD 105
000106  000          FLD(0,5,IPCS2A)=JTYPE                               PSD 106
000107  000          ONENUM=.FALSE.                                      PSD 107
000108  000          GO TO 130                                           PSD 108
000109  000      120 ITYPE=7                                             PSD 109
000110  000          GO TO 110                                           PSD 110
000111  000      125 ITYPE=10                                            VERS   5
000112  000          GO TO 110                                           VERS   5
```

```
000113  000     130 M=LOC(17)+LEN(17)                                          PSD 111
000114  000         IF (M.GE.LOC(18)) CALL FINDRM (17,M)                       PSD 112
000115  000         IB(M)=IPCS2A                                               PSD 113
000116  000         LEN(17)=LEN(17)+1                                          PSD 114
000117  000         IF (ONENUM) GO TO 140                                      PSD 115
000118  000         M=LOC(17)+LEN(17)                                          PSD 116
000119  000         IF (M.GE.LOC(18)) CALL FINDRM (17,M)                       PSD 117
000120  000         IB(M)=IPCS2B                                               PSD 118
000121  000         LEN(17)=LEN(17)+1                                          PSD 119
000122  000     C       PASS THRU NA/NB PAIRS                                  PSD 120
000123  000     140 J=0                                                        PSD 121
000124  000         NG=0                                                       PSD 122
000125  000         MPASS=0                                                    PSD 123
000126  000         MATCH=.FALSE.                                              PSD 124
000127  000         NOCOND=.TRUE.                                              PSD 125
000128  000         ISSET=.FALSE.                                              PSD 126
000129  000         NGV=0                                                      PSD 127
000130  000     150 IF (J.GE.LEN(7)) GO TO 490                                 PSD 128
000131  000         M=LOC(7)+J                                                 PSD 129
000132  000         J=J+1                                                      PSD 130
000133  000         NANB=IB(M)                                                 PSD 131
000134  000         MUSE=FLD(0,1,NANB)                                         PSD 132
000135  000         IF (MUSE.EQ.0) NGV=NGV+1                                   PSD 133
000136  000         NA=FLD(7,14,NANB)                                          PSD 134
000137  000         IF (NA.NE.1) GO TO 370                                     PSD 135
000138  000         NOCOND=.FALSE.                                             PSD 136
000139  000         NAIWAY=FLD(6,1,NANB)                                       PSD 137
000140  000         IF (NAIWAY.EQ.1) GO TO 370                                 PSD 138
000141  000         IPCS1=0                                                    PSD 139
000142  000         MATCH=.TRUE.                                               PSD 140
000143  000         FLD(5,16,IPCS1)=NGV                                        PSD 141
000144  000         NB=FLD(21,15,NANB)                                         PSD 142
000145  000         FLD(21,15,IPCS1)=NB                                        PSD 143
000146  000         IF (NB.GT.NNO) GO TO 160                                   PSD 144
000147  000         IF (.NOT.LONG) FLD(21,1,IB(M))=1                           PSD 145
000148  000     160 NRAD=FLD(1,1,NANB)                                         PSD 146
000149  000         FLD(3,1,IPCS1)=NRAD                                        PSD 147
000150  000     C       CHECK FOR NONLINEAR CONDUCTOR                          PSD 148
000151  000     170 NLGID=FLD(2,1,NANB)                                        PSD 149
000152  000         IF (NLGID.EQ.0) GO TO 460                                  PSD 150
000153  000         IF (FLD(3,1,NANB).EQ.1) GO TO 450                          PSD 151
000154  000     C       PROCESS NONLINEAR CONDUCTOR                            PSD 152
000155  000         FLD(2,1,IPCS1)=1                                           PSD 153
000156  000         M=LOC(9)+NG                                                PSD 154
000157  000         ITYPE=FLD(0,5,IB(M))                                       PSD 155
000158  000         GO TO (180,210,230,270,300,310,320,330,340,342,344,344),ITYPE  VER 6
000159  000     180 JTYPE=1                                                    PSD 157
000160  000     190 KNUM=FLD(23,13,IB(M))                                      PSD 158
000161  000         MM=LOC(12)+KNUM-1                                          PSD 159
000162  000         IF (B(MM).GE.0.0) GO TO 200                                PSD 160
000163  000         JTYPE=JTYPE+1                                              PSD 161
000164  000         IF (FLD(7,14,NANB).EQ.1) GO TO 200                         PSD 162
000165  000         JTYPE=12                                                   PSD 163
000166  000         IF (ITYPE.EQ.5) JTYPE=13                                   PSD 164
000167  000     200 CALL PCS2 (IB(M),IPCS2A,LITA)                              PSD 165
000168  000         FLD(0,5,IPCS2A)=JTYPE                                      PSD 166
000169  000         NG=NG+1                                                    PSD 167
```

```
000170  000          ONENUM=.TRUE.                                          PSD 168
000171  000          GO TO 350                                              PSD 169
000172  000      210 JTYPE=3                                                PSD 170
000173  000      220 CALL PCS2 (IB(M),IPCS2A,LITA)                          PSD 171
000174  000          IF (LITA.EQ.1) JTYPE=JTYPE+1                           PSD 172
000175  000          CALL PCS2 (IB(M+1),IPCS2B,LITA)                        PSD 173
000176  000          IF (LITA.EQ.1) JTYPE=JTYPE+2                           PSD 174
000177  000          FLD(0,5,IPCS2A)=JTYPE                                  PSD 175
000178  000          ONENUM=.FALSE.                                         PSD 176
000179  000          NG=NG+2                                                PSD 177
000180  000          GO TO 350                                              PSD 178
000181  000      230 JTYPE=1                                                PSD 179
000182  000      240 KNUM=FLD(23,13,IB(M))                                  PSD 180
000183  000          MM=LOC(12)+KNUM-1                                      PSD 181
000184  000          IF (B(MM).GE.0.0) GO TO 250                            PSD 182
000185  000          B(MM)=ABS(B(MM))                                       PSD 183
000186  000          JTYPE=JTYPE+1                                          PSD 184
000187  000          IF (FLD(7,14,NANB).EQ.1) GO TO 250                     PSD 185
000188  000          JTYPE=12                                               PSD 186
000189  000          IF (ITYPE.EQ.5) JTYPE=13                               PSD 187
000190  000      250 CALL PCS2 (IB(M),IPCS2A,LITA)                          PSD 188
000191  000          FLD(0,5,IPCS2A)=JTYPE                                  PSD 189
000192  000          ONENUM=.TRUE.                                          PSD 190
000193  000          MPASS=MPASS+1                                          PSD 191
000194  000          IF (MPASS.GT.1) GO TO 260                              PSD 192
000195  000          NUMGS=IB(M+1)                                          PSD 193
000196  000      260 IF (MPASS.LT.NUMGS) GO TO 350                          PSD 194
000197  000          MPASS=0                                                PSD 195
000198  000          NG=NG+2                                                PSD 196
000199  000          GO TO 350                                              PSD 197
000200  000      270 JTYPE=3                                                PSD 198
000201  000      280 CALL PCS2 (IB(M),IPCS2A,LITA)                          PSD 199
000202  000          IF (LITA.EQ.1) JTYPE=JTYPE+1                           PSD 200
000203  000          CALL PCS2 (IB(M+1),IPCS2B,LITA)                        PSD 201
000204  000          IF (LITA.EQ.1) JTYPE=JTYPE+2                           PSD 202
000205  000          FLD(0,5,IPCS2A)=JTYPE                                  PSD 203
000206  000          ONENUM=.FALSE.                                         PSD 204
000207  000          MPASS=MPASS+1                                          PSD 205
000208  000          IF (MPASS.GT.1) GO TO 290                              PSD 206
000209  000          NUMGS=IB(M+2)                                          PSD 207
000210  000      290 IF (MPASS.LT.NUMGS) GO TO 350                          PSD 208
000211  000          MPASS=0                                                PSD 209
000212  000          NG=NG+3                                                PSD 210
000213  000          GO TO 350                                              PSD 211
000214  000      300 JTYPE=6                                                PSD 212
000215  000          GO TO 190                                              PSD 213
000216  000      310 JTYPE=8                                                PSD 214
000217  000          GO TO 220                                              PSD 215
000218  000      320 JTYPE=6                                                PSD 216
000219  000          GO TO 240                                              PSD 217
000220  000      330 JTYPE=8                                                PSD 218
000221  000          GO TO 280                                              PSD 219
000222  000      340 JTYPE=11                                               PSD 220
000223  000          GO TO 200                                              PSD 221
000224  000      342 JTYPE = 14                                             VER 6
000225  000          GO TO 220                                              VER 6
000226  000      344 JTYPE = 14                                             VER 6
```

```
000227  000          GO TO 280                                          VER 6
000228  000    C        PLACE PCS2 IN BUCKET                           PSD 222
000229  000      350 M=LOC(17)+LEN(17)                                  PSD 223
000230  000          IF (M.GE.LOC(18)) CALL FINDRM (17,M)               PSD 224
000231  000          IB(M)=IPCS2A                                       PSD 225
000232  000          LEN(17)=LEN(17)+1                                  PSD 226
000233  000          IF (ONENUM) GO TO 360                              PSD 227
000234  000          M=LOC(17)+LEN(17)                                  PSD 228
000235  000          IF (M.GE.LOC(18)) CALL FINDRM (17,M)               PSD 229
000236  000          IB(M)=IPCS2B                                       PSD 230
000237  000          LEN(17)=LEN(17)+1                                  PSD 231
000238  000      360 CONTINUE                                           PSD 232
000239  000          M=LOC(7)+J-1                                       PSD 233
000240  000          IF (.NOT.LONG2) FLD(3,1,IB(M))=1                   VERS-005
000241  000          GO TO 460                                          PSD 235
000242  000    C        CHECK FOR MATCH ON NB                          PSD 236
000243  000      370 NA=FLD(22,14,NANB)                                 PSD 237
000244  000          IF (NA.NE.I) GO TO 390                             PSD 238
000245  000          NOCOND=.FALSE.                                     PSD 239
000246  000          NAIWAY=FLD(21,1,NANB)                              PSD 240
000247  000          IF (NAIWAY.EQ.1) GO TO 390                         PSD 241
000248  000          IPCS1=0                                            PSD 242
000249  000          MATCH=.TRUE.                                       PSD 243
000250  000          FLD(5,16,IPCS1)=NGV                                PSD 244
000251  000          NB=FLD(6,15,NANB)                                  PSD 245
000252  000          FLD(21,15,IPCS1)=NB                                PSD 246
000253  000          IF (NB.GT.NND) GO TO 380                           PSD 247
000254  000          IF (.NOT.LONG) FLD(6,1,IB(M))=1                    PSD 248
000255  000      380 NRAD=FLD(1,1,NANB)                                 PSD 249
000256  000          FLD(3,1,IPCS1)=NRAD                                PSD 250
000257  000          GO TO 170                                          PSD 251
000258  000    C        UPDATE POINTER NG IF NO MATCH                  PSD 252
000259  000      390 NLGIO=FLD(2,1,NANB)                                PSD 253
000260  000          IF (NLGIO.EQ.0) GO TO 150                          PSD 254
000261  000          M=LOC(9)+NG                                        PSD 255
000262  000          ITYPE=FLD(0,5,IB(M))                               PSD 256
000263  000          ITYPE=MOD(ITYPE,4)+1                               PSD 257
000264  000          GO TO (400,440,440,420), ITYPE                     PSD 258
000265  000      400 MPASS=MPASS+1                                      PSD 259
000266  000          IF (MPASS.GT.1) GO TO 410                          PSD 260
000267  000          NUMGS=IB(M+2)                                      PSD 261
000268  000      410 IF (MPASS.LT.NUMGS) GO TO 150                      PSD 262
000269  000          MPASS=0                                            PSD 263
000270  000          NG=NG+3                                            PSD 264
000271  000          GO TO 150                                          PSD 265
000272  000      420 MPASS=MPASS+1                                      PSD 266
000273  000          IF (MPASS.GT.1) GO TO 430                          PSD 267
000274  000          NUMGS=IB(M+1)                                      PSD 268
000275  000      430 IF (MPASS.LT.NUMGS) GO TO 150                      PSD 269
000276  000          MPASS=0                                            PSD 270
000277  000          NG=NG+2                                            PSD 271
000278  000          GO TO 150                                          PSD 272
000279  000      440 NG=NG+ITYPE-1                                      PSD 273
000280  000          GO TO 150                                          PSD 274
000281  000      450 ISSET=.TRUE.                                       PSD 275
000282  000    C        FLAG NLC AND NLO ON FIRST G ONLY               PSD 276
000283  000      460 IF (.NOT.NLC) GO TO 470                            PSD 277
```

```
000284  000          FLD(1,1,IPCS1)=1                                          PSD 278
000285  000          NLC=.FALSE.                                               PSD 279
000286  000     470  IF (.NOT.NLC) GO TO 480                                   PSD 280
000287  000          FLD(4,1,IPCS1)=1                                          PSD 281
000288  000          NLC=.FALSE.                                               PSD 282
000289  000     C       PLACE PCS1 IN BUCKET                                   PSD 283
000290  000     480  M=LOC(16)+LEN(16)                                         PSD 284
000291  000          IF (M.GE.LOC(17)) CALL FINDRM (16,M)                      PSD 285
000292  000          IB(M)=IPCS1                                               PSD 286
000293  000          LEN(16)=LEN(16)+1                                         PSD 287
000294  000          IF (.NOT.ISSET) GO TO 150                                 PSD 288
000295  000          ISSET=.FALSE.                                             PSD 289
000296  000          GO TO 490                                                 PSD 290
000297  000     C       FLAG LAST G FOR EACH NODE                              PSD 291
000298  000     490  CONTINUE                                                  PSD 292
000299  000     500  IF (MATCH) GO TO 510                                      PSD 297
000300  000          IPCS1=0                                                   PSD 298
000301  000          MATCH=.TRUE.                                              PSD 299
000302  000          GO TO 460                                                 PSD 300
000303  000     510  M=LOC(16)+LEN(16)-1                                       PSD 301
000304  000          FLD(0,1,IB(M))=1                                          PSD 302
000305  000     520  CONTINUE                                                  PSD 303
000306  000          IF (LONG) GO TO 530                                       PSD 304
000307  000     C       SET LAST G TO 0 IF SPCS                                PSD 305
000308  000          M=LOC(16)+LEN(16)                                         PSD 306
000309  000          IF (M.GE.LOC(17)) CALL FINDRM (16,M)                      PSD 307
000310  000          IB(M)=0                                                   PSD 308
000311  000          FLD(0,1,IB(M))=1                                          PSD 309
000312  000          LEN(16)=LEN(16)+1                                         PSD 310
000313  000     530  CONTINUE                                                  PSD 311
000314  000          IF(.NOT.LPRINT) GO TO 540
000315  000          WRITE (NOUT,550) LOC(16),LEN(16)                          PSD 313
000316  000          IST=LOC(16)                                               PSD 314
000317  000          IEND=IST+LEN(16)-1                                        PSD 315
000318  000          WRITE (NOUT,560) (I,IB(I),I=IST,IEND)                     PSD 316
000319  000          WRITE (NOUT,570) LOC(17),LEN(17)                          PSD 317
000320  000          IST=LOC(17)                                               PSD 318
000321  000          IEND=IST+LEN(17)-1                                        PSD 319
000322  000          WRITE (NOUT,560) (I,IB(I),I=IST,IEND)                     PSD 320
000323  000     540  CONTINUE                                                  PSD 321
000324  000          LSEQ1=LEN(16)                                             PSD 322
000325  000          LSEQ2=LEN(17)                                             PSD 323
000326  000          CALL WRTDTA (5)                                           PSD 324
000327  000          RETURN                                                    PSD 325
000328  000     C                                                              PSD 326
000329  000     550  FORMAT (/' PCS1  LOC =',I6,' LEN =',I6)                   PSD 327
000330  000     560  FORMAT (5(1X,I6,2X,O12,4X))                               PSD 328
000331  000     570  FORMAT (/' PCS2  LOC=',I6,' LEN =',I6)                    PSD 329
000332  000     580  FORMAT (' * * * RELATIVE NODE NUMBER (',I5,') IS NOT CONNECTED TO PSD 330
000333  000        1ANY OTHER NODE')                                           PSD 331
000334  000          END                                                       PSD 332-
```

END ELT.

@HOG,P OCOMB

```
@ELT,L QCOMB
ELTOT7 RLIB70 02/28-03:20:37-(0,)
000001      000          SUBROUTINE QCOMB(AC,F1,A1,F2,A2)
000002      000    C
000003      000          DIMENSION AC(1), A1(1), A2(1)
000004      000    C
000005      000          EQUIVALENCE (AC1,NC), (A11,N1), (A21,N2)
000006      000    C
000007      000    C
000008      000          AC1 = AC(1)
000009      000          A11 = A1(1)
000010      000          A21 = A2(1)
000011      000          NC = (NC/2)*2
000012      000          L = N1 + N2
000013      000          I1 = 2
000014      000          I2 = 2
000015      000          I  = 2
000016      000        5 IF(I1 .GT. N1) GO TO 70
000017      000          IF(I2 .GT. N2) GO TO 30
000018      000          IF(A1(I1) - A2(I2)) 20,10,50
000019      000       10 AC(I) = A1(I1)
000020      000          AC(I+1) = F1*A1(I1+1) + F2*A2(I2+1)
000021      000          I1 = I1 + 2
000022      000          I2 = I2 + 2
000023      000          L  = L  - 2
000024      000          GO TO 100
000025      000       20 CALL DIDEG1(A1(I1),A2,V)
000026      000          AC(I) = A1(I1)
000027      000          AC(I+1) = F1*A1(I1+1) + F2*V
000028      000          I1 = I1 + 2
000029      000          GO TO 100
000030      000       30 AC(I) = A1(I1)
000031      000          AC(I+1) = F1*A1(I+1) + F2*A2(N2+1)
000032      000          I1 = I1 + 2
000033      000          GO TO 100
000034      000       50 CALL DIDEG1(A2(I2),A1,V)
000035      000          AC(I) = A2(I2)
000036      000          AC(I+1) = F1*V + F2*A2(I2+1)
000037      000          I2 = I2 + 2
000038      000          GO TO 100
000039      000       70 AC(I) = A2(I2)
000040      000          AC(I+1) = F1*A1(N1+1) + F2*A2(I2+1)
000041      000          I2 = I2 + 2
000042      000      100 IF(I .EQ. L) GO TO 120
000043      000          I = I + 2
000044      000          IF(I .LE. NC) GO TO 5
000045      000          WRITE(6,110) AC(1)
000046      000      110 FORMAT(67HOINSUFFICIENT SPACE AVAILABLE IN AC ARRAY IN SUBROUTINE
000047      000         1QCOMB, IC = I5)
000048      000          CALL GENOUT(AC(2),1,NC,'QCOMBINED ARRAY')
000049      000          CALL WLKBCK
000050      000          CALL EXIT
000051      000      120 NC = I
000052      000          AC(1) = AC1
000053      000          CALL LINECK(2)
000054      000          CALL GENOUT(AC(2),1,NC,'QCOMBINED ARRAY')
000055      000          RETURN
```

```
QCOMB
000056      000         END
END ELT.

.*HDG,P REVPOL
```

```
SIQDUM                                                                          DATE 022875        PAGE     1

@ELT,L SIQDUM
ELTOT7 RLIB70 02/28-03:21:38-(0,)
000001      000    QDUM   PROC
000002      000           IF(FLD(4,1,NSQ1(K1+1)).EQ.0) GO TO 600
000003      000           NTYPE = FLD(0,5,NSQ2(K2))
000004      000           K2 = K2+1
000005      000           GO TO (600,600,600,599,599,599,599,599,599,599,599,599),NTYPE     VERS   5
000006      000      599  K2 = K2+1
000007      000      600  CONTINUE
000008      000     END

END ELT.


@HDG,P SIVARC
```

```
@ELT,L REVPOL
ELTOT7 RLIB70 02/28-03:20:38-(0,)
000001      000          SUBROUTINE REVPOL(Y,A,X)
000002      000     C
000003      000          DIMENSION A(1)
000004      000     C
000005      000          EQUIVALENCE (D,N)
000006      000     C
000007      000     C
000008      000          D = A(1)
000009      000          N = N
000010      000          IF(MOD(N,2) .GT. 0) GO TO 20
000011      000          IF(A(N+1) .GT. A(3)) GO TO 16
000012      000          X = A(2)
000013      000          IF(Y .GE. A(3)) RETURN
000014      000          X = A(N)
000015      000          IF(Y .LE. A(N+1)) RETURN
000016      000          DO 15 I=4,N,2
000017      000          IF(Y - A(I+1)) 15,10,5
000018      000        5 X = A(I-2) + (Y-A(I-1))*(A(I)-A(I-2))/(A(I+1)-A(I-1))
000019      000          RETURN
000020      000       10 X = A(I)
000021      000          RETURN
000022      000       15 CONTINUE
000023      000          GO TO 20
000024      000       16 X = A(2)
000025      000          IF(Y .LE. A(3)) RETURN
000026      000          X = A(N)
000027      000          IF(Y .GE. A(N+1)) RETURN
000028      000          DO 19 I=4,N,2
000029      000          IF(Y - A(I+1)) 17,18,19
000030      000       17 X = A(I-2) + (Y-A(I-1))*(A(I)-A(I-2))/(A(I+1)-A(I-1))
000031      000          RETURN
000032      000       18 X = A(I)
000033      000          RETURN
000034      000       19 CONTINUE
000035      000       20 WRITE(6,25) A(1)
000036      000       25 FORMAT(36H WRONG ARRAY LENGTH FOR REVPOL, IC = I5)
000037      000          CALL WLKBCK
000038      000          CALL EXIT
000039      000          END

END ELT.

@HDG,P RPOL
```

```
*ELT,L RPOL
ELTOTT RL1870 02/28-03:20:39-(4,1
000001      002          SUBROUTINE RPOL(L,Y,VAR)
000002      000    C
000003      000          DIMENSION RDATA(1)
000004      000    C
000005      000          COMMON /ARRAY / NDATA(1)
000006      002          COMMON /FDATA / DUM(17), TZERO
000007      000    C
000008      000          EQUIVALENCE (RDATA(1),NDATA(1))
000009      000    C
000010      000          DATA NOUT / 6 /
000011      000          DATA LOC  / 2 /
000012      002          DATA KERR / 0 /
000013      000    C
000014      000    C
000015      000    C
000016      004       10 M = 2
000017      000          NP = NDATA(L)
000018      000          K = L + M
000019      000          IF(RDATA(K)-Y) 20,100,50
000020      000       20 M = M + 2
000021      000          IF(M .GT. NP) GO TO 90
000022      000          DO 30 I=M,NP,2
000023      000          N = K + 2
000024      000          IF (RDATA(N)-Y) 25,100,80
000025      000       25 K = N
000026      000       30 CONTINUE
000027      000          GO TO 90
000028      000       50 M = M - 1
000029      000          IF(M .LT. 2) GO TO 90
000030      000          DO 60 I=1,M,2
000031      000          K = K - 2
000032      000          IF (RDATA(K)-Y) 80,100,60
000033      000       60 CONTINUE
000034      000          GO TO 90
000035      000       80 X = RDATA(K-1) + (Y-RDATA(K))*(RDATA(K+1)-RDATA(K-1))
000036      000         1                              /(RDATA(K+2)-RDATA(K  ))
000037      000          GO TO 110
000038      000       90 WRITE(NOUT,95) Y, L, NP, (RDATA(L+I),I=1,NP)
000039      000       95 FORMAT( 8H0* * *   G13.8, 40H IS OUT OF RANGE OF THE TABLE STORED
000040      000         1AT LOCATION  I9,  7H  * * * // 8X I5 // (8X 5G15.8))
000041      002          KERR = KERR+1
000042      002          IF(KERR.LE.10) GO TO 100
000043      002          CALL OUTCAL
000044      002          CALL EXIT
000045      000      100 X = RDATA(K-1)
000046      000      110 LOC = K - L
000047      002          VAR = X - TZERO
000048      000          RETURN
000049      000          END

END ELT.


