science for a changing world

Techniques of Water-Resources Investigations of the United States Geological Survey

Chapter A5

A MODULAR FINITE-ELEMENT MODEL (MODFE) FOR AREAL AND AXISYMMETRIC GROUND-WATER-FLOW PROBLEMS, PART 3: DESIGN PHILOSOPHY AND PROGRAMMING DETAILS

By Lynn J. Torak

Book 6

MODELING TECHNIQUES

List of Subroutines

```
      SUBROUTINE BAND(A,AT,B,JPT,IN,IBND)
C        SOLVES MATRIX EQUATION FOR HEAD CHANGE VECTOR, B, BY USING
C        TRIANGULAR DECOMPOSITION
      DIMENSION A(1),AT(1),B(1)
      DIMENSION JPT(1),IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
C *** DEFINE INDICES ***
      N=NNDS-NHDS
      MR=MBWC*N+1
      NR=IBND*(N-IBND)+(IBND*(IBND+1))/2
      IF(MR.GT.NR) NR=MR-1
      MBM1=MBWC-1
      NP=MBM1*NEQ
      NN=NEQ+1
      IBM1=IBND-1
C *** TRANSFORM CONDENSED COEFFICIENT MATRIX, A, INTO ROTATED
C     BAND-MATRIX, A ***
      DO 110 I=1,NEQ
      NP=NP-MBM1
      NN=NN-1
      K=IN(NN)
      IF(K.LT.0) GO TO 110
      IRBW=MIN0(IBND,N-K+1)
      DO 90 J=1,IRBW
   90 AT(J)=0.
      MR=MR-MBWC
      IF(A(MR).GT.1.E-20) GO TO 93
      A(MR)=1.E30
      B(K)=0.
   93 AT(1)=A(MR)
      DO 100 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 100,103,95
   95 AT(M-K+1)=A(MR+J)
  100 CONTINUE
  103 NR=NR-IRBW
      DO 105 J=1,IRBW
  105 A(NR+J)=AT(J)
  110 CONTINUE
C *** FACTOR A INTO PRODUCT OF UPPER TRIANGULAR, DIAGONAL, AND LOWER
C     TRIANGULAR MATRICES, ALL KEPT IN A ***
      NM1=N-1
      ID=NR+1
      DO 140 I=1,NM1
      C1=1./A(ID)
      LD=ID
      L=I
      IRBW=MIN0(IBM1,N-I)
      DO 130 J=1,IRBW
      L=L+1
      LD=LD+MIN0(IBND,N-L+2)
      IB=ID+J
```

```
            IF(A(IB)) 115,130,115
  115 C=A(IB)*C1
      LB=LD-1
      DO 120 K=J,IRBW
            LB=LB+1
      IF(A(ID+K)) 119,120,119
  119 A(LB)=A(LB)-C*A(ID+K)
  120 CONTINUE
      A(IB)=C
  130 CONTINUE
  140 ID=ID+IRBW+1
C *** REDUCE RIGHT-HAND SIDE, B ***
      ID=NR+1
      DO 160 I=1,NM1
      C=B(I)
      B(I)=C/A(ID)
      IRBW=MIN0(IBM1,N-I)
      DO 150 L=1,IRBW
      K=I+L
  150 B(K)=B(K)-A(ID+L)*C
  160 ID=ID+IRBW+1
C *** BACK-SOLVE FOR HEAD CHANGE, B ***
      B(N)=B(N)/A(ID)
      DO 180 I=1,NM1
      IRBW=MIN0(IBM1,I)
      ID=ID-IRBW-1
      L=N-I
      SUM=0.
      DO 170 J=1,IRBW
  170 SUM=SUM-A(ID+J)*B(L+J)
  180 B(L)=B(L)+SUM
      RETURN
      END
```

```
      SUBROUTINE CBADEQ(A,B,CH,CBQ,GMA,IN)
C        ADDS COEFFICIENTS FORMED IN CBFMEQ INTO MATRIX DIAGONAL AND
C        RIGHT-HAND-SIDE VECTOR
      DIMENSION A(1),B(1),CH(1),CBQ(1),GMA(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      L=0
      DO 10 I=1,NEQ
      IF(GMA(I).LT.1.E-30) GO TO 10
      L=L+1
      K=IN(I)
      IF(K.LT.0) GO TO 10
      NME=MBWC*(K-1)+1
      A(NME)=A(NME)+CH(L)
      B(K)=B(K)+CBQ(L)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CBADWT(DH,A,B,CH,CBQ,GMA,IN)
C        ADDS COEFFICIENTS FORMED IN CBFMEQ INTO MATRIX DIAGONAL AND
C        RIGHT-HAND-SIDE VECTOR FOR WATER-TABLE CORRECTOR STEP
      DIMENSION DH(1),A(1),B(1),CH(1),CBQ(1),GMA(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      L=0
      DO 10 I=1,NEQ
      IF(GMA(I).LT.1.E-30) GO TO 10
      L=L+1
      K=IN(I)
      IF(K.LT.0) GO TO 10
      NME=MBWC*(K-1)+1
      A(NME)=A(NME)+CH(L)
      B(K)=B(K)+CBQ(L)-CH(L)*DH(K)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CBCHG(HR,DHR,TIME,ISTP)
C        READS AND COMPUTES NEW VALUES FOR SOURCE-BED HEAD FOR TRANSIENT
C        LEAKAGE
      DIMENSION HR(1),DHR(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/CHG/NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (2I5)
    4 FORMAT (1H1,10X,40HCHANGES IN VALUES OF SOURCE-BED HEAD FOR/1H
     1,2X,57HLEAKAGE INVOLVING TRANSIENT EFFECTS FROM AQUITARD STORAGE
     2/1H ,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5
     3,11H TIME UNITS/1H ,35X,10HSOURCE-BED/1H ,19X,4HNODE,15X,4HHEAD)
    6 FORMAT (1H ,17X,I5,13X,G11.5)
    8 FORMAT (I5,F10.0)
C *** READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C     MADE AGAIN ***
      READ(IIN,2) N,NCBCH
      WRITE(IOUT,4) ISTP,TIME
      WF=2./3.
      DO 30 I=1,N
C *** READ AND WRITE NEW VALUES ***
      READ(IIN,8) J,HRJ
      WRITE(IOUT,6) J,HRJ
C *** FORM HEAD-CHANGE AND NEW-HEAD VECTORS ***
      DHR(J)=HRJ-HR(J)
   30 HR(J)=WF*DHR(J)+HR(J)
      RETURN
      END
```

```
      SUBROUTINE CBFMCO(A,AR,GMA,ALF,AC,BTA,BC,WVCN,ND,NCBZ)
C        FORMS COEFFICIENTS FOR TRANSIENT LEAKAGE
      DIMENSION A(1),AR(1),GMA(1),ALF(1),AC(1),BTA(1),BC(1),WVCN(1)
      DIMENSION ND(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (1H0,15X,40HPARAMETERS FOR TRANSIENT LEAKAGE BY ZONE/1H
     1,10X,5HFIRST,5X,6HNO. OF,4X,18HAQUITARD HYDRAULIC,3X
     2,17HAQUITARD SPECIFIC/1H ,6H  ZONE,3X,7HEL. NO.,3X,8HELEMENTS,6X
     3,12HCONDUCTIVITY,11X,7HSTORAGE)
    4 FORMAT (3I5,2F10.0)
    6 FORMAT (1H ,I5,4X,I5,5X,I5,10X,G11.5,8X,G11.5)
C *** DEFINE COEFFICIENTS FOR SERIES APPROXIMATIONS ***
      ALF(1)=13.656
      ALF(2)=436.53
      ALF(3)=49538.
      AC(1)=.26484
      AC(2)=.060019
      AC(3)=.008474
      BTA(1)=10.764
      BTA(2)=19.805
      BC(1)=-.25754
      BC(2)=.090873
      WRITE(IOUT,2)
C *** INITIALIZE NODAL EFFECTIVE AQUITARD HYDRAULIC CONDUCTIVITY ***
      DO 20 I=1,NNDS
   20 WVCN(I)=0.
C *** BEGIN ZONAL LOOP ***
      DO 50 I=1,NCBZ
C *** READ AND WRITE ZONAL DATA ***
      READ(IIN,4) L,NBE,NO,VCON,SPST
      WRITE(IOUT,6) L,NBE,NO,VCON,SPST
C *** BEGIN ELEMENT LOOP WITHIN ZONE ***
      NEND=NBE+NO-1
      NDC=4*(NBE-1)+1
      NTE=2*(NBE-1)
      DO 40 J=NBE,NEND
C *** BEGIN NODAL LOOP WITHIN ELEMENT ***
      NE=1
      IF(ND(NDC+3).GT.0) NE=2
      DO 30 IE=1,NE
C *** COMPUTE EFFECTIVE AQUITARD HYDRAULIC CONDUCTIVITY AND EFFECTIVE
C     AQUITARD SPECIFIC STORAGE ***
      NA=ND(NDC)
      NB=ND(NDC+IE)
      NC=ND(NDC+IE+1)
      TEVC=VCON*AR(NTE+IE)
      WVCN(NA)=WVCN(NA)+TEVC
      WVCN(NB)=WVCN(NB)+TEVC
      WVCN(NC)=WVCN(NC)+TEVC
      TESA=SPST*AR(NTE+IE)
      GMA(NA)=GMA(NA)+TESA
      GMA(NB)=GMA(NB)+TESA
      GMA(NC)=GMA(NC)+TESA
```

```
   30 CONTINUE
      NTE=NTE+2
   40 NDC=NDC+4
   50 CONTINUE
C *** COMPUTE GAMMA ***
      IW=MBWC+1
      NVL=2
      DO 60 I=1,NNDS
      IF(GMA(I).GT.1.E-30) GMA(I)=A(NVL)*A(NVL)/(GMA(I)*WVCN(I))
   60 NVL=NVL+IW
      RETURN
      END
```

```
      SUBROUTINE CBFMEQ(A,CH,CBQ,DHR,CBTQ,GMA,ALF,AC,BTA,BC,DT,IN)
C        FORMS COEFFICIENTS FOR TRANSIENT LEAKAGE
      DIMENSION A(1),CH(1),CBQ(1),DHR(1),CBTQ(1),GMA(1),ALF(1),AC(1)
     1,BTA(1),BC(1)
      DIMENSION IN(1)
      DIMENSION QOM1(3),QOM2(2)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      IW=MBWC+1
      NVL=2
      NQ=0
      L=0
      DO 50 I=1,NNDS
      IF(GMA(I).LT.1.E-30) GO TO 50
      DTD=GMA(I)*DT
C *** COMPUTE M1 SERIES AND EXPONENTIALLY REDUCE TRANSIENT-LEAKAGE
C     FLUX DUE TO TIME VARIANCE OF AQUIFER HEAD ***
      SM1=0.
      DO 30 NT=1,3
      XP=0.
      TMPA=ALF(NT)*DTD
      IF(TMPA.LT.50.) XP=EXP(-TMPA)
      SM1=SM1+AC(NT)*(1.-XP)
      NQ=NQ+1
      QOM1(NT)=CBTQ(NQ)
      CBTQ(NQ)=XP*CBTQ(NQ)
   30 CONTINUE
C *** COMPUTE NEW TRANSIENT-LEAKAGE FLUX DUE TO TIME VARIANCE OF
C     SOURCE-BED HEAD ***
      DO 40 NT=1,2
      XP=0.
      TMPA=BTA(NT)*DTD
      IF(TMPA.LT.50.) XP=EXP(-TMPA)
      NQ=NQ+1
      QOM2(NT)=CBTQ(NQ)
      CBTQ(NQ)=XP*CBTQ(NQ)+BC(NT)*(1.-XP)*DHR(I)/DTD
   40 CONTINUE
C *** COMPUTE COEFFICIENTS TO ADD INTO MATRIX DIAGONAL AND RIGHT-HAND-
C     SIDE VECTOR ***
      L=L+1
      CH(L)=SM1*A(NVL)/DTD
      CBQ(L)=(2.*(CBTQ(NQ-1)+CBTQ(NQ)-CBTQ(NQ-4)-CBTQ(NQ-3)-CBTQ(NQ-2))
     1+QOM2(1)+QOM2(2)-QOM1(1)-QOM1(2)-QOM1(3))*A(NVL)/3.
   50 NVL=NVL+IW
      RETURN
      END
```

```
      SUBROUTINE CBHRXT(HR,DHR)
C        EXTRAPOLATES SOURCE-BED HEADS AT ALL NODES FROM THE MEAN POINT
C        IN THE TIME STEP TO THE END OF THE TIME STEP, THEN ZEROS THE
C        HEAD CHANGES
      DIMENSION HR(1),DHR(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      TMPA=1./3.
      DO 10 I=1,NNDS
      HR(I)=TMPA*DHR(I)+HR(I)
      DHR(I)=0.
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CBINIT(G,ICHA,ICQA,IDHRA,ICTQA,IGMA,IALFA,IACA,IBTA
     1,IBCA,NCBZ)
C        READS PROBLEM SPECIFICATION, DEFINES AND INITIALIZES VARIABLES
C        FOR TRANSIENT LEAKAGE ROUTINES
      DIMENSION G(1)
      COMMON/GDIM/ISUM
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (1H0,40HTRANSIENT LEAKAGE FROM AQUITARD STORAGE:/1H
     1,38HNOW G MUST BE DIMENSIONED TO AT LEAST ,I6)
    4 FORMAT (2I5)
    6 FORMAT (1H0,54HNO. OF AQUITARD PROPERTY ZONES (NCBZ)..............
     1 = ,I5/
     $1H ,54HMAXIMUM NUMBER OF TRANSIENT LEAKAGE NODES (MCBN)... = ,I5)
C *** READ AND WRITE NUMBER OF AQUITARD PROPERTY ZONES AND TRANSIENT
C     LEAKAGE NODES ***
      READ(IIN,4) NCBZ,MCBN
      WRITE(IOUT,6) NCBZ,MCBN
C *** DEFINE ADDRESSES OF NEW ARRAYS WITHIN G ***
      ICHA=ISUM
      ISUM=ISUM+MCBN
      ICQA=ISUM
      ISUM=ISUM+MCBN
      IDHRA=ISUM
      ISUM=ISUM+NNDS
      ICTQA=ISUM
      ISUM=ISUM+5*MCBN
      IGMA=ISUM
      ISUM=ISUM+NNDS
      IALFA=ISUM
      ISUM=ISUM+3
      IACA=ISUM
      ISUM=ISUM+3
      IBTA=ISUM
      ISUM=ISUM+2
      IBCA=ISUM
      ISUM=ISUM+2
C *** WRITE NEW SIZE OF G ***
      WRITE(IOUT,2) ISUM
C *** INITIALIZE NEW ELEMENTS OF G ***
      DO 10 I=ICHA,ISUM
   10 G(I)=0.
      RETURN
      END
```

```
      SUBROUTINE CBTQC(B,DHB,CBTQ,GMA,ALF,AC,DT,IN)
C        COMPUTES NEW TRANSIENT-LEAKAGE FLUX DUE TO TIME VARIANCE OF
C        AQUIFER HEAD
      DIMENSION B(1),DHB(1),CBTQ(1),GMA(1),ALF(1),AC(1)
      DIMENSION IN(1)
      DIMENSION XP(3)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      DTM=3./(2.*DT)
      NQ=-4
      DO 30 I=1,NNDS
      IF(GMA(I).LT.1.E-30) GO TO 30
C *** COMPUTE EXPONENTIAL TERMS FOR M1 SERIES ***
      DTD=GMA(I)*DT
      DO 10 NT=1,3
      XP(NT)=0.
      TMPA=ALF(NT)*DTD
      IF(TMPA.LT.50.) XP(NT)=EXP(-TMPA)
   10 CONTINUE
C *** COMPUTE NEW TRANSIENT-LEAKAGE FLUX ***
      NQ=NQ+5
      K=IN(I)
      IF(K.LT.0) GO TO 20
      TMPA=DTM*B(K)/GMA(I)
      GO TO 25
   20 TMPA=DTM*DHB(-K)/GMA(I)
   25 CBTQ(NQ)=CBTQ(NQ)+AC(1)*(1.-XP(1))*TMPA
      CBTQ(NQ+1)=CBTQ(NQ+1)+AC(2)*(1.-XP(2))*TMPA
      CBTQ(NQ+2)=CBTQ(NQ+2)+AC(3)*(1.-XP(3))*TMPA
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COCHG(Q,AR,H,HR,DHB,HK,HL,ALPH,QBND,CFDK,CFDL,TIME,KQB
     1,LQB,ND,IN,ISTP)
C        READS AND COMPUTES NEW VALUES OF TIME-VARIANT QUANTITIES
      DIMENSION Q(1),AR(1),H(1),HR(1),DHB(1),HK(1),HL(1),ALPH(1),QBND(1)
     1,CFDK(1),CFDL(1)
      DIMENSION KQB(1),LQB(1),ND(1),IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/CHG/NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** FORMAT LIST ***
    2 FORMAT (16I5)
    4 FORMAT (1H1,14X,32HCHANGES IN VALUES OF POINT FLOWS/1H
     1,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5,11H TIME UNITS
     2/1H ,24X,3HOLD,11X,3HNEW/1H ,15X,15HNODE  DISCHARGE,5X
     3,9HDISCHARGE)
    6 FORMAT (I5,4F10.0)
    8 FORMAT (1H ,13X,I5,2(3X,G11.5))
   10 FORMAT (1H1,12X,38HCHANGES IN VALUES OF DISTRIBUTED FLOWS/1H
     1,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5,11H TIME UNITS
     2/1H ,16X,4HBEG.,3X,3HNO.,3X,8HOLD UNIT,6X,8HNEW UNIT/1H ,10X
     3,27HZONE   EL.  ELS.  DISCHARGE,5X,9HDISCHARGE)
   12 FORMAT (3I5,2F10.0)
   14 FORMAT (1H ,7X,3(1X,I5),2(3X,G11.5))
   16 FORMAT (1H1,12X,37HCHANGES IN VALUES OF CAUCHY-TYPE DATA/1H
     1,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5,11H TIME UNITS
     2/1H ,6X,8HEL. SIDE,3X,8HNEW UNIT,6X,8HEXTERNAL,6X,8HEXTERNAL
     3/1H ,9X,3HNO.,5X,9HDISCHARGE,6X,6HHEAD A,8X,6HHEAD B)
   18 FORMAT (1H ,6X,I5,2X,3(3X,G11.5))
   20 FORMAT (1H1,13X,35HCHANGES IN VALUES OF SPECIFIED HEAD/1H
     1,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5,11H TIME UNITS
     2/1H ,19X,4HNODE,15X,4HHEAD)
   22 FORMAT (1H ,17X,I5,13X,G11.5)
   24 FORMAT (1H1,12X,36HCHANGES IN VALUES OF SOURCE-BED HEAD/1H
     1,29H  BEGINNING ON TIME-STEP NO. ,I4,4H AT ,G11.5,11H TIME UNITS
     2/1H ,19X,4HNODE,15X,4HHEAD)
C *** FOR POINT SOURCES AND SINKS: ***
      IF(NWCH.NE.ISTP) GO TO 70
C *** (1) READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C     MADE AGAIN ***
      READ(IIN,2) N,NWCH
      WRITE(IOUT,4) ISTP,TIME
      DO 60 I=1,N
C *** (2) READ AND WRITE OLD AND NEW VALUES ***
      READ(IIN,6) J,QOLD,QNEW
      WRITE(IOUT,8) J,QOLD,QNEW
C *** (3) ADD VALUES INTO TOTAL INFLOW OR OUTFLOW PER UNIT TIME FOR
C     FLOW BALANCE ***
      IF(QOLD.GT.0.) GO TO 30
      WQO=WQO-QOLD
      GO TO 40
   30 WQI=WQI-QOLD
   40 IF(QNEW.GT.0.) GO TO 50
      WQO=WQO+QNEW
      GO TO 60
   50 WQI=WQI+QNEW
```

```
C *** (4) ADD VALUES INTO Q-VECTOR ***
   60 Q(J)=Q(J)+QNEW-QOLD
C *** FOR DISTRIBUTED RECHARGE OR DISCHARGE: ***
   70 IF(NQCH.NE.ISTP) GO TO 152
C *** (1) READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C       MADE AGAIN ***
      READ(IIN,2) N,NQCH
      WRITE(IOUT,10) ISTP,TIME
C *** (2) READ ELEMENT AREAS AND NODE NUMBERS ***
      REWIND ITB
      M1=2*NELS
      M2=4*NELS
      READ(ITB) (AR(I),I=1,M1),(ND(I),I=1,M2)
      TMPA=0.
      TMPB=0.