*HDG,P SINVRS
```

```
*ELT,L SINVRS
ELTOT7 RLI870 02/28-03:20:41-(0,)
000001      000          SUBROUTINE SINVRS(A,DET)
000002      000    C
000003      000          DOUBLE PRECISION D, PIV, SUM
000004      000    C
000005      000          DIMENSION A(1)
000006      000    C
000007      000          EQUIVALENCE (R,NC)
000008      000    C
000009      000    C
000010      000          R = A(1)
000011      000          IC = NC - 2
000012      000          R = A(2)
000013      000          N = NC
000014      000          ISTRT = 3
000015      000          M1 = N*N
000016      000          M2 = (M1+N)/2
000017      000          IF(IC.EQ.M1) GO TO 5
000018      000          IF(IC.EQ.M2) GO TO 20
000019      000          WRITE(6,1) IC, N
000020      000        1 FORMAT(61HOINCORRECT NUMBER OF ELEMENTS INPUT TO SUBROUTINE INVRS.
000021      000         1 IC = I5, 5H, N = I5)
000022      000          CALL WLKBCK
000023      000          CALL EXIT
000024      000    C
000025      000        5 LOC1=NC+4
000026      000          K=1
000027      000          DO 15 I=2,N
000028      000          DO 10 J=1,N
000029      000          A(LOC1)=A(LOC1+K)
000030      000          LOC1=LOC1+1
000031      000       10 CONTINUE
000032      000          K=K+I
000033      000       15 CONTINUE
000034      000          GO TO 20
000035      000    C
000036      000          ENTRY INVRS(A,M,DET)
000037      000          N=M
000038      000          IC=N*(N+1)/2
000039      000          M2=IC
000040      000          ISTRT=0
000041      000    C
000042      000       20 ASSIGN 120 TO II
000043      000          D = 1.000
000044      000          LOC1=ISTRT
000045      000          DO 300 I=1,N
000046      000          IM1 = I - 1
000047      000          ASSIGN 140 TO JJ
000048      000          DO 200 J=1,N
000049      000          LOC1 = LOC1 + 1
000050      000          SUM = A(LOC1)
000051      000          GO TO II
000052      000       90 LOC2 = I + ISTRT
000053      000          LOC3 = J + ISTRT
000054      000          DO 100 K=1,IM1
000055      000          SUM = SUM - A(LOC2)*A(LOC3)
```

```
000056  000          NMK = N - K
000057  000          LOC2 = LOC2 + NMK
000058  000          LOC3 = LOC3 + NMK
000059  000      100 CONTINUE
000060  000      120 GO TO JJ
000061  000      140 ASSIGN 160 TO JJ
000062  000          IF(SUM .LE. 0.000) GO TO 990
000063  000          PIV = DSQRT(SUM)
000064  000          D = D*PIV
000065  000          A(LOC1) = PIV
000066  000          GO TO 200
000067  000      160 A(LOC1) = SUM/PIV
000068  000      200 CONTINUE
000069  000          ASSIGN 90 TO II
000070  000      300 CONTINUE
000071  000          DET = D*D
000072  000      C
000073  000          LOC2S = LOC1 + 1
000074  000          LOC3S = LOC1
000075  000          A(LOC1) = 1.0/A(LOC1)
000076  000          LOC1 = LOC1 - 1
000077  000          PIV = A(LOC1-1)
000078  000          DO 600 I=2,N
000079  000          LOC2S = LOC2S - I
000080  000          L = I - 1
000081  000          DO 500 J=2,I
000082  000          SUM = 0.000
000083  000          LOC2 = LOC2S
000084  000          LOC3 = LOC3S
000085  000          DO 400 K=1,L
000086  000          SUM = SUM + A(LOC2)*A(LOC3)
000087  000          LOC2 = LOC2 + 1
000088  000          LOC3 = LOC3 + I - K - 1
000089  000      400 CONTINUE
000090  000          LOC3S = LOC3S - 1
000091  000          L = L - 1
000092  000          A(LOC1) = -SUM/PIV
000093  000          LOC1 = LOC1 - 1
000094  000      500 CONTINUE
000095  000          A(LOC1) = 1.0/A(LOC1)
000096  000          LOC1 = LOC1 - 1
000097  000          PIV = A(LOC1-1)
000098  000      600 CONTINUE
000099  000      C
000100  000          LOC1=ISTRT+1
000101  000          DO 900 I=1,N
000102  000          LOC2S = LOC1
000103  000          LOC3  = LOC1
000104  000          DO 800 J=I,N
000105  000          LOC2  = LOC2S
000106  000          LOC2S = LOC2S + I
000107  000          SUM = 0.000
000108  000          DO 700 K=J,N
000109  000          SUM = SUM + A(LOC2)*A(LOC3)
000110  000          LOC2 = LOC2 + 1
000111  000          LOC3 = LOC3 + 1
000112  000      700 CONTINUE
```

```
000113  000          A(LOC1) = SUM
000114  000          LOC1 = LOC1 + 1
000115  000      800 CONTINUE
000116  000      900 CONTINUE
000117  000    C
000118  000          IF(IC.EQ.M2) RETURN
000119  000          LOC1=M1+3
000120  000          DO 920 I=N,2,-1
000121  000          DO 910 J=N,1,-1
000122  000          II=MINO(I,J)
000123  000          JJ=MAXO(I,J)
000124  000          LOC2=ISTRT+(II-1)*N-(II-1)*II/2+JJ
000125  000          A(LOC1)=A(LOC2)
000126  000          LOC1=LOC1-1
000127  000      910 CONTINUE
000128  000      920 CONTINUE
000129  000          RETURN
000130  000    C
000131  000      990 WRITE(6,995)
000132  000      995 FORMAT(38HOSINGULAR MATRIX ENCOUNTERED BY SINVRS)
000133  000          CALL WLKBCK
000134  000          CALL EXIT
000135  000          RETURN
000136  000          END

END ELT.


@HOG,P SKP8
```

```
*ELT,L SKPB
ELTOT7 RL:B70 02/28-03:20:42-(1,)
000001      000          SUBROUTINE SKPB(JSW)
000002      000    C
000003      000          COMMON /TAPE  / NIN, NOUT
000004      000          COMMON /CARD  / KRD, KOL, MXKOL
000005      000          COMMON /C1MAGE/ KARD(80)
000006      000    C
000007      000          JSW = 1
000008      000          IF(KOL .GT. MXKOL) GO TO 30
000009      000       15 J = KOL
000010      000          DO 20 KOL=J,MXKOL
000011      000          IF(KARD(KOL) .NE. 1H ) GO TO 50
000012      000       20 CONTINUE
000013      000       30 CALL CARDIN(JSW)
000014      000          GO TO(15,50), JSW
000015      000       50 RETURN
000016      000          END

END ELT.


*HDG,P SKPTE
```

```
*ELT,L SKPTE
ELTOT7 RL1870 02/28-03:20:43-(1,)
000001      000          SUBROUTINE SKPTE(ISW)
000002      000    C
000003      000          COMMON /CARD  / KRD, KOL, MXKOL
000004      000          COMMON /CIMAGE/ KARD(80)
000005      000          DATA KE, KN, KD / 1HE, 1HN, 1HD /
000006      000          ISW = 1
000007      000        5 J = KOL
000008      000          M = MXKOL - 2
000009      000          DO 15 KOL=J,M
000010      000          IF(KARD(KOL) .NE. KE) GO TO 15
000011      000          IF(KARD(KOL+1) .NE. KN) GO TO 15
000012      000          IF(KARD(KOL+2) .EQ. KD) GO TO 20
000013      000       15 CONTINUE
000014      000          CALL CARDIN(ISW)
000015      000          GO TO(5,50), ISW
000016      000       20 J = KOL + 3
000017      000          DO 30 KOL=J,MXKOL
000018      000          IF(KARD(KOL) .EQ. 1H,) GO TO 40
000019      000       30 CONTINUE
000020      000          CALL CARDIN(ISW)
000021      000          GO TO 50
000022      000       40 KOL = KOL + 1
000023      000       50 RETURN
000024      000          END

END ELT.


*HDG,P SNDSNR
```

```
@ELT,L SNDSNR
ELTOTT RLI870 02/28-03:20:44-(1,)
000001      000          SUBROUTINE SNDSNR
000002      000    C
000003      000    C        STEADY STATE EXECUTION ROUTINE FOR SINDA    - FORTRAN 5
000004      000    C
000005      000    C        THE LONG PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000006      000    C
000007      000    C        ALGORITHM IS BASED ON NEWTON-RHAPSON METHOD
000008      000    C
000009      000    C        A GAUSS-JORDAN REDUCTION (SUBROUTINE GJR) IS USED
000010      000    C        TO SOLVE THE LINEAR SYSTEM EQUATIONS
000011      000    C
000012      000    C
000013      000          DIMENSION V(2)
000014      001          LOGICAL FLOW
000015      001          COMMON /FDINNS/ NTYP, NSYS
000016      000          INCLUDE COMM,LIST
000017      000          INCLUDE DEFF,LIST
000018      000          IF (KON(5). LE. 0) KON(5) =1000
000019      000          IF (CON(19) .LE. 0.) CON(19) =0.0001
000020      000          IF (CON(33) .LE. 0.) CON(33) = 1.0E-05
000021      000          IF (CON(50) .LE. 0.) CON(50)=1.0
000022      000          WRITE (6,801) KON(5),CON(19),CON(33)
000023      000          KON(28) = KON(28) +6
000024      000          IEBAL =0
000025      000          IMAXT =0
000026      000          PASS = -1.0
000027      000          NNC = NNA+NND
000028      001          FLOW = .FALSE.
000029      001          NSP = 0
000030      001          IXF = NTH
000031      001          IF(NSYS .LT. 1) GO TO 2
000032      001          FLOW = .TRUE.
000033      001          NSP = NNT
000034      001          DO 1 I=1,NNT
000035      001          NX(IXF+I) = 0
000036      001        1 CONTINUE
000037      001        2 IE1 = IXF + NSP
000038      000          NLA = NDIM
000039      000          NNCP1 = NNC+1
000040      001          JJ = NBC+ NNCP1 + NSP + NNCP1
000041      000          NTH = NTH+JJ
000042      000          NDIM = ND10-JJ
000043      000          WRITE (6,806) NDIM
000044      000          IF (NDIM .LT. 0) GO TO 995
000045      000          CON(1) = CON(13)
000046      000          CON(14) = CON(13)
000047      000          GO TO 15
000048      000       10 CON(1) = CON(13) + CON(18)
000049      000          IF (CON(1) - CON(3) .GT. 0.) CON(1) = CON(3)
000050      000          CON(14) = (CON(1) + CON(13))/2.0
000051      000          CON(2) = CON(1) - CON(13)
000052      000    C     COMPUTE STEADY STATE TEMPERATURES
000053      000       15 LAX = KON(5)
000054      000          DO 120 K1=1,LAX
000055      000          KON(20) =K1
```

```
000056  000        QIN = 0.
000057  000  C       EVALUATE TEMPERATURE VARYING PROPERTIES AND Q
000058  000        CALL NONLIN
000059  000  C       . HEAT INPUT Q(I) IS STORED IN LOCATIONS NNC*NNC+I, I=1,...NNC
000060  000  C         OF EXTRA
000061  001        KOP = KON(7)
000062  001        IF(FLOW) CALL FLUID(5,0,IXF,0.0,KOP)
000063  000        DO 21 I=1,NNC
000064  000        QIN = QIN + Q(I)
000065  000        DO 21 J=1,NNC
000066  000        L=(J-1)* NNC + I
000067  000        X(L) =0.
000068  000        L=NNC*NNC+I
000069  000     21 X(L) =Q(I)
000070  000        IF (PASS .GE. 0.) GO TO 25
000071  000        CALL OUTCAL
000072  000        PASS = 1.
000073  000     25 NPASS=1
000074  000        KJ=1
000075  000        J1 =0
000076  000        DO 70 I=1,NNC
000077  000        LE=IE1+I
000078  001        IF(.NOT. FLOW) GO TO 35
000079  001        LMP = NX(IXF+I)
000080  001        IF(LMP .EQ. 0)GO TO 35
000081  001        HA = X(IXF+LMP)
000082  001        J = (I-1)*NNC + I
000083  001        X(J) = X(J) + HA
000084  001        J = NNC*NNC + I
000085  001        X(J) = X(J) + HA * (T(LMP) - T(I))
000086  000     35 J1 = J1 +1
000087  000        LG = FLD(5,16,NSQ1(J1))
000088  001        IF(LG .EQ. 0) GO TO 70
000089  000        LTA = FLD(22,14,NSQ1(J1))
000090  000        IF (FLD(3,1,NSQ1(J1)) .EQ. 0) GO TO 40
000091  000        T1 = T(I)+460.
000092  000        T2 = T(LTA) +460.
000093  000  C          START BUILDING THE JACOBIAN MATRIX
000094  000        GLG = G(LG) *CON(50)
000095  000        CON1 = 4.0 * GLG * T1**3
000096  000        CON3 =GLG * (T1*T1 + T2*T2) * (T1 + T2)
000097  000        IF (FLD(21,1,NSQ1(J1)) .EQ. 1) GO TO 50
000098  000        IF (LTA .GT. NNC) GO TO 45
000099  000        J = (LTA -1) * NNC +I
000100  000        X(J) = -4.0 * GLG *T2**3
000101  000        GO TO 45
000102  000     40 CON1 = G(LG)
000103  000        CON3 = G(LG)
000104  000        IF (FLD(21,1,NSQ1(J1)) .EQ. 1) GO TO 50
000105  000        IF (LTA .GT. NNC) GO TO 45
000106  000        J =(LTA -1)*NNC +I
000107  000        X(J) = -G(LG)
000108  000     45 J = (I-1) * NNC +I
000109  000        X(J) =  X(J) +CON1
000110  000        J = NNC * NNC  +I
000111  000        X(J)  = X(J)  + CON3*(T(LTA) -T(I))
000112  000     50 IF (NSQ1(J1) .GT. 0) GO TO 35
```

```
000113  000      70 CONTINUE
000114  000         V(1) =4.
000115  000   C        WITH X(K) BEING THE KTH ITERATE AND J(K) THE JACOBIAN
000116  000   C        THE ALGORITHM IS THE FOLLOWING
000117  000   C        J(K)(X(K+1) -X(K)) = B - F(X(K))
000118  000   C        WHERE WE ARE SOLVING  F(I,X) = B(1)
000119  000   C        I = 1,...,N
000120  000         J=NNC* NNCP1 +1
000121  000         CALL GJR(X,NNCP1,NNC,NNC,NNCP1,S1004,X(J),V)
000122  000         DO 80 I=1,NNC
000123  000         J = NNC*NNC +I
000124  000      80 T(I) = T(I) + X(J)
000125  000         AAMAX = ABS(X(NNC*NNC +1))
000126  000         DO 90 I=1,NNC
000127  000         J = NNC* NNC +I
000128  000      90 AAMAX = AMAX1(ABS(X(J)),AAMAX)
000129  000         QOUT = 0.
000130  000         J1 = 0.
000131  000         DO 150 I=1,NNC
000132  001         IF(.NOT. FLOW) GO TO 135
000133  001         LMP = NX(IXF+I)
000134  001         IF(LMP .EQ. 0) GO TO 135
000135  001         QOUT = QOUT + X(IXF+LMP)*(T(I) - T(LMP))
000136  000     135 J1 = J1 +1
000137  000         LTA = FLD(22,14,NSQ1(J1))
000138  000         IF (LTA .LE. NNC) GO TO 145
000139  000         LG = FLD(5,16,NSQ1(J1))
000140  001         IF(LG .EQ. 0) GO TO 150
000141  000         IF (FLD(3,1,NSQ1(J1)) .EQ. 0) GO TO 140
000142  000         T1 = T(I) +460.
000143  000         T2 = T(LTA) + 460.
000144  000         QOUT = QOUT +G(LG)*CON(50)*(T1**4 -T2**4)
000145  000         GO TO 145
000146  000     140 QOUT = QOUT + G(LG)* (T(I) -T(LTA))
000147  000     145 IF (NSQ1(J1) .GT. 0) GO TO 135
000148  000     150 CONTINUE
000149  000         CON(32) = ABS(QIN-QOUT)
000150  000         IF (CON(32) .GT. CON(33) .OR. IEBAL .GT. 0) GO TO 160
000151  000         WRITE (6,882) KON(20),CON(32),AAMAX
000152  000         IEBAL =1
000153  000     160 CONTINUE
000154  000   C         CHECK FOR CONVERGENCE
000155  000         IF (AAMAX .GT. CON(19) .OR. IMAXT .GT. 0) GO TO 110
000156  000         WRITE (6,883) KON(20),CON(32),AAMAX
000157  000         IMAXT =1
000158  000     110 CONTINUE
000159  000         IF (CON(32) .LE. CON(33)  .AND.  AAMAX .LE. CON(19)) GO TO 130
000160  000         IF (KON(7) .LE. 0) GO TO 120
000161  000         WRITE (6,888) KON(20),AAMAX,CON(32)
000162  000         KON(28) =KON(28) +2
000163  000         CALL OUTCAL
000164  000     120 CONTINUE
000165  000     121 WRITE (6,885) LAX
000166  000     130 CALL VARBL2
000167  000         CON(13) = CON(1)
000168  000         WRITE (6,887) KON(5),KON(20),CON(19),AAMAX,CON(33),CON(32)
000169  000         KON(28) = KON(28) + 9
```

```
000170  000          CALL OUTCAL
000171  000          IF (CON(3) .GT. CON(1) * 1.000001) GO TO 10
000172  000          NTH=IE1
000173  000          NDIM = NLA
000174  000          RETURN
000175  000      995 WRITE (6,884)
000176  000          GO TO 1000
000177  000     1000 CALL OUTCAL
000178  000          RETURN
000179  000      881 FORMAT (/ 5X,19H CONTROL PARAMETERS,//10X,8H NLOOP =, I5,3X, 9H AR
000180  000         *LXCA =,F10.6,3X, 9H BALENG =,1PE10.5,/)
000181  000      882 FORMAT (/ 5X,63H THE DESIRED SYSTEM HEAT BALANCE HAS BEEN REACHED
000182  000         *  -- LOOPCT =, I5, 9H ENGBAL = E12.5, 9H ARLXCC =,E12.5/)
000183  000      883 FORMAT (/ 5X,76H THE MAXIMUM TEMPERATURE CHANGE IS NOW BELOW THE S
000184  000         *PECIFIED LIMIT -- LOOPCT =, I5, 9H ENGBAL =,E12.5, 9H ARLXCA =,E12
000185  000         *.5/)
000186  000      887 FORMAT (/ 5X,35H FINAL CONTROL PARAMETER COMPARISON,/10X,9H  NLOOP
000187  000         * =,I5,10X, 9H LOOPCT =,I5,/ 5X, 9H ARLXCA =,E12.5,3X, 9H ARLXCC =,
000188  000         *E12.5,/ 5X, 9H BALENG =,E12.5,3X, 9H ENGBAL =,E12.5,/)
000189  000      888 FORMAT (/ 5X,8HLOOPCT =, I5,3X,8HARLXCC =,E12.5,3X,8HENGBAL =,E12.
000190  000         *5)
000191  000      884                        FORMAT(' NO OF NODES HAS  EXCEEDED  DIMENSION
000192  000         1 LIMITS.  TO RERUN CHANGE DIMENSION STATEMENT AND NDIM STATEMENT T
000193  000         2O HANDLE LARGER NUMBER ')
000194  000      885 FORMAT(35H ITERATION COUNT EXCEEDED, NLOOP = ,I10)
000195  000      886 FORMAT(I6,20H LOCATIONS AVAILABLE)
000196  000     1003 FORMAT(1HO,' ERROR IN GIR')
000197  000     1004 WRITE (6,1003)
000198  000          RETURN
000199  000          END

END ELT.


*ROG,P SNFROL
```

```
@ELT,L SNFRDL
ELT077 RLIB70 02/28-03:20;48-(2,)
000001      000          SUBROUTINE SNFRDL
000002      000    C     EXPLICIT FORWARD DIFFERENCING EXECUTION SUBROUTINE FOR SINDA F-V
000003      000    C     THE LONG PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000004      002          COMMON /FDIMNS/ NTYP, NSYS
000005      000          INCLUDE COMM,LIST
000006      000          INCLUDE DEFF,LIST
000007      000          IF(CON(4).LT.1.0) CON(4) = 1.0
000008      000          IF (KON(5) .LE. 2) KON(5) =10
000009      000          IF(CON(6).LE.0.) CON(6) = 1.E+8
000010      000          IF(CON(8).LE.0.) CON(8) = 1.E+8
000011      000          IF(CON(9).LE.0.) CON(9) = 1.0
000012      000          IF(CON(11).LE.0.) CON(11) = 1.E+8
000013      000          IF(CON(18).LE.0.) GO TO 999
000014      000          IF (CON(19) .LE. 0.) CON(19) =.1
000015      000          IF(KON(31).NE.1) GO TO 995
000016      000          IF (CON(50) .LE. 0.) CON(50) =1.
000017      000          PASS = -1.0
000018      000          NNC = NND+NNA
000019      000          IE = NTH
000020      000          NLA = NDIM
000021      002          NTH = NTH+NNT
000022      002          NDIM = NDIM-NNT
000023      000    C     CHECK FOR EXTRA LOCATIONS FOR CALCULATED NODES
000024      000          I = NLA-NNC
000025      000          IF(I.LT.0) GO TO 998
000026      000          L1 = NND+1
000027      000          TSTEP = CON(18)
000028      000          TPRINT = CON(13)
000029      00     C     INITALIZE TIME SUM BETWEEN OUTPUT INTERVALS
000030      00         5 TSUM = 0.0
000031      000    C     DOES OLD TIME PLUS THE OUTPUT INTERVAL EXCEED THE STOP TIME
000032      000          IF(CON(13)+CON(18).LE.CON(3)) GO TO 10
000033      000    C     DONT EXCEED IT
000034      000          CON(18) = CON(3)-CON(13)
000035      000    C     IS THE TIME STEP LARGER THAN ALLOWED
000036      000       10 IF(TSTEP.LE.CON(8)) GO TO 15
000037      000          TSTEP = CON(8)
000038      000    C     DOES THE TIME SUM PLUS THE TIME STEP EXCEED OUTPUT INTERVAL
000039      000       15 IF(TSUM+TSTEP-CON(18)) 25,30,20
000040      000    C     DONT EXCEED IT
000041      000       20 TSTEP = CON(18)-TSUM
000042      000          GO TO 30
000043      000    C     DOES TIME SUM PLUS TWO TIME STEPS EXCEED OUTPUT INTERVAL
000044      000       25 IF(TSUM+2.0*TSTEP.LE.CON(18)) GO TO 30
000045      000    C     APPROACH THE OUTPUT INTERVAL GRADUALLY
000046      000          TSTEP = (CON(18)-TSUM)/2.0
000047      000    C     STORE DELTA TIME STEP IN THE CONSTANTS
000048      000       30 CON(2) = TSTEP
000049      000    C     IS THE TIME STEP USED LESS THAN THE TIME STEP ALLOWED
000050      000          IF(TSTEP.LT.CON(21)) GO TO 997
000051      000    C     CALCULATE THE NEW TIME
000052      000          CON(1) = TPRINT+TSUM+TSTEP
000053      000    C     COMPUTE THE MEAN TIME BETWEEN ITERATIONS
000054      000          CON(14) = (CON(1)+CON(13))/2.0
000055      000    C     ZERO OUT ALL SOURCE LOCATIONS AND EXTRA LOCATIONS
```

```
000056  000        DO 35 I = 1,NND
000057  000        LE = IE+I
000058  000        X(LE) = 0.0
000059  000        Q(I) = 0.0
000060  000     35 CONTINUE
000061  000  C     SHIFT THE ARITHMETIC TEMPERATURES INTO THE EXTRA LOCATIONS
000062  002        IF(NND .EQ. NNT) GO TO 45
000063  002        DO 36 I=L1,NNT
000064  002        X(IE+I) = T(I)
000065  002     36 CONTINUE
000066  000        DO 40 I = L1,NNC
000067  000        Q(I) = 0.0
000068  000     40 CONTINUE
000069  000     45 KON(12) = 0
000070  000        CALL VARBL1
000071  000        IF(KON(12).NE.0) GO TO 10
000072  000        J1 = 0
000073  000        J2 = 1
000074  000        TCGM = 0.0
000075  000        CKM = 1.E+8
000076  000  C     CALCULATE Q SUM AND G SUM
000077  000        DO 85 I = 1,NND
000078  000        LE = IE+I
000079  000        INCLUDE VARC,LIST
000080  000        INCLUDE VARD,LIST
000081  000     70 J1 = J1+1
000082  000        LG = FLD(5,16,NSQ1(J1))
000083  002        IF(LG .EQ. 0) GO TO 85
000084  000        LTA = FLD(22,14,NSQ1(J1))
000085  000        INCLUDE VARG,LIST
000086  000  C     CHECK FOR RADIATION CONDUCTOR
000087  000        IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 75
000088  000        T1 = T(I)+460.0
000089  000        T2 = T(LTA)+460.0
000090  000        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000091  000        GV =GV*CON(50)
000092  000        GO TO 80
000093  000     75 GV = G(LG)
000094  000  C     OBTAIN THE Q RATE THRU THE CONDUCTOR
000095  000     80 Q(I) = Q(I)+GV*(T(LTA)-T(I))
000096  000  C     SAVE SUMMATION OF CONDUCTORS
000097  000        X(LE) = X(LE)+GV
000098  000  C     CHECK FOR LAST CONDUCTOR
000099  000        IF(NSQ1(J1).GT.0) GO TO 70
000100  000     85 CONTINUE
000101  000  C     OBTAIN NEW DIFFUSION TEMPERATURES, DTMPCC AND CSGMIN
000102  000        DO 100 I = 1,NND
000103  000        LE = IE+I
000104  002        IF(.NOT. X(LE) .GT. 0.0) GO TO 90
000105  000  C     CALCULATE C/SK MINIMUM
000106  000        T1 = C(I)/X(LE)
000107  000        IF(T1.GE.CKM) GO TO 90
000108  000        CKM = T1
000109  000        KON(35) = 1
000110  000  C     COMPUTE NEW TEMPERATURES USING CALCULATED SOURCE TERMS
000111  000     90 T1 = TCTEP*Q(I)/C(I)
000112  000  C     CALCULATE THE ABSOLUTE VALUE TEMPERATURE CHANGE
```

```
000113  000          T2 = ABS(T1)
000114  000    C     SAVE THE LARGEST TEMPERATURE CHANGE
000115  000          IF(TCGM.GE.T2) GO TO 95
000116  000          TCGM = T2
000117  000          KON(36) = I
000118  000    C     STORE THE TEMPERATURES
000119  000       95 X(LE) = T(I)
000120  000          T(I) = T(I)+T1
000121  000      100 CONTINUE
000122  000          CON(17) = CKM
000123  000          DELTA = CKM/CON(4)
000124  002          KOP = KON(7)
000125  002          IF(NSYS .NE. 0) CALL FLUID(1,0,0,0.,KOP)
000126  000    C     CHECK FOR FIRST PASS
000127  000          IF(PASS.GT.0.0) GO TO 115
000128  000    C     UNDO THE TEMPERATURE CALCULATIONS
000129  002      105 DO 110 I = 1,NNT
000130  000          LE = IE+I
000131  000          T(I) = X(LE)
000132  000      110 CONTINUE
000133  000          IF(PASS.GT.0.0) GO TO 15
000134  000          PASS = 1.0
000135  000          CON(1) = TPRINT
000136  000          CON(2) = 0.0
000137  001          TSTEP = DELTA*0.75
000138  000          GO TO 195
000139  000    C     IS THE TIME STEP USED LESS THAN THE TIME STEP CALCULATED
000140  000      115 IF(TSTEP.LE.DELTA) GO TO 130
000141  000    C     COMPUTE THE TIME STEP
000142  001          TSTEP = DELTA*0.75
000143  000          GO TO 105
000144  001      120 TSTEP = 0.75*TSTEP*CON(6)/TCGM
000145  000          GO TO 105
000146  001      125 TSTEP = 0.75*TSTEP*CON(11)/TCGM
000147  000          GO TO 105
000148  000    C     SEE IF THE TEMPERATURE CHANGE WAS TOO LARGE
000149  000      130 IF(TCGM.GT.CON(6)) GO TO 120
000150  000    C     STORE THE MAXIMUM DIFFUSION TEMPERATURE CHANGE
000151  000          CON(15) = TCGM
000152  000    C     CHECK TO SEE IF THERE ARE ANY ARITHMETIC NODES
000153  000          IF(NNA.LE.0) GO TO 185
000154  000    C     COMPUTE ARITHMETIC TEMPERATURES BY SUCCESSIVE POINT OVER-RELAX
000155  000          BADAMP = 1.
000156  000          LAX = KON(5)
000157  000    C     STORE Q'S CALCULATED DURING DIFFUSION NODE LOOP
000158  000          DO 133 II=1,NNC
000159  000      133 X(II+NNA) = Q(II)
000160  000      131 CONTINUE
000161  000          DN = CON(9)
000162  000          DD = 1.0-DN
000163  000          DO 170 I = 1,LAX
000164  000          JJ1 = J1
000165  000          JJ2 = J2
000166  000          TCGM = 0.0
000167  000          KON(20) = I
000168  000          DO 165 L = L1,NNC
000169  000          SUMC = 0.0
```

```
000170  000          SUMCV = 0.0
000171  000          IF(I.GT.1) GO TO 6000
000172  000          INCLUDE VR02,LIST
000173  000      135 JJ1 = JJ1+1
000174  000          LG = FLD(5,16,NSO1(JJ1))
000175  000          LTA = FLD(22,14,NSO1(JJ1))
000176  000          IF(I.GT.1) GO TO 4000
000177  000          INCLUDE VRG2,LIST
000178  000      C   CHECK FOR RADIATION CONDUCTOR
000179  000          IF(FLD(3,1,NSO1(JJ1)).EQ.0) GO TO 155
000180  000          T1 = T(L)+460.0
000181  000          T2 = T(LTA)+460.0
000182  000          GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000183  000          GV =GV*CON(50)
000184  000          GO TO 160
000185  000      155 GV = G(LG)
000186  000      160 SUMC = SUMC+GV
000187  000          SUMCV = SUMCV+GV*T(LTA)
000188  000      C   CHECK FOR LAST CONDUCTOR
000189  000          IF(NSO1(JJ1).GT.0) GO TO 135
000190  000          T2 = DD*T(L)+DN*(SUMCV+Q(L))/SUMC
000191  000      C   OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000192  000          T1 = ABS(T(L)-T2)
000193  000      C   STORE THE NEW TEMPERATURE
000194  000          T(L) = T2
000195  000      C   SAVE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000196  000          IF(TCGM.GE.T1) GO TO 165
000197  000          TCGM = T1
000198  000          KON(37) = L
000199  000      165 CONTINUE
000200  000      C   SEE IF RELAXATION CRITERIA WAS MET
000201  000          OLD3 = OLD2
000202  000          OLD2 = OLD1
000203  000          OLD1 = TCGM
000204  000          LTL3 = LTL2
000205  000          LTL2 = LTL1
000206  000          LTL1 = L
000207  000          TEMP3 = TEMP2
000208  000          TEMP2 = TEMP1
000209  000          TEMP1 = T(L)
000210  000          IF(TCGM.LE.CON(19)) GO TO 175
000211  000      170 CONTINUE
000212  000          DELTAA= OLD2 - OLD1
000213  000          DELTAB= OLD3 - OLD2
000214  000          DO 173 II=1,NNC
000215  000      173 Q(II) = X(II+NNA)
000216  000          IF (DELTAA .LE. 0.) GO TO 174
000217  000          IF (DELTAB .LT. DELTAA) GO TO 174
000218  000          IF (DELTAB .LE. 0.) GO TO 174
000219  000          IF (LTL1 .NE. LTL2) GO TO 174
000220  000          IF (LTL2 .NE. LTL3) GO TO 174
000221  000          IF (TEMP1 .GT. TEMP2) GO TO 176
000222  000          IF (TEMP2 .GT. TEMP3) GO TO 174
000223  000          GO TO 177
000224  000      176 IF (TEMP1 .LT. TEMP3) GO TO 174
000225  000      177 CONTINUE
000226  000          KON(5) = KON(5) + LAX
```

```
000227      000          IF ((CON(9) +.1) .LT. BADAMP) CON(9) = CON(9) +.1
000228      000          IF (CON(9) .GT. 1.) CON(9) =1.
000229      000          WRITE (6,704) LAX,KON(5),DN,CON(9)
000230      000      704 FORMAT (//10X,40HTHE SOLUTION WAS CONVERGING WHEN NLOOP =,I5,43H W
000231      000         *AS EXCEEDED.   NLOOP WILL BE INCREASED TO,I5,1H,/10X, 20HDAMPA INC
000232      000         *REASED FROM,F5.3,3H TO,F5.3,23H AND THE LOOP CONTINUED,//)
000233      000          KON(28) = KON(28) +6
000234      000          GO TO 131
000235      000      C   REDUCE DAMPA AND TRY AGAIN
000236      000      174 DO 171 II=1,NNA
000237      000          JJ = II +L1 -1
000238      000          KK = II + IE + NND
000239      000      171 T(JJ) = X(KK)
000240      000          BADAMP = CON(9)
000241      000          CON(9) = CON(9) -.1
000242      000          IF (CON(9) .LT. .0001) GO TO 172
000243      000          WRITE (6,705) CON(9)
000244      000      705 FORMAT (//10X,76HNLOOP WAS EXCEEDED WITHOUT CONVERGENCE.  DAMPA WI
000245      000         *LL BE REDUCED TO A VALUE OF, F5.3,24H  AND THE LOOP REPEATED.,//)
000246      000          KON(28) = KON(28) +5
000247      000          GO TO 131
000248      000      172 CON(9) = CON(9) +.1
000249      000          WRITE (6,706) CON(9)
000250      000      706 FORMAT (//10X,20HDAMPA WAS REDUCED TO, F5.3,20H WITHOUT CONVERGENC
000251      000         *E)
000252      000          KON(28) = KON(28) +3
000253      000          GO TO 1000
000254      000      C   STORE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000255      000      175 CON(30) = TCGM
000256      000      C   COMPUTE THE ARITHMETIC TEMPERATURE CHANGE
000257      000          TCGM = 0.0
000258      000          DO 180 I = L1,NNC
000259      000          LE = IE+I
000260      000          T1 = ABS(T(I)-X(LE))
000261      000          IF(T1.LT.TCGM) GO TO 180
000262      000          TCGM = T1
000263      000          KON(38) = I
000264      000      180 CONTINUE
000265      000      C   SEE IF ATMPCA WAS SATISFIED
000266      000          IF(TCGM.GT.CON(11)) GO TO 125
000267      000          CON(16) = TCGM
000268      000      185 KON(12) = 0
000269      000          CALL VARBL2
000270      000      C   CHECK THE BACKUP SWITCH
000271      000          IF(KON(12).NE.0) GO TO 105
000272      000      C   ADVANCE TIME
000273      000          CON(13) = CON(1)
000274      000          TSUM = TSUM+TSTEP
000275      001          TSTEP = DELTA*0.75
000276      000      C   CHECK FOR TIME TO PRINT
000277      000          IF(TSUM.GE.CON(18)) GO TO 190
000278      000      C   CHECK FOR PRINT EVERY ITERATION
000279      000          IF(KON(7).EQ.0) GO TO 10
000280      000          CALL OUTCAL
000281      000          GO TO 10
000282      000      C   TRY TO EVEN THE OUTPUT INTERVALS
000283      000      190 TPRINT = TPRINT+TSUM
```

```
000284       000      195 CALL OUTCAL
000285       000    C     IS TIME GREATER THAN END COMPUTE TIME
000286       000          IF(CON(1)*1.000001.LT.CON(3)) GO TO 5
000287       000          NTH = IE
000288       000          NDIM = NLA
000289       000          RETURN
000290       000      995 WRITE(6,885)
000291       000          GO TO 1000
000292       000      996 WRITE(6,886)
000293       000          GO TO 1000
000294       000      997 WRITE(6,887)
000295       000          GO TO 1000
000296       000      998 WRITE(6,888) I
000297       000          GO TO 1000
000298       000      999 WRITE(6,889)
000299       000     1000 CALL OUTCAL
000300       000          CALL EXIT
000301       000      885 FORMAT(45H CNFRDL REQUIRES LONG PSEUDO-COMPUTE SEQUENCE)
000302       000      886 FORMAT(24H CSGMIN ZERO OR NEGATIVE)
000303       000      887 FORMAT(20H TIME STEP TOO SMALL)
000304       000      888 FORMAT(I8,20H LOCATIONS AVAILABLE)
000305       000      889 FORMAT(19H NO OUTPUT INTERVAL)
000306       000          END

END ELT.


*HOG,P SNFRWD
```

```
*ELT,L SNFRWD
ELT077 RL1870 02/28-03:20:52-(2,)
000001      000          SUBROUTINE SNFRWD
000002      000    C     EXPLICIT FORWARD DIFFERENCING EXECUTION SUBROUTINE FOR SINDA F-V
000003      000    C     THE SHORT PSEUDO-COMPUTE SEQUENCE IS REQUIRED
000004      002          COMMON /FDIMNS/ NTYP, NSYS
000005      000          INCLUDE COMM,LIST
000006      000          INCLUDE DEFF,LIST
000007      000          IF(CON(4).LT.1.0) CON(4) = 1.0
000008      000          IF (KON(5) .LE. 2) KON(5) =10
000009      000          IF(CON(6).LE.0.) CON(6) = 1.E+8
000010      000          IF(CON(8).LE.0.) CON(8) = 1.E+8
000011      000          IF(CON(9).LE.0.) CON(9) = 1.0
000012      000          IF(CON(11).LE.0.) CON(11) = 1.E+8
000013      000          IF(CON(18).LE.0.) GO TO 999
000014      000          IF (CON(19) .LE. 0.) CON(19) =.1
000015      000          IF(KON(31).NE.0) GO TO 995
000016      000           IF (CON(50) .LE. 0.) CON(50) =1.
000017      000          PASS = -1.0
000018      000          NNC = NND+NNA
000019      000          IE = NTH
000020      000          NLA = NDIM
000021      002          NTH = NTH+NNT
000022      002          NDIM = NDIM-NNT
000023      000    C     CHECK FOR EXTRA LOCATIONS FOR CALCULATED NODES
000024      000          I = NLA-NNC
000025      000          IF(I.LT.0) GO TO 998
000026      000          L1 = NND+1
000027      000          TSTEP = CON(18)
000028      000          TPRINT = CON(13)
000029      000    C     INITALIZE TIME SUM BETWEEN OUTPUT INTERVALS
000030      000        5 TSUM = 0.0
000031      000    C     DOES OLD TIME PLUS THE OUTPUT INTERVAL EXCEED THE STOP TIME
000032      000          IF(CON(13)+CON(18).LE.CON(3)) GO TO 10
000033      000    C     DONT EXCEED IT
000034      000          CON(18) = CON(3)-CON(13)
000035      000    C     IS THE TIME STEP LARGER THAN ALLOWED
000036      000       10 IF(TSTEP.LE.CON(8)) GO TO 15
000037      000          TSTEP = CON(8)
000038      000    C     DOES THE TIME SUM PLUS THE TIME STEP EXCEED OUTPUT INTERVAL
000039      000       15 IF(TSUM+TSTEP-CON(18)) 25,30,20
000040      000    C     DONT EXCEED IT
000041      000       20 TSTEP = CON(18)-TSUM
000042      000          GO TO 30
000043      000    C     DOES TIME SUM PLUS TWO TIME STEPS EXCEED OUTPUT INTERVAL
000044      000       25 IF(TSUM+2.0*TSTEP.LE.CON(18)) GO TO 30
000045      000    C     APPROACH THE OUTPUT INTERVAL GRADUALLY
000046      000          TSTEP = (CON(18)-TSUM)/2.0
000047      000    C     STORE DELTA TIME STEP IN THE CONSTANTS
000048      000       30 CON(2) = TSTEP
000049      000    C     IS THE TIME STEP USED LESS THAN THE TIME STEP ALLOWED
000050      000          IF(TSTEP.LT.CON(21)) GO TO 997
000051      000    C     CALCULATE THE NEW TIME
000052      000          CON(1) = TPRINT+TSUM+TSTEP
000053      000    C     COMPUTE THE MEAN TIME BETWEEN ITERATIONS
000054      000          CON(19) = (CON(1)+CON(13))/2.0
000055      000    C     ZERO OUT ALL SOURCE LOCATIONS AND EXTRA LOCATIONS
```

```
000056  000        DO 35 I = 1,NND
000057  000        LE = IE+I
000058  000        X(LE) = 0.0
000059  000        Q(I) = 0.0
000060  000     35 CONTINUE
000061  000  C     SHIFT THE ARITHMETIC TEMPERATURES INTO THE EXTRA LOCATIONS
000062  002        IF(NND.EQ.NNT) GO TO 45
000063  002        DO 31 I=L1,NNT
000064  002        X(IE+I) = T(I)
000065  002     31 CONTINUE
000066  000        IF(NNA.LE.0) GO TO 45
000067  000        DO 40 I = L1,NNC
000068  000        Q(I) = 0.0
000069  000     40 CONTINUE
000070  000     45 KON(12) = 0
000071  000        CALL VARBL1
000072  000        IF(KON(12).NE.0) GO TO 10
000073  000        J1 = 0
000074  000        J2 = 1
000075  000        TCGM = 0.0
000076  000        CKM = 1.E+8
000077  000  C     CALCULATE Q SUM AND G SUM
000078  000        DO 85 I = 1,NND
000079  000        LE = IE+I
000080  000        INCLUDE VARC,LIST
000081  000        INCLUDE VARD,LIST
000082  000     50 J1 = J1+1
000083  000        LG = FLD(5,16,NSO1(J1))
000084  000  C     CHECK FOR LAST CONDUCTOR
000085  000        IF(LG.EQ.0) GO TO 85
000086  000        LTA = FLD(22,14,NSO1(J1))
000087  000        INCLUDE VARG,LIST
000088  000  C     CHECK FOR RADIATION CONDUCTOR
000089  000        IF(FLD(3,1,NSO1(J1)).EQ.0) GO TO 55
000090  000        T1 = T(I)+460.0
000091  000        T2 = T(LTA)+460.0
000092  000        GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000093  000         GV = GV *CON(50)
000094  000        GO TO 60
000095  000     55 GV = G(LG)
000096  000  C     OBTAIN THE Q RATE THRU THE CONDUCTOR
000097  000     60 QDOT = GV*(T(LTA)-T(I))
000098  000        Q(I) = Q(I)+QDOT
000099  000  C     SAVE SUMMATION OF CONDUCTORS
000100  000        X(LE) = X(LE)+GV
000101  000  C     CHECK FOR ADJOINING DIFFUSION NODE
000102  000        IF(LTA.GT.NND.OR.FLD(21,1,NSO1(J1)).EQ.1) GO TO 65
000103  000  C     SAVE SUMMATION OF CONDUCTORS FOR ADJOINING NODE
000104  000        LEA = IE+LTA
000105  000        X(LEA) = X(LEA)+GV
000106  000        Q(LTA) = Q(LTA)-QDOT
000107  000  C     CHECK FOR LAST CONDUCTOR
000108  000     65 IF(NSO1(J1).GT.0) GO TO 50
000109  000        85 CONTINUE
000110  000  C     OBTAIN NEW DIFFUSION TEMPERATURES, DTMPCC AND CSGMIN
000111  000        DO 100 I = 1,NND
000112  000        LE = IE+I
```

```
000113      000   C     CALCULATE C/SK MINIMUM
000114      002         IF(.NOT. X(LE) .GT. 0) GO TO 90
000115      000         T1 = C(I)/X(LE)
000116      000         IF(T1.GE.CKM) GO TO 90
000117      000         CKM = T1
000118      000         KON(35) = I
000119      000   C     COMPUTE NEW TEMPERATURES USING CALCULATED SOURCE TERMS
000120      000      90 T1 = TSTEP*Q(I)/C(I)
000121      000   C     CALCULATE THE ABSOLUTE VALUE TEMPERATURE CHANGE
000122      000         T2 = ABS(T1)
000123      000   C     SAVE THE LARGEST TEMPERATURE CHANGE
000124      000         IF(TCGM.GE.T2) GO TO 95
000125      000         TCGM = T2
000126      000         KON(36) = I
000127      000   C     STORE THE TEMPERATURES
000128      000      95 X(LE) = T(I)
000129      000         T(I) = T(I)+T1
000130      000     100 CONTINUE
000131      000         CON(17) = CKM
000132      000         DELTA = CKM/CON(4)
000133      002         KOP = 0
000134      002         IF(CON(7).NE.0.0) KOP = 1
000135      002         IF(NSYS .NE. 0) CALL FLUID(1,0,0,0.,KOP)
000136      000   C     CHECK FOR FIRST PASS
000137      000         IF(PASS.GT.0.0) GO TO 115
000138      000   C     UNDO THE TEMPERATURE CALCULATIONS
000139      002     105 DO 110 I = 1,NNT
000140      000         LE = IE+I
000141      000         T(I) = X(LE)
000142      000     110 CONTINUE
000143      000         IF(PASS.GT.0.0) GO TO 15
000144      000         PASS = 1.0
000145      000         CON(1) = TPRINT
000146      000         CON(2) = 0.0
000147      001         TSTEP = DELTA*0.75
000148      000         GO TO 195
000149      000   C     IS THE TIME STEP USED LESS THAN THE TIME STEP CALCULATED
000150      000     115 IF(TSTEP.LE.DELTA) GO TO 130
000151      000   C     COMPUTE THE TIME STEP
000152      001         TSTEP = DELTA*0.75
000153      000         GO TO 105
000154      001     120 TSTEP = 0.75*TSTEP*CON(6)/TCGM
000155      000         GO TO 105
000156      001     125 TSTEP = 0.75*TSTEP*CON(11)/TCGM
000157      000         GO TO 105
000158      000   C     SEE IF THE TEMPERATURE CHANGE WAS TOO LARGE
000159      000     130 IF(TCGM.GT.CON(6)) GO TO 120
000160      000   C     STORE THE MAXIMUM DIFFUSION TEMPERATURE CHANGE
000161      000         CON(15) = TCGM
000162      000   C     CHECK TO SEE IF THERE ARE ANY ARITHMETIC NODES
000163      000         IF(NNA.LE.0) GO TO 185
000164      000   C     COMPUTE ARITHMETIC TEMPERATURES BY SUCCESSIVE POINT OVER-RELAX
000165      000         BADAMP = 1.
000166      000         LAX = KON(5)
000167      000   C     STORE Q'S CALCULATED DURING DIFFUSION NODE LOOP
000168      000         DO 133 II=1,NNC
000169      000     133 X(II+NNA) = Q(II)
```

```
000170  000     131 CONTINUE
000171  000         DN = CON(9)
000172  000         DD = 1.0-DN
000173  000         DO 170 I = 1,LAX
000174  000         JJ1 = J1
000175  000         JJ2 = J2
000176  000         TCGM = 0.0
000177  000         KON(20) = I
000178  000         DO 165 L = L1,NNC
000179  000         SUMC = 0.0
000180  000         SUMCV = 0.0
000181  000         IF(I.GT.1) GO TO 6000
000182  000         INCLUDE VRG2,LIST
000183  000     135 JJ1 = JJ1+1
000184  000         LG = FLD(5,16,NSO1(JJ1))
000185  000         LTA = FLD(22,14,NSO1(JJ1))
000186  000         IF(I.GT.1) GO TO 4000
000187  000         INCLUDE VRG2,LIST
000188  000   C     CHECK FOR RADIATION CONDUCTOR
000189  000         IF(FLD(3,1,NSO1(JJ1)).EQ.0) GO TO 140
000190  000         T1 = T(L)+460.0
000191  000         T2 = T(LTA)+460.0
000192  000         GV = G(LG)*(T1*T1+T2*T2)*(T1+T2)
000193  000         GV =GV*CON(50)
000194  000         GO TO 145
000195  000     140 GV = G(LG)
000196  000     145 SUMC = SUMC+GV
000197  000         SUMCV = SUMCV+GV*T(LTA)
000198  000   C     CHECK FOR LAST CONDUCTOR
000199  000         IF(NSO1(JJ1).GT.0) GO TO 135
000200  000         T2 = DD*T(L)+DN*(SUMCV+Q(L))/SUMC
000201  000   C     OBTAIN THE CALCULATED TEMPERATURE DIFFERENCE
000202  000         T1 = ABS(T(L)-T2)
000203  000   C     STORE THE NEW TEMPERATURE
000204  000         T(L) = T2
000205  000   C     SAVE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000206  000         IF(TCGM.GE.T1) GO TO 165
000207  000         TCGM = T1
000208  000         KON(37) = L
000209  000     165 CONTINUE
000210  000   C     SEE IF RELAXATION CRITERIA WAS MET
000211  000         OLD3 = OLD2
000212  000         OLD2 = OLD1
000213  000         OLD1 = TCGM
000214  000         LTL3 = LTL2
000215  000         LTL2 = LTL1
000216  000         LTL1 = L
000217  000         TEMP3 = TEMP2
000218  000         TEMP2 = TEMP1
000219  000         TEMP1 = T(L)
000220  000         IF(TCGM.LE.CON(19)) GO TO 175
000221  000     170 CONTINUE
000222  000         DELTAA= OLD2 - OLD1
000223  000         DELTAB= OLD3 - OLD2
000224  000         DO 173 II=1,NNC
000225  000     173 Q(II) = X(II+NNA)
000226  000         IF (DELTAA .LE. 0.) GO TO 174
```

```
000227  000          IF (DELTAB .LT. DELTAA) GO TO 174
000228  000          IF (DELTAB .LE. 0.) GO TO 174
000229  000          IF (LTL1 .NE. LTL2) GO TO 174
000230  000          IF (LTL2 .NE. LTL3) GO TO 174
000231  000          IF (TEMP1 .GT. TEMP2) GO TO 176
000232  000          IF (TEMP2 .GT. TEMP3) GO TO 174
000233  000          GO TO 177
000234  000      176 IF (TEMP2 .LT. TEMP3) GO TO 174
000235  000      177 CONTINUE
000236  000          KON(5) = KON(5) + LAX
000237  000          IF ((CON(9) +.1) .LT. BADAMP) CON(9) = CON(9) +.1
000238  000          IF (CON(9) .GT. 1.) CON(9) =1.
000239  000          WRITE (6,704) LAX,KON(5),DN,CON(9)
000240  000      704 FORMAT (//10X,40HTHE SOLUTION WAS CONVERGING WHEN NLOOP =,I5,43H W
000241  000         *AS EXCEEDED.  NLOOP WILL BE INCREASED TO,I5,1H,/10X, 20HDAMPA INC
000242  000         *REASED FROM,F5.3,3H TO,F5.3,23H AND THE LOOP CONTINUED,//)
000243  000          KON(28) = KON(28) +6
000244  000          GO TO 131
000245  000    C     REDUCE DAMPA AND TRY AGAIN
000246  000      174 DO 171 II=1,NNA
000247  000          JJ = II +L1 -1
000248  000          KK = II + IE + NND
000249  000      171 T(JJ) = X(KK)
000250  000          BADAMP = CON(9)
000251  000          CON(9) = CON(9) -.1
000252  000          IF (CON(9) .LT. .0001) GO TO 172
000253  000          WRITE (6,705) CON(9)
000254  000      705 FORMAT (//10X,76HNLOOP WAS EXCEEDED WITHOUT CONVERGENCE.  DAMPA WI
000255  000         *LL BE REDUCED TO A VALUE OF, F5.3,24H  AND THE LOOP REPEATED.,//)
000256  000          KON(28) = KON(28) +5
000257  000          GO TO 131
000258  000      172 CON(9) = CON(9) +.1
000259  000          WRITE (6,706) CON(9)
000260  000      706 FORMAT (//10X,20HDAMPA WAS REDUCED TO, F5.3,20H WITHOUT CONVERGENC
000261  000         *E)
000262  000          KON(28) = KON(28) +3
000263  000          GO TO 1000
000264  000    C     STORE THE MAXIMUM ARITHMETIC RELAXATION CHANGE
000265  000      175 CON(30) = TCGM
000266  000    C     COMPUTE THE ARITHMETIC TEMPERATURE CHANGE
000267  000          TCGM = 0.0
000268  000          DO 180 I = L1,NNC
000269  000          LE = IE+I
000270  000          T1 = ABS(T(I)-X(LE))
000271  000          IF(T1.LT.TCGM) GO TO 180
000272  000          TCGM = T1
000273  000          KON(38) = I
000274  000      180 CONTINUE
000275  000    C     SEE IF ATMPCA WAS SATISFIED
000276  000          IF(TCGM.GT.CON(11)) GO TO 125
000277  000          CON(16) = TCGM
000278  000      185 KON(12) = 0
000279  000          CALL VARBL2
000280  000    C     CHECK THE BACKUP SWITCH
000281  000          IF(KON(12).NE.0) GO TO 105
000282  000    C     ADVANCE TIME
000283  000          CON(13) = CON(1)
```

```
000284  000          TSUM = TSUM+TSTEP
000285  001          TSTEP = DELTA*0.75
000286  000    C     CHECK FOR TIME TO PRINT
000287  000          IF(TSUM.GE.CON(18)) GO TO 190
000288  000    C     CHECK FOR PRINT EVERY ITERATION
000289  000          IF(KON(7).EQ.0) GO TO 10
000290  000          CALL OUTCAL
000291  000          GO TO 10
000292  000    C     TRY TO EVEN THE OUTPUT INTERVALS
000293  000      190 TPRINT = TPRINT+TSUM
000294  000      195 CALL OUTCAL
000295  000    C     IS TIME GREATER THAN END COMPUTE TIME
000296  000          IF(CON(1)*1.000001.LT.CON(3)) GO TO 5
000297  000          NTH = IE
000298  000          NDIM = NLA
000299  000          RETURN
000300  000      995 WRITE(6,885)
000301  000          GO TO 1000
000302  000      996 WRITE(6,886)
000303  000          GO TO 1000
000304  000      997 WRITE(6,887)
000305  000          GO TO 1000
000306  000      998 WRITE(6,888) I
000307  000          GO TO 1000
000308  000     .999 WRITE(6,889)
000309  000     1000 CALL OUTCAL
000310  000          CALL EXIT
000311  000      885 FORMAT(46H CNFRWD REQUIRES SHORT PSEUDO-COMPUTE SEQUENCE)
000312  000      886 FORMAT(24H CSGMIN ZERO OR NEGATIVE)
000313  000      887 FORMAT(20H TIME STEP TOO SMALL)
000314  000      888 FORMAT(I8,20H LOCATIONS AVAILABLE)
000315  000      889 FORMAT(19H NO OUTPUT INTERVAL)
000316  000          END

END ELT.