      DO 150 I=1,N
C *** (3) READ AND WRITE OLD AND NEW VALUES ***
      READ(IIN,12) L,NBE,NO,QOLD,QNEW
      WRITE(IOUT,14) L,NBE,NO,QOLD,QNEW
C *** (4) BEGIN ELEMENT LOOP ***
      NEND=NBE+NO-1
      NDC=4*(NBE-1)+1
      NTE=2*(NBE-1)
      DO 140 J=NBE,NEND
      NE=1
      IF(ND(NDC+3).GT.0) NE=2
      DO 130 IE=1,NE
      AREA=AR(NTE+IE)
C *** (5) ADD VALUES INTO TOTAL INFLOW OR OUTFLOW PER UNIT TIME FOR
C       FLOW BALANCE ***
      IF(QOLD.GT.0.) GO TO 90
      TMPB=TMPB-QOLD*AREA
      GO TO 100
   90 TMPA=TMPA-QOLD*AREA
  100 IF(QNEW.GT.0.) GO TO 110
      TMPB=TMPB+QNEW*AREA
      GO TO 120
  110 TMPA=TMPA+QNEW*AREA
  120 DQ=(QNEW-QOLD)*AREA
C *** (6) ADD VALUES INTO Q-VECTOR ***
      NA=ND(NDC)
      NB=ND(NDC+IE)
      NC=ND(NDC+IE+1)
      Q(NA)=Q(NA)+DQ
      Q(NB)=Q(NB)+DQ
      Q(NC)=Q(NC)+DQ
  130 CONTINUE
      NTE=NTE+2
  140 NDC=NDC+4
  150 CONTINUE
C *** (7) MODIFY TOTAL INFLOW AND OUTFLOW PER UNIT TIME FOR FLOW
C       BALANCE ***
      DQI=3.*TMPA+DQI
      DQO=3.*TMPB+DQO
C *** FOR SOURCE-BED HEADS: ***
```

```
  152 IF(NHRCH.NE.ISTP) GO TO 160
C *** (1) READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C     MADE AGAIN ***
      READ(IIN,2) N,NHRCH
      WRITE(IOUT,24) ISTP,TIME
      DO 156 I=1,N
C *** (2) READ AND WRITE NEW VALUES ***
      READ (IIN,6) J,HR(J)
      WRITE(IOUT,22) J,HR(J)
  156 CONTINUE
C *** FOR CAUCHY-TYPE BOUNDARIES: ***
  160 IF(NBQCH.NE.ISTP) GO TO 210
C *** (1) READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C     MADE AGAIN ***
      READ(IIN,2) N,NBQCH
      WRITE(IOUT,16) ISTP,TIME
      DO 200 I=1,N
C *** (2) READ AND WRITE NEW VALUES OF KNOWN FLOW AND KNOWN HEADS ***
      READ(IIN,6) J,QNEW,HK(J),HL(J)
      WRITE(IOUT,18) J,QNEW,HK(J),HL(J)
C *** (3) COMPUTE NEW QBND AND ADD FLOWS INTO Q-VECTOR ***
  195 K=KQB(J)
      L=LQB(J)
      QOLD=QBND(J)
      IF(K.LT.0) GO TO 197
      QBND(J)=QNEW/ALPH(J)
      GO TO 198
  197 K=-K
      QBND(J)=QNEW
  198 Q(K)=Q(K)+(QBND(J)-QOLD)*CFDK(J)
  200 Q(L)=Q(L)+(QBND(J)-QOLD)*CFDL(J)
C *** FOR SPECIFIED-HEAD BOUNDARIES: ***
  210 IF(NHCH.NE.ISTP) RETURN
C *** (1) READ NUMBER OF CHANGES AND TIME STEP WHERE CHANGES ARE TO BE
C     MADE AGAIN ***
      READ(IIN,2) N,NHCH
      WRITE(IOUT,20) ISTP,TIME
      WF=2./3.
      DO 220 I=1,N
C *** (2) READ AND WRITE NEW VALUES ***
      READ(IIN,6) J,HB
      WRITE(IOUT,22) J,HB
C *** (3) FORM HEAD-CHANGE AND NEW-HEAD VECTORS ***
      K=-IN(J)
      DHB(K)=WF*(HB-H(J))
  220 H(J)=DHB(K)+H(J)
      RETURN
      END
```

```
      SUBROUTINE DATIN(TITLE,XG,YG,H,DHB,HR,HK,HL,ALPH,QBND,Q,KQB
     1,LQB,IN,IDZ,IDS,NBCZ)
C        READS DATA, SETS UP SOME INITIAL ARRAYS, AND COMPUTES SOME
C        FLOW-BALANCE COMPONENTS FOR BASIC VERSION OF PROGRAM
      DIMENSION TITLE(20),XG(1),YG(1),H(1),DHB(1),HR(1),HK(1),HL(1)
     1,ALPH(1),QBND(1),Q(1)
      DIMENSION KQB(1),LQB(1),IN(1),IDZ(1),IDS(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/IPRN/IPND
      COMMON/IND/IRAD,IUNIT,ISTD
      COMMON/SCLE/SCALE
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** FORMAT LIST ***
    2 FORMAT (I5,7F10.0)
    4 FORMAT (1H0,28X,17HNODAL COORDINATES/1H ,2X,2(4HNODE,5X,7HX COORD
     1,8X,7HY COORD,7X))
    5 FORMAT (1H0,10X,41HNODAL COORDINATES READ, OUTPUT SUPPRESSED)
    6 FORMAT (1H0,29X,13HINITIAL HEADS/1H ,2X,3(4HNODE,8X,4HHEAD,9X))
    7 FORMAT (1H0,10X,37HINITIAL HEADS READ, OUTPUT SUPPRESSED)
    8 FORMAT (1H0,27X,16HSOURCE BED HEADS/1H ,2X,3(4HNODE,8X,4HHEAD,9X))
    9 FORMAT (1H0,10X,40HSOURCE-BED HEADS READ, OUTPUT SUPPRESSED)
   10 FORMAT (1H0,35HSCALE CHANGE FOR NODAL COORDINATES:)
   12 FORMAT (20A4)
   14 FORMAT (1H ,20A4)
   16 FORMAT (8F10.0)
   22 FORMAT (1H0,6X,11HPOINT FLOWS/7H   NODE,5X,9HDISCHARGE)
   23 FORMAT (1H0,10X,35HPOINT FLOWS READ, OUTPUT SUPPRESSED)
   24 FORMAT (1H ,I5,6X,G11.5)
   28 FORMAT (16I5)
   32 FORMAT (1H0,17X,42HCAUCHY-TYPE BOUNDARY DATA BY BOUNDARY ZONE)
   33 FORMAT (1H0,10X,46HCAUCHY-TYPE BOUNDARIES READ, OUTPUT SUPPRESSED)
   34 FORMAT (3I5,4F10.0)
   35 FORMAT (1H0,5X,5HZONE ,I5,4X,9HCONTAINS ,I5,15H BOUNDARY SIDES
     1/1H ,5H SIDE,2X,18HBOUNDARY  BOUNDARY,13X,9HSPECIFIED,5X
     2,8HEXTERNAL,5X,8HEXTERNAL/1H ,5H  NO.,3X,6HNODE A,4X,6HNODE B,3X
     3,5HALPHA,9X,4HFLOW,8X,6HHEAD A,7X,6HHEAD B)
   36 FORMAT (1H ,I4,3X,I5,5X,I5,1X,4(2X,G11.5))
   38 FORMAT (1H0,4X,15HSPECIFIED HEADS/1H ,13X,8HBOUNDARY/7H   NODE,9X
     1,4HHEAD)
   39 FORMAT(1H0,10X,39HSPECIFIED HEADS READ, OUTPUT SUPPRESSED)
   40 FORMAT (1H ,I5,7X,G11.5)
   44 FORMAT (1H0,43H** AXI-SYMMETRIC RADIAL COORDINATES USED **)
   46 FORMAT (1H0,23H** STEADY-STATE FLOW **)
C *** READ INDICATORS FOR RADIAL FLOW, SCALING, AND
C     STEADY-STATE FLOW ***
      READ(IIN,28) IRAD,IUNIT,ISTD
      IF(IRAD.GT.0) WRITE(IOUT,44)
      IF(ISTD.GT.0) WRITE(IOUT,46)
C *** READ AND WRITE NODAL SCALE CHANGE, IF ANY ***
      SCALE=1.
      IF(IUNIT.LE.0) GO TO 60
      WRITE(IOUT,10)
      READ(IIN,12) (TITLE(I),I=1,20)
      WRITE(IOUT,14) (TITLE(I),I=1,20)
      READ(IIN,16) SCALE
```

```
C *** READ INDICATOR VARIABLES FOR SUPPRESSING PRINTOUT OF
C     INITIAL CONDITIONS ***
   60 READ(IIN,28) IPXY,IPH,IPHR,IPQW,IPCB,IPHB,IPND
C *** READ AND WRITE NODAL COORDINATES AND INITIAL HEADS ***
      DO 70 J=1,NNDS
      READ(IIN,2) I,XG(I),YG(I),H(I),HR(I)
   70 CONTINUE
      IF(IPXY.GT.0) GO TO 71
      WRITE(IOUT,4)
      CALL PRTOB(XG,YG,NNDS)
      GO TO 72
   71 WRITE(IOUT,5)
   72 IF(IPH.GT.0) GO TO 73
      WRITE(IOUT,6)
      CALL PRTOA(H,NNDS)
      GO TO 74
   73 WRITE(IOUT,7)
   74 IF(IPHR.GT.0) GO TO 75
      WRITE(IOUT,8)
      CALL PRTOA(HR,NNDS)
      GO TO 76
   75 WRITE(IOUT,9)
C *** FOR POINT SOURCES AND SINKS: ***
   76 WQI=0.
      WQO=0.
      IF(NWELS.LE.0) GO TO 100
      IF(IPQW.GT.0) GO TO 77
      WRITE(IOUT,22)
      GO TO 78
   77 WRITE(IOUT,23)
   78 DO 90 I=1,NWELS
C *** (1) READ AND WRITE VALUES ***
      READ(IIN,2) J,QWEL
      IF(IPQW.EQ.0) WRITE(IOUT,24) J,QWEL
C *** (2) COMPUTE TOTAL INFLOW OR OUTFLOW PER UNIT TIME
C     FOR FLOW BALANCE ***
      IF(QWEL.GT.0.) GO TO 85
      WQO=WQO+QWEL
      GO TO 90
   85 WQI=WQI+QWEL
C *** (3) ADD VALUES INTO Q-VECTOR ***
   90 Q(J)=Q(J)+QWEL
C *** READ AND WRITE DATA FOR CAUCHY-TYPE BOUNDARIES ***
  100 IF(NQBND.LE.0) GO TO 130
      IF(IPCB.GT.0) GO TO 101
      WRITE(IOUT,32)
      GO TO 102
  101 WRITE(IOUT,33)
  102 DO 128 NZ=1,NBCZ
      READ(IIN,28) KZ,NOS,IZIN
      IF(IPCB.EQ.0) WRITE(IOUT,35) KZ,NOS
      IDZ(NZ)=KZ
      IDS(KZ)=NOS
      IF(IZIN.GT.0) GO TO 124
      DO 120 I=1,NOS
```

```
      READ(IIN,34) J,KQB(J),LQB(J),ALPH(J),QBND(J),HK(J),HL(J)
      IF(IPCB.EQ.0) WRITE(IOUT,36) J,KQB(J),LQB(J),ALPH(J),QBND(J)
     1,HK(J),HL(J)
  120 CONTINUE
      GO TO 128
  124 READ(IIN,16) ALPHZ,QBNZ
      DO 126 I=1,NOS
      READ(IIN,34) J,KQB(J),LQB(J),HK(J),HL(J)
      ALPH(J)=ALPHZ
      QBND(J)=QBNZ
      IF(IPCB.EQ.0) WRITE(IOUT,36) J,KQB(J),LQB(J),ALPH(J),QBND(J)
     1,HK(J),HL(J)
  126 CONTINUE
  128 CONTINUE
C *** INITIALIZE NODAL POINTER ARRAY, IN
  130 DO 134 I=1,NNDS
  134 IN(I)=I
      IN(NNDS+1)=0
      NEQ=NNDS
C *** FOR SPECIFIED-HEAD BOUNDARIES: ***
      IF(NHDS.LE.0) RETURN
      WF=1.
      IF(ISTD.LT.1) WF=2./3.
      IF(IPHB.GT.0) GO TO 135
      WRITE(IOUT,38)
      GO TO 136
  135 WRITE(IOUT,39)
  136 DO 140 I=1,NHDS
C *** (1) READ AND WRITE VALUES ***
      READ(IIN,2) J,HB
      IF(IPHB.EQ.0) WRITE(IOUT,40) J,HB
C *** (2) MODIFY IN TO EXCLUDE SPECIFIED HEAD NODES FROM
C       MATRIX EQUATION ***
      DO 138 K=J,NNDS
  138 IF(IN(K).GT.0) IN(K)=IN(K)-1
      IN(J)=-I
C *** (3) DEFINE HEAD-CHANGE AND MEAN-HEAD VECTORS ***
      DHB(I)=WF*(HB-H(J))
      H(J)=DHB(I)+H(J)
  140 CONTINUE
C *** COMPUTE NUMBER OF SPECIFIED HEADS AT END OF HEAD VECTOR ***
      NLHS=0
      J=NNDS+1
      DO 150 I=1,NNDS
      J=J-1
      IF(IN(J).GT.0) GO TO 160
  150 NLHS=NLHS+1
C *** COMPUTE REDUCED NUMBER OF EQUATIONS ***
  160 NEQ=NNDS-NLHS
      RETURN
      END
```

```
      SUBROUTINE DATOUT(H,DT,TIME,ISTP)
C        UPDATES TIME AND WRITES HEAD VECTOR AT END OF TIME STEP
      DIMENSION H(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
   10 FORMAT (1H1,9X,25HOUTPUT FOR TIME STEP NO. ,I3,4H AT ,G11.5
     1,11H TIME UNITS)
   12 FORMAT (1H0,19X,33HCOMPUTED VALUES OF HYDRAULIC HEAD
     1/1H ,3(6H  NODE,8X,4HHEAD,7X))
      TIME=TIME+DT
      WRITE(IOUT,10) ISTP,TIME
      WRITE(IOUT,12)
      CALL PRTOA(H,NNDS)
      RETURN
      END
```

```
      SUBROUTINE EXTRAP(H,DHB,B,IN)
C        EXTRAPOLATES HEADS AT ALL NODES FROM THE MEAN POINT IN THE TIME
C        STEP TO THE END OF THE TIME STEP, THEN ZEROES THE HEAD CHANGES
      DIMENSION H(1),DHB(1),B(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      DO 30 I=1,NNDS
      J=IN(I)
      IF(J.GT.0) GO TO 20
      J=-J
      H(I)=.5*DHB(J)+H(I)
      DHB(J)=0.
      GO TO 30
   20 H(I)=.5*B(J)+H(I)
      B(J)=0.
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FMCO(XG,YG,A,Q,AR,ALPH,QBND,CFDK,CFDL,ND,IZN,JPT,KQB
     1,LQB)
C        READS ELEMENT DATA AND FORMS COEFFICIENT ARRAYS FOR MATRIX
C        EQUATION
      DIMENSION XG(1),YG(1),A(1),Q(1),AR(1),ALPH(1),QBND(1),CFDK(1)
     1,CFDL(1)
      DIMENSION ND(1),IZN(1),JPT(1),KQB(1),LQB(1)
      DIMENSION XL(4),YL(4),TFL(3),NDID(4)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/IPRN/IPND
      COMMON/IND/IRAD,IUNIT,ISTD
      COMMON/SCLE/SCALE
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** FORMAT LIST ***
    2 FORMAT (1H0,31X,18HPARAMETERS BY ZONE/1H ,31X,8HROTATION,6X
     1,8HAQUITARD,5X,7HSTORAGE,4X,8HRECHARGE/7H   ZONE,2X,8HX TRANS.
     2,4X,8HY TRANS.,5X,5HANGLE,6X,10HHYD. COND.,2X,11HCOEFFICIENT,4X
     3,4HRATE)
    4 FORMAT (2I5,6F10.0)
    6 FORMAT (1H ,I5,1X,6(1X,G11.5))
    8 FORMAT (16I5)
   10 FORMAT (1H0,18X,12HELEMENT DATA/9H  ELEMENT,2X,6HNODE 1,2X
     1,6HNODE 2,2X,6HNODE 3,2X,6HNODE 4,3X,4HZONE)
   11 FORMAT (1H0,8X,36HELEMENT DATA READ, OUTPUT SUPPRESSED)
   12 FORMAT (1H ,I6,5I8)
C *** FOR CAUCHY-TYPE BOUNDARIES: ***
      IF(NQBND.LE.0) GO TO 35
      CA=.5*SCALE
      DO 30 J=1,NQBND
C *** (1) FORM COEFFICIENTS ***
      K=KQB(J)
      L=LQB(J)
      TMPA=XG(K)-XG(L)
      TMPB=YG(K)-YG(L)
      DIST=CA*(TMPA*TMPA+TMPB*TMPB)**.5
      TMPA=DIST
      TMPB=DIST
      IF(IRAD.LE.0) GO TO 20
      TMPA=DIST*(2.*XG(K)+XG(L))/3.
      TMPB=DIST*(2.*XG(L)+XG(K))/3.
   20 ALF=ALPH(J)
      QB=QBND(J)
      IF(ALF.LT.1.E-30) GO TO 24
      CFDK(J)=ALF*TMPA
      CFDL(J)=ALF*TMPB
      QBND(J)=QB/ALF
      GO TO 26
   24 CFDK(J)=TMPA
      CFDL(J)=TMPB
      KQB(J)=-K
C *** (2) ADD KNOWN-FLUX COMPONENTS INTO Q-VECTOR ***
   26 Q(K)=Q(K)+QB*TMPA
   30 Q(L)=Q(L)+QB*TMPB
C *** INITIALIZE POINTER ARRAY, JPT, AND VARIABLES FOR ZONAL LOOP ***
   35 MBM1=MBWC-1
```

```
      N=MBM1*NNDS
      L=NNDS+1
      DO 40 I=1,N
   40 JPT(I)=L
      DQI=0.
      DQO=0.
      NDC=0
      NTE=0
      CA=.5
      CB=SCALE*SCALE/6.
      IW=MBWC+1
      KNT=0
C *** BEGIN ZONAL LOOP ***
      WRITE(IOUT,2)
      DO 150 IZ=1,NZNS
C *** READ AND WRITE DATA FOR EACH ZONE ***
      READ(IIN,4) KZ,NO,XTR,YTR,ANG,VLC,STR,QD
      WRITE(IOUT,6) KZ,XTR,YTR,ANG,VLC,STR,QD
C *** SCALE DATA ***
      XTR=XTR*CA
      YTR=YTR*CA
      VLC=VLC*CB
      STR=STR*CB
      QD=QD*CB
C *** COMPUTE SINE AND COSINE OF ROTATION ANGLE ***
      IF(ANG.LE.0.) GO TO 42
      ANG=.01745329*ANG
      SN=SIN(ANG)
      CS=COS(ANG)
C *** BEGIN ELEMENT LOOP WITHIN ZONE ***
   42 DO 140 I=1,NO
C *** READ NODE NUMBERS FOR EACH ELEMENT IN ZONE ***
      READ(IIN,8) IEL,(ND(NDC+J),J=1,4)
      KNT=KNT+1
      IZN(KNT)=KZ
      NE=2
      N=4
      IF(ND(NDC+4).GT.0) GO TO 45
      NE=1
      N=3
C *** COMPUTE LOCAL, ROTATED COORDINATES ***
   45 DO 50 J=1,N
      K=ND(NDC+J)
      XL(J)=XG(K)
   50 YL(J)=YG(K)
      IF(ANG.LT.1.E-20) GO TO 60
      DO 55 J=1,N
      TMPA=XL(J)
      TMPB=YL(J)
      XL(J)=TMPA*CS+TMPB*SN
   55 YL(J)=-TMPA*SN+TMPB*CS
C *** BEGIN NODAL LOOP WITHIN ELEMENT ***
   60 NT=NDC+1
      DO 135 IE=1,NE
C *** COMPUTE COEFFICIENTS FOR COORDINATE FUNCTIONS ***
```

```
      NA=ND(NT)
      NB=ND(NT+IE)
      NC=ND(NT+IE+1)
      XNA=XL(1)
      YNA=YL(1)
      XNB=XL(IE+1)
      YNB=YL(IE+1)
      XNC=XL(IE+2)
      YNC=YL(IE+2)
      BJ=YNB-YNC
      BK=YNC-YNA
      BL=YNA-YNB
      CJ=XNC-XNB
      CK=XNA-XNC
      CL=XNB-XNA
      AREA=BJ*CK-BK*CJ
      AR(NTE+IE)=CB*AREA
C *** COMPUTE ELEMENT CONTRIBUTION TO DISTRIBUTED RECHARGE-DISCHARGE ***
      TEQ=QD*AREA
C *** COMPUTE TOTAL DISTRIBUTED RECHARGE OR DISCHARGE PER UNIT
C     TIME FOR FLOW BALANCE ***
      IF(QD.GT.0.) GO TO 65
      DQO=DQO+TEQ
      GO TO 70
   65 DQI=DQI+TEQ
C *** ADD DISTRIBUTED RECHARGE-DISCHARGE INTO Q-VECTOR ***
   70 Q(NA)=Q(NA)+TEQ
      Q(NB)=Q(NB)+TEQ
      Q(NC)=Q(NC)+TEQ
C *** COMPUTE ELEMENT CONTRIBUTIONS TO CAPACITANCE, LEAKANCE,
C     AND FLOW DIVERGENCE ***
      TESJ=STR*AREA
      TESK=TESJ
      TESL=TESJ
      XT=XTR/AREA
      YT=YTR/AREA
      IF(IRAD.LE.0) GO TO 75
      TESJ=TESJ*.25*(2.*XNA+XNB+XNC)
      TESK=TESK*.25*(2.*XNB+XNA+XNC)
      TESL=TESL*.25*(2.*XNC+XNA+XNB)
      TMPA=(XNA+XNB+XNC)/3.
      XT=XT*TMPA
      YT=YT*TMPA
   75 TEL=VLC*AREA
      TXJ=XT*BJ
      TYJ=YT*CJ
      TXL=XT*BL
      TYL=YT*CL
      TFL(1)=TXJ*BK+TYJ*CK
      TFL(2)=TXL*BK+TYL*CK
      TFL(3)=TXJ*BL+TYJ*CL
C *** ADD ELEMENT CONTRIBUTIONS FOR CAPACITANCE AND LEAKANCE INTO
C     A-MATRIX ***
      ICA=IW*(NA-1)+1
      ICB=IW*(NB-1)+1
      ICC=IW*(NC-1)+1
```

```
      A(ICA)=A(ICA)+TESJ
      A(ICB)=A(ICB)+TESK
      A(ICC)=A(ICC)+TESL
      A(ICA+1)=A(ICA+1)+TEL
      A(ICB+1)=A(ICB+1)+TEL
      A(ICC+1)=A(ICC+1)+TEL
C *** ADD ELEMENT CONTRIBUTIONS FOR FLOW DIVERGENCE INTO A-MATRIX AND
C     FORM JPT ***
      NDID(1)=NA
      NDID(2)=NB
      NDID(3)=NC
      NDID(4)=NA
      DO 130 J=1,3
      NA=NDID(J)
      NB=NDID(J+1)
      IF(NB.GT.NA) GO TO 80
      NA=NB
      NB=NDID(J)
   80 M=IW*(NA-1)+1
      M1=MBM1*(NA-1)
      K=0
      DO 90 KK=1,MBM1
      K=K+1
      L=JPT(M1+K)
      IF(L-NB) 90,100,110
   90 CONTINUE
  100 A(M+K+1)=A(M+K+1)+TFL(J)
      GO TO 130
  110 KK=MBM1-K
      L=MBWC
      DO 120 LL=1,KK
      L=L-1
      JPT(M1+L)=JPT(M1+L-1)
  120 A(M+L+1)=A(M+L)
      JPT(M1+K)=NB
      A(M+K+1)=TFL(J)
  130 CONTINUE
  135 CONTINUE
      NTE=NTE+2
  140 NDC=NDC+4
  150 CONTINUE
C *** WRITE ELEMENT NODE AND ZONE NUMBERS ***
      IF(IPND.GT.0) GO TO 162
      WRITE(IOUT,10)
      NDC=0
      DO 160 I=1,NELS
      WRITE(IOUT,12) I,(ND(NDC+J),J=1,4),IZN(I)
      IZN(I)=0
  160 NDC=NDC+4
      GO TO 166
  162 WRITE(IOUT,11)
      DO 164 I=1,NELS
  164 IZN(I)=0
C *** ADJUST TOTAL RECHARGE AND DISCHARGE PER UNIT TIME ***
  166 DQI=3.*DQI
      DQO=3.*DQO
      RETURN
      END
```

```
      SUBROUTINE FMECWT(H,DH,DHB,HR,HK,HL,CFDK,CFDL,A,AD,Q,B,THK,DTK
     1,TOP,ASY,DT,JPT,IN,KQB,LQB)
C        FORMS CONDENSED COEFFICIENT MATRIX, A, AND RIGHT-HAND-SIDE
C        VECTOR, B, FOR WATER-TABLE CORRECTOR STEP
      DIMENSION H(1),DH(1),DHB(1),HR(1),HK(1),HL(1),CFDK(1),CFDL(1)
     1,A(1),AD(1),Q(1),B(1),THK(1),DTK(1),TOP(1),ASY(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** FORMAT LIST ***
    1 FORMAT (1H0,5HNODE ,I5,20H PREDICTED TO BE DRY/1H ,3X
     1,38HPREDICTED AQUIFER THICKNESS AT NODE = ,G11.5/1H ,3X
     2,19HNET FLOW AT NODE = ,G11.5)
    2 FORMAT (1H ,3X,30HNET FLUX FROM NODE REDUCED TO ,G11.5)
      WF=2./3.
      IW=MBWC+1
      NC=1
      DO 25 I=1,NNDS
C *** INITIALIZE  DTK-VECTOR ***
      DTK(I)=0.
      J=IN(I)
      IF(J.GT.0) GO TO 12
C *** COMPUTE DTK-VECTOR AT SPECIFIED-HEAD NODES ***
      HO=H(I)-DHB(-J)
      DHP=1.5*DHB(-J)
      HP=HO+DHP
      IF(HO.LT.TOP(I)) GO TO 8
      IF(HP.GT.TOP(I)) GO TO 25
      DTK(I)=HP-TOP(I)
      GO TO 20
    8 IF(HP.LT.TOP(I)) GO TO 10
      DTK(I)=TOP(I)-HO
      GO TO 20
   10 DTK(I)=DHP
      GO TO 20
C *** COMPUTE OLD AND PREDICTED HEADS AT ACTIVE NODES ***
   12 HO=H(I)-DH(J)
      DHP=1.5*DH(J)
      HP=HO+DHP
C *** INITIALIZE MATRIX DIAGONAL, AD, AND B-VECTOR ***
      AD(J)=0.
      B(J)=0.
      IF(HO.LT.TOP(I)) GO TO 14
      IF(HP.GE.TOP(I)) GO TO 25
C *** FOR ACTIVE NODES CONVERTING FROM CONFINED TO UNCONFINED
C     CONDITIONS: ***
C *** (1) COMPUTE PREDICTED HEAD AND DTK-VECTOR FOR NODES THAT ARE
C     SATURATED ***
      HP=(A(NC)/ASY(I))*(HP-TOP(I))+TOP(I)
      DHP=HP-HO
      DH(J)=WF*DHP
      H(I)=HO+DH(J)
      DTK(I)=HP-TOP(I)
C *** (2) MODIFY B-VECTOR AND DEFINE CAPACITANCE TERM, A, AS ASY ***
      B(J)=B(J)+(ASY(I)-A(NC))*(TOP(I)-HO)/DT
```

```
      A(NC)=ASY(I)
C *** (3) COMPUTE DTK-VECTOR AND IDENTIFY AND REDUCE DISCHARGE AT NODES
C     THAT ARE PREDICTED TO GO DRY ***
      IF(THK(I)+DTK(I).GT.0.) GO TO 20
      THKP=THK(I)+DTK(I)
      WRITE(IOUT,1) I,THKP,Q(I)
      DTK(I)=-THK(I)
      IF(Q(I).GE.0.) GO TO 20
      Q(I)=.5*Q(I)
      WQO=WQO-Q(I)
      WRITE(IOUT,2) Q(I)
      GO TO 20
C *** COMPUTE DTK-VECTOR AND MODIFY B-VECTOR AT ACTIVE NODES
C     CONVERTING FROM UNCONFINED TO CONFINED CONDITIONS ***
   14 IF(HP.LT.TOP(I)) GO TO 16
      DTK(I)=TOP(I)-HO
      B(J)=B(J)+(A(NC)-ASY(I))*DTK(I)/DT
      GO TO 20
C *** FOR UNCONFINED ACTIVE NODES: ***
C *** (1) DEFINE CAPACITANCE TERM, A, AS ASY ***
   16 A(NC)=ASY(I)
C *** (2) COMPUTE DTK-VECTOR AND IDENTIFY AND REDUCE DISCHARGE AT
C     NODES THAT ARE DRY ***
      IF(THK(I)+DHP.GT.0.) GO TO 18
      DTK(I)=0.
      IF(THK(I).GT.0.) DTK(I)=-THK(I)
      THKP=THK(I)+DHP
      WRITE(IOUT,1) I,THKP,Q(I)
      IF(Q(I).GE.0..OR.DHP.GE.0.) GO TO 20
      Q(I)=.5*Q(I)
      WQO=WQO-Q(I)
      WRITE(IOUT,2) Q(I)
      GO TO 20
   18 DTK(I)=DHP
      IF(THK(I).LT.0.) DTK(I)=DHP+THK(I)
C *** MODIFY DTK-VECTOR AND DEFINE PH-VECTOR ***
   20 DTK(I)=WF*DTK(I)
   25 NC=NC+IW
C *** INITIALIZE FOR NODAL ASSEMBLY LOOP ***
      DTM=3./(2.*DT)
      MBM1=MBWC-1
      C1=9./16.
      C2=8./9.
      C3=1./16.
      NC=1
      NME=1
      NP=0
      DO 80 I=1,NNDS
      NVL=NC+1
      ND=NVL
C *** COMPUTE A-MATRIX AND B-VECTOR EXCEPT FOR BOUNDARY CONDITION
C     CONTRIBUTIONS ***
      THKI=THK(I)
      IF(THKI.LT.0.) THKI=0.
      K=IN(I)
```

```
      IF(K.LT.0) GO TO 64
      TMPA=A(NC)*DTM
      B(K)=B(K)-TMPA*DH(K)+A(NVL)*(HR(I)-H(I))+Q(I)
      AD(K)=AD(K)+TMPA+A(NVL)
      DO 50 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 60
      THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      TMPA=C1*(DTK(I)+DTK(L)+C2*(THKI+THKL))*A(ND+J)
      AD(K)=AD(K)-TMPA
      M=IN(L)
      IF(M.GT.0) GO TO 40
      M=-M
      B(K)=B(K)-(C3*(DTK(I)+DTK(L))*(DHB(M)-DH(K))
     1+.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I)))*A(ND+J)
      A(NME+J)=0.
      GO TO 50
   40 TMPB=(C3*(DTK(I)+DTK(L))*(DH(M)-DH(K))
     1+.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I)))*A(ND+J)
      B(K)=B(K)-TMPB
      B(M)=B(M)+TMPB
      A(NME+J)=TMPA
      AD(M)=AD(M)-TMPA
   50 CONTINUE
   60 NME=NME+MBWC
      GO TO 70
   64 K=-K
      DO 66 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 66,70,65
   65 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      B(M)=B(M)-(C3*(DTK(I)+DTK(L))*(DHB(K)-DH(M))
     1+.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(I)-H(L)))*A(ND+J)
      AD(M)=AD(M)-C1*(DTK(I)+DTK(L)+C2*(THKI+THKL))*A(ND+J)
   66 CONTINUE
   70 NC=NC+IW
   80 NP=NP+MBM1
      N=NNDS-NHDS
      NME=1
      J=0
      DO 85 I=1,N
      A(NME)=AD(I)
      J=J+1
   85 NME=NME+MBWC
C *** ADD CONTRIBUTIONS FROM CAUCHY-TYPE BOUNDARIES INTO A AND B ***
      IF(NQBND.LE.0) RETURN
      DO 130 I=1,NQBND
      K=KQB(I)
      IF(K.LT.0) GO TO 130
      L=LQB(I)
      KE=IN(K)
      IF(KE.LT.0) GO TO 100
      M=MBWC*(KE-1)+1
```

```
      A(M)=A(M)+CFDK(I)
      B(KE)=B(KE)+CFDK(I)*(HK(I)-H(K))
  100 LE=IN(L)
      IF(LE.LT.0) GO TO 130
      N=MBWC*(LE-1)+1
      A(N)=A(N)+CFDL(I)
      B(LE)=B(LE)+CFDL(I)*(HL(I)-H(L))
  130 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FMEPWT(H,HR,HK,HL,CFDK,CFDL,A,AD,Q,B,THK,TOP,ASY,DT,JPT
     1,IN,KQB,LQB)
C        FORMS CONDENSED COEFFICIENT MATRIX, A, AND RIGHT-HAND-SIDE
C        VECTOR, B, FOR WATER-TABLE PREDICTOR STEP
      DIMENSION H(1),HR(1),HK(1),HL(1),CFDK(1),CFDL(1),A(1),AD(1),Q(1)
     1,B(1),THK(1),TOP(1),ASY(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      IW=MBWC+1
      NC=1
      DO 10 I=1,NEQ
      J=IN(I)
      IF(J.LT.0) GO TO 10
C *** INITIALIZE MATRIX DIAGONAL, AD, AND B-VECTOR ***
      AD(J)=0.
      B(J)=0.
C *** DEFINE CAPACITANCE FOR WATER-TABLE NODES AS THE NODAL SPECIFIC
C     YIELD COEFFICIENTS, ASY ***
      IF(H(I).LE.TOP(I)) A(NC)=ASY(I)
   10 NC=NC+IW
C *** INITIALIZE FOR NODAL ASSEMBLY LOOP ***
      DTM=3./(2.*DT)
      MBM1=MBWC-1
      NC=1
      NME=1
      NP=0
C *** COMPUTE A-MATRIX AND B-VECTOR EXCEPT FOR BOUNDARY-CONDITION
C     CONTRIBUTIONS ***
      DO 80 I=1,NNDS
      NVL=NC+1
      ND=NVL
      THKI=THK(I)
      IF(THKI.LT.0.) THKI=0.
      K=IN(I)
      IF(K.LT.0) GO TO 64
      B(K)=B(K)+A(NVL)*(HR(I)-H(I))+Q(I)
      AD(K)=AD(K)+A(NC)*DTM+A(NVL)
      DO 50 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 60
      THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      TMPA=.5*(THKI+THKL)*A(ND+J)
      TMPB=TMPA*(H(L)-H(I))
      B(K)=B(K)-TMPB
      AD(K)=AD(K)-TMPA
      M=IN(L)
      IF(M.GT.0) GO TO 40
      A(NME+J)=0.
      GO TO 50
   40 B(M)=B(M)+TMPB
      A(NME+J)=TMPA
      AD(M)=AD(M)-TMPA
   50 CONTINUE
   60 NME=NME+MBWC
      GO TO 70
```

```
   64 DO 66 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 66,70,65
   65 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      TMPA=.5*(THKI+THKL)*A(ND+J)
      B(M)=B(M)-TMPA*(H(I)-H(L))
      AD(M)=AD(M)-TMPA
   66 CONTINUE
   70 NC=NC+IW
   80 NP=NP+MBM1
      N=NNDS-NHDS
      NME=1
      DO 85 I=1,N
      A(NME)=AD(I)
   85 NME=NME+MBWC
C *** ADD CONTRIBUTIONS FROM CAUCHY-TYPE BOUNDARIES INTO A AND B ***
      IF(NQBND.LE.0) RETURN
      DO 130 I=1,NQBND
      K=KQB(I)
      IF(K.LT.0) GO TO 130
      L=LQB(I)
      KE=IN(K)
      IF(KE.LT.0) GO TO 100
      M=MBWC*(KE-1)+1
      A(M)=A(M)+CFDK(I)
      B(KE)=B(KE)+CFDK(I)*(HK(I)-H(K))
  100 LE=IN(L)
      IF(LE.LT.0) GO TO 130
      N=MBWC*(LE-1)+1
      A(N)=A(N)+CFDL(I)
      B(LE)=B(LE)+CFDL(I)*(HL(I)-H(L))
  130 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FMEQ(H,HR,HK,HL,CFDK,CFDL,A,AD,Q,B,DT,JPT,IN,KQB,LQB)
C        FORMS CONDENSED COEFFICIENT MATRIX, A, AND RIGHT-HAND-SIDE
C        VECTOR, B
      DIMENSION H(1),HR(1),HK(1),HL(1),CFDK(1),CFDL(1),A(1),AD(1),Q(1)
     1,B(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
C *** INITIALIZE MATRIX DIAGONAL, AD, AND B-VECTOR ***
      N=NNDS-NHDS
      DO 20 I=1,N
      AD(I)=0.
   20 B(I)=0.
C *** INIALIZE FOR NODAL LOOP ***
      DTM=3./(2.*DT)
      MBM1=MBWC-1
      IW=MBWC+1
      NC=1
      NME=1
      NP=0
C *** COMPUTE A-MATRIX AND B-VECTOR EXCEPT FOR BOUNDARY-CONDITION
C     CONTRIBUTIONS ***
      DO 80 I=1,NNDS
      NVL=NC+1
      ND=NVL
      K=IN(I)
      IF(K.LT.0) GO TO 64
      B(K)=B(K)+A(NVL)*(HR(I)-H(I))+Q(I)
      AD(K)=AD(K)+A(NC)*DTM+A(NVL)
      DO 50 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 60
      TMPB=A(ND+J)*(H(L)-H(I))
      B(K)=B(K)-TMPB
      AD(K)=AD(K)-A(ND+J)
      M=IN(L)
      IF(M.GT.0) GO TO 40
      A(NME+J)=0.
      GO TO 50
   40 B(M)=B(M)+TMPB
      A(NME+J)=A(ND+J)
      AD(M)=AD(M)-A(ND+J)
   50 CONTINUE
   60 NME=NME+MBWC
      GO TO 70
   64 DO 66 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 66,70,65
   65 B(M)=B(M)-A(ND+J)*(H(I)-H(L))
      AD(M)=AD(M)-A(ND+J)
   66 CONTINUE
   70 NC=NC+IW
   80 NP=NP+MBM1
      NME=1
      DO 85 I=1,N
      A(NME)=AD(I)
```

```
   85 NME=NME+MBWC
C *** ADD CONTRIBUTIONS FROM CAUCHY-TYPE BOUNDARIES INTO A AND B ***
      IF(NQBND.LE.0) RETURN
      DO 130 I=1,NQBND
      K=KQB(I)
      IF(K.LT.0) GO TO 130
      L=LQB(I)
      KE=IN(K)
      IF(KE.LT.0) GO TO 100
      M=MBWC*(KE-1)+1
      A(M)=A(M)+CFDK(I)
      B(KE)=B(KE)+CFDK(I)*(HK(I)-H(K))
  100 LE=IN(L)
      IF(LE.LT.0) GO TO 130
      N=MBWC*(LE-1)+1
      A(N)=A(N)+CFDL(I)
      B(LE)=B(LE)+CFDL(I)*(HL(I)-H(L))
  130 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GNBAL(H,DH,DHB,R,GC,HRK,HRL,ZRK,ZRL,ZP,VQK,VQL,DT,IN,KR
     1,LR,KP,NBNC,NPNB)
C        COMPUTES FLOW-BALANCE COMPONENTS FOR NONLINEAR CAUCHY-TYPE
C        BOUNDARIES AND NONLINEAR POINT SINKS
      DIMENSION H(1),DH(1),DHB(1),R(1),GC(1),HRK(1),HRL(1),ZRK(1),ZRL(1)
     1,ZP(1),VQK(1),VQL(1)
      DIMENSION IN(1),KR(1),LR(1),KP(1),NL(2)
      DIMENSION HR(2),ZR(2),QR(2)
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/GNBL/BNQI,BNQO,TBNQI,TBNQO,PNQO,TPNQO
C *** SET VOLUMETRIC-FLOW RATES TO ZERO ***
      BNQI=0.
      BNQO=0.
      PNQO=0.
C *** COMPUTE WEIGHTING FACTORS FOR GALERKIN-IN-TIME FORMULATION ***
      CA=1./3.
      CB=2./3.