*HDG,M,SPLIT3PLIT/P
```

```
@ELT,L SPLIT/P
ELTOTT RL1870 02/28-03:20:55-(0,)
000001      000          SUBROUTINE SPLIT (ID)                                    SPL   1
000002      000          COMMON /BUCKET/ B(1)                                     SPL   2
000003      000          COMMON /TAPE/ NIN,NOUT,INTERN,LB3D,LB4P,LUT1,LUT2,LUT3,LUT4,LUT7  SPL   3
000004      000          COMMON /DATA/ NND,NNA,NNB,NNT,NGL,NGR,NGT,NUC,NEC1,NEC2,NCT,LENA,  SPL   4
000005      000         1   ERDATA,PROGRM,ENDRUN,LSEQ1,LSEQ2                      SPL   5
000006      000          DATA RECALL /6HRECALL/                                   SPL   6
000007      000    C     ..INITIALIZATION                                         SPL   7
000008      000          REWIND LUT1                                              SPL   8
000009      000          REWIND LUT3                                              SPL   9
000010      000          REWIND LB3D                                              SPL  10
000011      000    C     ..PROBLEM IDENTIFICATION                                 SPL  11
000012      000       10 CONTINUE                                                 SPL  12
000013      000          READ (LUT7) IDENT                                        SPL  13
000014      000          IF (ID.EQ.IDENT) GO TO 20                                SPL  14
000015      000          CALL SKIP                                                SPL  15
000016      000          GO TO 10                                                 SPL  16
000017      000    C     ..TITLE                                                  SPL  17
000018      000       20 CONTINUE                                                 SPL  18
000019      000          WRITE (LB3D) RECALL                                      SPL  19
000020      000          READ (LUT7) (B(I),I=1,20)                                SPL  20
000021      000          WRITE (LUT3) (B(I),I=1,20)                               SPL  21
000022      000          WRITE (LB3D) (B(I),I=1,20)                               SPL  22
000023      000    C     ..PROBLEM DIMENSIONS                                     SPL  23
000024      000          READ (LUT7) NND,NNA,NNT,NGT,NCT,NAT,LSEQ1,LSEQ2,LENA     SPL  24
000025      000    C     ..CHECK FOR GENERAL PROBLEM                              SPL  25
000026      000          IF ((NNT.EQ.0).AND.(NGT.EQ.0)) GO TO 40                  SPL  26
000027      000    C     ..NODE INFO                                              SPL  27
000028      000          READ (LUT7) (B(I),I=1,NNT)                               SPL  28
000029      000          WRITE (LUT1) NNT,(B(I),I=1,NNT)                          SPL  29
000030      000          READ (LUT7) (B(I),I=1,NNT)                               SPL  30
000031      000          WRITE (LUT3) NND,(B(I),I=1,NNT)                          SPL  31
000032      000          WRITE (LB3D) NND,NNA,NNT,(B(I),I=1,NNT)                  SPL  32
000033      000          IF (NND.EQ.0) GO TO 30                                   SPL  33
000034      000          READ (LUT7) (B(I),I=1,NND)                               SPL  34
000035      000          WRITE (LUT3) (B(I),I=1,NND)                              SPL  35
000036      000          WRITE (LB3D) (B(I),I=1,NND)                              SPL  36
000037      000    C     ..CONDUCTOR INFO                                         SPL  37
000038      000       30 CONTINUE                                                 SPL  38
000039      000          READ (LUT7) (B(I),I=1,NGT)                               SPL  39
000040      000          WRITE (LUT1) NGT,(B(I),I=1,NGT)                          SPL  40
000041      000          READ (LUT7) (B(I),I=1,NGT)                               SPL  41
000042      000          WRITE (LUT3) (B(I),I=1,NGT)                              SPL  42
000043      000          WRITE (LB3D) NGT,(B(I),I=1,NGT)                          SPL  43
000044      000    C     ..CONSTANTS INFO                                         SPL  44
000045      000       40 CONTINUE                                                 SPL  45
000046      000          READ (LUT7) (B(I),I=1,50)                                SPL  46
000047      000          NUC=0                                                    SPL  47
000048      000          IF (NCT.EQ.0) GO TO 50                                   SPL  48
000049      000          IST=51                                                   SPL  49
000050      000          IEND=IST+NCT-1                                           SPL  50
000051      000          READ (LUT7) NUC,(B(I),I=IST,IEND)                        SPL  51
000052      000       50 CONTINUE                                                 SPL  52
000053      000          WRITE (LUT3) NUC,NCT,(B(I),I=1,50)                       SPL  53
000054      000          WRITE (LB3D) NCT,(B(I),I=1,50)                           SPL  54
000055      000          IF (NCT.EQ.0) GO TO 60                                   SPL  55
```

```
000056  000        WRITE (LUT1) NUC,NCT,(B(I),I=IST,IEND)                SPL  56
000057  000        READ (LUT7) (B(I),I=1,NCT)                            SPL  57
000058  000        WRITE (LUT3) (B(I),I=1,NCT)                           SPL  58
000059  000        WRITE (LB3D) (B(I),I=1,NCT)                           SPL  59
000060  000  C     ..ARRAY INFO                                          SPL  60
000061  000     60 CONTINUE                                              SPL  61
000062  000        IF (NAT.EQ.0) GO TO 70                                SPL  62
000063  000        READ (LUT7) (B(I),I=1,NAT)                            SPL  63
000064  000        WRITE (LUT1) NAT,(B(I),I=1,NAT)                       SPL  64
000065  000        READ (LUT7) (B(I),I=1,NAT)                            SPL  65
000066  000        WRITE (LUT1) NAT,(B(I),I=1,NAT)                       SPL  66
000067  000        READ (LUT7) (B(I),I=1,LENA)                           SPL  67
000068  000     70 CONTINUE                                              SPL  68
000069  000        WRITE (LUT3) NAT,LENA                                 SPL  69
000070  000        WRITE (LB3D) NAT,LENA                                 SPL  70
000071  000        IF (NAT.EQ.0) GO TO 80                                SPL  71
000072  000        WRITE (LUT3) (B(I),I=1,LENA)                          SPL  72
000073  000        WRITE (LB3D) (B(I),I=1,LENA)                          SPL  73
000074  000  C     ..PSEUDO COMPUTE SEQUENCES                            SPL  74
000075  000     80 CONTINUE                                              SPL  75
000076  000        IF ((NNT.EQ.0).AND.(NGT.EQ.0)) GO TO 90               SPL  76
000077  000        READ(LUT7) NTYP, NSYS, NTB, NP, NV, NFD
000078  000        IF(NSYS .LT. 1) GO TO 88
000079  000        READ(LUT7)  (B(I),I=1,NTB)
000080  000        WRITE(LUT1) (B(I),I=1,NTB)
000081  000        READ(LUT7)  (B(I),I=1,NP )
000082  000        WRITE(LUT1) (B(I),I=1,NP )
000083  000        IF(NV .EQ. 0) GO TO 82
000084  000        READ(LUT7)  (B(I),I=1,NV )
000085  000        WRITE(LUT1) (B(I),I=1,NV )
000086  000     82 READ(LUT7)  (B(I),I=1,NFD)
000087  000        WRITE(LB3D) (B(I),I=1,NFD)
000088  000        NSP = 15*NSYS
000089  000        READ(LUT7)  (B(I),I=1,NSP)
000090  000        WRITE(LB3D) (B(I),I=1,NSP)
000091  000        NSP = 10*NTYP
000092  000        READ(LUT7)  (B(I),I=1,NSP)
000093  000        WRITE(LB3D) (B(I),I=1,NSP)
000094  000        READ(LUT7)  (B(I),I=1,NTB)
000095  000        WRITE(LB3D) (B(I),I=1,NTB)
000096  000        IF(NV .EQ. 0) GO TO 84
000097  000        READ(LUT7)  (B(I),I=1,NV)
000098  000        WRITE(LB3D) (B(I),I=1,NV)
000099  000     84 READ(LUT7)  (B(I),I=1,NP)
000100  000        WRITE(LB3D) (B(I),I=1,NP)
000101  000        READ(LUT7)  (B(I),I=1,NP)
000102  000        WRITE(LB3D) (B(I),I=1,NP)
000103  000     88 CONTINUE
000104  000        READ (LUT7) (B(I),I=1,LSEQ1)                          SPL  77
000105  000        WRITE (LB3D) LSEQ1,LSEQ2,(B(I),I=1,LSEQ1)             SPL  78
000106  000        IF (LSEQ2.EQ.0) GO TO 90                              SPL  79
000107  000        READ (LUT7) (B(I),I=1,LSEQ2)                          SPL  80
000108  000        WRITE (LB3D) (B(I),I=1,LSEQ2)                         SPL  81
000109  000  C                                                           SPL  82
000110  000     90 RETURN                                                SPL  83
000111  000        END                                                   SPL  84-
```

```
SNFRWD
END ELT.

#HOG,P SRTLST
```

```
SRTLST                                                          DATE 022875        PAGE     1

*ELT,L SRTLST
ELTOT7 RL1870 02/28-03:20:59-(3,)
000001       000          SUBROUTINE SRTLST(IC,LIST)
000002       000    C
000003       000    C
000004       000          DIMENSION LIST(1)
000005       000    C
000006       000    C
000007       000          JC = IC - 1
000008       000          DO 200 I=1,JC
000009       000          ASSIGN 300 TO J
000010       002          KC = IC-I
000011       001          DO 100 K=1,KC
000012       000          IF(LIST(K+1)-LIST(K)) 50,100,100
000013       000       50 KEEP = LIST(K)
000014       000          LIST(K) = LIST(K+1)
000015       000          LIST(K+1) = KEEP
000016       001          ASSIGN 200 TO J
000017       000      100 CONTINUE
000018       001          GO TO J
000019       000      200 CONTINUE
000020       000      300 RETURN
000021       000          END

END ELT.


*HDG,P SYDSTL
```

```
@ELT,L STDSTL
ELTOT7 RL1870 02/28-03:21:01-(1,)
000001      000          SUBROUTINE STDSTL                                          STDSTL
000002      000    C                                                                STDSTL
000003      000    C*****************************************************************STDSTL
000004      000    C                                                                STDSTL
000005      000    C         THIS EXECUTION SUBROUTINE CALCULATES THE STEADY STATE  STDSTL
000006      000    C      SOLUTION OF A NETWORK.  THE LONG PSEUDO COMPUTE SEQUENCE  STDSTL
000007      000    C      OPTION IS REQUIRED.  ALL NODES RECEIVE A -SUCCESSIVE POINT-STDSTL
000008      000    C      ITERATION.  THE CONVERGENCE IS EITHER A MAXIMUM TEMPERATURE STDSTL
000009      000    C      CHANGE OR AN ENERGY BALANCE OR BOTH.                       STDSTL
000010      000    C                                                                STDSTL
000011      000    C*********************                                         STDSTL
000012      000    C                                                                STDSTL
000013      000    C      CONTROL CONSTANT 1 CONTAINS THE NEW PROBLEM TIME          (TIMEN)
000014      000    C      CONTROL CONSTANT 2 CONTAINS THE TIME STEP USED            (DTIMEU)
000015      000    C      CONTROL CONSTANT 3 CONTAINS THE PROBLEM STOP TIME         (TIMEND)
000016      000    C      CONTROL CONSTANT 4 CONTAINS THE TIME STEP FACTOR,EXPLICIT (CSGFAC)
000017      000    C      CC5 IS THE INPUT NUMBER OF ITERATION DO LOOPS, INTEGER    (NLOOP)
000018      000    C      CC6 CONTAINS THE DIFFUSION TEMPERATURE CHANGE ALLOWED     (DTMPCA)
000019      000    C      CC7 CONTAINS THE OUTPUT EACH ITERATION SWITCH             (OPEITR)
000020      000    C      CC8 CONTAINS THE MAXIMUM ALLOWED TIME STEP                (DTIMEH)
000021      000    C      CC9 CONTAINS THE NEW ARITHMETIC TEMP. DAMPING FACTOR      (DAMPA)
000022      000    C      CC10 CONTAINS THE NEW DIFFUSION TEMP. DAMPING FACTOR      (DAMPD)
000023      000    C      CC11 CONTAINS THE MAXIMUM ALLOWED ARITHMETIC TEMP. CHANGE (ATMPCA)
000024      000    C      CC12 CONTAINS THE BACKUP SWITCH CHECKED AFTER VARIABLES   (BACKUP)
000025      000    C      CC13 CONTAINS THE PRESENT TIME OR PROBLEM START TIME       (TIMEO)
000026      000    C      CC14 CONTAINS THE MEAN TIME BETWEEN AN ITERATION           (TIMEM)
000027      000    C      CC15 CONTAINS THE DIFFUSION TEMPERATURE CHANGE CALCULATED (DTMPCC)
000028      000    C      CC16 CONTAINS ARITHMETIC TEMPERATURE CHANGE CALCULATED    (ATMPCC)
000029      000    C      CONTROL CONSTANT 17 IS RESERVED FOR THE C/SG MINIMUM      (CSGMIN)
000030      000    C      CONTROL CONSTANT 18 CONTAINS THE OUTPUT INTERVAL          (OUTPUT)
000031      000    C      CC19 CONTAINS THE ARITHMETIC RELAXATION CRITERIA ALLOWED  (ARLXCA)
000032      000    C      CC20 CONTAINS THE NUMBER OF RELAXATION LOOPS USED,INTEGER (LOOPCT)
000033      000    C      CC21 CONTAINS THE MINIMUM ALLOWED TIME STEP                (DTIMEL)
000034      000    C      CC22 IS FOR THE INPUT TIME STEP IMPLICIT                   (DTIMEI)
000035      000    C      CC23 CONTAINS THE C/SG MAXIMUM                             (CSGMAX)
000036      000    C      CC24 CONTAINS THE C/SG RANGE ALLOWED                       (CSGRAL)
000037      000    C      CC25 CONTAINS THE C/SG RANGE CALCULATED                    (CSGRCL)
000038      000    C      CC26 CONTAINS THE DIFFUSION RELAXATION CRITERIA ALLOWED    (DRLXCA)
000039      000    C      CC27 CONTAINS THE DIFFUSION RELAXATION CHANGE CALCULATED   (DRLXCC)
000040      000    C      CC28 CONTAINS THE LINE COUNTER, INTEGER                    (LINECT)
000041      000    C      CC29 CONTAINS THE PAGE COUNTER, INTEGER                    (PAGECT)
000042      000    C      CC30 CONTAINS ARITHMETIC RELAXATION CHANGE CALCULATED      (ARLXCC)
000043      000    C      CC31 IS INDICATOR, 0=THERMAL SPCS,1=THERMAL LPCS,2=GENERAL (LSPCS)
000044      000    C      CC32 CONTAINS THE ENERGY BALANCE OF THE SYSTEM, IN - OUT   (ENGBAL)
000045      000    C      CC33 CONTAINS THE DESIRED ENERGY BALANCE, USER INPUT       (BALENG)
000046      000    C      CC34 CONTAINS THE NOCOPY SWITCH FOR MATRIX USERS           (NOCOPY)
000047      000    C      CC35 CONTAINS RELATIVE NODE NUMBER OF CSGMIN
000048      000    C      CC36 CONTAINS RELATIVE NODE NUMBER OF DTMPCC
000049      000    C      CC37 CONTAINS RELATIVE NODE NUMBER OF ARLXCC
000050      000    C      CC38 CONTAINS RELATIVE NODE NUMBER OF ATMPCC
000051      000    C      CC39-40-41-42-43 CONTAIN DUMMY INTEGER CONSTANTS   (I-J-K-L-MTEST)
000052      000    C      CC44-45-46-47-48 CONTAIN DUMMY FLOATING CONSTANTS  (R-S-T-U-VTEST)
000053      000    C      CC49 IS THE QUASI-LINEARIZATION INTERVAL FOR CINDSM        (LAXFAC)
000054      000    C      CC50 IS NOT USED AT PRESENT
000055      000    C*****************************************************************STDSTL
```

```
000056  000  C                                                                STDSTL
000057  001        LOGICAL FLOW
000058  001        COMMON /FDIMNS/ NTYP, NSYS
000059  001        COMMON /POINTN/ LNODE
000060  000        COMMON /TITLE/R(1) /TEMP/T(1) /CAP/C(1) /SOURCE/Q(1) /COND/G(1)
000061  000        COMMON /PC1/NSQ1(1) /PC2/NSQ2(1) /RONST/K(1) /ARRAY/A(1)
000062  000        COMMON /DIMENS/ NND,NNA,NNT,NGT,NCT,NAT,LSQ1,LSQ2
000063  000        DIMENSION  KON(50)
000064  000        DIMENSION CON(1),XK(1),NX(1)
000065  000        EQUIVALENCE (KON(1),CON(1)),(K(1),XK(1)),(X(1),NX(1))
000066  000         COMMON /CONAP /    ITRAN   , ITNTH   , IDIM    , ITH          STDSTL
000067  000         COMMON /XSPACE/    NDIM    , NTH     , X(1)                   STDSTL
000068  000         COMMON /FIXCON/ TIMEN , DTIMEU , TIMEND , CSGFAC , NLOOP  ,  STDSTL
000069  000       1     DTMPCA  , OPEITR  , DTIMEN  , DAMPA   , DAMPB   , ATMPCA  ,  STDSTL
000070  000       2     NBACK   , TIMEO   , TIMEM   , DTMPCC  , ATMPCC  , CSGMIN  ,  STDSTL
000071  000       3     OUTPUT  , ARLXCA  , LOOPCT  , DTIMEL  , DTIMEI  , EXTLIM  ,  STDSTL
000072  000       4     PERIOD  , NPRD    , DRLXCA  , DRLXCC  , NLINE   , NPAGE   ,  STDSTL
000073  000       5     ARLXCC  , LSPCS   , ENGBAL  , BALENG  , NOCOPY  , NCSGM   ,  STDSTL
000074  000       6     NDTMPC  , NARLXC  , NATMPC  , NDRLXC  , DUMMY(11)
000075  000  C                                                                STDSTL
000076  000  C                                                                STDSTL
000077  000        DOUBLE PRECISION R16,R3,RSR2,CS,FF,GG,ZZ
000078  000        CON(14) = TIMEO
000079  000        NOUT =6
000080  000        ICNT=0
000081  000  C++++++++++                                                       STDSTL
000082  000  C                                                                STDSTL
000083  000  C      INITIALIZING CONTROL CONSTANTS.                           STDSTL
000084  000  C                                                                STDSTL
000085  001        TZERO = -460.
000086  000         IF(NLOOP.LE.0)                  GO TO 200                  STDSTL
000087  000         IF(NNA.LE.0)                    GO TO 100                  STDSTL
000088  000         IF(ARLXCA.GT.0.0)               GO TO 100                  STDSTL
000089  000         IF(BALENG.LE.0.0)               GO TO 300                  STDSTL
000090  000         IF(NND.GT.0.0.AND.DRLXCA.LE.0.0) GO TO 350                 STDSTL
000091  000         GO TO 150                                                  STDSTL
000092  000    100  IF(NND.LE.0.0)                  GO TO 150                  STDSTL
000093  000         IF(DRLXCA.GT.0.0)               GO TO 150                  STDSTL
000094  000         IF(BALENG.LE.0.0)               GO TO 400                  STDSTL
000095  000         IF(NNA.GT.0.0.AND.ARLXCA.LE.0.0) GO TO 450                 STDSTL
000096  000    150  IF(LSPCS.NE.1)                  GO TO 500
000097  000        CON(50) = DUMMY(11)
000098  000        IF (CON(50) .LE. 0.) CON(50) =1.
000099  000        SIGMA=CON(50)
000100  000         GO TO 700                                                 STDSTL
000101  000    200  WRITE (NOUT,201)                                          STDSTL
000102  000    201 FORMAT(1H1,32H ERROR - NLOOP MUST BE SPECIFIED)
000103  000         GO TO 600                                                 STDSTL
000104  000    300  WRITE (NOUT,301)                                          STDSTL
000105  000    301 FORMAT(1H1,50H ERROR - EITHER ARLXCA OR BALENG MUST BE SPECIFIED)
000106  000         GO TO 600                                                 STDSTL
000107  000    350  WRITE(NOUT,351)                                           STDSTL
000108  000    351 FORMAT(1H1,67H ERROR - ARLXCA MUST ALSO BE SPECIFIED, SINCE DRLXCA
000109  000       1AND BALENG WERE)
000110  000         GO TO 600                                                 STDSTL
000111  000    400  WRITE (NOUT,401)                                          STDSTL
000112  000    401 FORMAT(1H1,50H ERROR - EITHER DRLXCA OR BALENG MUST BE SPECIFIED)
```

```
000113  000          GO TO 600                                                STDSTL
000114  000     450  WRITE(NOUT,451)                                          STDSTL
000115  000     451 FORMAT(1H1,67H ERROR - DRLXCA MUST ALSO BE SPECIFIED, SINCE ARLXCA
000116  000        1AND BALENG WERE)
000117  000          GO TO 600                                                STDSTL
000118  000     500  WRITE (NOUT,501)                                         STDSTL
000119  000     501 FORMAT(1H1,33H ERROR - STDSTL REQUIRES THE LPCS)
000120  000     600  CALL OUTCAL                                              STDSTL
000121  000          GO TO 5300
000122  000     700  CALL TOPLIN                                              STDSTL
000123  000          ITRAN       = 0                                          STDSTL
000124  000          ITNTH       = NTH                                        STDSTL
000125  000          NN          = NND+1                                      STDSTL
000126  000          NNC         = NNA+NND                                    STDSTL
000127  000         R16 = 1.000/16.000
000128  000         R3  = 1.000/3.000
000129  000         RSR2= 1.000/SQRT(2.000)
000130  001         FLOW = .FALSE.
000131  001         NSP = 0
000132  001         IXF = NTH
000133  001         IF(NSYS.LT.1) GO TO 750
000134  001         FLOW = .TRUE.
000135  001         NSP = NNT
000136  001         DO 725 I = 1,NNT
000137  001         NX(IXF+I) = 0
000138  001     725 CONTINUE
000139  000     C                                                            STDSTL
000140  000     C++++++++++                                                  STDSTL
000141  000     C                                                            STDSTL
000142  000     C       SETTING UP THE DYNAMIC STORAGE (2*NNC)               STDSTL
000143  000     C                                                            STDSTL
000144  001     750 IX1 = IXF+NSP
000145  000          IX2         = IX1+NNC                                    STDSTL
000146  000          IXL         = NDIM                                       STDSTL
000147  000          J           = IX2+NNC                                    STDSTL
000148  000          WRITE (NOUT,801)   J                                     STDSTL
000149  000     801 FORMAT(1H1,23H NOTE - STDSTL REQUIRES,I6,25H WORDS OF DYNAMIC STOR
000150  000        1AGE)
000151  000          NLINE       = NLINE+4                                    STDSTL
000152  000     840  NTH         = NTH+J                                      STDSTL
000153  000          NDIM        = NDIM-J                                     STDSTL
000154  000          IF(NDIM.GE.0)                  GO TO 1000                STDSTL
000155  000          NDIM        = IABS(NDIM)                                 STDSTL
000156  000          WRITE (NOUT,901)   NDIM                                  STDSTL
000157  000     901 FORMAT(1H1,24H ERROR - STDSTL REQUIRES,I5,30H MORE WORDS OF DYNAMI
000158  000        1C STORAGE)
000159  000          GO TO 600                                                STDSTL
000160  000     C                                                            STDSTL
000161  000     C++++++++++                                                  STDSTL
000162  000     C         SET UP INITIAL VALUES                              STDSTL
000163  000     C                                                            STDSTL
000164  000     C                                                            STDSTL
000165  000    1000  IDIM        = NDIM                                       STDSTL
000166  000          ITH         = NTH                                        STDSTL
000167  000         TIMEN = TIMEM + TIMEO
000168  000         WRITE (6,705)  ARLXCA,DRLXCA,BALENG,DAMPD,DAMPA,NLOOP
000169  000     705 FORMAT (///22H  CONTROL CONSTANTS  ,5E12.5,I10)
```

```
000170  000  C                                                                  STDSTL
000171  000  C**********                                                        STDSTL
000172  000  C                                                                  STDSTL
000173  000  C       INITIALIZE ITERATION CONSTANTS AND SOURCE LOCATIONS, CALL VARBLS STDSTL
000174  000  C          AND ON 1ST PASS CALL OUTCAL.                            STDSTL
000175  000  C                                                                  STDSTL
000176  000   1100  CONTINUE                                                    STDSTL
000177  000         NQ          = 0                                            STDSTL
000178  000         IF(ARLXCA.LE.0.0.AND.                                      STDSTL
000179  000        .   ORLXCA.LE.0.0)                NQ = 1                    STDSTL
000180  000         ASSIGN 1350 TO IPASS                                       STDSTL
000181  000         ASSIGN 4900 TO ITER                                        STDSTL
000182  000        ASSIGN 2410 TO JPASS
000183  000        ASSIGN 2410 TO NPASS
000184  000         DO 4800 LOOPCT = 1,NLOOP                                   STDSTL
000185  000  C                                                                  STDSTL
000186  000  C       ZERO OUT ALL SOURCE TERMS                                  STDSTL
000187  000  C                                                                  STDSTL
000188  000         DO 1300   I = 1,NNC                                        STDSTL
000189  000   1300  Q(I)         = 0.0                                         STDSTL
000190  000         CALL VARBLS                                                STDSTL
000191  000         GO TO IPASS                                                STDSTL
000192  000   1350  ASSIGN 1400 TO IPASS                                       STDSTL
000193  000         CALL OUTCAL                                                STDSTL
000194  000   1400  DO 1410   I = 1,NNT                                        STDSTL
000195  000   1410  T(I)         = T(I)+460.0                                  STDSTL
000196  000         IF(NND.LE.0)                      GO TO 3100               STDSTL
000197  000  C                                                                  STDSTL
000198  000  C       PERFORM A SUCCESSIVE POINT ITERATION ON THE DIFFUSION NODES. STDSTL
000199  000  C                                                                  STDSTL
000200  001         ARLXCC = 0.0
000201  001         KOP = 0
000202  001         IF(OPEITR.NE.0.0) KOP  = 1
000203  001         IF(FLOW) CALL FLUID(5,0,IXF,TZERO,KOP)
000204  000         J1 =0
000205  000         J2 =1
000206  000          ORLXCC      = 0.0                                         STDSTL
000207  000         DO 3001  I=1,NND
000208  000         T(I) = T(I) -460.
000209  000         IF(FLD(1,1,NSO1(J1+1)).EQ.0) GO TO 2000
000210  000         NTYPE = FLD(0,5,NSO2(J2))
000211  000         GO TO (1999,1998,1998,1998,1999,1998,1998,1998,1999), NTYPE
000212  000   1998  J2 = J2+1
000213  000   1999  J2 = J2+1
000214  000   2000  CONTINUE
000215  000         IF(FLD(4,1,NSO1(J1+1)).EQ.0) GO TO 5000
000216  000         NTYPE = FLD(0,5,NSO2(J2))
000217  000         LA = FLD(5,17,NSO2(J2))
000218  000         LK = FLD(22,14,NSO2(J2))
000219  000         GO TO (4005,4010,4015,4020,4025,4030,4035,4040,4030,          VERS   5
000220  000        *       4050,4050,4050),NTYPE                                 VERS   5
000221  000   4005  Q(I) = XK(LK)+Q(I)
000222  000         GO TO 4999
000223  000   4010  Q1 = 0.0
000224  000   4012  CALL DIDIWM(T(I),A(LA),XK(LK),Q2)
000225  000         GO TO 4998
000226  000   4015  Q1 = 0.0
```

```
000227  000   4017 CALL D1D1WM(CON(14),A(LA),XK(LK),Q2)
000228  000        GO TO 4998
000229  000   4020 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000230  000   4022 J2 = J2+1
000231  000        LA = FLD(5,17,NSQ2(J2))
000232  000        LK = FLD(22,14,NSQ2(J2))
000233  000        GO TO 4017
000234  000   4025 Q1 = XK(LK)*XK(LA)
000235  000        GO TO 4022
000236  000   4030 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000237  000        J2 = J2+1
000238  000        LA = FLD(5,17,NSQ2(J2))
000239  000        LK = FLD(22,14,NSQ2(J2))
000240  000        Q2 = XK(LK)*XK(LA)
000241  000        GO TO 4998
000242  000   4035 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000243  000   4037 J2 = J2+1
000244  000        LA = FLD(5,17,NSQ2(J2))
000245  000        LK = FLD(22,14,NSQ2(J2))
000246  000        GO TO 4012
000247  000   4040 Q1 = XK(LK)*XK(LA)
000248  000        GO TO 4037
000249  000   4050 J2=J2+1                                              VERS  5
000250  000        JPSLK=FLD(5,17,NSQ2(J2))                             VERS  5
000251  000        JPSLK=FLD(22,14,NSQ2(J2))                            VERS  5
000252  000        SPJTIM=CON(14)+XK(JPSLA)*XK(JPSLK)                   VERS  5
000253  000        CALL D1IMCY(XK(JPSLK),SPJTIM,A(LA),XK(LK),Q1)        VERS  5
000254  000        Q2=0.0                                               VERS  5
000255  000        GO TO 4998                                           VERS  5
000256  000   4998 Q(I) = Q1+Q2+Q(I)
000257  000   4999 J2 = J2+1
000258  000   5000 CONTINUE
000259  000         QSUM        = Q(I)                                 STDSTL
000260  000         GSUML       = 0.0                                  STDSTL
000261  000         GSUMR       = 0.0                                  STDSTL
000262  001        IF(.NOT.FLOW) GO TO 25
000263  001        LMP = NX(IXF+1)
000264  001        IF(LMP .EQ. 0) GO TO 25
000265  001        HA = X(IXF+LMP)
000266  001        GSUML = HA
000267  001        Q(I) = Q(I)+HA*T(LMP)
000268  001        QSUM = Q(I)
000269  000     25 J1 = J1+1
000270  000        LG = FLD(5,16,NSQ1(J1))
000271  001        IF(LG .EQ. 0) GO TO 2310
000272  000        LTA = FLD(22,14,NSQ1(J1))
000273  000        T(LTA) = T(LTA) -460.
000274  000        IF(FLD(2,1,NSQ1(J1)).EQ.0) GO TO 3000
000275  000        NTYPE = FLD(0,5,NSQ2(J2))
000276  000        LA = FLD(5,17,NSQ2(J2))
000277  000        LK = FLD(22,14,NSQ2(J2))
000278  000        GOTO(2005,2010,2015,2020,2025,2030,2035,2040,2045,2050,2055,
000279  000       1     2060,2065,2070,2073,2070) , NTYPE                VER 6
000280  000   2005 TM = (T(I)+T(LTA))/2.0
000281  000   2007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000282  000        GO TO 2999
000283  000   2010 TM = T(I)
```

```
000284  000        GO TO 2007
000285  000   2015 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000286  000   2017 J2 = J2+1
000287  000        LA = FLD(5,17,NSQ2(J2))
000288  000        LK = FLD(22,14,NSQ2(J2))
000289  000        CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000290  000        GO TO 2998
000291  000   2020 G1 = XK(LK)*XK(LA)
000292  000        GO TO 2017
000293  000   2025 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000294  000        J2 = J2+1
000295  000        LA = FLD(5,17,NSQ2(J2))
000296  000        LK = FLD(22,14,NSQ2(J2))
000297  000        G2 = XK(LK)*XK(LA)
000298  000        GO TO 2998
000299  000   2030 TM = (T(I)+T(LTA))/2.0
000300  000   2032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000301  000        GO TO 2999
000302  000   2035 TM = T(I)
000303  000        GO TO 2032
000304  000   2040 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),G1)
000305  000   2042 J2 = J2+1
000306  000        LA = FLD(5,17,NSQ2(J2))
000307  000        LK = FLD(22,14,NSQ2(J2))
000308  000        CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000309  000        GO TO 2998
000310  000   2045 G1 = XK(LK)*XK(LA)
000311  000        GO TO 2042
000312  000   2050 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),G1)
000313  000        J2 = J2+1
000314  000        LA = FLD(5,17,NSQ2(J2))
000315  000        LK = FLD(22,14,NSQ2(J2))
000316  000        G2 = XK(LK)*XK(LA)
000317  000        GO TO 2998
000318  000   2055 TM = (T(I)+T(LTA))/2.0
000319  000        CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000320  000        GO TO 2999
000321  000   2060 TM = T(LTA)
000322  000        GO TO 2007
000323  000   2065 TM = T(LTA)
000324  000        GO TO 2032
000325  000   2070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                    VER 6
000326  000   2071 TM  = (T(I) + T(LTA)) / 2.0                            VER 6
000327  000         J2 =  J2 + 1                                          VER 6
000328  000        LA  = FLD(5,17,NSQ2( J2))                              VER 6
000329  000        LK  = FLD(22,14,NSQ2( J2))                             VER 6
000330  000        IF(NTYPE .EQ. 16)  GO TO 2075                          VER 6
000331  000        CALL D2D1WM( TM, G1, A(LA), XK(LK), G(LG))             VER 6
000332  000        GO TO 2999                                             VER 6
000333  000   2073 G1 = XK(LA) *  XK(LK)                                  VER 6
000334  000        GO TO 2071                                             VER 6
000335  000   2075 G(LG) = G1 * XK(LA)  *  XK(LK)                         VER 6
000336  000        GO  TO 2999                                            VER 6
000337  000   2998 G(LG) = 1./(1./G1+1./G2)
000338  000        IF(FLD(3,1,NSQ1(J1)).EQ.1) G(LG) = G1-G2
000339  000   2999 J2 = J2+1
000340  000   3000 CONTINUE
```

```
000341  000        T(LTA) = T(LTA) +460.
000342  000         TNT         = T(LTA)
000343  000         GNG         = G(LG)
000344  000  C     CHECK FOR RADIATION CONDUCTOR
000345  000        IF(FLD(3,1,NSO1(J1)).NE.0) GO TO 2200
000346  000         GSUML       = GSUML+GNG                                  STDSTL
000347  000         QSUM        = QSUM+GNG*TNT                               STDSTL
000348  000         GO TO 2300                                               STDSTL
000349  000   2200  GSUMR = GSUMR +GNG*SIGMA
000350  000         QSUM  = QSUM  +GNG*SIGMA*TNT**4
000351  000  C     CHECK FOR LAST CONDUCTOR TO THIS NODE
000352  000   2300 IF(NSO1(J1).GT.0) GO TO 25
000353  001   2310 T(I) = T(I)+460.
000354  000  C                                                              STDSTL
000355  000  C        CHECK TO DETERMINE THAT THE SUM OF CONDUCTORS IS POSITIVE STDSTL
000356  000  C                                                              STDSTL
000357  000         IF(GSUML.GT.0.0.OR.GSUMR.GT.0.0) GO TO 2420               STDSTL
000358  000        T2 = T(I)
000359  000        GO TO NPASS
000360  000   2410  WRITE(NOUT,2411) I                                       STDSTL
000361  000   2411 FORMAT   (,60H ERROR - THE SUM OF THE CONDUCTORS ATTACHED TO RELAT
000362  000       1IVE NODE,I5,27H IS EQUAL OR LESS THAN ZERO)
000363  000        IF (LOOPCT .GT. 1) ASSIGN 2700 TO NPASS
000364  000        IF (I .LE. NNO) GO TO 2700
000365  000        IF (LOOPCT .GT. 1) ASSIGN 3600 TO JPASS
000366  000        GO TO 3600
000367  000  C                                                              STDSTL
000368  000  C        CHECK TO SEE IF MORE ENERGY IS BEING REMOVED THAN THE  STDSTL
000369  000  C        NODE CAN SUPPLY.                                      STDSTL
000370  000  C                                                              STDSTL
000371  000   2420  IF(QSUM.GT.0.0)                GO TO 2430               STDSTL
000372  000         T2          = 0.0                                        STDSTL
000373  000         GO TO 2700                                               STDSTL
000374  000   2430  IF(GSUMR.LE.0.0)               GO TO 2600               STDSTL
000375  000         IF(GSUML.LE.0.0)               GO TO 2500               STDSTL
000376  000  C                                                              STDSTL
000377  000  C        SOLVE FOR THE TEMPERATURE BY GENERAL QUARTIC EQUATION  STDSTL
000378  000  C                                                              STDSTL
000379  000         CC          = GSUML/GSUMR                                STDSTL
000380  000         FF          = -CC**2*R16                                 STDSTL
000381  000         GG          = -QSUM/GSUMR*R3                             STDSTL
000382  000         GG          = SQRT(FF**2-GG**3)                          STDSTL
000383  000         ZZ          = SQRT((GG-FF)**R3-(GG+FF)**R3)              STDSTL
000384  000         IF(ZZ.EQ.0.0)                  GO TO 2500                STDSTL
000385  000         T2          = RSR2*(-ZZ+SQRT(-ZZ**2+CC*RSR2/ZZ))         STDSTL
000386  000         GO TO 2700                                               STDSTL
000387  000  C                                                              STDSTL
000388  000  C        SOLVE FOR THE TEMPERATURE BY QUARTIC EQUATION          STDSTL
000389  000  C                                                              STDSTL
000390  000   2500  T2          = (QSUM/ GSUMR)**0.25                        STDSTL
000391  000         GO TO 2700                                               STDSTL
000392  000  C                                                              STDSTL
000393  000  C        SOLVE FOR THE TEMPERATURE BY LINEAR EQUATION           STDSTL
000394  000  C                                                              STDSTL
000395  000   2600  T2          = QSUM/GSUML                                 STDSTL
000396  000  C                                                              STDSTL
000397  000  C        COMPUTE TEMPERATURE CHANGE AND SET THE NEW TEMPERATURE STDSTL
```

```
000398  000  C                                                                STDSTL
000399  000   2700  T1          = T2-T(I)                                      STDSTL
000400  000         T(I)        = T2                                           STDSTL
000401  000         IF(ABS(DRLXCC).GT.ABS(T1))       GO TO 3001
000402  000        .DRLXCC      = T1                                           STDSTL
000403  000         NDRLXC      = I                                            STDSTL
000404  000   3001  CONTINUE
000405  000  C                                                                STDSTL
000406  000  C      DO SUCCESSIVE POINT ITERATION ON ARITHMETIC NODES          STDSTL
000407  000  C                                                                STDSTL
000408  000         IF(NNA.LE.0)                     GO TO 3800                STDSTL
000409  001   3100 JJ1 = J1
000410  000        JJ2 =J2
000411  000         DO 3700   I = NN,NNC                                       STDSTL
000412  000        T(I) = T(I) -460.
000413  000        L = I
000414  000        IF(FLD(4,1,NSO1(JJ1+1)).EQ.0) GO TO 6000
000415  000        NTYPE = FLD(0,5,NSO2(JJ2))
000416  000        LA = FLD(5,17,NSO2(JJ2))
000417  000        LK = FLD(22,14,NSO2(JJ2))
000418  000        GO TO (5005,5010,5015,5020,5025,5030,5035,5040,5050,           VERS  5
000419  000     *       5050,5050,5050),NTYPE                                     VERS  5
000420  000   5005 Q(L) = XK(LK)+Q(L)
000421  000        GO TO 5999
000422  000   5010 Q1 = 0.0
000423  000   5012 CALL D1D1WM(T(L),A(LA),XK(LK),Q2)
000424  000        GO TO 5998
000425  000   5015 Q1 = 0.0
000426  000   5017 CALL D1D1WM(CON(14),A(LA),XK(LK),Q2)
000427  000        GO TO 5998
000428  000   5020 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000429  000   5022 JJ2 = JJ2+1
000430  000        LA = FLD(5,17,NSO2(JJ2))
000431  000        LK = FLD(22,14,NSO2(JJ2))
000432  000        GO TO 5017
000433  000   5025 Q1 = XK(LK)*XK(LA)
000434  000        GO TO 5022
000435  000   5030 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000436  000        JJ2 = JJ2+1
000437  000        LA = FLD(5,17,NSO2(JJ2))
000438  000        LK = FLD(22,14,NSO2(JJ2))
000439  000        Q2 = XK(LK)*XK(LA)
000440  000        GO TO 5998
000441  000   5035 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000442  000   5037 JJ2 = JJ2+1
000443  000        LA = FLD(5,17,NSO2(JJ2))
000444  000        LK = FLD(22,14,NSO2(JJ2))
000445  000        GO TO 5012
000446  000   5040 Q1 = XK(LK)*XK(LA)
000447  000        GO TO 5037
000448  000   5050 JJ2=JJ2+1                                                   VERS  5
000449  000        JPSLA=FLD(5,17,NSO2(JJ2))                                  VERS  5
000450  000        JPSLK=FLD(22,14,NSO2(JJ2))                                 VERS  5
000451  000        SPJTIM=CON(14)*XK(JPSLA)*XK(JPSLK)                         VERS  5
000452  000        CALL D11MCY(XK(JPSLK),SPJTIM,A(LA),XK(LK),Q1)              VERS  5
000453  000        Q2=0.0                                                     VERS  5
000454  000        GO TO 5998                                                 VERS  5
```

```
000455  000    5998 Q(L) = Q1+Q2+Q(L)
000456  000    5999 JJ2 = JJ2+1
000457  000    6000 CONTINUE
000458  000         GSUML       = 0.0                                    STDSTL
000459  000         GSUMR       = 0.0                                    STDSTL
000460  000         QSUM        = Q(I)                                   STDSTL
000461  000      80 JJ1 = JJ1+1
000462  000         LG = FLD(5,16,NSQ1(JJ1))
000463  000         LTA = FLD(22,14,NSQ1(JJ1))
000464  000         T(LTA) = T(LTA) -460.
000465  000         IF(FLD(2,1,NSQ1(JJ1)).EQ.0) GO TO 4000
000466  000         NTYPE = FLD(0,5,NSQ2(JJ2))
000467  000         LA = FLD(5,17,NSQ2(JJ2))
000468  000         LK = FLD(22,14,NSQ2(JJ2))
000469  000         GOTO(3005,3010,3015,3020,3025,3030,3035,3040,3045,3050,3055,
000470  000        1     3060,3065,3070,3073,3070) , NTYPE                 VER 6
000471  000    3005 TM = (T(L)+T(LTA))/2.0
000472  000    3007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000473  000         GO TO 3999
000474  000    3010 TM = T(L)
000475  000         GO TO 3007
000476  000    3015 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000477  000    3017 JJ2 = JJ2+1
000478  000         LA = FLD(5,17,NSQ2(JJ2))
000479  000         LK = FLD(22,14,NSQ2(JJ2))
000480  000         CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000481  000         GO TO 3998
000482  000    3020 G1 = XK(LK)*XK(LA)
000483  000         GO TO 3017
000484  000    3025 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000485  000         JJ2 = JJ2+1
000486  000         LA = FLD(5,17,NSQ2(JJ2))
000487  000         LK = FLD(22,14,NSQ2(JJ2))
000488  000         G2 = XK(LK)*XK(LA)
000489  000         GO TO 3998
000490  000    3030 TM = (T(L)+T(LTA))/2.0
000491  000    3032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000492  000         GO TO 3999
000493  000    3035 TM = T(L)
000494  000         GO TO 3032
000495  000    3040 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000496  000    3042 JJ2 = JJ2+1
000497  000         LA = FLD(5,17,NSQ2(JJ2))
000498  000         LK = FLD(22,14,NSQ2(JJ2))
000499  000         CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000500  000         GO TO 3998
000501  000    3045 G1 = XK(LK)*XK(LA)
000502  000         GO TO 3042
000503  000    3050 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000504  000         JJ2 = JJ2+1
000505  000         LA = FLD(5,17,NSQ2(JJ2))
000506  000         LK = FLD(22,14,NSQ2(JJ2))
000507  000         G2 = XK(LK)*XK(LA)
000508  000         GO TO 3998
000509  000    3055 TM = (T(L)+T(LTA))/2.0
000510  000         CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000511  000         GO TO 3999
```

```
000512  000   3060 TM = T(LTA)
000513  000        GO TO 3007
000514  000   3065 TM = T(LTA)
000515  000        GO TO 3032
000516  000   3070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                    VER 6
000517  000   3071 TM  = (T(L) + T(LTA)) / 2.0                            VER 6
000518  000        JJ2 = JJ2 + 1                                          VER 6
000519  000        LA  = FLD(5,17,NSO2(JJ2))                              VER 6
000520  000        LK  = FLD(22,14,NSO2(JJ2))                             VER 6
000521  000        IF(NTYPE .EQ.16) GO TO 3075                            VER 6
000522  000        CALL D2D1WM(TM,  G1,  A(LA), XK(LK), G(LG))            VER 6
000523  000        GO TO  3999                                            VER 6
000524  000   3073 G1 = XK(LA) * XK(LK)                                   VER 6
000525  000        GO TO 3071                                             VER 6
000526  000   3075 G(LG) = G1 * XK(LA) * XK(LK)                           VER 6
000527  000        GO TO 3999                                             VER 6
000528  000   3998 G(LG) = 1./(1./G1+1./G2)
000529  000        IF(FLD(3,1,NSO1(JJ1)).EQ.1) G(LG) = G1*G2
000530  000   3999 JJ2 = JJ2+1
000531  000   4000 CONTINUE
000532  000        T(LTA) = T(LTA) +460.
000533  000         GNG           = G(LG)
000534  000         TNT           = T(LTA)
000535  000   C    CHECK FOR RADIATION CONDUCTOR
000536  000        IF(FLD(3,1,NSO1(JJ1)).NE.0) GO TO 3400
000537  000         QSUM          = QSUM+GNG*TNT                           STDSTL
000538  000         GSUML         = GSUML+GNG                              STDSTL
000539  000         GO TO 3500                                             STDSTL
000540  000   3400  QSUM  = QSUM  +GNG*SIGMA*TNT**4
000541  000         GSUMR = GSUMR +GNG*SIGMA
000542  000   C    CHECK FOR LAST CONDUCTOR TO THIS NODE
000543  000   3500 IF(NSO1(JJ1).GT.0) GO TO 80
000544  000        T(I) = T(I) +460.
000545  000   C                                                           STDSTL
000546  000   C      CHECK TO DETERMINE THAT THE SUM OF CONDUCTORS IS POSITIVE  STDSTL
000547  000   C                                                           STDSTL
000548  000         IF(GSUML.GT.0.0 .OR.GSUMR.GT.0.0)GO TO 3501
000549  000        T2 = T(I)
000550  000        GO TO JPASS
000551  000   C                                                           STDSTL
000552  000   C      CHECK TO SEE IF MORE ENERGY IS BEING REMOVED THAN THE    STDSTL
000553  000   C      NODE CAN SUPPLY.                                     STDSTL
000554  000   C                                                           STDSTL
000555  000   3501  IF(QSUM.GT.0.0)                GO TO 3505             STDSTL
000556  000         T2          = 0.0                                      STDSTL
000557  000         GO TO 3600                                             STDSTL
000558  000   3505  IF(GSUMR.LE.0.0)               GO TO 3520             STDSTL
000559  000         IF(GSUML.LE.0.0)               GO TO 3510             STDSTL
000560  000   C                                                           STDSTL
000561  000   C      SOLVE FOR THE TEMPERATURE BY GENERAL QUARTIC EQUATION    STDSTL
000562  000   C                                                           STDSTL
000563  000         CC          = GSUML/GSUMR                              STDSTL
000564  000         FF          = -CC**2*R16                               STDSTL
000565  000         GG          = -QSUM/GSUMR*R3                           STDSTL
000566  000         GG          = SQRT(FF**2-GG**3)                        STDSTL
000567  000         ZZ          = SQRT((GG-FF)**R3-(GG+FF)**R3)            STDSTL
000568  000         IF(ZZ.EQ.0.0)                  GO TO 3510             STDSTL
```