C ***
      IF(NBNC.LT.1) GO TO 65
C *** FORM COEFFICIENTS FOR EACH SIDE J ON A NONLINEAR CAUCHY-TYPE
C     BOUNDARY ***
      DO 60 J=1,NBNC
      HR(1)=HRK(J)
      HR(2)=HRL(J)
      ZR(1)=ZRK(J)
      ZR(2)=ZRL(J)
      NL(1)=KR(J)
      NL(2)=LR(J)
C *** FOR EACH NODE I ON THE SIDE: ***
      DO 50 I=1,2
C *** (1) COMPUTE HEAD AT N LEVEL, HO, AND HEAD AT N+1 LEVEL, HC ***
      L=NL(I)
      LE=IN(L)
      IF(LE.LT.0) GO TO 10
      DHC=DH(LE)
      GO TO 15
   10 DHC=DHB(-LE)
   15 HO=H(L)-DHC
      DHC=1.5*DHC
      HC=HO+DHC
C *** (2) REPRESENT ELEVATIONS, AND HEAD AND ELEVATION DIFFERENCES, BY
C         TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      ZRI=ZR(I)
      HRI=HR(I)
      DRZ=HRI-ZRI
      DHZ=HO-ZRI
C *** (3) COMPARE HEADS WITH CONTROLLING ELEVATIONS TO IDENTIFY LEAKAGE
C         CASE. COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-FLOW RATES
C         FOR: ***
      IF(HO.LT.ZRI) GO TO 30
      IF(HC.LT.ZRI) GO TO 20
C *** CASE (1) ***
      TMPA=HRI-H(L)
      GO TO 40
C *** CASE (2) ***
   20 PHI=DHZ/DHC
```

```
      TMPA=DRZ-CA*PHI*PHI*DHZ
      GO TO 40
   30 IF(HC.GT.ZRI) GO TO 35
C *** CASE (4) ***
      TMPA=DRZ
      GO TO 40
C *** CASE (3) ***
   35 PHC=1.+DHZ/DHC
      TMPA=DRZ-CB*PHC*(HC-ZRI)
C *** (4) COMPUTE VOLUMETRIC-FLOW RATE ***
   40 TMPB=GC(J)*TMPA
      QR(I)=TMPB
C *** (5) INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C         SPECIFIED-HEAD NODE ***
      IF(LE.LT.0) R(-LE)=R(-LE)-TMPB
C *** (6) SUM VOLUMETRIC-FLOW RATES ACCORDING TO SIGN, POSITIVE
C         FOR INFLOW ***
      IF(TMPB.LT.0.) GO TO 45
      BNQI=BNQI+TMPB
      GO TO 50
   45 BNQO=BNQO+TMPB
   50 CONTINUE
      VQK(J)=QR(1)
      VQL(J)=QR(2)
   60 CONTINUE
C *** COMPUTE TOTAL VOLUMES TO BE RECHARGED (+) AND DICHARGED (-) ACROSS
C     BOUNDARY FOR ENTIRE SIMULATION ***
      TBNQI=TBNQI+BNQI*DT
      TBNQO=TBNQO+BNQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES IN THE FLOW IMBALANCE ***
      ER=ER-BNQI-BNQO
C ***
   65 IF(NPNB.LT.1) RETURN
C *** FORM COEFFICIENTS FOR NONLINEAR POINT SINKS ***
C *** FOR EACH POINT I: ***
      DO 100 I=1,NPNB
      K=KP(I)
      KE=IN(K)
      IF(KE.LT.0) GO TO 70
      DHC=DH(KE)
      GO TO 75
   70 DHC=DHB(-KE)
   75 IP=I+NBNC
C *** (1) COMPUTE HEAD AT N LEVEL, HO, AND HEAD AT N+1 LEVEL, HC ***
      HO=H(K)-DHC
      DHC=1.5*DHC
      HC=HO+DHC
C *** (2) REPRESENT ELEVATION, AND HEAD AND ELEVATION DIFFERENCE, BY
C         TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      ZPI=ZP(I)
      DHZ=ZPI-HO
C *** (3) COMPARE HEADS WITH CONTROLLING ELEVATION TO IDENTIFY LEAKAGE
C         CASE. COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-FLOW RATES
C         FOR: ***
      IF(HO.GT.ZPI) GO TO 80
```

```
      IF(HC.LT.ZPI) GO TO 100
C *** CASE (3) ***
      PHC=1.-DHZ/DHC
      TMPA=CB*PHC*(ZPI-HC)
      GO TO 90
   80 IF(HC.GT.ZPI) GO TO 85
C *** CASE (2) ***
      PHI=DHZ/DHC
      TMPA=CA*PHI*PHI*DHZ
      GO TO 90
C *** CASE (1) ***
   85 TMPA=ZPI-H(K)
C *** (4) COMPUTE VOLUMERIC-FLOW RATE ***
   90 TMPB=GC(IP)*TMPA
C *** (5) INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C           SPECIFIED-HEAD NODE ***
      IF(KE.LT.0) R(-KE)=R(-KE)-TMPB
C *** (6) SUM VOLUMETRIC-FLOW RATES ***
      PNQO=PNQO+TMPB
  100 CONTINUE
C *** COMPUTE TOTAL VOLUME DISCHARGED ACROSS POINTS FOR ENTIRE
C       SIMULATION ***
      TPNQO=TPNQO+PNQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES IN THE FLOW IMBALANCE ***
      ER=ER-PNQO
      RETURN
      END
```

```
      SUBROUTINE GNBLSS(H,R,GC,HRK,HRL,ZRK,ZRL,ZP,VQK,VQL,DT,IN,KR,LR,KP
     1,NBNC,NPNB)
C        COMPUTES FLOW-BALANCE COMPONENTS FOR NONLINEAR CAUCHY-TYPE
C        BOUNDARIES AND NONLINEAR POINT SINKS FOR STEADY-STATE
C        SIMULATIONS
      DIMENSION H(1),R(1),GC(1),HRK(1),HRL(1),ZRK(1),ZRL(1),ZP(1),VQK(1)
     1,VQL(1)
      DIMENSION IN(1),KR(1),LR(1),KP(1),NL(2)
      DIMENSION HR(2),ZR(2),QR(2)
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/GNBL/BNQI,BNQO,TBNQI,TBNQO,PNQO,TPNQO
C *** SET VOLUMETRIC-FLOW RATES TO ZERO ***
      BNQI=0.
      BNQO=0.
      PNQO=0.
C ***
      IF(NBNC.LT.1) GO TO 65
C *** FORM COEFFICIENTS FOR EACH SIDE J ON A NONLINEAR CAUCHY-TYPE
C     BOUNDARY ***
      DO 60 J=1,NBNC
      HR(1)=HRK(J)
      HR(2)=HRL(J)
      ZR(1)=ZRK(J)
      ZR(2)=ZRL(J)
      NL(1)=KR(J)
      NL(2)=LR(J)
C *** FOR EACH NODE I ON THE SIDE: ***
      DO 50 I=1,2
      L=NL(I)
C *** (1) REPRESENT HEADS AND CONTROLLING ELEVATION BY TEMPORARY
C         (UNSUBSCRIPTED) VARIABLES ***
      HL=H(L)
      HRI=HR(I)
      ZRI=ZR(I)
C *** (2) COMPARE HEAD WITH CONTROLLING ELEVATION TO IDENTIFY LEAKAGE
C         CASE. COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-FLOW RATES
C         FOR: ***
      IF(HL.GT.ZRI) GO TO 20
C *** CASE (4) ***
      TMPA=HRI-ZRI
      GO TO 30
C *** CASE (1) ***
   20 TMPA=HRI-HL
C *** (3) COMPUTE VOLUMETRIC-FLOW RATE ***
   30 TMPB=GC(J)*TMPA
      QR(I)=TMPB
C *** (4) INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C         SPECIFIED-HEAD NODE ***
      LE=-IN(L)
      IF(LE.GT.0) R(LE)=R(LE)-TMPB
C *** (5) SUM VOLUMETRIC-FLOW RATES ACCORDING TO SIGN, POSITIVE FOR
C         INFLOW ***
      IF(TMPB.LT.0.) GO TO 40
      BNQI=BNQI+TMPB
      GO TO 50
   40 BNQO=BNQO+TMPB
```

```
   50 CONTINUE
      VQK(J)=QR(1)
      VQL(J)=QR(2)
   60 CONTINUE
C *** COMPUTE TOTAL VOLUMES RECHARGED (+) AND DISCHARGED (-) ACROSS
C     BOUNDARY FOR SIMULATION ***
      TBNQI=BNQI*DT
      TBNQO=BNQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES IN THE FLOW IMBALANCE ***
      ER=ER-BNQI-BNQO
C ***
   65 IF(NPNB.LT.1) RETURN
C *** FORM COEFFICIENTS FOR NONLINEAR POINT SINKS ***
C *** FOR EACH POINT I: ***
      DO 80 I=1,NPNB
      K=KP(I)
      IP=I+NBNC
C *** REPRESENT HEAD AND CONTOLLING ELEVATION BY TEMPORARY
C     (UNSUBSCRIPTED) VARIABLES ***
      HK=H(K)
      ZPI=ZP(I)
C *** COMPARE HEAD WITH CONTROLLING ELEVATION TO DETERMINE LEAKAGE
C     CASE ***
      IF(HK.GT.ZPI) GO TO 70
C *** NO LEAKAGE, THUS NO COMPUTATION ***
      GO TO 80
C *** LEAKAGE, THUS COMPUTE VOLUMETRIC-FLOW RATE ***
   70 TMPB=GC(IP)*(ZPI-HK)
C *** INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C     SPECIFIED-HEAD NODES ***
      KE=-IN(K)
      IF(KE.GT.0) R(KE)=R(KE)-TMPB
C *** SUM VOLUMETRIC-FLOW RATES ***
      PNQO=PNQO+TMPB
   80 CONTINUE
C *** COMPUTE TOTAL VOLUME DISCHARGED ACROSS BOUNDARY FOR SIMULATION ***
      TPNQO=PNQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES INTO FLOW IMBALANCE ***
      ER=ER-PNQO
      RETURN
      END
```

```
      SUBROUTINE GNCHG(HRK,HRL,TIME,KR,LR,ISTP)
C        CHANGES VALUES OF EXTERNAL HEADS FOR NONLINEAR CAUCHY-TYPE
C        BOUNDARIES FOR TIME-VARIANT BOUNDARY CONDITIONS
      DIMENSION HRK(1),HRL(1)
      DIMENSION KR(1),LR(1)
      COMMON/CHG/NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (16I5)
    4 FORMAT (1H1,13X,34HCHANGES IN VALUES OF EXTERNAL HEAD
     1/1H ,12X,36HFOR NONLINEAR CAUCHY-TYPE BOUNDARIES
     2/1H ,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5
     3,11H TIME UNITS/1H ,15X,4HSIDE,19X,8HEXTERNAL/1H ,16X,3HNO.,8X
     4,4HNODE,9X,4HHEAD)
    6 FORMAT (I5,2F10.0)
    8 FORMAT (1H ,13X,2(I5,7X),G11.5/1H ,25X,I5,7X,G11.5)
C *** READ NUMBER OF BOUNDARIES TO BE CHANGED AND TIME-STEP NUMBER FOR
C     NEXT CHANGE ***
      READ(IIN,2) N,NGNCH
C *** WRITE HEADING ***
      WRITE(IOUT,4) ISTP,TIME
C *** READ AND WRITE VALUES OF NEW EXTERNAL HEADS ***
      DO 20 I=1,N
      READ(IIN,6) J,HRK(J),HRL(J)
      WRITE(IOUT,8) J,KR(J),HRK(J),LR(J),HRL(J)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GNCORR(H,DH,A,B,GC,HRK,HRL,ZRK,ZRL,ZP,IN,KR,LR,KP
     1,NBNC,NPNB)
C        FORMS COEFFICIENTS FOR NONLINEAR CAUCHY-TYPE BOUNDARIES AND
C        NONLINEAR POINT SINKS; ADDS TERMS TO MATRIX EQUATION FOR
C        THE CORRECTOR
      DIMENSION H(1),DH(1),A(1),B(1),GC(1),HRK(1),HRL(1),ZRK(1),ZRL(1)
     1,ZP(1)
      DIMENSION IN(1),KR(1),LR(1),KP(1)
      DIMENSION HR(2),ZR(2),NL(2)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
C *** COMPUTE WEIGHTING FACTORS FOR GALERKIN-IN-TIME FORMULATION ***
      CA=1./3.
      CB=2./3.
C ***
      IF(NBNC.LT.1) GO TO 110
C *** FORM COEFFICIENTS FOR EACH SIDE J ON A NONLINEAR CAUCHY-TYPE
C     BOUNDARY ***
      DO 100 J=1,NBNC
      HR(1)=HRK(J)
      HR(2)=HRL(J)
      ZR(1)=ZRK(J)
      ZR(2)=ZRL(J)
      NL(1)=KR(J)
      NL(2)=LR(J)
C *** FOR EACH NODE I ON THE SIDE: ***
      DO 50 I=1,2
      L=NL(I)
      LE=IN(L)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(LE.LT.0) GO TO 50
      N=MBWC*(LE-1)+1
C *** (2) REPRESENT HEADS, ELEVATIONS, AND DIFFERENCES BY TEMPORARY
C          (UNSUBSCRIPTED) VARIABLES ***
      HO=H(L)-DH(LE)
      DHP=1.5*DH(LE)
      HP=HO+DHP
      ZRI=ZR(I)
      HRI=HR(I)
      DRZ=HRI-ZRI
      DHZ=HO-ZRI
C *** (3) COMPARE HEADS WITH CONTROLLING ELEVATIONS TO IDENTIFY LEAKAGE
C         CASE. ADD LEAKAGE COEFFICIENT TO MAIN DIAGONAL, IF APPLICABLE,
C         AND COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT-SIDE TERM
C         FOR: ***
      IF(HO.LT.ZRI) GO TO 30
      IF(HP.LT.ZRI) GO TO 20
C *** CASE (1) ***
      A(N)=A(N)+GC(J)
      TMPA=HRI-H(L)
      GO TO 40
C *** CASE (2) ***
   20 PHI=DHZ/DHP
      TMPA=DRZ-CA*PHI*PHI*DHZ
      GO TO 40
   30 IF(HP.GT.ZRI) GO TO 35
C *** CASE (4) ***
```

```
      TMPA=DRZ
      GO TO 40
C *** CASE (3) ***
   35 PHC=1.+DHZ/DHP
      A(N)=A(N)+PHC*GC(J)
      TMPA=DRZ-CB*PHC*(HP-ZRI)
C *** (4) COMPUTE RIGHT-SIDE TERM AND ADD TO B-VECTOR ***
   40 B(LE)=B(LE)+GC(J)*TMPA
   50 CONTINUE
  100 CONTINUE
C ***
  110 IF(NPNB.LT.1) RETURN
C *** FORM COEFFICIENTS FOR NONLINEAR POINT SINKS ***
C *** FOR EACH POINT I: ***
      DO 150 I=1,NPNB
      K=KP(I)
      KE=IN(K)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(KE.LT.0) GO TO 150
      N=MBWC*(KE-1)+1
      IP=I+NBNC
C *** (2) REPRESENT HEADS, ELEVATION, AND DIFFERENCE BY TEMPORARY
C         (UNSUBSCRIPTED) VARIABLES ***
      HO=H(K)-DH(KE)
      DHP=1.5*DH(KE)
      HP=HO+DHP
      ZPI=ZP(I)
      DHZ=ZPI-HO
C *** (3) COMPARE HEADS WITH CONTROLLING ELEVATION TO IDENTIFY LEAKAGE
C         CASE. ADD LEAKAGE COEFFICIENT TO MAIN DIAGONAL, IF APPLICABLE,
C         AND COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT-SIDE TERM
C         FOR: ***
      IF(HO.GT.ZPI) GO TO 120
      IF(HP.LT.ZPI) GO TO 150
C *** CASE (3) ***
      PHC=1.-DHZ/DHP
      A(N)=A(N)+GC(IP)*PHC
      TMPA=CB*PHC*(ZPI-HP)
      GO TO 140
  120 IF(HP.GT.ZPI) GO TO 130
C *** CASE (2) ***
      PHI=DHZ/DHP
      TMPA=CA*PHI*PHI*DHZ
      GO TO 140
C *** CASE (1) ***
  130 A(N)=A(N)+GC(IP)
      TMPA=ZPI-H(K)
C *** (4) COMPUTE RIGHT-SIDE TERM AND ADD TO B-VECTOR ***
  140 B(KE)=B(KE)+GC(IP)*TMPA
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GNFMCO(XG,YG,GC,HRK,HRL,ZRK,ZRL,ZP,KR,LR,KP,INLZ,INLS
     1,NBNC,NPNB,NLCZ)
C        READS AND WRITES DATA AND COMPUTES LEAKAGE COEFFICIENTS FOR
C        NONLINEAR CAUCHY-TYPE BOUNDARIES AND NONLINEAR POINT SINKS
      DIMENSION XG(1),YG(1),GC(1),HRK(1),HRL(1),ZRK(1),ZRL(1),ZP(1)
      DIMENSION KR(1),LR(1),KP(1),INLZ(1),INLS(1)
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/SCLE/SCALE
C *** FORMAT LIST ***
    2 FORMAT (1H0,13X,52HNONLINEAR CAUCHY-TYPE BOUNDARY DATA BY BOUNDARY
     1 ZONE)
    3 FORMAT (1H0,5X,5HZONE ,I5,4X,9HCONTAINS ,I5,15H BOUNDARY SIDES
     1/1H ,7X,4HSIDE,6X,7HLEAKAGE,19X,8HEXTERNAL,4X,11HCONTROLLING/1H
     2,8X,3HNO.,4X,11HCOEFFICIENT,7X,4HNODE,8X,4HHEAD,7X,9HELEVATION)
    4 FORMAT (F10.0)
    5 FORMAT (1H0,8X,50HNONLINEAR CAUCHY-TYPE DATA READ, OUTPUT SUPPRESS
     1ED)
    6 FORMAT (3I5,5F10.0)
    7 FORMAT (1H ,2(5X,I5,6X,G11.5),3X,G11.5/1H ,32X,I5,3X,2(3X,G11.5))
    8 FORMAT (1H0,16X,25HNONLINEAR POINT SINK DATA/1H ,6X,5HPOINT,5X
     1,4HNODE,8X,7HLEAKAGE,6X,11HCONTROLLING/1H ,7X,3HNO.,7X,3HNO.,6X
     2,11HCOEFFICIENT,5X,9HELEVATION)
    9 FORMAT(1H0,8X,48HNONLINEAR POINT SINK DATA READ,OUTPUT SUPPRESSED)
   10 FORMAT (2I5,2F10.0)
   12 FORMAT (1H ,4X,I5,5X,I5,2X,2(5X,G11.5))
C *** COMPUTE FACTORING AND SCALING TERM FOR BOUNDARY LENGTHS ***
      CA=.5*SCALE
C *** READ INDICATOR VARIABLES FOR SUPRESSING PRINTOUT OF NONLINEAR-
C     BOUNDARY DATA ***
      READ(IIN,6) IPNC,IPNP
      IF(NBNC.LT.1) GO TO 30
C *** FOR NONLINEAR CAUCHY-TYPE BOUNDARIES: ***
      IF(IPNC.GT.0) GO TO 14
      WRITE(IOUT,2)
      GO TO 15
   14 WRITE(IOUT,5)
C *** (1) READ AND WRITE DATA DEFINING BOUNDARY SIDE I ***
   15 DO 19 NZ=1,NLCZ
      READ(IIN,6) KZ,NOS,IZIN
      IF(IPNC.EQ.0) WRITE(IOUT,3) KZ,NOS
      INLZ(NZ)=KZ
      INLS(NZ)=NOS
      IF(IZIN.GT.0) GO TO 17
      DO 16 I=1,NOS
      READ(IIN,6) J,KR(J),LR(J),GC(J),HRK(J),HRL(J),ZRK(J),ZRL(J)
      IF(IPNC.EQ.0) WRITE(IOUT,7) J,GC(J),KR(J),HRK(J),ZRK(J),LR(J)
     1,HRL(J),ZRL(J)
   16 CONTINUE
      GO TO 19
   17 READ(IIN,4) GCZ
      DO 18 I=1,NOS
      READ(IIN,6) J,KR(J),LR(J),HRK(J),HRL(J),ZRK(J),ZRL(J)
      GC(J)=GCZ
      IF(IPNC.EQ.0) WRITE(IOUT,7) J,GC(J),KR(J),HRK(J),ZRK(J),LR(J)
     1,HRL(J),ZRL(J)
   18  CONTINUE
```

```
  19  CONTINUE
C *** (2) COMPUTE LENGTH OF BOUNDARY SIDE ***
      DO 20 J=1,NBNC
      K=KR(J)
      L=LR(J)
      TMPA=XG(K)-XG(L)
      TMPB=YG(K)-YG(L)
      DIST=CA*(TMPA*TMPA+TMPB*TMPB)**.5
C *** (3) COMPUTE LEAKAGE COEFFICIENT ***
   20 GC(J)=GC(J)*DIST
C ***
   30 IF(NPNB.LT.1) RETURN
C *** FOR NONLINEAR POINT SINKS: ***
      IF(IPNP.GT.0) GO TO 32
      WRITE(IOUT,8)
      GO TO 34
   32 WRITE(IOUT,9)
C *** (1) READ AND WRITE DATA DEFINING THE BOUNDARY ***
   34 DO 40 J=1,NPNB
      READ(IIN,10) I,KP(I),GCP,ZP(I)
      IF(IPNP.EQ.0) WRITE(IOUT,12) I,KP(I),GCP,ZP(I)
C *** (2) STORE LEAKAGE COEFFICIENT IN GC ***
      GC(I+NBNC)=GCP
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GNINIT(G,IGCA,IHRK,IHRL,IZRK,IZRL,IZPA,IKRA,ILRA,IKPA
     1,INZA,INSA,NBNC,NPNB,NLCZ)
C        READS DATA, ASSIGNS STORAGE, AND INITIALIZES STORAGE LOCATIONS
C        TO ZERO FOR NONLINEAR CAUCHY-TYPE BOUNDARIES AND NONLINEAR
C        POINT SINKS
      DIMENSION G(1)
      COMMON/GDIM/ISUM
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (16I5)
    4 FORMAT (1H0,54HNO. OF NONLINEAR CAUCHY-TYPE BOUNDARIES (NBNC).....
     $ = ,I5/
     $1H ,54HNO. OF NONLINEAR CAUCHY-TYPE BOUNDARY ZONES (NLCZ). = ,I5/
     $1H ,54HNO. OF NONLINEAR POINT SINKS (NPNB)................ = ,I5)