```
000569  000          T2          = RSR2*(-ZZ+SQRT(-ZZ**2+CC*RSR2/ZZ))                STDSTL
000570  000          GO TO 3600                                                       STDSTL
000571  000  C                                                                       STDSTL
000572  000  C         SOLVE FOR THE TEMPERATURE BY QUARTIC EQUATION                  STDSTL
000573  000  C                                                                       STDSTL
000574  000   3510   T2          = (QSUM/ GSUMR)**0.25                              STDSTL
000575  000          GO TO 3600                                                       STDSTL
000576  000  C                                                                       STDSTL
000577  000  C         SOLVE FOR THE TEMPERATURE BY LINEAR EQUATION                   STDSTL
000578  000  C                                                                       STDSTL
000579  000   3520   T2          = QSUM/GSUML                                       STDSTL
000580  000  C                                                                       STDSTL
000581  000  C         COMPUTE TEMPERATURE CHANGE AND SET THE NEW TEMPERATURE         STDSTL
000582  000  C                                                                       STDSTL
000583  000   3600   T1          = T2-T(I)                                          STDSTL
000584  000          T(I)        = T2                                               STDSTL
000585  000          IF(ABS(ARLXCC).GT.ABS(T1))       GO TO 3700                    STDSTL
000586  000          ARLXCC      = T1                                               STDSTL
000587  000          NARLXC      = I                                                STDSTL
000588  000   3700   CONTINUE                                                       STDSTL
000589  000  C                                                                       STDSTL
000590  000  C**********                                                            STDSTL
000591  000  C                                                                       STDSTL
000592  000  C       CHECK IF RELAXATION CRITERIA IS MET.                            STDSTL
000593  000  C                                                                       STDSTL
000594  000  C         THE VARIABLE -NQ- INDICATES THE RELAXATION CRITERIA           STDSTL
000595  000  C         BEING WORKED TOWARD---                                        STDSTL
000596  000  C              NQ = 0  TRYING TO COLSE ON DRLXCA AND/OR ARLXCA          STDSTL
000597  000  C                 = 1  TRYING TO CLOSE ON BALENG FOR THE SYSTEM BALANCE STDSTL
000598  000  C                 = 2  TRYING TO CLOSE ON BALENG FOR THE MAXIMUM NODAL  STDSTL
000599  000  C                      BALANCE                                          STDSTL
000600  000  C                 = 3  HAVE ACHIEVED CLOSURE FOR ALL CRITERIA INPUT    STDSTL
000601  000  C                                                                       STDSTL
000602  000  C                                                                       STDSTL
000603  000   3800   IF(NQ.GT.0)                        GO TO 155
000604  000          IF(ABS(DRLXCC).GT.DRLXCA.OR.                                   STDSTL
000605  000       .     ABS(ARLXCC).GT.ARLXCA)          GO TO 4390                 STDSTL
000606  000          WRITE(NOUT,3801) LOOPCT                                        STDSTL
000607  000   3801 FORMAT(1H1,58H NOTE - BOTH DRLXCA AND ARLXCA HAVE BEEN MET WITH LO
000608  000      1OPCT =,I10)
000609  000          NL = NL+2                                                      STDSTL
000610  000          NQ          = 1                                                STDSTL
000611  000          DO 3810  I = 1,NNT                                             STDSTL
000612  000   3810   T(I)        = T(I)-460.0                                       STDSTL
000613  000          CALL OUTCAL                                                    STDSTL
000614  000          DO 3830  I = 1,NNT                                             STDSTL
000615  000   3830   T(I)        = T(I)+460.0                                       STDSTL
000616  000  C                                                                       STDSTL
000617  000  C       TEMPERATURE CHANGE CRITERIA HAS BEEN MET, NOW COMPUTE          STDSTL
000618  000  C            THE ENERGY BALANCE FROM THE SYSTEM TO THE BOUNDARIES OR    STDSTL
000619  000  C            THE LARGEST INBALANCE ON A NODE.                           STDSTL
000620  000  C                                                                       STDSTL
000621  000          IF(BALENG.LE.0.0)                GO TO 4150                    STDSTL
000622  000    155 DO 3840 I=1,NNT
000623  000   3840 T(I) = T(I) -460.
000624  000          CALL NONLIN
000625  000          QOUT = 0.0
```

```
000626  000        QIN = 0.0
000627  000        J1 = 0
000628  000        DO 195 I = 1,NNC
000629  000        QIN = QIN+Q(I)
000630  001        IF(.NOT.FLOW) GO TO 165
000631  001        LMP = NX(IXF+I)
000632  001        IF(LMP .EQ. 0) GO TO 165
000633  001        QIN = QIN+X(IXF+LMP)*(T(LMP)-T(I))
000634  000    165 J1 = J1+1
000635  000        LTA = FLD(22,16,NSQ1(J1))
000636  000        IF(LTA.LE.NNC) GO TO 175
000637  000        LG = FLD(5,16,NSQ1(J1))
000638  001        IF(LG .EQ. 0) GO TO 195
000639  000        IF(FLD(3,1,NSQ1(J1)).EQ.0) GO TO 170
000640  000        T1 = T(I)+460.0
000641  000        T2 = T(LTA)+460.0
000642  000        QOUT = QOUT+G(LG)*(T1**4-T2**4)
000643  000        QOUT = QOUT +G(LG)*(T1**4-T2**4) *CON(50)
000644  000        GO TO 175
000645  000    170 QOUT = QOUT+G(LG)*(T(I)-T(LTA))
000646  000    C   CHECK FOR LAST CONDUCTOR TO THIS NODE
000647  000    175 IF(NSQ1(J1).GT.0) GO TO 165
000648  000    195 CONTINUE
000649  000        DO 3850 I=1,NNT
000650  000   3850 T(I) = T(I) +460.
000651  000        ENGBAL  = ABS(QIN-QOUT)
000652  000        IF (ICNT .EQ. 0) WRITE (NOUT,3853) ENGBAL,LOOPCT
000653  000   3853 FORMAT (// 13H     ENGBAL =,E12.5,13H  AT LOOPCT =,I10)
000654  000        ICNT =1
000655  000    C                                                              STDSTL
000656  000    C       CHECK TO SEE IF ENERGY BALANCE CRITERIA HAS BEEN MET.  STDSTL
000657  000    C                                                              STDSTL
000658  000         IF(ABS(ENGBAL).GE.BALENG)       GO TO 4390                STDSTL
000659  000         IF(NQ.EQ.2)                     GO TO 4110                STDSTL
000660  000         WRITE(NOUT,4101) ENGBAL, LOOPCT                           STDSTL
000661  000   4101 FORMAT(1H1,62H NOTE - SYSTEM ENERGY BALANCE CRITERIA HAS BEEn MET,
000662  000       1 ENGBAL = ,10H ENGBAL = ,G13.6,10H  LOOPCT =,I10)
000663  000         NLINE       = NLINE+2                                     STDSTL
000664  000         NQ          = 2                                           STDSTL
000665  000         DO 4105   I = 1,NNT                                       STDSTL
000666  000   4105  T(I)        = T(I)-460.0                                  STDSTL
000667  000         CALL OUTCAL                                               STDSTL
000668  000         DO 4107   I = 1,NNT                                       STDSTL
000669  000   4107  T(I)        = T(I)+460.0                                  STDSTL
000670  000         GO TO 5000                                                STDSTL
000671  000   4110  WRITE(NOUT,4111) ENGBAL, LOOPCT
000672  000   4111 FORMAT(1H1,62H NOTE - NODAL ENERGY BALANCE CRITERIA HAS BEEN MET,
000673  000       1 ENGBAL = ,G13.6,                              8HLOOPCT =,I10)
000674  000         NLINE       = NLINE+2                                     STDSTL
000675  000   4150  NQ          = 3                                           STDSTL
000676  000         DO 4200   I = 1,NNT                                       STDSTL
000677  000   4200  T(I)        = T(I)-460.0                                  STDSTL
000678  000         GO TO 5001
000679  000    C                                                              STDSTL
000680  000    C**********                                                    STDSTL
000681  000    C                                                              STDSTL
000682  000    C       IF THREE ITERATIONS HAVE BEEN PERFORMED,EXTRAPOLATE USING STDSTL
```

```
000683  000  C         -AITKENS DELTA SQUARED METHOD-.                                  STDSTL
000684  000  C                                                                          STDSTL
000685  000  C              X(E) = X(3)-(X(3)-X(2))**2/((X(3)-X(2))-(X(2)-X(1)))       STDSTL
000686  000  C                                                                          STDSTL
000687  000   4390  GO TO ITER                                                         STDSTL
000688  000  C                                                                          STDSTL
000689  000  C        FIRST ITERATION                                                   STDSTL
000690  000  C                                                                          STDSTL
000691  000   4400  ASSIGN 4440 TO ITER                                               STDSTL
000692  000         DO 4410   I = 1,NNC                                                STDSTL
000693  000   4410  X(IX1+I)    = T(I)                                                 STDSTL
000694  000         GO TO 4720                                                         STDSTL
000695  000  C                                                                          STDSTL
000696  000  C        SECOND ITERATION                                                  STDSTL
000697  000  C                                                                          STDSTL
000698  000   4440  ASSIGN 4500 TO ITER                                               STDSTL
000699  000         DO 4450   I = 1,NNC                                                STDSTL
000700  000   4450  X(IX2+I)    = T(I)                                                 STDSTL
000701  000         GO TO 4720                                                         STDSTL
000702  000  C                                                                          STDSTL
000703  000  C        THIRD ITERATION                                                   STDSTL
000704  000  C                                                                          STDSTL
000705  000   4500  ASSIGN 4400 TO ITER                                               STDSTL
000706  000        DIFMAX = 50.
000707  000        IF(ABS(EXTLIM) .NE. 0.)  DIFMAX = ABS(EXTLIM)
000708  000         DO 4710   I = 1,NNC                                               STDSTL
000709  000         T1          = X(IX1+I)                                            STDSTL
000710  000         T2          = X(IX2+I)                                            STDSTL
000711  000         T3          = T(I)                                                STDSTL
000712  000         IF(T1.LT.T2.AND.T2.LT.T3)       GO TO 4600                        STDSTL
000713  000         IF(T1.GT.T2.AND.T2.GT.T3)       GO TO 4600                        STDSTL
000714  000         GO TO 4710                                                        STDSTL
000715  000   4600  ANUM         = T3-T2                                              STDSTL
000716  000         R1           = T2-T1                                              STDSTL
000717  000         IF(ABS(ANUM).GE.ABS(R1))        GO TO 4710                        STDSTL
000718  000         ADEN         = ANUM-R1                                            STDSTL
000719  000         IF(ABS(ADEN).LT.1.0E-10)        GO TO 4710                        STDSTL
000720  000         TE           = T3-ANUM**2/ADEN                                    STDSTL
000721  000         TDIF         = TE-T3                                              STDSTL
000722  000  C                                                                        STDSTL
000723  000  C        LIMIT THE EXTRAPOLATION TO + OR -  DIFMAX DEG., AND RESULTANT   STDSTL
000724  000  C        TEMPERATURES MUST BE POSITIVE                                   STDSTL
000725  000  C                                                                        STDSTL
000726  000        IF(ABS(TDIF) .GT. DIFMAX)  TE = T3+SIGN(DIFMAX,TDIF)
000727  000         IF(TE.LT.0.0)                   TE = 0.0                         STDSTL
000728  000         TDIF         = TE-T3                                             STDSTL
000729  000         IF(I.GT.NNB)                    GO TO 4680                       STDSTL
000730  000         IF(ABS(TDIF).LE.ABS(DRLXCC))    GO TO 4700                       STDSTL
000731  000         DRLXCC       = TDIF                                              STDSTL
000732  000         NDRLXC       = I                                                 STDSTL
000733  000         GO TO 4700                                                       STDSTL
000734  000   4680  IF(ABS(TDIF).LE.ABS(ARLXCC))    GO TO 4700                       STDSTL
000735  000         ARLXCC       = TDIF                                              STDSTL
000736  000         NARLXC       = I                                                 STDSTL
000737  000   4700  T(I)         = TE                                                STDSTL
000738  000   4710  CONTINUE                                                        STDSTL
000739  000  C                                                                        STDSTL
```

```
000740  000   C        CONVERT TEMPERATURES BACK TO FARENHEIT AND DETERMINE IF        STDSTL
000741  000   C        -OUTCAL- IS TO BE CALLED.                                     STDSTL
000742  000   C                                                                      STDSTL
000743  000    4720  DO 4730   I = 1,NNT                                            STDSTL
000744  000    4730  T(I)        = T(I)-460.0                                        STDSTL
000745  000          IF(OPEITR.EQ.0.0)                GO TO 4800                   STDSTL
000746  000          IF(MOD(LOOPCT,IFIX(OPEITR)).NE.0)GO TO 4800                   STDSTL
000747  000          CALL OUTCAL                                                   STDSTL
000748  000    4800  CONTINUE                                                      STDSTL
000749  000   C                                                                    STDSTL
000750  000   C++++++++++                                                          STDSTL
000751  000   C                                                                    STDSTL
000752  000   C        FAILURE TO ACHIEVE CLOSURE WITHIN THE MAXIMUM NUMBER        STDSTL
000753  000   C        OF ITERATIONS.                                              STDSTL
000754  000   C                                                                    STDSTL
000755  000   C                                                                    STDSTL
000756  001         IF(LNODE .EQ. 0) CALL NNREAD(1)
000757  001         IF(NDRLXC .GT. 0) NDRLXC = NX(LNODE+NDRLXC)
000758  001         IF(NARLXC .GT. 0) NARLXC = NX(LNODE+NARLXC)
000759  000          WRITE (NOUT,4901) NLOOP , NDRLXC, DRLXCC, DRLXCA, NARLXC, ARLXCC,STDSTL
000760  000        1                   ARLXCA, ENGBAL, BALENG                      STDSTL
000761  000    4901 FORMAT(1H1,77H CAUTION - ITERATION COUNT EXCEEDED BEFORE RELAXATIO
000762  000        1N CRITERIA MET.  LOOPCT =,I10/13X, 7HDRLXCC(,I6,2H)=,G13.6,13H VS.
000763  000        2 DRLXCA =,G13.6/13X 7HARLXCC(,I6,2H)=,G13.6,13H VS. ARLXCA =,G13.6
000764  000        3/13X, 6HENGBAL,8X,1H=,G13.6,13H VS. BALENG =,G13.6)
000765  000          NLINE       = NLINE+5                                       STDSTL
000766  000   C                                                                    STDSTL
000767  000   C++++++++++                                                          STDSTL
000768  000   C                                                                    STDSTL
000769  000   C        CALL -VARBL2- AND -OUTCAL-, THEN DETERMINE IF THE           STDSTL
000770  000   C        PROBLEM IS TO STOP OR CONTINUE.                             STDSTL
000771  000   C                                                                    STDSTL
000772  000   C                                                                    STDSTL
000773  000    5001  CALL VARBL2
000774  000          TIMEO       = TIMEN                                         STDSTL
000775  000          CALL OUTCAL                                                 STDSTL
000776  000          IF(TIMEN+.01*OUTPUT.GE.TIMEND)  GO TO 5300                  STDSTL
000777  000   C                                                                    STDSTL
000778  000   C++++++++++                                                          STDSTL
000779  000   C                                                                    STDSTL
000780  000   C       INCREMENTING TIME CONSTANTS.                                 STDSTL
000781  000   C                                                                    STDSTL
000782  000          TIMEN       = TIMEO+OUTPUT                                  STDSTL
000783  000          IF(TIMEN.GT.TIMEND)              TIMEN = TIMEND             STDSTL
000784  000          TIMEM       = (TIMEN+TIMEO)/2.0                             STDSTL
000785  000          DTIMEU      = TIMEN-TIMEO                                   STDSTL
000786  000          GO TO 1100                                                  STDSTL
000787  000   C                                                                    STDSTL
000788  001    5300  NTH         = IXF                                           STDSTL
000789  000          NDIM        = IXL                                           STDSTL
000790  000          RETURN                                                      STDSTL
000791  000          END                                                         STDSTL

END ELT.
```

#HDG,R,STQ&EBTOREP

```
*ELT,L STOREP
ELTOT7 RL1870 02/28-03:21:09-(0,)
000001      000          COMPILER (XM=1)
000002      000          SUBROUTINE STOREP(IDENT)
000003      000          COMMON /FDIMNS/ NTYP, NSYS, NTB, NP, NV, NFD
000004      000          COMMON /FLODAT/ FLOW(1)
000005      000          COMMON /SYSDAT/ SYSTEM(15,1)
000006      000          COMMON /TYPDAT/ TYPE(10,1)
000007      000          COMMON /WDOT  / W(1)
000008      000          COMMON /PRESS / P(1)
000009      000          COMMON /FLOWG / GF(1)
000010      000          COMMON /VALVP / VP(1)
000011      000          COMMON /WDOTI / WI(1)
000012      000          COMMON /FLOWR / AFR(1)
000013      000          COMMON /DELTAP/ DP(1)
000014      000          INCLUDE COMM,LIST
000015      000          DIMENSION DUMMY(1)
000016      000          EQUIVALENCE (DUMMY(1),NND), (DUMMY(9),LENA)
000017      000          INTEGER STAPE
000018      000          DATA STAPE /22/, NOUT /6/, LUT1 /4/
000019      000    C
000020      000          REWIND LUT1
000021      000    C..PROBLEM IDENTIFICATION
000022      000          WRITE (STAPE) IDENT
000023      000    C..TITLE
000024      000          WRITE (STAPE) (H(I),I=1,20)
000025      000    C..DIMENSIONS
000026      000          WRITE (STAPE) NND,NNA,NNT,NGT,NCT,NAT,LSQ1,LSQ2,LENA
000027      000    C..CHECK FOR GENERAL PROBLEM
000028      000          IF (KON(31).EQ.2) GO TO 50
000029      000    C..NODE DATA
000030      000          J=NDIM-(NNT+NTH)
000031      000          IF (J.LT.0) GO TO 200
000032      000          READ (LUT1) NNT,(X(I+NTH),I=1,NNT)
000033      000          WRITE (STAPE) (X(I+NTH),I=1,NNT)
000034      000          WRITE (STAPE) (T(I),I=1,NNT)
000035      000          IF (NND.GT.0)  WRITE (STAPE) (C(I),I=1,NND)
000036      000    C..CONDUCTOR DATA
000037      000          J=NDIM-(NGT+NTH)
000038      000          IF (J.LT.0) GO TO 200
000039      000          READ (LUT1) NGT,(X(I+NTH),I=1,NGT)
000040      000          WRITE (STAPE) (X(I+NTH),I=1,NGT)
000041      000          WRITE (STAPE) (G(I),I=1,NGT)
000042      000    C..CONSTANTS DATA
000043      000     50   WRITE (STAPE) (KON(I),I=1,50)
000044      000          IF (NCT.EQ.0) GO TO 60
000045      000          J=NDIM-(NCT+NTH)
000046      000          IF (J.LT.0) GO TO 200
000047      000          READ (LUT1) NUC,NCT,(X(I+NTH),I=1,NCT)
000048      000          WRITE (STAPE) NUC,(X(I+NTH),I=1,NCT)
000049      000          WRITE (STAPE) (K(I),I=1,NCT)
000050      000    C..ARRAY DATA
000051      000     60   CONTINUE
000052      000          IF (NAT.EQ.0) GO TO 70
000053      000          J=NDIM-(NAT+NTH)
000054      000          IF (J.LT.0) GO TO 200
000055      000          READ (LUT1) NAT,(X(I+NTH),I=1,NAT)
```

```
000056  000          WRITE (STAPE) (X(I+NTH),I=1,NAT)
000057  000          READ (LUT1) NAT,(X(I+NTH),I=1,NAT)
000058  000          WRITE (STAPE) (X(I+NTH),I=1,NAT)
000059  000          WRITE (STAPE) (A(I),I=1,LENA)
000060  000    C..PSEUDO COMPUTE SEQUENCE DATA
000061  000     70   CONTINUE
000062  000          IF (KON(31).EQ.2) GO TO 100
000063  000          WRITE(STAPE) NTYP, NSYS, NTB, NP, NV, NFD
000064  000          IF(NSYS .LT. 1) GO TO 80
000065  000          NSP = MAXO(NTB,MAXO(NP,NV))
000066  000          J = NDIM - (NSP+NTH)
000067  000          IF(J .LT. 0) GO TO 200
000068  000          READ(LUT1)   (X(NTH+I),I=1,NTB)
000069  000          WRITE(STAPE) (X(NTH+I),I=1,NTB)
000070  000          READ(LUT1)   (X(NTH+I),I=1,NP )
000071  000          WRITE(STAPE) (X(NTH+I),I=1,NP )
000072  000          IF(NV .EQ. 0) GO TO 75
000073  000          READ(LUT1)   (X(NTH+I),I=1,NV )
000074  000          WRITE(STAPE) (X(NTH+I),I=1,NV )
000075  000       75 WRITE(STAPE) (FLOW(I),I=1,NFD)
000076  000          WRITE(STAPE)((SYSTEM(I,J),I=1,15),J=1,NSYS)
000077  000          WRITE(STAPE)((TYPE  (I,J),I=1,10),J=1,NTYP)
000078  000          WRITE(STAPE) (W(I),I=1,NTB)
000079  000          IF(NV .GT. 0) WRITE(STAPE) (VP(I),I=1,NV)
000080  000          WRITE(STAPE) (WI(I),I=1,NP)
000081  000          WRITE(STAPE) (P (I),I=1,NP)
000082  000       80 CONTINUE
000083  000          WRITE (STAPE) (NSQ1(I),I=1,LSQ1)
000084  000          IF (LSQ2.EQ.0) GO TO 100
000085  000          WRITE (STAPE) (NSQ2(I),I=1,LSQ2)
000086  000    C
000087  000     100  CONTINUE
000088  000          WRITE (NOUT,109) IDENT
000089  000     109  FORMAT (' THE PROBLEM IDENTIFIED AS ',A6,' HAS BEEN STORED AT THIS-
000090  000         1 POINT')
000091  000     120  RETURN
000092  000    C
000093  000     200  WRITE (NOUT,209) J
000094  000     209  FORMAT (' STOREP SHORT',I5,' DYNAMIC CORE LOCATIONS')
000095  000          GO TO 120
000096  000          END

END ELT.


*NOG,F SUBN
```

```
@ELT,L SUBN
ELTOT7 RLIB70 02/28-03:21:11-(2,)
000001      000          SUBROUTINE SUBN(VALUE,ISW)
000002      000    C
000003      000    C SUBROUTINE SUBN BUILDS NUMBERS FROM CHARACTERS
000004      000    C
000005      000          DIMENSION ITABC(6), ITABN(10)
000006      000          COMMON /CIMAGE/ KARD(80)
000007      000          COMMON /CARD  / KRD, KOL, MXKOL
000008      000          COMMON /TAPE  / NIN, NOUT
000009      000          EQUIVALENCE (RN,IN)
000010      000    C
000011      000          DATA ITABC / 1H+, 1H., 1H-, 1HE, 1HN, 1HD /
000012      000          DATA ITABN / 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9 /
000013      000    C
000014      000    C
000015      000          KSAVE = KOL
000016      000          VALUE = 0.0
000017      000          INT1  = 0
000018      000          IEXP1 = 0
000019      000          IEXP2 = 0
000020      000          ISIGN1 = 1
000021      000          ISIGN2 = 1
000022      000          ISW = 1
000023      000          IF(KOL .LT. MXKOL) GO TO 4
000024      000          CALL CARDIN(ISW)
000025      000          GO TO(4,1), ISW
000026      000        1 ISW = 5
000027      000          RETURN
000028      000        4 KOL = KOL - 1
000029      000        5 I = KOL + 1
000030      000          IF(I .GT. MXKOL) GO TO(40,50,45), ISW
000031      000          DO 6 KOL=I,MXKOL
000032      000          IF(KARD(KOL) .NE. 1H ) GO TO 10
000033      000        6 CONTINUE
000034      000        7 KOL = MXKOL
000035      000          GO TO(40,50,45), ISW
000036      000       10 DO 20 I=1,10
000037      000          IF(KARD(KOL) .EQ. ITABN(I)) GO TO(240,230,250), ISW
000038      000       20 CONTINUE
000039      000          DO 30 I=1,4
000040      000          IF(KARD(KOL) .EQ. ITABC(I)) GO TO(60,100,140,200), I
000041      000       30 CONTINUE
000042      000          GO TO (40,50,45),ISW
000043      000       40 IN = INT1*ISIGN1
000044      000          VALUE = RN
000045      000          RETURN
000046      000       45 ISW = 2
000047      000       50 VALUE =INT1*ISIGN1*10.0**(IEXP1+IEXP2*ISIGN2)
000048      000          RETURN
000049      000       60 GO TO(70,80,80), ISW
000050      000       70 IF(INT1.NE. 0) ISW = 2
000051      000          GO TO 5
000052      000       80 IF(IEXP2 .NE. 0) GO TO 270
000053      000          ISW = 3
000054      000          GO TO 5
000055      000    C
```

```
000056  000     100 GO TO(110,290,280), ISW
000057  000     110 ISW = 2
000058  000         GO TO 5
000059  000   C
000060  000     140 GO TO(150,180,180), ISW
000061  000     150 IF(INT1.EQ. 0) GO TO 170
000062  000         ISW = 2
000063  000         GO TO 5
000064  000     170 ISIGN1 = -1
000065  000         GO TO 5
000066  000     180 IF(IEXP2 .NE. 0) GO TO 270
000067  000         ISW = 3
000068  000         ISIGN2 = -1
000069  000         GO TO 5
000070  000   C
000071  000     200 IF(KARD(KOL+1) .EQ. ITABC(5) .AND. KARD(KOL+2) .EQ. ITABC(6))
000072  000        1GO TO 220
000073  000         GO TO(210,210,260), ISW
000074  000     210 ISW = 3
000075  000         GO TO 5
000076  001     220 ISW = 3
000077  000         RETURN
000078  000   C
000079  000     230 IEXP1 = IEXP1 - 1
000080  000     240 INT1 = INT1 * 10 + I - 1
000081  000         GO TO 5
000082  000     250 IEXP2 = IEXP2*10 + I - 1
000083  000         GO TO 5
000084  000   C
000085  000     260 WRITE(NOUT,265) KOL
000086  000     265 FORMAT(48HOE ENCOUNTERED AFTER START OF EXPONENT AT COLUMN I3,
000087  000        1        14H OF ABOVE CARD /)
000088  000         GO TO 400
000089  000     270 WRITE(NOUT,275) KOL
000090  000     275 FORMAT(51HOSIGN ENCOUNTERED AFTER START OF EXPONENT AT COLUMN I3,
000091  000        1        14H OF ABOVE CARD /)
000092  000         GO TO 400
000093  000     280 WRITE(NOUT,285) KOL
000094  000     285 FORMAT(48H0DECIMAL POINT ENCOUNTERED IN EXPONENT AT COLUMN I3,
000095  000        1        14H OF ABOVE CARD /)
000096  000         GO TO 400
000097  000     290 WRITE(NOUT,295) KOL
000098  000     295 FORMAT(43HOSECOND DECIMAL POINT ENCOUNTERED AT COLUMN I3,
000099  000        1        14H OF ABOVE CARD /)
000100  000     400 ISW = 4
000101  000         RETURN
000102  000         END

END ELT.