    6 FORMAT (1H0,64HNONLINEAR CAUCHY-TYPE BOUNDARIES AND (OR) NONLINEAR
     1 POINT SINKS:/1H ,38HNOW G MUST BE DIMENSIONED TO AT LEAST ,I6)
C *** READ AND WRITE NUMBER OF NONLINEAR CAUCHY-TYPE BOUNDARIES AND
C     NONLINEAR POINT SINKS ***
      READ(IIN,2) NBNC,NLCZ,NPNB
      WRITE(IOUT,4) NBNC,NLCZ,NPNB
C *** DEFINE ADDRESSES OF NEW ARRAYS WITHIN G ***
      IGCA=ISUM
      ISUM=ISUM+NBNC+NPNB
      IHRK=ISUM
      ISUM=ISUM+NBNC
      IHRL=ISUM
      ISUM=ISUM+NBNC
      IZRK=ISUM
      ISUM=ISUM+NBNC
      IZRL=ISUM
      ISUM=ISUM+NBNC
      IKRA=ISUM
      ISUM=ISUM+NBNC
      ILRA=ISUM
      ISUM=ISUM+NBNC
      INZA=ISUM
      ISUM=ISUM+NLCZ
      INSA=ISUM
      ISUM=ISUM+NLCZ
      IZPA=ISUM
      ISUM=ISUM+NPNB
      IKPA=ISUM
      ISUM=ISUM+NPNB
C *** WRITE SIZE OF G ***
      WRITE(IOUT,6) ISUM
C *** INITIALIZE G ***
      DO 10 I=IGCA,ISUM
   10 G(I)=0.
      RETURN
      END
```

```
      SUBROUTINE GNPRED(H,A,B,GC,HRK,HRL,ZRK,ZRL,ZP,IN,KR,LR,KP,NBNC
     1,NPNB)
C        FORMS COEFFICIENTS FOR NONLINEAR CAUCHY-TYPE BOUNDARIES AND
C        NONLINEAR POINT SINKS; ADDS TERMS TO MATRIX EQUATION FOR THE
C        PREDICTOR
      DIMENSION H(1),A(1),B(1),GC(1),HRK(1),HRL(1),ZRK(1),ZRL(1),ZP(1)
      DIMENSION IN(1),KR(1),LR(1),KP(1)
      DIMENSION HR(2),ZR(2),NL(2)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
C ***
      IF(NBNC.LT.1) GO TO 50
C *** FORM COEFFICIENTS FOR EACH SIDE J ON A NONLINEAR CAUCHY-TYPE
C     BOUNDARY ***
      DO 40 J=1,NBNC
      HR(1)=HRK(J)
      HR(2)=HRL(J)
      ZR(1)=ZRK(J)
      ZR(2)=ZRL(J)
      NL(1)=KR(J)
      NL(2)=LR(J)
C *** FOR EACH NODE I ON THE BOUNDARY: ***
      DO 30 I=1,2
      L=NL(I)
      LE=IN(L)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(LE.LT.0) GO TO 30
C *** (2) COMPARE AQUIFER HEAD WITH CONTROLLING ELEVATION TO IDENTIFY
C         LEAKAGE CASE. ADD LEAKAGE COEFFICIENT TO MAIN DIAGONAL, IF
C         APPLICABLE, AND COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT SIDE
C         FOR: ***
      IF(H(L).LT.ZR(I)) GO TO 10
      N=MBWC*(LE-1)+1
C *** CASE (1) ***
      A(N)=A(N)+GC(J)
      TMPA=HR(I)-H(L)
      GO TO 20
C *** CASE (4) ***
   10 TMPA=HR(I)-ZR(I)
C *** (3) COMPUTE RIGHT-SIDE TERM AND ADD TO B-VECTOR ***
   20 B(LE)=B(LE)+GC(J)*TMPA
   30 CONTINUE
   40 CONTINUE
C ***
   50 IF(NPNB.LT.1) RETURN
C *** FORM COEFFICIENTS FOR NONLINEAR POINT SINKS ***
C *** FOR EACH POINT I: ***
      DO 60 I=1,NPNB
      IP=I+NBNC
      K=KP(I)
      KE=IN(K)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(KE.LT.0) GO TO 60
C *** (2) COMPARE AQUIFER HEAD WITH CONTROLLING ELEVATION TO DETERMINE
C         IF LEAKAGE CASE (1) APPLIES ***
      IF(H(K).LT.ZP(I)) GO TO 60
      N=MBWC*(KE-1)+1
```

```
C *** (3) COMPUTE AND ADD TERMS TO MAIN DIAGONAL AND RIGHT SIDE ***
      A(N)=A(N)+GC(IP)
      B(KE)=B(KE)+GC(IP)*(ZP(I)-H(K))
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HCALC(H,B,IN)
C        COMPUTES MEAN HEAD DURING TIME STEP AT ALL ACTIVE NODES
      DIMENSION H(1),B(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      DO 10 I=1,NEQ
      J=IN(I)
      IF(J.GT.0) H(I)=H(I)+B(J)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HCALWT(H,DH,B,IN)
C        COMPUTES MEAN HEAD AND MEAN HEAD CHANGE DURING TIME STEP AT
C        ALL ACTIVE NODES
      DIMENSION H(1),DH(1),B(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      DO 10 I=1,NEQ
      J=IN(I)
      IF(J.LT.0) GO TO 10
      H(I)=H(I)+B(J)
      DH(J)=DH(J)+B(J)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INITB(TITLE,G,TIME,NBCZ)
C        READS PROBLEM SPECIFICATIONS, DEFINES AND INITIALIZES VARIABLES
C        FOR BASIC VERSION OF PROGRAM THAT USES BAND MATRIX SOLVER
      DIMENSION TITLE(20),G(1)
      COMMON/GDIM/ISUM
      COMMON/ADR/IAA,IARA,IXGA,IYGA,IATA,IQA,IBA,IHA,IHRA,IHBA,IALA,IQBA
     1,ICKA,ICLA,IHKA,IHLA,IDTA,IJPA,INA,INDA,IKA,ILA,IDZA,IDSA
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/TBAL/TSA,TWQI,TWQO,TDQI,TDQO,TLQI,TLQO,TBQI,TBQO,THBQI
     1,THBQO,TER
      COMMON/VNLBL/VNLQI,VNLQO,TNLQI,TNLQO
      COMMON/GNBL/BNQI,BNQO,TBNQI,TBNQO,PNQO,TPNQO
C *** FORMAT LIST ***
    8 FORMAT (1H1)
   10 FORMAT (20A4)
   12 FORMAT (1H ,20A4)
   14 FORMAT (16I5)
   16 FORMAT (1H0,54HNO. OF ELEMENTS (NELS)...........................
     1 = ,I5/
     $1H ,54HNO. OF NODES (NNDS)................................ = ,I5/
     $1H ,54HMAX. NO. OF TIME STEPS PER STRESS PERIOD (MXSTPS).. = ,I5/
     $1H ,54HNO. OF STRESS PERIODS (NPER)....................... = ,I5/
     $1H ,54HNO. OF ZONES (NZNS)................................ = ,I5/
     $1H ,54HNO. OF POINT FLOWS (NWELS)......................... = ,I5/
     $1H ,54HNO. OF CAUCHY-TYPE BOUNDARY ELEMENT SIDES (NQBND).. = ,I5/
     $1H ,54HNO. OF CAUCHY-TYPE BOUNDARY ZONES (NBCZ)........... = ,I5/
     $1H ,54HNO. OF SPECIFIED HEADS (NHDS)...................... = ,I5/
     $1H ,54HMAXIMUM CONDENSED BAND WIDTH (MBWC)................ = ,I5/
     $1H ,54HMAXIMUM MATRIX BAND WIDTH (MBW).................... = ,I5)
   18 FORMAT (33H0DIMENSION OF G MUST BE AT LEAST ,I6)
C *** DEFINE FILE NUMBERS ***
      IIN=50
      IOUT=60
      ITA=55
      ITB=56
C *** READ AND WRITE PROBLEM TITLE ***
      WRITE(IOUT,8)
      DO 20 I=1,3
      READ(IIN,10) (TITLE(K),K=1,20)
      WRITE(IOUT,12) (TITLE(K),K=1,20)
   20 CONTINUE
C *** READ AND WRITE PROBLEM SPECIFICATIONS ***
      READ(IIN,14) NELS,NNDS,MXSTPS,NPER,NZNS,NWELS,NQBND,NBCZ,NHDS,MBWC
     1,MBW
      WRITE(IOUT,16) NELS,NNDS,MXSTPS,NPER,NZNS,NWELS,NQBND,NBCZ,NHDS
     1,MBWC,MBW
C *** DEFINE ADDRESSES OF ARRAYS WITHIN G ***
      NA=(MBWC+1)*NNDS
      NB=2*NELS
      NC=MBW*(NNDS-NHDS-MBW)+(MBW*(MBW+1))/2
      ND=(MBWC-1)*NNDS
      NE=4*NELS
      NF=NNDS-NHDS
      IF(NELS.GT.NF) NF=NELS
```

```
      ISUM=1
      IAA=ISUM
      ISUM=ISUM+NA
      IARA=ISUM
      ISUM=ISUM+NB
      INDA=ISUM
      ISUM=ISUM+NE
      IXGA=ISUM
      ISUM=ISUM+NNDS
      IYGA=ISUM
      ISUM=ISUM+NNDS
      IF(NC.GE.ISUM) ISUM=NC+1
      IATA=ISUM
      ISUM=ISUM+MBW
      IQA=ISUM
      ISUM=ISUM+NNDS
      IBA=ISUM
      ISUM=ISUM+NF
      IHA=ISUM
      ISUM=ISUM+NNDS
      IHRA=ISUM
      ISUM=ISUM+NNDS
      IHBA=ISUM
      ISUM=ISUM+NHDS
      IALA=ISUM
      ISUM=ISUM+NQBND
      IQBA=ISUM
      ISUM=ISUM+NQBND
      ICKA=ISUM
      ISUM=ISUM+NQBND
      ICLA=ISUM
      ISUM=ISUM+NQBND
      IHKA=ISUM
      ISUM=ISUM+NQBND
      IHLA=ISUM
      ISUM=ISUM+NQBND
      IDZA=ISUM
      ISUM=ISUM+NBCZ
      IDSA=ISUM
      ISUM=ISUM+NBCZ
      IDTA=ISUM
      ISUM=ISUM+MXSTPS
      IJPA=ISUM
      ISUM=ISUM+ND
      INA=ISUM
      ISUM=ISUM+NNDS+1
      IKA=ISUM
      ISUM=ISUM+NQBND
      ILA=ISUM
      ISUM=ISUM+NQBND
C *** WRITE SIZE OF G ***
      WRITE(IOUT,18) ISUM
C *** INITIALIZE G ***
      DO 30 I=1,ISUM
```

```
   30 G(I)=0.
C *** INITIALIZE TIME AND FLOW-BALANCE COMPONENTS ***
      TIME=0.
      TSA=0.
      TWQI=0.
      TWQO=0.
      TDQI=0.
      TDQO=0.
      TLQI=0.
      TLQO=0.
      TBQI=0.
      TBQO=0.
      THBQI=0.
      THBQO=0.
      PNQO=0.
      TPNQO=0.
      VNLQI=0.
      VNLQO=0.
      TNLQI=0.
      TNLQO=0.
      BNQI=0.
      BNQO=0.
      TBNQI=0.
      TBNQO=0.
      TER=0.
      RETURN
      END
```

```
      SUBROUTINE INITCG(TITLE,G,TIME,TOL,IAFA,IXA,IPA,IRA,NBCZ)
C        READS PROBLEM SPECIFICATIONS, DEFINES AND INITIALIZES VARIABLES
C        FOR BASIC VERSION OF PROGRAM THAT USES MICCG MATRIX SOLVER
      DIMENSION TITLE(20),G(1)
      COMMON/GDIM/ISUM
      COMMON/ADR/IAA,IARA,IXGA,IYGA,IATA,IQA,IBA,IHA,IHRA,IHBA,IALA,IQBA
     1,ICKA,ICLA,IHKA,IHLA,IDTA,IJPA,INA,INDA,IKA,ILA,IDZA,IDSA
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/TBAL/TSA,TWQI,TWQO,TDQI,TDQO,TLQI,TLQO,TBQI,TBQO,THBQI
     1,THBQO,TER
      COMMON/VNLBL/VNLQI,VNLQO,TNLQI,TNLQO
      COMMON/GNBL/BNQI,BNQO,TBNQI,TBNQO,PNQO,TPNQO
C *** FORMAT LIST ***
    8 FORMAT (1H1)
   10 FORMAT (20A4)
   12 FORMAT (1H ,20A4)
   14 FORMAT (16I5)
   15 FORMAT (F10.0)
   16 FORMAT (1H0,54HNO. OF ELEMENTS (NELS)...........................
     1 = ,I5/
     $1H ,54HNO. OF NODES (NNDS)................................ = ,I5/
     $1H ,54HMAX. NO. OF TIME STEPS PER STRESS PERIOD (MXSTPS).. = ,I5/
     $1H ,54HNO. OF STRESS PERIODS (NPER)....................... = ,I5/
     $1H ,54HNO. OF ZONES (NZNS)................................ = ,I5/
     $1H ,54HNO. OF POINT FLOWS (NWELS)......................... = ,I5/
     $1H ,54HNO. OF CAUCHY-TYPE BOUNDARY ELEMENT SIDES (NQBND).. = ,I5/
     $1H ,54HNO. OF CAUCHY-TYPE BOUNDARY ZONES (NBCZ)........... = ,I5/
     $1H ,54HNO. OF SPECIFIED HEADS (NHDS)...................... = ,I5/
     $1H ,54HMAXIMUM CONDENSED BAND WIDTH (MBWC)................ = ,I5/
     $1H ,54HMAXIMUM NO. OF ITERATIONS FOR MICCG (NIT).......... = ,I5/
     $1H ,54HCLOSURE TOLERANCE FOR MICCG (TOL).................. =
     $,G11.5)
   18 FORMAT (33H0DIMENSION OF G MUST BE AT LEAST ,I6)
C *** DEFINE FILE NUMBERS ***
      IIN=50
      IOUT=60
      ITA=55
      ITB=56
C *** READ AND WRITE PROBLEM TITLE ***
      WRITE(IOUT,8)
      DO 20 I=1,3
      READ(IIN,10) (TITLE(K),K=1,20)
      WRITE(IOUT,12) (TITLE(K),K=1,20)
   20 CONTINUE
C *** READ AND WRITE PROBLEM SPECIFICATIONS ***
      READ(IIN,14) NELS,NNDS,MXSTPS,NPER,NZNS,NWELS,NQBND,NBCZ,NHDS,MBWC
     1,NIT
      READ(IIN,15) TOL
      WRITE(IOUT,16) NELS,NNDS,MXSTPS,NPER,NZNS,NWELS,NQBND,NBCZ,NHDS
     1,MBWC,NIT,TOL
C *** DEFINE ADDRESSES OF ARRAYS WITHIN G ***
      NA=(MBWC+1)*NNDS
      NB=2*NELS
      NC=NA+(MBWC+3)*(NNDS-NHDS)
```

```
ND=(MBWC-1)*NNDS
NE=4*NELS
NF=NNDS-NHDS
NG=NF
IF(NELS.GT.NF) NF=NELS
ISUM=1
IAA=ISUM
ISUM=ISUM+NA
IARA=ISUM
ISUM=ISUM+NB
INDA=ISUM
ISUM=ISUM+NE
IXGA=ISUM
ISUM=ISUM+NNDS
IYGA=ISUM
ISUM=ISUM+NNDS
IAFA=NA+1
IXA=IAFA+NG*MBWC
IPA=IXA+NG
IRA=IPA+NG
IF(NC.GE.ISUM) ISUM=NC+1
IQA=ISUM
ISUM=ISUM+NNDS
IBA=ISUM
ISUM=ISUM+NF
IHA=ISUM
ISUM=ISUM+NNDS
IHRA=ISUM
ISUM=ISUM+NNDS
IHBA=ISUM
ISUM=ISUM+NHDS
IALA=ISUM
ISUM=ISUM+NQBND
IQBA=ISUM
ISUM=ISUM+NQBND
ICKA=ISUM
ISUM=ISUM+NQBND
ICLA=ISUM
ISUM=ISUM+NQBND
IHKA=ISUM
ISUM=ISUM+NQBND
IHLA=ISUM
ISUM=ISUM+NQBND
IDZA=ISUM
ISUM=ISUM+NBCZ
IDSA=ISUM
ISUM=ISUM+NBCZ
IDTA=ISUM
ISUM=ISUM+MXSTPS
IJPA=ISUM
ISUM=ISUM+ND
INA=ISUM
ISUM=ISUM+NNDS+1
IKA=ISUM
ISUM=ISUM+NQBND
```

```
      ILA=ISUM
      ISUM=ISUM+NQBND
C *** WRITE SIZE OF G ***
      WRITE(IOUT,18) ISUM
C *** INITIALIZE G ***
      DO 30 I=1,ISUM
   30 G(I)=0.
C *** INITIALIZE TIME AND FLOW-BALANCE COMPONENTS ***
      TIME=0.
      TSA=0.
      TWQI=0.
      TWQO=0.
      TDQI=0.
      TDQO=0.
      TLQI=0.
      TLQO=0.
      TBQI=0.
      TBQO=0.
      THBQI=0.
      THBQO=0.
      PNQO=0.
      TPNQO=0.
      VNLQI=0.
      VNLQO=0.
      TNLQI=0.
      TNLQO=0.
      BNQI=0.
      BNQO=0.
      TBNQI=0.
      TBNQO=0.
      TER=0.
      RETURN
      END
```

```
      SUBROUTINE MASBAL(H,DHB,HR,HK,HL,QBND,CFDK,CFDL,A,Q,B,R,VQK,VQL,DT
     1,JPT,IN,KQB,LQB)
C        COMPUTES FLOW BALANCE FOR BASIC, LINEAR VERSION OF PROGRAM
      DIMENSION H(1),DHB(1),HR(1),HK(1),HL(1),QBND(1),CFDK(1),CFDL(1)
     1,A(1),Q(1),B(1),R(1),VQK(1),VQL(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** INITIALIZE FLOW-BALANCE COMPONENTS ***
      IF(NHDS.LE.0) GO TO 15
      DO 10 I=1,NHDS
   10 R(I)=0.
   15 SA=0.
      VLQI=0.
      VLQO=0.
      MBM1=MBWC-1
      IW=MBWC+1
      NC=1
      NP=0
      DTM=3./(2.*DT)
C *** COMPUTE TOTAL ACCUMULATION IN STORAGE PER UNIT TIME AND LEAKAGE
C     ACROSS AN AQUITARD PER UNIT TIME, AND ADD THESE CONTRIBUTIONS
C     AND Q INTO THE NODAL FLUX, R, AT SPECIFIED-HEAD NODES ***
      DO 90 I=1,NNDS
      NVL=NC+1
      ND=NVL
      TMPB=A(NVL)*(HR(I)-H(I))
      IF(TMPB.GT.0.) GO TO 20
      VLQO=VLQO+TMPB
      GO TO 25
   20 VLQI=VLQI+TMPB
   25 K=IN(I)
      IF(K.GT.0) GO TO 35
      K=-K
      TMPA=A(NC)*DTM*DHB(K)
      SA=SA+TMPA
      R(K)=R(K)-Q(I)+TMPA-TMPB
      DO 30 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 80
      TMPA=A(ND+J)*(H(L)-H(I))
      M=-IN(L)
      IF(M.GT.0) R(M)=R(M)-TMPA
   30 R(K)=R(K)+TMPA
      GO TO 80
   35 SA=SA+A(NC)*DTM*B(K)
      DO 70 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 80
      M=-IN(L)
      IF(M.GT.0) R(M)=R(M)+A(ND+J)*(H(I)-H(L))
   70 CONTINUE
   80 NC=NC+IW
   90 NP=NP+MBM1
C *** ADD HEAD-DEPENDENT FLUX INTO TOTAL INFLOW OR OUTFLOW PER UNIT TIME
C     ACROSS CAUCHY-TYPE BOUNDARIES AND INTO R ***
```

```
      BQI=0.
      BQO=0.
      IF(NQBND.LE.0) GO TO 140
      DO 130 I=1,NQBND
      TMPA=0.
      TMPB=0.
      K=KQB(I)
      IF(K.LT.0) GO TO 95
      L=LQB(I)
      M=-IN(K)
      N=-IN(L)
      TMPA=CFDK(I)*(HK(I)-H(K))
      TMPB=CFDL(I)*(HL(I)-H(L))
      IF(M.GT.0) R(M)=R(M)-TMPA
      IF(N.GT.0) R(N)=R(N)-TMPB
   95 TMPA=QBND(I)*CFDK(I)+TMPA
      VQK(I)=TMPA
      IF(TMPA.GT.0.) GO TO 100
      BQO=BQO+TMPA
      GO TO 110
  100 BQI=BQI+TMPA
  110 TMPB=QBND(I)*CFDL(I)+TMPB
      VQL(I)=TMPB
      IF(TMPB.GT.0.) GO TO 120
      BQO=BQO+TMPB
      GO TO 130
  120 BQI=BQI+TMPB
  130 CONTINUE
C *** COMPUTE TOTAL FLOW IMBALANCE PER UNIT TIME ***
  140 ER=SA-WQI-WQO-DQI-DQO-VLQI-VLQO-BQI-BQO
      RETURN
      END
```

```
      SUBROUTINE MASOUT(R,DT,ISTP)
C        COMPUTES TOTAL FLOW BALANCE ACROSS SPECIFIED-HEAD BOUNDARIES
C        AND WRITES TOTAL FLOW-BALANCE COMPONENTS
      DIMENSION R(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/TBAL/TSA,TWQI,TWQO,TDQI,TDQO,TLQI,TLQO,TBQI,TBQO,THBQI
     1,THBQO,TER
      COMMON/VNLBL/VNLQI,VNLQO,TNLQI,TNLQO
      COMMON/GNBL/BNQI,BNQO,TBNQI,TBNQO,PNQO,TPNQO
C *** FORMAT LIST ***
    2 FORMAT (1H1,20X,21HWATER BALANCE SUMMARY/1H0,12X,35HVOLUMETRIC RAT
     1ES FOR TIME STEP NO. ,I3)
    4 FORMAT (56H ACCUMULATION OF WATER IN STORAGE.................... =
     1 ,G11.5/
     $56H RECHARGE FROM POINT SOURCES......................... = ,G11.5/
     $56H DISCHARGE FROM POINT SINKS.......................... = ,G11.5/
     $56H RECHARGE FROM DISTRIBUTED SOURCES................... = ,G11.5/
     $56H DISCHARGE FROM DISTRIBUTED SINKS.................... = ,G11.5/
     $56H RECHARGE FROM STEADY OR TRANSIENT LEAKAGE........... = ,G11.5/
     $56H DISCHARGE FROM STEADY OR TRANSIENT LEAKAGE.......... = ,G11.5/
     $56H RECHARGE ACROSS CAUCHY-TYPE BOUNDARIES.............. = ,G11.5/
     $56H DISCHARGE ACROSS CAUCHY-TYPE BOUNDARIES............. = ,G11.5/
     $56H RECHARGE ACROSS SPECIFIED-HEAD BOUNDARIES........... = ,G11.5/
     $56H DISCHARGE ACROSS SPECIFIED-HEAD BOUNDARIES.......... = ,G11.5/
     $56H DISCHARGE FROM NONLINEAR POINT SINKS................ = ,G11.5/
     $56H RECHARGE FROM NONLINEAR STEADY LEAKAGE.............. = ,G11.5/
     $56H DISCHARGE FROM NONLINEAR STEADY LEAKAGE............. = ,G11.5/
     $56H RECHARGE FROM NONLINEAR CAUCHY-TYPE BOUNDARIES...... = ,G11.5/
     $56H DISCHARGE FROM NONLINEAR CAUCHY-TYPE BOUNDARIES..... = ,G11.5/
     $56H FLOW IMBALANCE...................................... = ,G11.5)
    6 FORMAT (1H0,10X,43HTOTAL VOLUMES SINCE BEGINNING OF SIMULATION)
C *** COMPUTE TOTAL INFLOW OR OUTFLOW PER UNIT TIME ACROSS SPECIFIED-
C     HEAD BOUNDARIES ***
      HBQO=0.
      HBQI=0.
      IF(NHDS.LE.0) GO TO 170
      DO 160 I=1,NHDS
      IF(R(I).GT.0.) GO TO 150
      HBQO=HBQO+R(I)
      GO TO 160
  150 HBQI=HBQI+R(I)
  160 CONTINUE
C *** MODIFY FLOW IMBALANCE AND COMPUTE TOTAL INFLOW OR OUTFLOW
C     SINCE T=0 ACROSS SPECIFIED-HEAD BOUNDARIES ***
      ER=ER-HBQO-HBQI
      THBQO=THBQO+HBQO*DT
      THBQI=THBQI+HBQI*DT
C *** WRITE TOTAL INFLOWS AND OUTFLOWS PER UNIT TIME ***
  170 WRITE(IOUT,2) ISTP
      WRITE(IOUT,4) SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,HBQI,HBQO
     1,PNQO,VNLQI,VNLQO,BNQI,BNQO,ER
C *** COMPUTE TOTAL INFLOWS AND OUTFLOWS SINCE T=0 ***
      TSA=TSA+SA*DT
      TWQI=TWQI+WQI*DT
```

```
      TWQO=TWQO+WQO*DT
      TDQI=TDQI+DQI*DT
      TDQO=TDQO+DQO*DT
      TLQI=TLQI+VLQI*DT
      TLQO=TLQO+VLQO*DT
      TBQI=TBQI+BQI*DT
      TBQO=TBQO+BQO*DT
      TER=TER+ER*DT
      WRITE(IOUT,6)
C *** WRITE TOTAL INFLOWS AND OUTFLOWS SINCE T=0 ***
      WRITE(IOUT,4) TSA,TWQI,TWQO,TDQI,TDQO,TLQI,TLQO,TBQI,TBQO,THBQI
     1,THBQO,TPNQO,TNLQI,TNLQO,TBNQI,TBNQO,TER
      RETURN
      END
```

```
      SUBROUTINE MBALCB(H,DHB,HR,HK,HL,QBND,CFDK,CFDL,A,Q,B,R,VQK,VQL
     1,CH,CBQ,GMA,DT,JPT,IN,KQB,LQB)
C        COMPUTES FLOW BALANCE FOR BASIC, LINEAR VERSION OF PROGRAM
C        WITH TRANSIENT LEAKAGE
      DIMENSION H(1),DHB(1),HR(1),HK(1),HL(1),QBND(1),CFDK(1),CFDL(1)
     1,A(1),Q(1),B(1),R(1),VQK(1),VQL(1),CH(1),CBQ(1),GMA(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** INITIALIZE FLOW-BALANCE COMPONENTS ***
      IF(NHDS.LE.0) GO TO 15
      DO 10 I=1,NHDS
   10 R(I)=0.
   15 SA=0.
      VLQI=0.
      VLQO=0.
      MBM1=MBWC-1
      IW=MBWC+1
      NC=1
      NP=0
      N=0
      DTM=3./(2.*DT)
C *** COMPUTE TOTAL ACCUMULATION IN STORAGE PER UNIT TIME AND LEAKAGE
C     ACROSS AN AQUITARD PER UNIT TIME, AND ADD THESE CONTRIBUTIONS
C     AND Q INTO THE NODAL FLUX, R, AT SPECIFIED-HEAD NODES ***
      DO 90 I=1,NNDS
      NVL=NC+1
      ND=NVL
      TMPB=A(NVL)*(HR(I)-H(I))
      K=IN(I)
      IF(K.GT.0) GO TO 35
      K=-K
      TMPA=A(NC)*DTM*DHB(K)
      SA=SA+TMPA
      IF(GMA(I).LT.1.E-30) GO TO 20
      N=N+1
      TMPB=TMPB+CBQ(N)-CH(N)*DHB(K)
   20 R(K)=R(K)-Q(I)+TMPA-TMPB
      DO 30 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 75
      TMPA=A(ND+J)*(H(L)-H(I))
      M=-IN(L)
      IF(M.GT.0) R(M)=R(M)-TMPA
   30 R(K)=R(K)+TMPA
      GO TO 75
   35 SA=SA+A(NC)*DTM*B(K)
      IF(GMA(I).LT.1.E-30) GO TO 40
      N=N+1
      TMPB=TMPB+CBQ(N)-CH(N)*B(K)
   40 DO 70 J=1,MBM1
      L=JPT(NP+J)
      IF(L.GT.NNDS) GO TO 75
      M=-IN(L)
      IF(M.GT.0) R(M)=R(M)+A(ND+J)*(H(I)-H(L))
   70 CONTINUE
```

```
   75 IF(TMPB.GT.0.) GO TO 80
      VLQO=VLQO+TMPB
      GO TO 85
   80 VLQI=VLQI+TMPB
   85 NC=NC+IW
   90 NP=NP+MBM1
C *** ADD HEAD-DEPENDENT FLUX INTO TOTAL INFLOW OR OUTFLOW PER UNIT TIME
C     ACROSS CAUCHY-TYPE BOUNDARIES AND INTO R ***
      BQI=0.
      BQO=0.
      IF(NQBND.LE.0) GO TO 140
      DO 130 I=1,NQBND
      TMPA=0.
      TMPB=0.
      K=KQB(I)
      IF(K.LT.0) GO TO 95
      L=LQB(I)
      M=-IN(K)
      N=-IN(L)
      TMPA=CFDK(I)*(HK(I)-H(K))
      TMPB=CFDL(I)*(HL(I)-H(L))
      IF(M.GT.0) R(M)=R(M)-TMPA
      IF(N.GT.0) R(N)=R(N)-TMPB
   95 TMPA=QBND(I)*CFDK(I)+TMPA
      VQK(I)=TMPA
      IF(TMPA.GT.0.) GO TO 100
      BQO=BQO+TMPA
      GO TO 110
  100 BQI=BQI+TMPA
  110 TMPB=QBND(I)*CFDL(I)+TMPB
      VQL(1)=TMPB
      IF(TMPB.GT.0.) GO TO 120
      BQO=BQO+TMPB
      GO TO 130
  120 BQI=BQI+TMPB
  130 CONTINUE
C *** COMPUTE TOTAL FLOW IMBALANCE PER UNIT TIME ***
  140 ER=SA-WQI-WQO-DQI-DQO-VLQI-VLQO-BQI-BQO
      RETURN
      END
```

```
      SUBROUTINE MBALWT(H,DH,DHB,HR,HK,HL,QBND,CFDK,CFDL,A,Q,R,VQL,THK
     1,DTK,TOP,ASY,DT,JPT,IN,KQB,LQB)
C        COMPUTES FLOW BALANCE FOR BASIC, WATER-TABLE AQUIFER VERSION OF
C        PROGRAM
      DIMENSION H(1),DH(1),DHB(1),HR(1),HK(1),HL(1),QBND(1),CFDK(1)
     1,CFDL(1),A(1),Q(1),R(1),VQL(1),THK(1),DTK(1),TOP(1),ASY(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** COMPUTE DTK-VECTOR, AND COMPUTE CAPACITANCE AT STORAGE CONVERSION
C     NODES ***
      WF=2./3.
      NC=1
      IW=MBWC+1
      DO 14 I=1,NNDS
      DTK(I)=0.
      J=IN(I)
      IF(J.GT.0) GO TO 2
      DHC=DHB(-J)
      GO TO 4
    2 DHC=DH(J)
    4 HO=H(I)-DHC
      DHC=1.5*DHC
      HC=HO+DHC
      IF(HO.LT.TOP(I)) GO TO 6
      IF(HC.GE.TOP(I)) GO TO 14
      DTK(I)=HC-TOP(I)
      IF(THK(I)+DTK(I).LT.0.) DTK(I)=-THK(I)
      THET=(TOP(I)-HO)/DHC
      A(NC)=THET*(A(NC)-ASY(I))+ASY(I)
      GO TO 12
    6 IF(HC.LT.TOP(I)) GO TO 8
      DTK(I)=TOP(I)-HO
      THET=DTK(I)/DHC
      A(NC)=THET*(ASY(I)-A(NC))+A(NC)
      GO TO 12
    8 IF(THK(I)+DHC.GT.0.) GO TO 10
      DTK(I)=0.
      IF(THK(I).GT.0.) DTK(I)=-THK(I)
      GO TO 11
   10 DTK(I)=DHC
      IF(THK(I).LT.0.) DTK(I)=DHC+THK(I)
   11 A(NC)=ASY(I)
   12 DTK(I)=WF*DTK(I)
   14 NC=NC+IW
C *** INITIALIZE FLOW-BALANCE COMPONENTS ***
      IF(NHDS.LE.0) GO TO 18
      DO 16 I=1,NHDS
   16 R(I)=0.
   18 SA=0.
      VLQI=0.
      VLQO=0.
      C=1./16.
      MBM1=MBWC-1
      NC=1
      NP=0
```

```
      DTM=3./(2.*DT)
C *** COMPUTE TOTAL ACCUMULATION IN STORAGE PER UNIT TIME AND LEAKAGE
C     ACROSS AN AQUITARD PER UNIT TIME, AND ADD THESE CONTRIBUTIONS
C     AND Q INTO THE NODAL FLUX, R, AT SPECIFIED-HEAD NODES ***
      DO 90 I=1,NNDS
      NVL=NC+1
      ND=NVL
      TMPB=A(NVL)*(HR(I)-H(I))
      IF(TMPB.GT.0.) GO TO 20
      VLQO=VLQO+TMPB
      GO TO 25
   20 VLQI=VLQI+TMPB
   25 THKI=THK(I)
      IF(THKI.LT.0.) THKI=0.
      K=IN(I)
      IF(K.GT.0) GO TO 35
      K=-K
      TMPA=A(NC)*DTM*DHB(K)
      SA=SA+TMPA
      R(K)=R(K)-Q(I)+TMPA-TMPB
      DO 30 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 26,80,28
   26 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      M=-M
      TMPA=(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I))
     1+C*(DTK(I)+DTK(L))*(DHB(M)-DHB(K)))*A(ND+J)
      R(K)=R(K)+TMPA
      R(M)=R(M)-TMPA
      GO TO 30
   28 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      R(K)=R(K)+(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I))
     1+C*(DTK(I)+DTK(L))*(DH(M)-DHB(K)))*A(ND+J)
   30 CONTINUE
      GO TO 80
   35 SA=SA+A(NC)*DTM*DH(K)
      DO 70 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 60,80,70
   60 M=-M
      THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      R(M)=R(M)+(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(I)-H(L))
     1+C*(DTK(I)+DTK(L))*(DH(K)-DHB(M)))*A(ND+J)
   70 CONTINUE
   80 NC=NC+IW
   90 NP=NP+MBM1
C *** ADD HEAD-DEPENDENT FLUX INTO TOTAL INFLOW OR OUTFLOW PER UNIT
C     TIME ACROSS CAUCHY-TYPE BOUNDARIES AND INTO R ***
      BQI=0.
```

```
      BQO=0.
      IF(NQBND.LE.0) GO TO 140
      DO 130 I=1,NQBND
      TMPA=0.
      TMPB=0.
      K=KQB(I)
      IF(K.LT.0) GO TO 95
      L=LQB(I)
      M=-IN(K)
      N=-IN(L)
      TMPA=CFDK(I)*(HK(I)-H(K))
      TMPB=CFDL(I)*(HL(I)-H(L))
      IF(M.GT.0) R(M)=R(M)-TMPA
      IF(N.GT.0) R(N)=R(N)-TMPB
   95 TMPA=QBND(I)*CFDK(I)+TMPA
      DTK(I)=TMPA
      IF(TMPA.GT.0.) GO TO 100
      BQO=BQO+TMPA
      GO TO 110
  100 BQI=BQI+TMPA
  110 TMPB=QBND(I)*CFDL(I)+TMPB
      VQL(I)=TMPB
      IF(TMPB.GT.0.) GO TO 120
      BQO=BQO+TMPB
      GO TO 130
  120 BQI=BQI+TMPB
  130 CONTINUE
C *** COMPUTE TOTAL FLOW IMBALANCE PER UNIT TIME ***
  140 ER=SA-WQI-WQO-DQI-DQO-VLQI-VLQO-BQI-BQO
      RETURN
      END
```

```
      SUBROUTINE MBWTCB(H,DH,DHB,HR,HK,HL,QBND,CFDK,CFDL,A,Q,R,VQL,THK
     1,DTK,TOP,ASY,CH,CBQ,GMA,DT,JPT,IN,KQB,LQB)
C        COMPUTES FLOW BALANCE FOR BASIC, WATER-TABLE AQUIFER VERSION OF
C        PROGRAM WITH TRANSIENT LEAKAGE
      DIMENSION H(1),DH(1),DHB(1),HR(1),HK(1),HL(1),QBND(1),CFDK(1)
     1,CFDL(1),A(1),Q(1),R(1),VQL(1),THK(1),DTK(1),TOP(1),ASY(1),CH(1)
     2,CBQ(1),GMA(1)
      DIMENSION JPT(1),IN(1),KQB(1),LQB(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** COMPUTE DTK-VECTOR, AND COMPUTE CAPACITANCE AT STORAGE CONVERSION
C     NODES ***
      WF=2./3.
      NC=1
      IW=MBWC+1
      DO 14 I=1,NNDS
      DTK(I)=0.
      J=IN(I)
      IF(J.GT.0) GO TO 2
      DHC=DHB(-J)
      GO TO 4
    2 DHC=DH(J)
    4 HO=H(I)-DHC
      DHC=1.5*DHC
      HC=HO+DHC
      IF(HO.LT.TOP(I)) GO TO 6
      IF(HC.GE.TOP(I)) GO TO 14
      DTK(I)=HC-TOP(I)
      IF(THK(I)+DTK(I).LT.0.) DTK(I)=-THK(I)
      THET=(TOP(I)-HO)/DHC
      A(NC)=THET*(A(NC)-ASY(I))+ASY(I)
      GO TO 12
    6 IF(HC.LT.TOP(I)) GO TO 8
      DTK(I)=TOP(I)-HO
      THET=DTK(I)/DHC
      A(NC)=THET*(ASY(I)-A(NC))+A(NC)
      GO TO 12
    8 IF(THK(I)+DHC.GT.0.) GO TO 10
      DTK(I)=0.
      IF(THK(I).GT.0.) DTK(I)=-THK(I)
      GO TO 11
   10 DTK(I)=DHC
      IF(THK(I).LT.0.) DTK(I)=DHC+THK(I)
   11 A(NC)=ASY(I)
   12 DTK(I)=WF*DTK(I)
   14 NC=NC+IW
C *** INITIALIZE FLOW-BALANCE COMPONENTS ***
      IF(NHDS.LE.0) GO TO 18
      DO 16 I=1,NHDS
   16 R(I)=0.
   18 SA=0.
      VLQI=0.
      VLQO=0.
      C=1./16.
      MBM1=MBWC-1
      NC=1
```

```
      NP=0
      N=0
      DTM=3./(2.*DT)
C *** COMPUTE TOTAL ACCUMULATION IN STORAGE PER UNIT TIME AND LEAKAGE
C     ACROSS AN AQUITARD PER UNIT TIME, AND ADD THESE CONTRIBUTIONS
C     AND Q INTO THE NODAL FLUX, R, AT SPECIFIED-HEAD NODES ***
      DO 90 I=1,NNDS
      NVL=NC+1
      ND=NVL
      TMPB=A(NVL)*(HR(I)-H(I))
      THKI=THK(I)
      IF(THKI.LT.0.) THKI=0.
      K=IN(I)
      IF(K.GT.0) GO TO 35
      K=-K
      TMPA=A(NC)*DTM*DHB(K)
      SA=SA+TMPA
      IF(GMA(I).LT.1.E-30) GO TO 20
      N=N+1
      TMPB=TMPB+CBQ(N)-CH(N)*DHB(K)
   20 R(K)=R(K)-Q(I)+TMPA-TMPB
      DO 30 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 26,75,28
   26 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      M=-M
      TMPA=(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I))
     1+C*(DTK(I)+DTK(L))*(DHB(M)-DHB(K)))*A(ND+J)
      R(K)=R(K)+TMPA
      R(M)=R(M)-TMPA
      GO TO 30
   28 THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      R(K)=R(K)+(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(L)-H(I))
     1+C*(DTK(I)+DTK(L))*(DH(M)-DHB(K)))*A(ND+J)
   30 CONTINUE
      GO TO 75
   35 SA=SA+A(NC)*DTM*DH(K)
      IF(GMA(I).LT.1.E-30) GO TO 40
      N=N+1
      TMPB=TMPB+CBQ(N)-CH(N)*DH(K)
   40 DO 70 J=1,MBM1
      L=JPT(NP+J)
      M=IN(L)
      IF(M) 60,75,70
   60 M=-M
      THKL=THK(L)
      IF(THKL.LT.0.) THKL=0.
      R(M)=R(M)+(.5*(DTK(I)+DTK(L)+THKI+THKL)*(H(I)-H(L))
     1+C*(DTK(I)+DTK(L))*(DH(K)-DHB(M)))*A(ND+J)
   70 CONTINUE
   75 IF(TMPB.GT.0.) GO TO 80
```

```
      VLQO=VLQO+TMPB
      GO TO 85
   80 VLQI=VLQI+TMPB
   85 NC=NC+IW
   90 NP=NP+MBM1
C *** ADD HEAD-DEPENDENT FLUX INTO TOTAL INFLOW OR OUTFLOW PER UNIT
C     TIME ACROSS CAUCHY-TYPE BOUNDARIES AND INTO R ***
      BQI=0.
      BQO=0.
      IF(NQBND.LE.0) GO TO 140
      DO 130 I=1,NQBND
      TMPA=0.
      TMPB=0.
      K=KQB(I)
      IF(K.LT.0) GO TO 95
      L=LQB(I)
      M=-IN(K)
      N=-IN(L)
      TMPA=CFDK(I)*(HK(I)-H(K))
      TMPB=CFDL(I)*(HL(I)-H(L))
      IF(M.GT.0) R(M)=R(M)-TMPA
      IF(N.GT.0) R(N)=R(N)-TMPB
   95 TMPA=QBND(I)*CFDK(I)+TMPA
      DTK(I)=TMPA
      IF(TMPA.GT.0.) GO TO 100
      BQO=BQO+TMPA
      GO TO 110
  100 BQI=BQI+TMPA
  110 TMPB=QBND(I)*CFDL(I)+TMPB
      VQL(I)=TMPB
      IF(TMPB.GT.0.) GO TO 120
      BQO=BQO+TMPB
      GO TO 130
  120 BQI=BQI+TMPB
  130 CONTINUE
C *** COMPUTE TOTAL FLOW IMBALANCE PER UNIT TIME ***
  140 ER=SA-WQI-WQO-DQI-DQO-VLQI-VLQO-BQI-BQO
      RETURN
      END
```

```
      SUBROUTINE MICCG(A,AF,X,P,R,B,TOL,JPT,IN)
C        SOLVES MATRIX EQUATION FOR HEAD CHANGE VECTOR, B, BY USING
C        THE MODIFIED, INCOMPLETE CHOLESKY, CONJUGATE GRADIENT METHOD
      DIMENSION A(1),AF(1),X(1),P(1),R(1),B(1)
      DIMENSION JPT(1),IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (23H0SOLUTION CONVERGED IN ,I4,11H ITERATIONS)
    4 FORMAT (32H0SOLUTION FAILED TO CONVERGE IN ,I4,11H ITERATIONS
     1,3X,7HRMAX = ,G11.5)
    6 FORMAT (1H0,14X,42HCURRENT VALUES OF CHANGE IN HYDRAULIC HEAD
     1/1H ,3(6H  NODE,8X,4HHEAD,7X))
      N=NNDS-NHDS
      MBM1=MBWC-1
C *** COMPUTE AUGMENTATION FACTOR, IF NEEDED, FOR DIAGONAL ELEMENTS OF
C     AF MATRIX, WHICH IS THE MATRIX TO BE FACTORED ***
      DEL=.9993333
   10 DEL=1.5*DEL-.499
C *** DEFINE AF MATRIX AS THE A MATRIX WITH THE DIAGONALS AUGMENTED
C     IF NECESSARY ***
      ID=1
      DO 20 I=1,N
      IF(A(ID).GT.1.E-20) GO TO 12
      A(ID)=1.E30
      B(I)=0.