*HDG,P SYSPAR
```

```
#ELT,L SYSPAR
ELTGT7 RLIB70 02/28-03:21:12-(1,)
000001      000          SUBROUTINE SYSPAR(SYS,L,SPR,JSW)
000002      000    C
000003      000          LOGICAL LPAR, ERR
000004      000    C
000005      000          DIMENSION SYS(1), CODE(6), SET(15), LPAR(15), SPR(1)
000006      000          DIMENSION KEY1(10), LOC1(7), KEY2(5), LOC2(5)
000007      000    C
000008      000          COMMON /TAPE  / NIN, NOUT
000009      000          COMMON /CARD  / KRD, KOL, MXKOL
000010      000          COMMON /CIMAGE/ KARD(80)
000011      001           COMMON /FLOERR/ERR
000012      000    C
000013      000          DATA KEY1 / 1HC, 1HR, 1HG, 1HT, 1HF, 1HH  1HP, 1HE, 1HM, 1HK /
000014      000          DATA LOC1 /   1,   2,   5,   7,   9,  13,  15                 /
000015      000          DATA KEY2 / 1HU, 1HT, 1HP, 1HX, 1HQ                           /
000016      000          DATA LOC2 /   3,   4,   6,   8,  10                           /
000017      000    C
000018      000          DATA SET  / 4*0, 4.1696208E8, 1, 0.01, 100, 0.5, 6*0          /
000019      000          DATA CODE / 2HCP, 2HRO, 2HMU, 2HKT, 1HH, 1HP                  /
000020      000    C
000021      000          L = 0
000022      000          ISW = 1
000023      000        5 DO 10 I=1,15
000024      000          LPAR(I) = .FALSE.
000025      000          SYS(I) = SET(I)
000026      000       10 CONTINUE
000027      000       15 CALL SKPB(JSW)
000028      000          GO TO(25,720), JSW
000029      000       25 DO 50 I=7,10
000030      000          IF(KARD(KOL) .NE. KEY1(I)) GO TO 50
000031      000          IF(I-8) 60,30,200
000032      000       30 IF(KARD(KOL+1) .NE. 1HN) GO TO 230
000033      000          IF(KARD(KOL+2) .NE. 1HO) GO TO 230
000034      000          CALL SKPTE(JSW)
000035      000          GO TO 720
000036      000       50 CONTINUE
000037      000          DO 55 I=1,6
000038      000          IF (KARD(KOL).EQ.KEY1(I)) GO TO 245
000039      000       55 CONTINUE
000040      000          GO TO 230
000041      000    C
000042      000       60 J = KOL + 1
000043      000          DO 70 KOL=J,MXKOL
000044      000          IF(KARD(KOL) .EQ. 1H ) GO TO 80
000045      000       70 CONTINUE
000046      000          WRITE(NOUT,75)
000047      000       75 FORMAT(48H0* * *  PRESSURE NODE NUMBER NOT SUPPLIED  * * * /)
000048      000          GO TO 650
000049      000       80 KOL = KOL + 1
000050      000          CALL SUBN (SPR(L+1),ISW)
000051      000          GO TO(240,85,95,650,710), ISW
000052      000       85 WRITE(NOUT,90)
000053      000       90 FORMAT(67H0* * *  FLOATING POINT NUMBER INPUT FOR PRESSURE NODE NU
000054      000         1MBER  * * * /)
000055      000          GO TO 650
```

```
000056  000      95 WRITE(NOUT,100)
000057  000     100 FORMAT(62H0* * *  END FOUND BEFORE SPECIFIED PRESSURE NODE NUMBER
000058  000        1 * * * /)
000059  000         GO TO 650
000060  000     200 KOL = KOL + 1
000061  000         DO 220 I=1,5
000062  000         IF(KARD(KOL) .EQ. KEY2(I)) GO TO 250
000063  000     220 CONTINUE
000064  000     230 WRITE(NOUT,235) KOL
000065  000     235 FORMAT(36H0* * *  INVALID DESIGNATOR IN COLUMN I3,.7H  * * * /)
000066  000         GO TO 650
000067  000     240 LPAR(19) = .TRUE.
000068  000     245 LOC = LOC1(I)
000069  000         GO TO 400
000070  000     250 LOC = LOC2(I)
000071  000     400 J = KOL + 1
000072  000         DO 420 KOL=J,MXKOL
000073  000         IF(KARD(KOL) .EQ. 1H= .OR. KARD(KOL) .EQ. 1H,) GO TO 450
000074  000     420 CONTINUE
000075  000     450 KOL = KOL + 1
000076  000         CALL SKPB(JSW)
000077  000         GO TO(475,720), JSW
000078  000     475 IF (KARD(KOL).NE.1HA) GO TO 550
000079  000         KOL = KOL + 1
000080  000     480 CALL SUBN(SYS(LOC),ISW)
000081  000         GO TO(700,485,500,650,500), ISW
000082  000     485 WRITE(NOUT,490)
000083  000     490 FORMAT(59H0* * *  FLOATING POINT NUMBER INPUT FOR ARRAY NUMBER  *
000084  000        1* * /)
000085  000         GO TO 705
000086  000     500 WRITE(NOUT,510)
000087  000     510 FORMAT(44H0* * *  END FOUND BEFORE ARRAY NUMBER  * * * /)
000088  000         IF(ISW .EQ. 5) GO TO 710
000089  000         GO TO 650
000090  000     550 IF (LOC.NE.15) GO TO 600
000091  000         CALL SUBN (SPAR(L+2),ISW)
000092  000         GO TO (610,555,630,650,630),ISW
000093  000     555 L = L + 2
000094  000         GO TO 700
000095  000     600 CALL SUBN(SYS(LOC),ISW)
000096  000         IF (LOC.EQ.6.OR.LOC.EQ.8.OR.LOC.EQ.10)
000097  000        1GO TO (700,625,630,650,630),ISW
000098  000         GO TO(610,700,630,650,630), ISW
000099  000     610 WRITE(NOUT,620)
000100  000     620 FORMAT(52H0* * *  INTEGER INPUT IN FLOATING POINT FIELD  * * * /)
000101  000         GO TO 705
000102  000     625 WRITE (NOUT,626)
000103  000     626 FORMAT (52H0* * *  FLOATING POINT INPUT IN INTEGER FIELD  * * */)
000104  000         GO TO 705
000105  000     630 WRITE(NOUT,635)
000106  000     635 FORMAT(50H0* * *  END FOUND WITHIN NETWORK PARAMETERS  * * * /)
000107  000         IF(ISW .EQ. 5) GO TO 710
000108  000     650 CALL SKPTE(JSW)
000109  000         GO TO 720
000110  000     700 LPAR(LOC) = .TRUE.
000111  000     705 KOL = KOL + 1
000112  000         GO TO 15
```

```
000113      000      710 JSW = 2
000114      000      720 DO 750 I=1,4
000115      000          IF(LPAR(I)) GO TO 750
000116      000          WRITE(NOUT,725) CODE(I)
000117      000      725 FORMAT(8H0* * *  A2, 20H NOT SUPPLIED  * * * /)
000118      000          ERR = .TRUE.
000119      000      750 CONTINUE
000120      000          DO 760 I=13,14
000121      000          IF(LPAR(I)) GO TO 760
000122      000          WRITE(NOUT,725) CODE(I-8)
000123      000          ERR = .TRUE.
000124      000      760 CONTINUE
000125      000          RETURN
000126      000          END

END ELT.


*MOG,P TIMCHK
```

```
*ELT,L TIMCHK
ELTOT7 RL1870 02/28-03:21:14-(0,)
000001       000          SUBROUTINE TIMCHK(RTIME,KODE)
000002       000    C
000003       000          COMMON / FIXCON / CON(1)
000004       000    C
000005       000          EQUIVALENCE (CON(1),TIMEN), (CON(3),TIMEND), (CON(18),OUTPUT)
000006       000    C
000007       000          DATA CTIME1 / 0.0 /
000008       000    C
000009       000          IF(CTIME1 .GT. 0.0) GO TO 100
000010       000          CALL CLOCK(CTIME1)
000011       000          ETIME = 0.0
000012       000          GO TO 200
000013       000      100 CALL CLOCK(CTIME)
000014       000          ETIME = CTIME - CTIME1
000015       000      200 IF(KODE .EQ. 0) GO TO 350
000016       000          CALL LINECK(3)
000017       000          WRITE(6,300) ETIME
000018       000      300 FORMAT(16HOCOMPUTER TIME = F9.3, 8H MINUTES)
000019       000      350 IF(ETIME .LT. RTIME) RETURN
000020       000          CALL LINECK(2)
000021       000          IF(KODE .EQ. 0) GO TO 450
000022       000          WRITE(6,400) RTIME
000023       000      400 FORMAT(67HOEXECUTION TERMINATED BECAUSE COMPUTER TIME EXCEEDS TIME
000024       000         1 REQUESTED, F9.3, 8H MINUTES)
000025       000          GO TO 500
000026       000      450 WRITE(6,475) RTIME
000027       000      475 FORMAT(38HOCOMPUTER TIME EXCEEDS TIME REQUESTED, F9.3, 8H MINUTES)
000028       000      500 TIMEND = TIMEN
000029       000          OUTPUT = 0.0
000030       000          RETURN
000031       000          END

END ELT.


*HOG,P TPOL
```

```
*ELT,L TPOL
ELTOT7 RLIB70 02/28-03:21:15-(4,)
000001      002          FUNCTION TPOL(KODE,X)
000002      000    C
000003      000          DIMENSION RDATA(1), LOC(5), CODE(5)
000004      000    C
000005      000          COMMON /ARRAY / NDATA(1)
000006      000          COMMON /FDATA / CDP, LCP, NCP, RCP, LRO, NRO, RRO
000007      000          COMMON /FDATA / NH , LMU, NMU, RMU, LKT, NKT, RKT
000008      002          COMMON /FDATA / DUM(3),TZERO
000009      000    C
000010      000          EQUIVALENCE (RDATA(1),NDATA(1))
000011      000    C
000012      002          DATA KERR / 0 /
000013      000          DATA NOUT / 6 /
000014      000          DATA LOC  / 5*1 /
000015      000          DATA CODE / 2HCP, 2HRO, 2HMU, 2HKT, 2HH  /
000016      000    C
000017      000    C
000018      002          VAR = X + TZERO
000019      000          GO TO (1,2,3,4,5), KODE
000020      000        1 L = NCP
000021      000          GO TO 10
000022      000        2 L = NRO
000023      000          GO TO 10
000024      000        3 L = NMU
000025      000          GO TO 10
000026      000        4 L = NKT
000027      000          GO TO 10
000028      000        5 L = NH
000029      000    C
000030      004       10 M = 1
000031      000          NP = NDATA(L)
000032      000          K = L + M
000033      000          IF(RDATA(K)-VAR) 20,100,50
000034      000       20 M = M + 2
000035      000          IF(M .GT. NP) GO TO 90
000036      000          DO 30 I=M,NP,2
000037      000          N = K + 2
000038      000          IF (RDATA(N)-VAR) 25,100,80
000039      000       25 K = N
000040      000       30 CONTINUE
000041      000          GO TO 90
000042      000       50 M = M - 2
000043      000          IF(M .LT. 1) GO TO 90
000044      000          DO 60 I=1,M,2
000045      000          K = K - 2
000046      000          IF (RDATA(K)-VAR) 80,100,60
000047      000       60 CONTINUE
000048      000          GO TO 90
000049      000       80 TPOL = RDATA(K+1) + (VAR-RDATA(K))*(RDATA(K+3)-RDATA(K+1))
000050      000         1                                 /(RDATA(K+2)-RDATA(K  ))
000051      000          GO TO 110
000052      003       90  IF(KERR .GT. 10) GO TO 100
000053      003          WRITE (NOUT,95) VAR, CODE(KODE), L, NP, (RDATA(L+I),I=1,NP)
000054      000       95 FORMAT(33H0* * *  THE INDEPENDENT VARIABLE  G13.8, 74H IS LESS THA
000055      000         1N THE FIRST POINT OR GREATER THAN THE LAST POINT ON THE  * * * /
```

```
000056      000        2 8X A2, 25H CURVE STORED AT LOCATION  I9 // 8X I5 // (8X 5G15.8))
000057      002          KERR = KERR+1
000058      000      100 TPOL = RDATA(K+1)
000059      000      110 LOC(KODE) = K - L
000060      000          RETURN
000061      000          END

END ELT.


*HDG,P TPRNT
```

```
@ELT,L TPRNT
ELT07 AL1870 02/28-03:21:17-(0,)
000001      000          SUBROUTINE TPRNT
000002      000    C
000003      000          LOGICAL LSRT, CHK
000004      000    C
000005      000          DIMENSION EXT(1)
000006      000    C
000007      000          COMMON /TEMP   / T(1)
000008      000          COMMON /XSPACE/ NDIM, NTH, NEXT(1)
000009      000          COMMON /FIXCON/ KON(1)
000010      000          COMMON /DIMENS/ NND, NNA, NNT
000011      000          COMMON /POINTR/ LNODE
000012      000    C
000013      000          EQUIVALENCE (NEXT,EXT)
000014      000    C
000015      000          DATA LSRT / .FALSE. /
000016      000          DATA HT   / 1HT /
000017      000    C
000018      000          IF(LNODE .EQ. 0) CALL NNREAD(1)
000019      000          CALL STNDRD
000020      000          IF(LSRT) GO TO 50
000021      000          LSRT = .TRUE.
000022      000          NDIM = NDIM - NNT
000023      000          IF(NDIM .LT. 0) GO TO 100
000024      000          NNODE = NDIM + NTH
000025      000          DO 10 I=1,NNT
000026      000          NEXT(NNODE+I) = I
000027      000       10 CONTINUE
000028      000          DO 30 J=2,NNT
000029      000          K = NNT - J + 1
000030      000          CHK = .TRUE.
000031      000          DO 20 N=1,K
000032      000          NN  = NEXT(NNODE+N  )
000033      000          NN1 = NEXT(NNODE+N+1)
000034      000          IF(NEXT(LNODE+NN) .LE. NEXT(LNODE+NN1)) GO TO 20
000035      000          CHK = .FALSE.
000036      000          NEXT(NNODE+N  ) = NN1
000037      000          NEXT(NNODE+N+1) = NN
000038      000       20 CONTINUE
000039      000          IF(CHK) GO TO 50
000040      000       30 CONTINUE
000041      000       50 IF(NDIM .LT. 12) GO TO 100
000042      000          J = 1
000043      000          L = 6
000044      000          M = NTH + 1
000045      000       60 IF(L .GT. NNT) L = NNT
000046      000          K = M
000047      000          DO 70 I=J,L
000048      000          N = NEXT(NNODE+I)
000049      000          NEXT(K) = NEXT(LNODE+N)
000050      000          EXT(K+1) = T(N)
000051      000          K = K + 2
000052      000       70 CONTINUE
000053      000          K = K - 1
000054      000          IF(KON(28) .LT. 60) GO TO 80
000055      000          CALL TOPLIN
```

```
000056      000          WRITE(6,75)
000057      000       75 FORMAT(1H )
000058      000          KON(28) = KON(28) + 1
000059      000       80 WRITE(6,90) (HT, NEXT(I), EXT(I+1),I=M,K,2)
000060      000       90 FORMAT(6(1X, A1, I6, 1H=, G12.5, 1X))
000061      000          KON(28) = KON(28) + 1
000062      000          IF(L .EQ. NNT) RETURN
000063      000          J = L + 1
000064      000          L = L + 6
000065      000          GO TO 60
000066      000      100 WRITE(6,110) NOIM
000067      000      110 FORMAT(75H0* * *  INSUFFICIENT DYNAMIC STORAGE AVAILABLE FOR SUBRO
000068      000         1UTINE TPRNT, NOIM = I5, 7H  * * *)
000069      000          STOP
000070      000          END

END ELT.


*HDG,P TUBIN
```

```
*ELT,L TUBIN
ELT0T7 RL1870 02/28-03:21:18-(1,)
000001       000          SUBROUTINE TUBIN(KON,JSW)
000002       000    C
000003       000          LOGICAL ERR
000004       000    C
000005       000          DIMENSION KON(1)
000006       000    C
000007       000          COMMON /TAPE   / NIN, NOUT
000008       000          COMMON /CARD   / KRD, KOL, MXKOL
000009       000          COMMON /CIMAGE/ KARD(80)
000010       001           COMMON /FLOERR/ERR
000011       000    C
000012       000       15 CALL SKPB(JSW)
000013       000          GO TO(25,100), JSW
000014       000       25 CALL SUBN(KON(1),ISW)
000015       000          GO TO(30,50,70,80,90), ISW
000016       000       30 KOL = KOL + 1
000017       000          CALL SUBN(KON(2),ISW)
000018       000          GO TO(35,60,70,80,90), ISW
000019       000       35 KOL = KOL + 1
000020       000          CALL SUBN(KON(3),ISW)
000021       000          GO TO (40,60,70,80,90), ISW
000022       000       40 KOL = KOL + 1
000023       000          RETURN
000024       000       50 WRITE(NOUT,55)
000025       000       55 FORMAT(58H0* * *  FLOATING POINT NUMBER INPUT FOR TUBE NUMBER  * *
000026       000         1 * /)
000027       000          GO TO 80
000028       000       60 WRITE(NOUT,65)
000029       000       65 FORMAT(67H0* * *  FLOATING POINT NUMBER INPUT FOR PRESSURE NODE NU
000030       000         1MBER  * * * /)
000031       000          GO TO 80
000032       000       70 WRITE(NOUT,75)
000033       000       75 FORMAT(48H0* * *  END FOUND WITHIN TUBE CONNECTIONS  * * * /)
000034       000       80 ERR = .TRUE.
000035       000          CALL SKPTE(JSW)
000036       000          GO TO(15,100), JSW
000037       000       90 JSW = 2
000038       000      100 RETURN
000039       000          END

END ELT.


*HDG,P VAR
```

```
#ELT,L VAR
ELT077 RLIB70 02/28-03:21:19-(0,)
000001      000          SUBROUTINE VAR
000002      000    C
000003      000          INCLUDE COMM,LIST
000004      000    C
000005      000    C
000006      000          ENTRY CVAR(J1,J2,I)
000007      000          INCLUDE VARC,LIST
000008      000          RETURN
000009      000    C
000010      000    C
000011      000          ENTRY QVAR(J1,J2,I)
000012      000          INCLUDE VARQ,LIST
000013      000          RETURN
000014      000    C
000015      000    C
000016      000          ENTRY GVAR(J1,J2,I,LTA,LG)
000017      000          INCLUDE VARG,LIST
000018      000          RETURN
000019      000    C
000020      000          END

END ELT.


#HOG,P VLVIN
```

```
*ELT,L SIVARC
ELTOT7 RLIB70 02/28-03:21:39-(0,)
000001      000    VARC    PROC
000002      000          IF(FLD(1,1,NSQ1(J1+1)).EQ.0) GO TO 2000
000003      000          NTYPE = FLD(0,5,NSQ2(J2))
000004      000          LA = FLD(5,17,NSQ2(J2))
000005      000          LK = FLD(22,14,NSQ2(J2))
000006      000          GO TO (1005,1010,1015,1020,1025,1030,1035,1040,1045), NTYPE
000007      000     1005 CALL D1D1WM(T(I),A(LA),XK(LK),C(I))
000008      000          GO TO 1999
000009      000     1010 CALL D1D1WM(T(I),A(LA),XK(LK),C1)
000010      000     1012 J2 = J2+1
000011      000          LA = FLD(5,17,NSQ2(J2))
000012      000          LK = FLD(22,14,NSQ2(J2))
000013      000          CALL D1D1WM(T(I),A(LA),XK(LK),C2)
000014      000          GO TO 1998
000015      000     1015 C1 = XK(LK)*XK(LA)
000016      000          GO TO 1012
000017      000     1020 CALL D1D1WM(T(I),A(LA),XK(LK),C1)
000018      000          J2 = J2+1
000019      000          LA = FLD(5,17,NSQ2(J2))
000020      000          LK = FLD(22,14,NSQ2(J2))
000021      000          C2 = XK(LK)*XK(LA)
000022      000          GO TO 1998
000023      000     1025 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C(I))
000024      000          GO TO 1999
000025      000     1030 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C1)
000026      000     1032 J2 = J2+1
000027      000          LA = FLD(5,17,NSQ2(J2))
000028      000          LK = FLD(22,14,NSQ2(J2))
000029      000          CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C2)
000030      000          GO TO 1998
000031      000     1035 C1 = XK(LK)*XK(LA)
000032      000          GO TO 1032
000033      000     1040 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),C1)
000034      000          J2 = J2+1
000035      000          LA = FLD(5,17,NSQ2(J2))
000036      000          LK = FLD(22,14,NSQ2(J2))
000037      000          C2 = XK(LK)*XK(LA)
000038      000          GO TO 1998
000039      000     1045 CALL D2D1WM(T(I),CON(14),A(LA),XK(LK),C(I))
000040      000          GO TO 1999
000041      000     1998 C(I) = C1+C2
000042      000     1999 J2 = J2+1
000043      000     2000 CONTINUE
000044      000     END

END ELT.


*HDG,P SIVARG
```

```
SIVARG                                                                  DATE 022875        PAGE      1

#ELT,L SIVARG
ELTOT7 RLIB70 02/28-03:21:41-(0,)
000001       000    VARG     PROC
000002       000          IF(FLD(2,1,NSQ1(J1)).EQ.0) GO TO 3000
000003       000          NTYPE = FLD(0,5,NSQ2(J2))
000004       000          LA = FLD(5,17,NSQ2(J2))
000005       000          LK = FLD(22,14,NSQ2(J2))
000006       000          GOTO(2005,2010,2015,2020,2025,2030,2035,2040,2045,2050,2055,
000007       000         1     2060,2065,2070,2073,2070) , NTYPE                    VER 6
000008       000     2005 TM = (T(I)+T(LTA))/2.0
000009       000     2007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000010       000          GO TO 2999
000011       000     2010 TM = T(I)
000012       000          GO TO 2007
000013       000     2015 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000014       000     2017 J2 = J2+1
000015       000          LA = FLD(5,17,NSQ2(J2))
000016       000          LK = FLD(22,14,NSQ2(J2))
000017       000          CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000018       000          GO TO 2998
000019       000     2020 G1 = XK(LK)*XK(LA)
000020       000          GO TO 2017
000021       000     2025 CALL D1D1WM(T(I),A(LA),XK(LK),G1)
000022       000          J2 = J2+1
000023       000          LA = FLD(5,17,NSQ2(J2))
000024       000          LK = FLD(22,14,NSQ2(J2))
000025       000          G2 = XK(LK)*XK(LA)
000026       000          GO TO 2998
000027       000     2030 TM = (T(I)+T(LTA))/2.0
000028       000     2032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000029       000          GO TO 2999
000030       000     2035 TM = T(I)
000031       000          GO TO 2032
000032       000     2040 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),G1)
000033       000     2042 J2 = J2+1
000034       000          LA = FLD(5,17,NSQ2(J2))
000035       000          LK = FLD(22,14,NSQ2(J2))
000036       000          CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000037       000          GO TO 2998
000038       000     2045 G1 = XK(LK)*XK(LA)
000039       000          GO TO 2042
000040       000     2050 CALL PLYAWM(A(LA),T(I),A(LA+1),XK(LK),G1)
000041       000          J2 = J2+1
000042       000          LA = FLD(5,17,NSQ2(J2))
000043       000          LK = FLD(22,14,NSQ2(J2))
000044       000          G2 = XK(LK)*XK(LA)
000045       000          GO TO 2998
000046       000     2055 TM = (T(I)+T(LTA))/2.0
000047       000          CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000048       000          GO TO 2999
000049       000     2060 TM = T(LTA)
000050       000          GO TO 2007
000051       000     2065 TM = T(LTA)
000052       000          GO TO 2032
000053       000     2070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                    VER 6
000054       000     2071 TM  = (T(I) + T(LTA)) / 2.0                            VER 6
000055       000           J2 =  J2 + 1                                          VER 6
```

```
000056  000          LA  = FLD(5,17,NS02( J2))                                  VER 6
000057  000          LK  = FLD(22,14,NS02( J2))                                 VER 6
000058  000          IF(NTYPE .EQ. 16)  GO TO 2075                              VER 6
000059  000          CALL D2D1WM( TM, G1, A(LA), XK(LK), G(LG))                 VER 6
000060  000          GO TO 2999                                                 VER 6
000061  000     2073 G1 = XK(LA) *  XK(LK)                                      VER 6
000062  000          GO TO 2071                                                 VER 6
000063  000     2075 G(LG) = G1 * XK(LA)  *  XK(LK)                             VER 6
000064  000          GO  TO 2999                                                VER 6
000065  000     2998 G(LG) = 1./(1./G1+1./G2)
000066  000          IF(FLD(3,1,NS01(J1)).EQ.1) G(LG) = G1*G2
000067  000     2999 J2 = J2+1
000068  000     3000 CONTINUE
000069  000          END
```

END ELT.

#HDG,P SIVARG

```
SIVARQ                                                                  DATE 022875        PAGE     1

@ELT,L SIVARQ
ELTGT7 RL1B70 02/28-03:21:42-(0,)
000001       000    VARQ    PROC
000002       000            IF(FLD(4,1,NSQ1(J1+1)).EQ.0) GO TO 5000
000003       000            NTYPE = FLD(0,5,NSQ2(J2))
000004       000            LA = FLD(5,17,NSQ2(J2))
000005       000            LK = FLD(22,14,NSQ2(J2))
000006       000            GO TO (4005,4010,4015,4020,4025,4030,4035,4040,4030,          VERS   5
000007       000           *       4050,4050,4050),NTYPE                                  VERS   5
000008       000     4005 Q(I) = XK(LK)+Q(I)
000009       000          GO TO 4999
000010       000     4010 Q1 = 0.0
000011       000     4012 CALL D101WM(T(I),A(LA),XK(LK),Q2)
000012       000          GO TO 4998
000013       000     4015 Q1 = 0.0
000014       000     4017 CALL D101WM(CON(1),A(LA),XK(LK),Q2)
000015       000          GO TO 4998
000016       000     4020 CALL D101WM(CON(14),A(LA),XK(LK),Q1)
000017       000     4022 J2 = J2+1
000018       000          LA = FLD(5,17,NSQ2(J2))
000019       000          LK = FLD(22,14,NSQ2(J2))
000020       000          GO TO 4017
000021       000     4025 Q1 = XK(LK)*XK(LA)
000022       000          GO TO 4022
000023       000     4030 CALL D101WM(CON(14),A(LA),XK(LK),Q1)
000024       000          J2 = J2+1
000025       000          LA = FLD(5,17,NSQ2(J2))
000026       000          LK = FLD(22,14,NSQ2(J2))
000027       000          Q2 = XK(LK)*XK(LA)
000028       000          GO TO 4998
000029       000     4035 CALL D101WM(CON(14),A(LA),XK(LK),Q1)
000030       000     4037 J2 = J2+1
000031       000          LA = FLD(5,17,NSQ2(J2))
000032       000          LK = FLD(22,14,NSQ2(J2))
000033       000          GO TO 4012
000034       000     4040 Q1 = XK(LK)-XK(LA)
000035       000          GO TO 4037
000036       000     4050 J2=J2+1                                                       VERS   5
000037       000          JPSLA=FLD(5,17,NSQ2(J2))                                      VERS   5
000038       000          JPSLK=FLD(22,14,NSQ2(J2))                                     VERS   5
000039       000          SPJTIN=CON(14)+XK(JPSLA)*XK(JPSLK)                            VERS   5
000040       000          CALL D1.MCY(XK(JPSLK),SPJTIN,A(LA),XK(LK),Q1)                 VERS   5
000041       000          Q2=0.0                                                        VERS   5
000042       000          GO TO 4998                                                    VERS   5
000043       000     4998 Q(I) = Q1+Q2+Q(I)
000044       000     4999 J2 = J2+1
000045       000     5000 CONTINUE
000046       000     END

END ELT.