   12 AF(ID)=DEL*A(ID)
      DO 15 J=1,MBM1
   15 AF(ID+J)=A(ID+J)
   20 ID=ID+MBWC
      NEM1=NEQ-1
C *** BEGIN FACTORIZATION OF AF INTO SPARSE PRODUCT OF UPPER TRIANGULAR,
C     DIAGONAL, AND LOWER TRIANGULAR MATRICES, ALL KEPT IN AF ***
      NP=0
      ID=1
      DO 90 I=1,NEM1
      IF(IN(I).LT.0) GO TO 90
C *** DEFINE PIVOT ***
      IF(AF(ID).LE.0.) GO TO 10
      PIV=1./AF(ID)
      DO 70 J=1,MBM1
      KJ=JPT(NP+J)
      K=IN(KJ)
      IF(K) 70,80,25
   25 KD=MBWC*(K-1)+1
C *** DEFINE ROW MULTIPLIER ***
      C=AF(ID+J)*PIV
C *** REDUCE DIAGONAL OF AF ***
      AF(KD)=AF(KD)-C*AF(ID+J)
      LBG=J+1
      IF(LBG.GT.MBM1) GO TO 60
      KJ=MBM1*(KJ-1)
      MBG=1
      DO 50 L=LBG,MBM1
      KL=JPT(NP+L)
      IF(IN(KL)) 50,60,30
```

```
   30 TMPA=C*AF(ID+L)
C *** SEARCH TO DETERMINE IF NODE KL LIES IN THE ROW BEING REDUCED ***
      DO 40 M=MBG,MBM1
      MS=M
      IF(JPT(KJ+M).EQ.KL) GO TO 45
   40 CONTINUE
C *** REDUCE APPROPRIATE DIAGONAL ELEMENTS BY SUBTRACTING FILL-IN IF
C     KL IS NOT IN THE ROW BEING REDUCED ***
      AF(KD)=AF(KD)-TMPA
      LD=MBWC*(IN(KL)-1)+1
      AF(LD)=AF(LD)-TMPA
      GO TO 50
C *** REDUCE OFF-DIAGONAL IF KL IS IN THE ROW BEING REDUCED ***
   45 AF(KD+MS)=AF(KD+MS)-TMPA
      MBG=MS
   50 CONTINUE
   60 AF(ID+J)=C
   70 CONTINUE
   80 ID=ID+MBWC
   90 NP=NP+MBM1
      IF(AF(ID).LE.0.) GO TO 10
C *** DEFINE INITIAL RESIDUAL VECTOR, R, AS THE RIGHT-HAND-SIDE VECTOR,
C     B, OF THE ORIGINAL MATRIX EQUATION, THEN ZERO OUT INITIAL HEAD
C     CHANGE VECTOR, B ***
      DO 95 I=1,N
      R(I)=B(I)
   95 B(I)=0.
C *** BEGIN CONJUGATE GRADIENT ITERATION LOOP ***
      DO 270 KNT=1,NIT
      IT=KNT
C *** READ RESIDUAL VECTOR, R, INTO X ***
      DO 100 I=1,N
  100 X(I)=R(I)
C *** REDUCE RESIDUAL VECTOR, X ***
      ID=1
      NP=0
      DO 130 I=1,NEM1
      M=IN(I)
      IF(M.LT.0) GO TO 130
      DO 110 J=1,MBM1
      K=JPT(NP+J)
      K=IN(K)
      IF(K) 110,120,105
  105 X(K)=X(K)-AF(ID+J)*X(M)
  110 CONTINUE
  120 X(M)=X(M)/AF(ID)
      ID=ID+MBWC
  130 NP=NP+MBM1
C *** BACK-SOLVE FOR FIRST-ORDER DISPLACEMENT VECTOR, X ***
      X(N)=X(N)/AF(ID)
      K=NEQ
      DO 150 I=1,NEM1
      NP=NP-MBM1
      K=K-1
      M=IN(K)
```

```
      IF(M.LT.0) GO TO 150
      ID=ID-MBWC
      DO 140 J=1,MBM1
      L=JPT(NP+J)
      L=IN(L)
      IF(L) 140,150,135
  135 X(M)=X(M)-AF(ID+J)*X(L)
  140 CONTINUE
  150 CONTINUE
C *** FORM INNER PRODUCT OF RESIDUAL AND X VECTORS ***
      RX=0.
      DO 160 I=1,N
  160 RX=RX+R(I)*X(I)
      IF(IT.EQ.1) GO TO 180
C *** COMPUTE ITERATION PARAMETER, BETA ***
      S=RX/RXO
C *** COMPUTE SECOND-ORDER DISPLACEMENT VECTOR, P ***
      DO 170 I=1,N
  170 P(I)=X(I)+S*P(I)
      GO TO 200
  180 DO 190 I=1,N
  190 P(I)=X(I)
  200 DO 210 I=1,N
  210 X(I)=0.
C *** COMPUTE PRODUCT OF MATRIX A AND VECTOR P, AND KEEP PRODUCT
C     IN X ***
      ID=1
      NP=0
      DO 240 I=1,NEQ
      M=IN(I)
      IF(M.LT.0) GO TO 240
      X(M)=X(M)+A(ID)*P(M)
      DO 220 J=1,MBM1
      K=JPT(NP+J)
      K=IN(K)
      IF(K) 220,230,215
  215 X(M)=X(M)+A(ID+J)*P(K)
      X(K)=X(K)+A(ID+J)*P(M)
  220 CONTINUE
  230 ID=ID+MBWC
  240 NP=NP+MBM1
C *** FORM INNER PRODUCT OF P AND X VECTORS ***
      PX=0.
      DO 250 I=1,N
  250 PX=PX+P(I)*X(I)
C *** COMPUTE ITERATION PARAMETER, ALPHA ***
      W=1.
      IF(PX.GT.0.) W=RX/PX
C *** COMPUTE NEW HEAD CHANGE VECTOR, B, AND NEW RESIDUAL VECTOR, R,
C     THEN TEST FOR CONVERGENCE ***
      PMAX=0.
      RMAX=0.
      ID=1
      DO 260 I=1,N
      TMPA=W*P(I)
      B(I)=B(I)+TMPA
```

```
      R(I)=R(I)-W*X(I)
      TMPA=ABS(TMPA)
      IF(TMPA.GT.PMAX) PMAX=TMPA
      TMPA=ABS(R(I)/A(ID))
      IF(TMPA.GT.RMAX) RMAX=TMPA
  260 ID=ID+MBWC
      IF(PMAX.LE.TOL.AND.RMAX.LE.TOL) GO TO 280
      RXO=RX
  270 CONTINUE
C *** WRITE MAXIMUM SCALED RESIDUAL AND CURRENT HEAD CHANGE VECTOR IF
C     SOLUTION DID NOT CONVERGE ***
      WRITE(IOUT,4) NIT,RMAX
      WRITE(IOUT,6)
      CALL PRTOA(B,N)
      STOP
C *** WRITE NUMBER OF ITERATIONS IF SOLUTION CONVERGED ***
  280 WRITE(IOUT,2) IT
      RETURN
      END
```

```
      SUBROUTINE NXTPD(DELT,JPER)
C        READS TIME-STEP DATA FOR NEXT PUMPING PERIOD
      DIMENSION DELT(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/CHG/NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT(16I5)
    4 FORMAT(8F10.0)
    6 FORMAT(1H0,19X,14HSTRESS PERIOD ,I3,19H:   TIME STEP SIZES
     1/1H ,3X,3(3HNO.,6X,7HDELTA T,9X))
C *** READ NUMBER OF TIME STEPS AND TIME-STEP NUMBERS WHERE VALUES OF
C     TIME-VARIANT QUANTITIES ARE CHANGED ***
      READ(IIN,2) NTMP,NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
C *** READ AND WRITE TIME-STEP SIZES IF DIFFERENT FROM LAST PERIOD ***
      IF(NTMP.LE.0) GO TO 10
      NSTEPS=NTMP
      READ(IIN,4) (DELT(I),I=1,NSTEPS)
   10 WRITE(IOUT,6) JPER
      CALL PRTOA(DELT,NSTEPS)
      RETURN
      END
```

```
      SUBROUTINE PRTCBV(VALA,VALB,DT,NVLA,NVLB,INZ,INS,NBZ)
C        PRINTS OUT NODAL VOLUMETRIC FLOW RATES FROM (NONLINEAR) CAUCHY-
C        TYPE BOUNDARIES AND SUMS NODAL RATES BY ZONE.  FLOW RATES FOR
C        NODES K AND L ON BOUNDARIES ARE STORED IN TWO VECTORS, WHICH
C        ARE WRITTEN OUT FOR EACH BOUNDARY SIDE.
      DIMENSION VALA(1),VALB(1)
      DIMENSION NVLA(1),NVLB(1),INZ(1),INS(1)
      COMMON/ITP/IIN,IOUT,ITA,ITB
   20 FORMAT(1H0,5HZONE ,I5/
     $56H VOLUMETRIC RECHARGE RATE............................ = ,G11.5/
     $56H VOLUMETRIC DISCHARGE RATE........................... = ,G11.5/
     $56H TOTAL RECHARGE VOLUME............................... = ,G11.5/
     $56H TOTAL DISCHARGE VOLUME.............................. = ,G11.5/
     $56H NET VOLUMETRIC FLOW RATE, POSITIVE FOR RECHARGE..... = ,G11.5/
     $56H NET VOLUME, POSITIVE FOR ACCUMULATION............... = ,G11.5)
   22 FORMAT(/1H0,4X
     1,48HVOLUMETRIC FLOW RATES BY BOUNDARY SIDE FOR ZONE ,I5/
     2/1H ,4X,8HBOUNDARY,4X,4HNODE,4X,4HNODE,5X,21HVOLUMETRIC FLOW RATES
     3/1H ,6X,4HSIDE,8X,1HK,7X,1HL,6X,6HNODE K,9X,6HNODE L)
   24 FORMAT(' ',4X,I5,6X,I5,3X,I5,2(4X,G11.5))
      J=0
      JB=1
      DO 150 NZ=1,NBZ
      KZ=INZ(NZ)
      NOS=INS(KZ)
      SMQI=0.
      SMQO=0.
      SMVI=0.
      SMVO=0.
      QNET=0.
      VNET=0.
      DO 130 I=1,NOS
      J=J+1
      VQK=VALA(J)
      VQL=VALB(J)
      IF(VQK.GT.0.) GO TO 100
      SMQO=SMQO+VQK
      GO TO 110
  100 SMQI=SMQI+VQK
  110 IF(VQL.GT.0.) GO TO 120
      SMQO=SMQO+VQL
      GO TO 130
  120 SMQI=SMQI+VQL
  130 CONTINUE
      SMVI=SMQI*DT
      SMVO=SMQO*DT
      QNET=SMQI+SMQO
      VNET=SMVI+SMVO
      WRITE(IOUT,20) KZ,SMQI,SMQO,SMVI,SMVO,QNET,VNET
      WRITE(IOUT,22) KZ
      L=JB
      DO 145 I=1,NOS
      NVA=NVLA(L)
      IF(NVA.LT.0) NVA=-NVA
      WRITE(IOUT,24) L,NVA,NVLB(L),VALA(L),VALB(L)
  145 L=L+1
```

```
      JB=L
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRTOA(VAL,NO)
C        WRITES VECTOR IN THREE COLUMNS
      DIMENSION VAL(1)
      COMMON/ITP/IIN,IOUT,ITA,ITB
      NR=NO/3
      IF(3*NR.NE.NO) NR=NR+1
      DO 10 K=1,NR
      WRITE(IOUT,20) (L,VAL(L),L=K,NO,NR)
   10 CONTINUE
      RETURN
   20 FORMAT (1H ,3(I5,6X,G11.5,3X))
      END


      SUBROUTINE PRTOB(VALA,VALB,NO)
C        WRITES TWO VECTORS IN TWO DOUBLE COLUMNS
      DIMENSION VALA(1),VALB(1)
      COMMON/ITP/IIN,IOUT,ITA,ITB
      NR=NO/2
      IF(2*NR.NE.NO) NR=NR+1
      DO 10 K=1,NR
      WRITE(IOUT,20) (L,VALA(L),VALB(L),L=K,NO,NR)
   10 CONTINUE
      RETURN
   20 FORMAT (1H ,2(I5,5X,G11.5,4X,G11.5,2X))
      END
```

```
      SUBROUTINE RDTP(A)
C        READS COEFFICIENT ARRAY FROM ITA
      DIMENSION A(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
      IE=(MBWC+1)*NNDS
      REWIND ITA
      READ(ITA) (A(I),I=1,IE)
      RETURN
      END
```

```
      SUBROUTINE SETB(A,IN,ND,IBND)
C        FINDS MATRIX BAND WIDTH FOR BAND MATRIX SOLVER AND WRITES
C        ARRAYS INTO FILES
      DIMENSION A(1)
      DIMENSION IN(1),ND(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (1H0,27HMATRIX BAND WIDTH (IBND) = ,I3)
      M=NNDS-NHDS
      NDC=0
      IBND=0
C *** FIND AND WRITE MAXIMUM MATRIX BAND WIDTH ***
      DO 30 I=1,NELS
      NE=2
      IF(ND(NDC+4).LE.0) NE=1
      JB=1
      JE=3
      DO 25 IE=1,NE
      MINND=M
      MAXND=0
      DO 20 J=JB,JE
      K=ND(NDC+1)
      IF(J.NE.5) K=ND(NDC+J)
      L=IN(K)
      IF(L.LT.0) GO TO 20
      IF(L.GT.MAXND) MAXND=L
      IF(L.LT.MINND) MINND=L
   20 CONTINUE
      NTMP=MAXND-MINND
      IF(NTMP.GT.IBND) IBND=NTMP
      JB=3
   25 JE=5
   30 NDC=NDC+4
      IBND=IBND+1
      WRITE(IOUT,2) IBND
C *** REWIND FILES ITA AND ITB ***
      REWIND ITA
      REWIND ITB
C *** WRITE COEFFICIENT ARRAY TO ITA ***
      IE=(MBWC+1)*NNDS
      WRITE(ITA) (A(I),I=1,IE)
C *** WRITE ELEMENT AREAS AND NODE POINTS TO ITB ***
      IB=IE+1
      IE=IB+2*NELS
      IB1=IE+1
      IE1=IB1+4*NELS
      WRITE(ITB) (A(I),I=IB,IE),(A(I),I=IB1,IE1)
      RETURN
      END
```

```
      SUBROUTINE SETCG(A)
C        WRITES ARRAYS INTO FILES
      DIMENSION A(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** REWIND FILES ITA AND ITB ***
      REWIND ITA
      REWIND ITB
C *** WRITE COEFFICIENT ARRAY TO ITA ***
      IE=(MBWC+1)*NNDS
      WRITE(ITA) (A(I),I=1,IE)
C *** WRITE ELEMENT AREAS AND NODE POINTS TO ITB ***
      IB=IE+1
      IE=IB+2*NELS
      IB1=IE+1
      IE1=IB1+4*NELS
      WRITE(ITB) (A(I),I=IB,IE),(A(I),I=IB1,IE1)
      RETURN
      END
```

```
      SUBROUTINE SWBDMP(B,DSMX,DSP,DSPO,DSPA,RP,ITER)
C        COMPUTES DAMPING PARAMETER AND DAMPS B-VECTOR
      DIMENSION B(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      N=NNDS-NHDS
      DSPA=0.
      DSPO=DSP
      DO 10 I=1,N
C *** COMPUTE MAXIMUM ABSOLUTE VALUE OF B-VECTOR ***
      TMPA=ABS(B(I))
      IF(TMPA.LE.DSPA) GO TO 10
      DSPA=TMPA
      DSP=B(I)
   10 CONTINUE
C *** COMPUTE DAMPING PARAMETER, RP ***
      IF(ITER.LE.1) GO TO 30
      SPR=DSP/(RP*DSPO)
      IF(SPR.LT.-1.) GO TO 20
      RP=(3.+SPR)/(3.+ABS(SPR))
      GO TO 30
   20 RP=.5/ABS(SPR)
   30 IF(RP*DSPA.GT.DSMX) RP=DSMX/DSPA
C *** DAMP B-VECTOR ***
      DO 40 I=1,N
   40 B(I)=RP*B(I)
      RETURN
      END
```

```
      SUBROUTINE SWFMCO(A,THK,TOP,JPT,IN)
C        READS AND WRITES INITIAL AQUIFER THICKNESS AND ELEVATION OF
C        AQUIFER TOP THEN FORMS INITIAL FLOW COEFFICIENTS
      DIMENSION A(1),THK(1),TOP(1)
      DIMENSION JPT(1),IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    4 FORMAT (8F10.0)
    5 FORMAT (16I5)
    6 FORMAT (1H0,24X,25HINITIAL AQUIFER THICKNESS/1H ,3(6H  NODE,5X
     1,9HTHICKNESS,5X))
    7 FORMAT (1H0,8X,49HINITIAL AQUIFER THICKNESS READ, OUTPUT SUPPRESSE
     1D)
    8 FORMAT (1H0,25X,24HELEVATION OF AQUIFER TOP/1H ,3(6H  NODE,5X
     1,9HELEVATION,5X))
    9 FORMAT (1H0,8X,45HAQUIFER TOP ELEVATION READ, OUTPUT SUPPRESSED)
C *** READ AND WRITE INITIAL AQUIFER THICKNESS, THK, AND ELEVATION OF
C     AQUIFER TOP, TOP ***
      READ(IIN,5) IPTK,IPTP
      READ(IIN,4) (THK(I),I=1,NNDS)
      IF(IPTK.GT.0) GO TO 16
      WRITE(IOUT,6)
      CALL PRTOA(THK,NNDS)
      GO TO 17
   16 WRITE(IOUT,7)
   17 READ(IIN,4) (TOP(I),I=1,NNDS)
      IF(IPTP.GT.0) GO TO 18
      WRITE(IOUT,8)
      CALL PRTOA(TOP,NNDS)
      GO TO 19
   18 WRITE(IOUT,9)
C *** COMPUTE INITIAL FLOW COEFFICIENTS, A ***
   19 IW=MBWC+1
      MBM1=MBWC-1
      ND=2
      NP=0
      DO 40 I=1,NNDS
      DO 20 J=1,MBM1
      K=JPT(NP+J)
      IF(IN(K).EQ.0) GO TO 30
   20 A(ND+J)=.5*(THK(K)+THK(I))*A(ND+J)
   30 ND=ND+IW
   40 NP=NP+MBM1
      RETURN
      END
```

```
      SUBROUTINE SWINIT(G,TOLSW,DSMX,DSP,RP,NITSW,ITKA,ITPA)
C        READS PROBLEM SPECIFICATIONS, DEFINES AND INITIALIZES VARIABLES
C        FOR STEADY-STATE WATER-TABLE AQUIFER VERSION OF PROGRAM
      DIMENSION G(1)
      COMMON/GDIM/ISUM
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (I5,2F10.0)
    4 FORMAT (1H0,54HMAX. NO. OF WATER-TABLE ITERATIONS (NITSW) ........
     $ = ,I5/
     $1H ,54HCLOSURE TOL. FOR WATER-TABLE ITERATIONS (TOLSW) ... =
     $,G11.5/
     $1H ,54HMAXIMUM ALLOWABLE DISPLACEMENT (DSMX) ............ =
     $,G11.5)
    6 FORMAT (1H0,43HSTEADY-STATE FLOW IN A WATER-TABLE AQUIFER:/1H
     1,38HNOW G MUST BE DIMENSIONED TO AT LEAST ,I6)
C *** READ AND WRITE PROBLEM SPECIFICATIONS ***
      READ(IIN,2) NITSW,TOLSW,DSMX
      WRITE(IOUT,4) NITSW,TOLSW,DSMX
C *** DEFINE INITIAL MAXIMUM HEAD DISPLACEMENT AND DAMPING PARAMETER ***
      DSP=DSMX
      RP=1.