@HDG,P SIVAR2
```

```
SIVAR2                                                                  DATE 022875        PAGE     1

#ELT,L SIVAR2
ELTOTT RLI870 02/28-03:21:44-(0,)
000001      000    VAR2    PROC
000002      000         IF(FLD(2,1,NSQ1(JJ1)).EQ.0) GO TO 4000
000003      000         NTYPE = FLD(0,5,NSQ2(JJ2))
000004      000         LA = FLD(5,17,NSQ2(JJ2))
000005      000         LK = FLD(2?,14,NSQ2(JJ2))
000006      000         GOTO(3005,3010,3015,3020,3025,3030,3035,3040,3045,3050,3055,
000007      000        1     3060,3065,3070,3073,3070) , NTYPE                          VER 6
000008      000    3005 TM = (T(L)+T(LTA))/2.0
000009      000    3007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000010      000         GO TO 3999
000011      000    3010 TM = T(L)
000012      000         GO TO 3007
000013      000    3015 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000014      000    3017 JJ2 = JJ2+1
000015      000         LA = FLD(5,17,NSQ2(JJ2))
000016      000         LK = FLD(22,14,NSQ2(JJ2))
000017      000         CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000018      000         GO TO 3998
000019      000    3020 G1 = XK(LK)*XK(LA)
000020      000         GO TO 3017
000021      000    3025 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000022      000         JJ2 = JJ2+1
000023      000         LA = FLD(5,17,NSQ2(JJ2))
000024      000         LK = FLD(22,14,NSQ2(JJ2))
000025      00          G2 = XK(LK)*XK(LA)
000026      0..         GO TO 3998
000027      000    3030 TM = (T(L)+T(LTA))/2.0
000028      000    3032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000029      000         GO TO 3999
000030      000    3035 TM = T(L)
000031      000         GO TO 3032
000032      000    3040 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000033      000    3042 JJ2 = JJ2+1
000034      000         LA = FLD(5,17,NSQ2(JJ2))
000035      000         LK = FLD(22,14,NSQ2(JJ2))
000036      000         CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000037      000         GO TO 3998
000038      000    3045 G1 = XK(LK)*XK(LA)
000039      000         GO TO 3042
000040      000    3050 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000041      000         JJ2 = JJ2+1
000042      000         LA = FLD(5,17,NSQ2(JJ2))
000043      000         LK = FLD(22,14,NSQ2(JJ2))
000044      000         G2 = XK(LK)*XK(LA)
000045      000         GO TO 3998
000046      000    3055 TM = (T(L)+T(LTA))/2.0
000047      000         CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000048      000         GO TO 3999
000049      000    3060 TM = T(LTA)
000050      000         GO TO 3007
000051      000    3065 TM = T(LTA)
000052      000         GO TO 3032
000053      000    3070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                          VER 6
000054      000    3071 TM  = (T(L) + T(LTA)) / 2.0                                   VER 6
000055      000         JJ2 = JJ2 + 1                                                 VER 6
```

SIVAR2 DATE 022875 PAGE 2

```
000056  000        LA  = FLD(5,17,NSQ2(JJ2))                          VER 6
000057  000        LK  = FLD(22,17,NSQ2(JJ2))                         VER 6
000058  000        IF(NTYPE .EQ.16) GO TO 3075                        VER 6
000059  000        CALL D2D1WM(TM,  G1,  A(LA), XK(LK), G(LG))        VER 6
000060  000        GO TO  3999                                        VER 6
000061  000   3073 G1 = XK(LA) * XK(LK)                               VER 6
000062  000        GO TO 3071                                         VER 6
000063  000   3075 G(LG) = G1 * XK(LA) * XK(LK)                       VER 6
000064  000        GO TO 3999                                         VER 6
000065  000   3998 G(LG) = 1./(1./G1+1./G2)
000066  000        IF(FLD(3,1,NSQ1(JJ1)).EQ.1) G(LG) = G1*G2
000067  000   3999 JJ2 = JJ2+1
000068  000   4000 CONTINUE
000069  000   END
```

END ELT.

*HDG,P SIVAR2

```
*ELT,L VLVIN
ELTOT7 RLIB70 02/28-03:21:21-(2,)
000001      000          SUBROUTINE VLVIN(L,VALVE,XI,JSW)
000002      001           LOGICAL ERR
000003      000    C
000004      000          DIMENSION VALVE(1)
000005      000    C
000006      000          COMMON /TAPE  / NIN, NOUT
000007      000          COMMON /CARD  / KRD, KOL, MXKOL
000008      000          COMMON /CIMAGE/ KARD(80)
000009      001           COMMON /FLOERR/ERR
000010      000    C
000011      000          L = 0
000012      000       15 CALL SKPB(JSW)
000013      000          GO TO(50,290), JSW
000014      000       50 DO 75 I=1,3
000015      000          K = I - 1
000016      000          CALL SUBN(VALVE(I),ISW)
000017      000          GO TO(70,55,261,275,262), ISW
000018      000       55 WRITE(NOUT,60)
000019      000       60 FORMAT(60H0* * *  FLOATING POINT NUMBER INPUT FOR INTEGER FIELD  *
000020      000         1 * * /)
000021      000          GO TO 275
000022      000       70 KOL = KOL + 1
000023      000       75 CONTINUE
000024      000          CALL SUBN(XI,ISW)
000025      000          K = 3
000026      000          GO TO(100,110,262,275,262), ISW
000027      000      100 WRITE(NOUT,105)
000028      000      105 FORMAT(78H0* * *  INITIAL VALVE POSITION MUST BE INPUT AS A FLOATI
000029      000         ING POINT NUMBER  * * * /)
000030      000          GO TO 275
000031      000      110 IF(XI .LE. 0.0) GO TO 115
000032      000          IF(XI .LT. 1.0) GO TO 130
000033      000      115 WRITE(NOUT,120) XI
000034      000      120 FORMAT(72H0* * *  INITIAL VALVE POSITION MUST BE WITHIN THE RANGE
000035      000         0.0 TO 1.0, XI = G10.5, 7H  * * * /)
000036      000          GO TO 275
000037      000      130 KOL = KOL + 1
000038      000    C MODE
000039      000          CALL SUBN(VALVE(4),ISW)
000040      000          K = 4
000041      000          GO TO(150,135,262,275,262), ISW
000042      000      135 WRITE(NOUT,140)
000043      000      140 FORMAT(54H0* * *  MODE MUST BE INPUT AS AN INTEGER NUMBER  * * */)
000044      000          GO TO 275
000045      000      150 KOL = KOL + 1
000046      000    C XMINI
000047      000          CALL SUBN(VALVE(5),ISW)
000048      000          K = 5
000049      000          GO TO(155,165,262,275,262), ISW
000050      000      155 WRITE(NOUT,160)
000051      000      160 FORMAT(61H0* * *  XMINI MUST BE INPUT AS A FLOATING POINT NUMBER
000052      000         1* * * /)
000053      000          GO TO 275
000054      000      165 IF(VALVE(5) .LE. 0.0) GO TO 170
000055      000          IF(VALVE(5) .LE. 1.0) GO TO 180
```

```
000056  000     170 WRITE(NOUT,175) VALVE(5)
000057  000     175 FORMAT(58H0* * *  XMIN1 MUST BE WITHIN THE RANGE 0.0 TO 1.0, XMIN1
000058  000        1 = G10.5, 7H  * * * /)
000059  000         GO TO 275
000060  000     180 KOL = KOL + 1
000061  000   C XMAX1
000062  000         CALL SUBN(VALVE(6),ISW)
000063  000         K = 6
000064  000         GO TO(185,195,262,275,262), ISW
000065  000     185 WRITE(NOUT,190)
000066  000     190 FORMAT(61H0* * *  XMAX1 MUST BE INPUT AS A FLOATING POINT NUMBER
000067  000        1* * * /)
000068  000         GO TO 275
000069  000     195 IF(VALVE(6) .LE. 0.0) GO TO 200
000070  000         IF(VALVE(6) .LT. 1.0) GO TO 210
000071  000     200 WRITE(NOUT,205) VALVE(6)
000072  000     205 FORMAT(58H0* * *  XMAX1 MUST BE WITHIN THE RANGE 0.0 TO 1.0, XMAX1
000073  000        1 = G10.5, 7H  * * * /)
000074  000         GO TO 275
000075  000     210 KOL = KOL + 1
000076  000   C E
000077  000         CALL SUBN(VALVE(7),ISW)
000078  000         K = 7
000079  000         GO TO(215,225,262,275,262), ISW
000080  000     215 IF(.NOT. ABS(VALVE(7)) .GT. 0.0) GO TO 225
000081  000         WRITE(NOUT,220)
000082  000     220 FORMAT(57H0* * *  E MUST BE INPUT AS A FLOATING POINT NUMBER  * *
000083  000        1* /)
000084  000         GO TO 275
000085  000     225 KOL = KOL + 1
000086  000         DO 250 I=8,16
000087  000         CALL SUBN(VALVE(I),ISW)
000088  000         GO TO(230,240,260,275,260), ISW
000089  000     230 IF(I .LT. 10) GO TO 240
000090  000         K = I + 1
000091  000         WRITE(NOUT,235) K
000092  000     235 FORMAT(13H0* * *  ENTRY 13, 48H MUST BE INPUT AS A FLOATING POINT
000093  000        1NUMBER  * * * /)
000094  000         GO TO 275
000095  000     240 KOL = KOL + 1
000096  000         L = I
000097  000     250 CONTINUE
000098  002         GO TO 270
000099  000     260 IF(L .EQ. 10 .OR. L .EQ. 12) GO TO 270
000100  000     261 K = I - 1
000101  000     262 WRITE(NOUT,265) K
000102  000     265 FORMAT(56H0* * *  INCORRECT NUMBER OF ENTRIES FOR VALVE DATA, IC =
000103  000        1 I3, 7H  * * * /)
000104  001     275  ERR=.TRUE.
000105  000     270 IF(ISW .EQ. 5) GO TO 280
000106  001          CALL SKPTE(JSW)
000107  000         GO TO(276,290), JSW
000108  000     276 IF(L .EQ. 0) GO TO 15
000109  000         RETURN
000110  000     280 JSW = 2
000111  000     290 RETURN
000112  000         END
```

```
#ELT,L SIVRG2
ELTOTT RL1870 02/28-03:21:45-(0,)
000001      000   VRG2    PROC
000002      000         IF(FLD(2,1,NSQ1(JJ1)).EQ.0) GO TO 4000
000003      000         NTYPE = FLD(0,5,NSQ2(JJ2))
000004      000         LA = FLD(5,17,NSQ2(JJ2))
000005      000         LK = FLD(22,14,NSQ2(JJ2))
000006      000         GOTO(3005,3010,3015,3020,3025,3030,3035,3040,3045,3050,3055,
000007      000        1      3060,3065,3070,3073,3070) , NTYPE                          VER 6
000008      000    3005 TM = (T(L)+T(LTA))/2.0
000009      000    3007 CALL D1D1WM(TM,A(LA),XK(LK),G(LG))
000010      000         GO TO 3999
000011      000    3010 TM = T(L)
000012      000         GO TO 3007
000013      000    3015 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000014      000    3017 JJ2 = JJ2+1
000015      000         LA = FLD(5,17,NSQ2(JJ2))
000016      000         LK = FLD(22,14,NSQ2(JJ2))
000017      000         CALL D1D1WM(T(LTA),A(LA),XK(LK),G2)
000018      000         GO TO 3998
000019      000    3020 G1 = XK(LK)*XK(LA)
000020      000         GO TO 3017
000021      000    3025 CALL D1D1WM(T(L),A(LA),XK(LK),G1)
000022      000         JJ2 = JJ2+1
000023      000         LA = FLD(5,17,NSQ2(JJ2))
000024      000         LK = FLD(22,14,NSQ2(JJ2))
000025      000         G2 = XK(LK)*XK(LA)
000026      000         GO TO 3998
000027      000    3030 TM = (T(L)+T(LTA))/2.0
000028      000    3032 CALL PLYAWM(A(LA),TM,A(LA+1),XK(LK),G(LG))
000029      000         GO TO 3999
000030      000    3035 TM = T(L)
000031      000         GO TO 3032
000032      000    3040 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000033      000    3042 JJ2 = JJ2+1
000034      000         LA = FLD(5,17,NSQ2(JJ2))
000035      000         LK = FLD(22,14,NSQ2(JJ2))
000036      000         CALL PLYAWM(A(LA),T(LTA),A(LA+1),XK(LK),G2)
000037      000         GO TO 3998
000038      000    3045 G1 = XK(LK)*XK(LA)
000039      000         GO TO 3042
000040      000    3050 CALL PLYAWM(A(LA),T(L),A(LA+1),XK(LK),G1)
000041      000         JJ2 = JJ2+1
000042      000         LA = FLD(5,17,NSQ2(JJ2))
000043      000         LK = FLD(22,14,NSQ2(JJ2))
000044      000         G2 = XK(LK)*XK(LA)
000045      000         GO TO 3998
000046      000    3055 TM = (T(L)+T(LTA))/2.0
000047      000         CALL D2D1WM(TM,CON(14),A(LA),XK(LK),G(LG))
000048      000         GO TO 3999
000049      000    3060 TM = T(LTA)
000050      000         GO TO 3007
000051      000    3065 TM = T(LTA)
000052      000         GO TO 3032
000053      000    3070 CALL D1D1WM(CON(14),A(LA),XK(LK),G1)                           VER 6
000054      000    3071 TM  = (T(L) + T(LTA)) / 2.0                                    VER 6
000055      000         JJ2 = JJ2 + 1                                                  VER 6
```

```
000056  000        LA  = FLD(5,17,NSQ2(JJ2))                          VER 6
000057  000        LK  = FLD(22,14,NSQ2(JJ2))                         VER 6
000058  000        IF(NTYPE .EQ.16) GO TO 3075                        VER 6
000059  000        CALL D2D1WM(TM,  G1,  A(LA), XK(LK), G(LG))        VER 6
000060  000        GO TO  3999                                        VER 6
000061  000    3073 G1 = XK(LA) * XK(LK)                              VER 6
000062  000        GO TO 3071                                         VER 6
000063  000    3075 G(LG) = G1 * XK(LA) * XK(LK)                      VER 6
000064  000        GO TO 3999                                         VER 6
000065  000    3998 G(LG) = 1./(1./G1+1./G2)
000066  000        IF(FLD(3,1,NSQ1(JJ1)).EQ.1) G(LG) = G1*G2
000067  000    3999 JJ2 = JJ2+1
000068  000    4000 CONTINUE
000069  000    END

END ELT.


@ADD,P SIVRG2
```

```
SIVRQ2                                                                  DATE 022875        PAGE     1

@ELT,L SIVRQ2
ELTOTT KLIB:U 02/28-03:21:47-(0,)
000001      000    VRQ2     PROC
000002      000          IF(FLD(4,1,NSQ1(JJ1+1)).EQ.0) GO TO 6000
000003      000          NTYPE = FLD(0,5,NSQ2(JJ2))
000004      000          LA = FLD(5,17,NSQ2(JJ2))
000005      000          LK = FLD(22,14,NSQ2(JJ2))
000006      000          GO TO (5005,5010,5015,5020,5025,5030,5035,5040,5030,          VERS   5
000007      000         *       5050,5050,5050),NTYPE                                  VERS   5
000008      000     5005 Q(L) = XK(LK)+Q(L)
000009      000          GO TO 5999
000010      000     5010 Q1 = 0.0
000011      000     5012 CALL D1D1WM(T(L),A(LA),XK(LK),Q2)
000012      000          GO TO 5998
000013      000     5015 Q1 = 0.0
000014      000     5017 CALL D1D1WM(CON(14),A(LA),XK(LK),Q2)
000015      000          GO TO 5998
000016      000     5020 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000017      000     5022 JJ2 = JJ2+1
000018      000          LA = FLD(5,17,NSQ2(JJ2))
000019      000          LK = FLD(22,14,NSQ2(JJ2))
000020      000          GO TO 5017
000021      000     5025 Q1 = XK(LK)*XK(LA)
000022      000          GO TO 5022
000023      000     5030 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000024      000          JJ2 = JJ2+1
000025      000          LA = FLD(5,17,NSQ2(JJ2))
000026      000          LK = FLD(22,14,NSQ2(JJ2))
000027      000          Q2 = XK(LK)*XK(LA)
000028      000          GO TO 5998
000029      000     5035 CALL D1D1WM(CON(14),A(LA),XK(LK),Q1)
000030      000     5037 JJ2 = JJ2+1
000031      000          LA = FLD(5,17,NSQ2(JJ2))
000032      000          LK = FLD(22,14,NSQ2(JJ2))
000033      000          GO TO 5012
000034      000     5040 Q1 = XK(LK)*XK(LA)
000035      000          GO TO 5037
000036      000     5050 JJ2=JJ2+1                                                    VERS   5
000037      000          JPSLA=FLD(5,17,NSQ2(JJ2))                                 VERS   5
000038      000          JPSLK=FLD(22,14,NSQ2(JJ2))                                VERS   5
000039      000          SPJTIM=CON(14)+XK(JPSLA)*XK(JPSLK)                        VERS   5
000040      000          CALL D11MCV(XK(JPSLK),SPJTIM,A(LA),XK(LK),Q1)             VERS   5
000041      000          Q2=0.0                                                    VERS   5
000042      000          GO TO 5998                                                VERS   5
000043      000     5998 Q(L) = Q1+Q2+Q(L)
000044      000     5999 JJ2 = JJ2+1
000045      000     6000 CONTINUE
000046      000     END

END ELT.


@BRKPT PRINTS
```

```
*ELT,L UPSUM1
ELTOT7 RL1870 02/28-03:21:22-(1,)
000001      000          SUBROUTINE UPSUM1(NFRM,LOC1,J1,SUM1,SUM2)
000002      000    C
000003      000          LOGICAL ERR
000004      000    C
000005      000          COMMON /FLODAT/ NFLOW(1)
000006      000    C
000007      000          DATA NOUT / 6 /
000008      000    C
000009      000    C
000010      000          IF(J1 .GT. 4) GO TO 20
000011      000          NTB = NFLOW(LOC1+J1)
000012      000          WRITE(NOUT,10) NTB
000013      000       10 FORMAT(40H0* * *  NO UPSTREAM FLUID COMP FOR TUBE  I6,7H  * * * /)
000014      000          ERR = .TRUE.
000015      000          GO TO 700
000016      000       20 JJ1 = J1
000017      000      100 JJ1 = JJ1 - 4
000018      000          IF(JJ1 .LT. 4) GO TO 700
000019      000          L = LOC1 + JJ1
000020      000          IF(NFLOW(L+2) .NE. NFRM) GO TO 100
000021      000          LOCO = NFLOW(L+3)
000022      000          IF(LOCO) 120,100,500
000023      000      120 LOC2 = -LOCO
000024      000          J2   = NFLOW(LOC2) + 1
000025      000          CALL DNSUM2(NFRM,LOC2,J2,SUM1,SUM2,N2)
000026      000          GO TO 100
000027      000    C
000028      000      500 CALL FLOSUM(NFLOW(L),LOCO,SUM1,SUM2)
000029      000          GO TO 100
000030      000    C
000031      000      700 RETURN
000032      000          END

END ELT.


*HDG,P UPSUM2
```

```
*ELT,L UPSUM2
ELTOT7 RLIB70 02/28-03:21:23-(1,)
000001       000          SUBROUTINE UPSUM2(NFRM,LOC1,J1,LOC2,J2,SUM1,SUM2)
000002       000    C
000003       000    C
000004       000          IF(J2 .LE. 4) GO TO 20
000005       000          CALL DNSUM2(NFRM,LOC2,J2,SUM1,SUM2,N2)
000006       000          IF (NFRM .NE. N2) GO TO 700
000007       000    C
000008       000       20 CALL UPSUM1(NFRM,LOC1,J1,SUM1,SUM2)
000009       000    C
000010       000      700 RETURN
000011       000          END

END ELT.


*HDG,P UPSUM3
```

```
@ELT,L UPSUM3
ELTOT7 RL1870 02/28-03:21:24-(1,)
000001      000          SUBROUTINE UPSUM3(NFRM,LOC1,J1,LOC2,J2,LOC3,J3,SUM1,SUM2)
000002      000    C
000003      000    C
000004      000          IF(J3 .LE. 4) GO TO 20
000005      000          CALL DNSUM3(NFRM,LOC3,J3,SUM1,SUM2,N3)
000006      000          IF (NFRM .NE. N3) GO TO 700
000007      000    C
000008      000       20 CALL UPSUM2(NFRM,LOC1,J1,LOC2,J2,SUM1,SUM2)
000009      000    C
000010      000      700 RETURN
000011      000          END

END ELT.


@HDG,P WLKBCK
```

WLKBCK

DATE 022875 PAGE 1

```
@ELT,L WLKBCK
ELT0T7 RLIB70 02/28-03:21:25-(0,)
000001      000       SUBROUTINE WLKBCK
000002      000       X = X
000003      000       RETURN
000004      000       END
```

END ELT.

@HOG,P WPRINT

D-267

```
@ELT,L WPRINT
ELT077 RL1870 02/28-03:21:26-(1,)
000001      001          SUBROUTINE WPRINT(K1,K2,K3,K4)
000002      000    C
000003      000          COMMON /FDIMNS/ NTYPE, NSYS, NTB, NP, NV
000004      000          COMMON /WDOT  / W (1)
000005      000          COMMON /PRESS / P (1)
000006      000          COMMON /BELTAP/ DP(1)
000007      000          COMMON /VALVP / VP(1)
000008      000          COMMON /POINTN/ LNODE, LCOND, LCONS, LARRY, ICOMP, LNTB, LNP, LV
000009      000    C
000010      000          DATA HW, HD, HP, HV / 1HW, 2HDP, 1HP, 2HVP /
000011      000    C
000012      000    C
000013      000          IF(LNTB .EQ. 0) CALL NNREAD(5)
000014      001          IF(K1.NE.0) CALL PRINTW(LNTB,W ,NTB,HW)
000015      001          IF(K2.NE.0) CALL PRINTW(LNTB,DP,NTB,HD)
000016      001          IF(K3.NE.0) CALL PRINTW(LNP ,P ,NP ,HP)
000017      001          IF(K4.NE.0) CALL PRINTW(LV  ,VP,NV ,HV)
000018      000          RETURN
000019      000          END

END ELT.


@HDG,P SICOUM
```