C *** DEFINE ADDRESSES OF ARRAYS WITHIN G ***
      ITKA=ISUM
      ISUM=ISUM+NNDS
      ITPA=ISUM
      ISUM=ISUM+NNDS
C *** WRITE NEW SIZE OF G ***
      WRITE(IOUT,6) ISUM
C *** INITIALIZE NEW ELEMENTS OF G ***
      DO 10 I=ITKA,ISUM
   10 G(I)=0.
      RETURN
      END
```

```
      SUBROUTINE SWTHK(H,A,B,Q,THK,TOP,DSPA,TOLSW,JPT,IN,ITER)
C        COMPUTES CURRENT AQUIFER THICKNESS AND FLOW COEFFICIENT ARRAY
      DIMENSION H(1),A(1),B(1),Q(1),THK(1),TOP(1)
      DIMENSION JPT(1),IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,NIT
      COMMON/ITP/IIN,IOUT,ITA,ITB
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
C *** FORMAT LIST ***
    1 FORMAT (1H0,5HNODE ,I5,23H WENT DRY ON ITERATION ,I3/1H ,3X
     1,38HPREDICTED AQUIFER THICKNESS AT NODE = ,G11.5/1H ,3X
     2,19HNET FLOW AT NODE = ,G11.5)
    2 FORMAT (1H ,3X,30HNET FLUX FROM NODE REDUCED TO ,G11.5)
C *** COMPUTE AQUIFER THICKNESS, THK ***
      DO 20 I=1,NNDS
      J=IN(I)
      IF(J.LT.0) GO TO 20
      HO=H(I)-B(J)
      IF(HO.LT.TOP(I)) GO TO 5
      IF(H(I).GT.TOP(I)) GO TO 20
      THK(I)=THK(I)+H(I)-TOP(I)
      GO TO 20
    5 IF(H(I).LT.TOP(I)) GO TO 10
      THK(I)=THK(I)-HO+TOP(I)
      GO TO 20
   10 THK(I)=THK(I)+B(J)
C *** REDUCE DISCHARGE AT NODES THAT ARE DRY ***
      IF(THK(I).GT.0.) GO TO 20
      WRITE(IOUT,1) I,ITER,THK(I),Q(I)
      IF(Q(I).GE.0..OR.B(J).GE.0.) GO TO 20
      Q(I)=.5*Q(I)
      WQO=WQO-Q(I)
      WRITE(IOUT,2) Q(I)
C *** INCREASE MAXIMUM ABSOLUTE DISPLACEMENT TO PERMIT
C     ANOTHER ITERATION USING REDUCED Q VECTOR ***
      DSPA=TOLSW
   20 CONTINUE
C *** COMPUTE FLOW COEFFICIENT ARRAY, A ***
      IW=MBWC+1
      MBM1=MBWC-1
      ND=2
      NP=0
      DO 50 I=1,NNDS
      THKI=THK(I)
      IF(THKI.LT.0.) THKI=0.
      DO 30 J=1,MBM1
      K=JPT(NP+J)
      IF(IN(K).EQ.0) GO TO 40
      THKK=THK(K)
      IF(THKK.LT.0.) THKK=0.
   30 A(ND+J)=.5*(THKK+THKI)*A(ND+J)
   40 ND=ND+IW
   50 NP=NP+MBM1
      RETURN
      END
```

```
      SUBROUTINE TKOUT(THK)
C        WRITES SATURATED THICKNESS VECTOR, THK
      DIMENSION THK(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
    2 FORMAT (1H0,27X,19HSATURATED THICKNESS/1H ,3(6H  NODE,5X
     1,9HTHICKNESS,5X))
      WRITE(IOUT,2)
      CALL PRTOA(THK,NNDS)
      RETURN
      END
```

```
      SUBROUTINE VNBAL(H,DH,DHB,R,TOP,E,HS,DT,IN)
C        COMPUTES FLOW-BALANCE COMPONENTS FOR NONLINEAR STEADY-
C        LEAKAGE FUNCTIONS
      DIMENSION H(1),DH(1),DHB(1),R(1),TOP(1),E(1),HS(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/VNLBL/VNLQI,VNLQO,TNLQI,TNLQO
C *** INITIALIZE VOLUMETRIC-FLOW RATES TO ZERO ***
      VNLQI=0.
      VNLQO=0.
C *** COMPUTE WEIGHTING FACTORS FOR GALERKIN-IN-TIME FORMULATION ***
      CA=1./3.
      CB=2./3.
C *** FOR EACH NODE I ON THE LEAKAGE BOUNDARY: ***
      DO 170 I=1,NNDS
C *** (1) COMPUTE HEAD AT N LEVEL, HO, AND HEAD AT N+1 LEVEL, HC ***
      K=IN(I)
      IF(K.LT.0) GO TO 5
      DHC=DH(K)
      GO TO 10
    5 DHC=DHB(-K)
   10 HO=H(I)-DHC
      DHC=1.5*DHC
      HC=HO+DHC
C *** (2) DETERMINE FUNCTION TYPE BY SIGN OF E(I) ***
      IF(E(I)) 15,170,90
C ***
C *** DISCHARGE-ONLY TYPE ***
   15 EI=-E(I)
C *** (3) REPRESENT ELEVATIONS, AND HEAD AND ELEVATION DIFFERENCES, BY
C         TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      ZE=HS(I)
      ZT=TOP(I)
      DEH=ZE-HO
      DHT=HO-ZT
      DET=ZE-ZT
C *** (4) COMPARE HEADS WITH CONTROLLING ELEVATIONS TO IDENTIFY LEAKAGE
C         CASE. COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC FLOW RATES
C         FOR: ***
      IF(HO.GT.ZE) GO TO 30
      IF(HC.LT.ZE) GO TO 170
      IF(HC.LT.ZT) GO TO 20
C *** CASE (9) ***
      DMHT=DEH-DHT
      PHCF=1.-CA*(DMHT*DMHT+DHT*DEH)/(DHC*DHC)
      TMPA=PHCF*DET
      GO TO 150
C *** CASE (6) ***
   20 PHC=1.-DEH/DHC
      TMPA=CB*PHC*(ZE-HC)
      GO TO 150
   30 IF(HO.LT.ZT) GO TO 40
      IF(HC.LT.ZT) GO TO 70
C *** CASE (1) ***
      TMPA=DET
```

```
      GO TO 150
   40 IF(HC.GT.ZT) GO TO 60
      IF(HC.GT.ZE) GO TO 50
C *** CASE (5) ***
      PHE=DEH/DHC
      TMPA=CA*PHE*PHE*DEH
      GO TO 150
C *** CASE (4) ***
   50 TMPA=ZE-H(I)
      GO TO 150
C *** CASE (3) ***
   60 PHT=DHT/DHC
      TMPA=DET-CA*PHT*PHT*DHT
      GO TO 150
   70 IF(HC.GT.ZE) GO TO 80
C *** CASE (8) ***
      DMHT=DEH-DHT
      PHCF=(DMHT*DMHT+DHT*DEH)/(DHC*DHC)
      TMPA=CA*PHCF*DET
      GO TO 150
C *** CASE (2) ***
   80 PHC=1.+DHT/DHC
      TMPA=DET-CB*PHC*(HC-ZT)
      GO TO 150
C ***
C *** STEADY VERTICAL-LEAKAGE TYPE ***
   90 EI=E(I)
C *** (3) REPRESENT CONTROLLING HEAD, AQUIFER TOP ELEVATION, AND
C         DIFFERENCES BY TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      HA=HS(I)
      ZT=TOP(I)
      DHT=HO-ZT
      DAT=HA-ZT
C *** (4) COMPARE HEADS WITH AQUIFER TOP TO IDENTIFY LEAKAGE CASE.
C         COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-FLOW RATES
C         FOR: ***
      IF(HO.LT.ZT) GO TO 130
      IF(HC.LT.ZT) GO TO 120
C *** CASE (1) ***
      TMPA=HA-H(I)
      GO TO 150
C *** CASE (2) ***
  120 PHI=DHT/DHC
      TMPA=DAT-CA*PHI*PHI*DHT
      GO TO 150
  130 IF(HC.GT.ZT) GO TO 140
C *** CASE (4) ***
      TMPA=DAT
      GO TO 150
C *** CASE (3) ***
  140 PHC=1.+DHT/DHC
      TMPA=DAT-CB*PHC*(HC-ZT)
C ***
C *** (5) COMPARE VOLUMETRIC-FLOW RATES FOR BOTH TYPES OF NONLINEAR
C         STEADY-LEAKAGE FUNCTIONS ***
```

```
  150 TMPB=EI*TMPA
C *** INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C     SPECIFIED-HEAD NODE ***
      IF(K.LT.0) R(-K)=R(-K)-TMPB
C *** SUM VOLUMETRIC-FLOW RATES ACCORDING TO SIGN, POSITIVE FOR
C     INFLOW ***
      IF(TMPB.LT.0.) GO TO 160
      VNLQI=VNLQI+TMPB
      GO TO 170
  160 VNLQO=VNLQO+TMPB
  170 CONTINUE
C *** COMPUTE TOTAL VOLUMES RECHARGED (+) AND DISCHARGED (-) ACROSS
C     BOUNDARY FOR ENTIRE SIMULATION ***
      TNLQI=TNLQI+VNLQI*DT
      TNLQO=TNLQO+VNLQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES IN THE FLOW IMBALANCE ***
      ER=ER-VNLQI-VNLQO
      RETURN
      END
```

```
      SUBROUTINE VNBLSS(H,R,TOP,E,HS,DT,IN)
C        COMPUTES FLOW-BALANCE COMPONENTS FOR NONLINEAR STEADY-LEAKAGE
C        FUNCTIONS FOR STEADY-STATE SIMULATIONS
      DIMENSION H(1),R(1),TOP(1),E(1),HS(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/BAL/SA,WQI,WQO,DQI,DQO,VLQI,VLQO,BQI,BQO,ER
      COMMON/VNLBL/VNLQI,VNLQO,TNLQI,TNLQO
C *** INITIALIZE VOLUMETRIC FLOW RATES TO ZERO ***
      VNLQI=0.
      VNLQO=0.
C *** FOR EACH NODE I ON THE LEAKAGE BOUNDARY: ***
      DO 90 I=1,NNDS
C *** (1) DETERMINE FUNCTION TYPE BY SIGN OF E(I) ***
      IF(E(I)) 10,90,40
C ***
C *** DISCHARGE-ONLY TYPE ***
   10 EI=-E(I)
C *** (2) REPRESENT HEADS AND ELEVATIONS BY TEMPORARY (UNSUBSCRIPTED)
C         VARIABLES ***
      ZE=HS(I)
      ZT=TOP(I)
      HI=H(I)
C *** (3) COMPARE HEADS WITH CONTROLLING ELEVATIONS TO IDENTIFY
C         LEAKAGE CASE. COMPUTE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-
C         FLOW RATES FOR: ***
      IF(HI.GT.ZE) GO TO 20
C *** CASE (7) (NO FORMULATION) ***
      GO TO 90
   20 IF(HI.LT.ZT) GO TO 30
C *** CASE (1) ***
      TMPA=ZE-ZT
      GO TO 60
C *** CASE (4) ***
   30 TMPA=ZE-HI
      GO TO 60
C ***
C *** STEADY VERTICAL-LEAKAGE TYPE ***
   40 EI=E(I)
C *** (2) REPRESENT HEADS AND ELEVATIONS BY TEMPORARY (UNSUBSCRIPTED)
C         VARIABLES ***
      HA=HS(I)
      ZT=TOP(I)
      HI=H(I)
C *** (3) COMPARE HEAD WITH AQUIFER TOP TO IDENTIFY LEAKAGE CASE.
C         COMPARE HEAD DIFFERENCE, TMPA, FOR VOLUMETRIC-FLOW RATES
C         FOR: ***
      IF(HI.GT.ZT) GO TO 50
C *** CASE (4) ***
      TMPA=HA-ZT
      GO TO 60
C *** CASE (1) ***
   50 TMPA=HA-HI
C ***
C *** (4) COMPUTE VOLUMETRIC-FLOW RATE ***
   60 TMPB=EI*TMPA
```

```
C *** (5) SUM VOLUMETRIC-FLOW RATES ACCORDING TO SIGN, POSITIVE
C         FOR INFLOW ***
      IF(TMPB.LT.0.) GO TO 70
      VNLQI=VNLQI+TMPB
      GO TO 80
   70 VNLQO=VNLQO+TMPB
C *** (6) INCLUDE VOLUMETRIC-FLOW RATE INTO FLOW-BALANCE EQUATION AT
C         SPECIFIED-HEAD NODE ***
   80 K=-IN(I)
      IF(K.GT.0) R(K)=R(K)-TMPB
   90 CONTINUE
C *** COMPUTE TOTAL VOLUMES RECHARGED (+) AND DISCHARGED (-) ACROSS
C     BOUNDARY FOR SIMULATION ***
      TNLQI=TNLQI+VNLQI*DT
      TNLQO=TNLQO+VNLQO*DT
C *** INCLUDE VOLUMETRIC-FLOW RATES IN THE FLOW IMBALANCE ***
      ER=ER-VNLQI-VNLQO
      RETURN
      END
```

```
      SUBROUTINE VNCHG(HS,TIME,ISTP)
C        CHANGES VALUES OF CONTROLLING HEAD FOR NONLINEAR STEADY
C        VERTICAL-LEAKAGE FUNCTIONS FOR TIME-VARIANT BOUNDARY CONDITIONS
      DIMENSION HS(1)
      COMMON/CHG/NWCH,NQCH,NHRCH,NBQCH,NHCH,NCBCH,NVNCH,NGNCH
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (16I5)
    4 FORMAT (1H1,11X,37HCHANGES IN VALUES OF CONTROLLING HEAD/1H
     1,10X,38HFOR NONLINEAR STEADY-LEAKAGE FUNCTIONS
     2/1H ,29H  BEGINNING ON TIME STEP NO. ,I4,4H AT ,G11.5
     3,11H TIME UNITS/1H ,34X,11HCONTROLLING/1H ,19X,4HNODE,15X,4HHEAD)
    6 FORMAT (I5,F10.0)
    8 FORMAT (1H ,17X,I5,13X,G11.5)
C *** READ NUMBER OF CONTROLLING HEADS TO BE CHANGED AND TIME-STEP
C     NUMBER FOR NEXT CHANGE ***
      READ(IIN,2) N,NVNCH
C *** WRITE HEADING ***
      WRITE(IOUT,4) ISTP,TIME
C *** READ AND WRITE VALUES OF NEW CONTROLLING HEADS ***
      DO 20 I=1,N
      READ(IIN,6) J,HS(J)
      WRITE(IOUT,8) J,HS(J)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VNCORR(H,DH,A,B,TOP,E,HS,IN)
C        FORMS COEFFICIENTS FOR NONLINEAR STEADY-LEAKAGE FUNCTIONS;
C        ADDS TERMS TO MATRIX EQUATION FOR THE CORRECTOR
      DIMENSION H(1),DH(1),A(1),B(1),TOP(1),E(1),HS(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      NME=1
C *** COMPUTE WEIGHTING FACTORS FOR GALERKIN-IN-TIME FORMULATION ***
      CA=1./3.
      CB=2./3.
C *** FOR EACH NODE I ON THE LEAKAGE BOUNDARY: ***
      DO 140 I=1,NEQ
      K=IN(I)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(K.LT.0) GO TO 140
C *** (2) COMPUTE HEAD AT N LEVEL, HO, CHANGE IN HEAD, DHP, AND
C         PREDICTED HEAD, HP ***
      HO=H(I)-DH(K)
      DHP=1.5*DH(K)
      HP=HO+DHP
C *** (3) DETERMINE FUNCTION TYPE BY SIGN OF E(I) ***
      IF(E(I)) 10,140,90
C ***
C *** DISCHARGE-ONLY TYPE ***
   10 EI=-E(I)
C *** (4) REPRESENT ELEVATIONS, AND HEAD AND ELEVATION DIFFERENCES, BY
C         TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      ZE=HS(I)
      ZT=TOP(I)
      DEH=ZE-HO
      DHT=HO-ZT
      DET=ZE-ZT
C *** (5) COMPARE HEADS WITH CONTROLLING ELEVATIONS TO IDENTIFY LEAKAGE
C         CASE. ADD LEAKAGE COEFFICIENT TO MAIN DIAGONAL, IF
C         APPLICABLE, AND COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT-SIDE
C         TERM FOR: ***
      IF(HO.GT.ZE) GO TO 30
      IF(HP.LT.ZE) GO TO 140
      IF(HP.LT.ZT) GO TO 20
C *** CASE (9) ***
      DMHT=DEH-DHT
      PHCF=1.-CA*(DMHT*DMHT+DHT*DEH)/(DHP*DHP)
      TMPA=PHCF*DET
      GO TO 130
C *** CASE (6) ***
   20 PHC=1.-DEH/DHP
      A(NME)=A(NME)+PHC*EI
      TMPA=CB*PHC*(ZE-HP)
      GO TO 130
   30 IF(HO.LT.ZT) GO TO 40
      IF(HP.LT.ZT) GO TO 70
C *** CASE (1) ***
      TMPA=DET
      GO TO 130
   40 IF(HP.GT.ZT) GO TO 60
      IF(HP.GT.ZE) GO TO 50
```

```
C *** CASE (5) ***
      PHE=DEH/DHP
      TMPA=CA*PHE*PHE*DEH
      GO TO 130
C *** CASE (4) ***
   50 A(NME)=A(NME)+EI
      TMPA=ZE-H(I)
      GO TO 130
C *** CASE (3) ***
   60 PHT=DHT/DHP
      TMPA=DET-CA*PHT*PHT*DHT
      GO TO 130
   70 IF(HP.GT.ZE) GO TO 80
C *** CASE (8) ***
      DMHT=DEH-DHT
      PHCF=(DMHT*DMHT+DHT*DEH)/(DHP*DHP)
      TMPA=CA*PHCF*DET
      GO TO 130
C *** CASE (2) ***
   80 PHC=1.+DHT/DHP
      A(NME)=A(NME)+PHC*EI
      TMPA=DET-CB*PHC*(HP-ZT)
      GO TO 130
C ***
C *** STEADY VERTICAL-LEAKAGE TYPE ***
   90 EI=E(I)
C *** (4) REPRESENT CONTROLLING HEAD, AQUIFER TOP ELEVATION, AND
C         DIFFERENCES BY TEMPORARY (UNSUBSCRIPTED) VARIABLES ***
      HA=HS(I)
      ZT=TOP(I)
      DHT=HO-ZT
      DAT=HA-ZT
C *** (5) COMPARE HEADS WITH AQUIFER TOP TO IDENTIFY LEAKAGE CASE. ADD
C         LEAKAGE COEFFICIENTS TO MAIN DIAGONAL, IF APPLICABLE, AND
C         COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT-SIDE TERM FOR: ***
      IF(HO.LT.ZT) GO TO 110
      IF(HP.LT.ZT) GO TO 100
C *** CASE (1) ***
      A(NME)=A(NME)+EI
      TMPA=HA-H(I)
      GO TO 130
C *** CASE (2) ***
  100 PHI=DHT/DHP
      TMPA=DAT-CA*PHI*PHI*DHT
      GO TO 130
  110 IF(HP.GT.ZT) GO TO 120
C *** CASE (4) ***
      TMPA=DAT
      GO TO 130
C *** CASE (3) ***
  120 PHC=1.+DHT/DHP
      A(NME)=A(NME)+PHC*EI
      TMPA=DAT-CB*PHC*(HP-ZT)
C ***
C *** (6) COMPUTE RIGHT-SIDE TERM AND ADD TO B-VECTOR FOR BOTH TYPES OF
```

```
C           NONLINEAR STEADY-LEAKAGE FUNCTIONS ***
  130 B(K)=B(K)+EI*TMPA
      NME=NME+MBWC
  140 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VNFMCO(H,AR,E,HS,ND,NVNZ)
C        READS AND WRITES DATA AND COMPUTES LEAKAGE COEFFICIENTS
C        FOR NONLINEAR STEADY-LEAKAGE FUNCTIONS
      DIMENSION H(1),AR(1),E(1),HS(1)
      DIMENSION ND(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    6 FORMAT (1H0,15X,14HPARAMETERS FOR/1H ,6X,32HNONLINEAR STEADY LEAKA
     1GE BY ZONE/1H ,10X,5HFIRST,5X,6HNO. OF,5X,11HCOEFFICIENT/1H
     2,6H  ZONE,3X,7HEL. NO.,3X,8HELEMENTS,7X,5HVALUE)
    7 FORMAT (1H0,8X,53HNONLINEAR STEADY-LEAKAGE DATA READ, OUTPUT SUPPR
     1ESSED)
    8 FORMAT (3I5,F10.0)
   10 FORMAT (1H ,I5,4X,I5,5X,I5,7X,G11.5)
   12 FORMAT (8F10.0)
   14 FORMAT (1H0,12X,46HCONTROL ELEVATION FOR NONLINEAR STEADY LEAKAGE
     1/1H ,2X,3(4HNODE,7X,5HVALUE,9X))
   15 FORMAT (1H0,8X,70HCONTROL ELEVATION FOR NONLINEAR STEADY LEAKAGE R
     1EAD, OUTPUT SUPPRESSED)
C *** READ INDICATOR VARIABLES FOR SUPPRESSING OUTPUT OF INITIAL VALUES
C     OF NONLINEAR STEADY-LEAKAGE FUNCTIONS ***
      READ(IIN,8) IPNV,IPHS
C *** FOR EACH ZONE SIMULATING NONLINEAR STEADY LEAKAGE: ***
      IF(IPNV.GT.0) GO TO 16
      WRITE(IOUT,6)
      GO TO 18
   16 WRITE(IOUT,7)
   18 DO 40 I=1,NVNZ
C *** (1) READ AND WRITE ZONE DATA ***
      READ(IIN,8) L,NBE,NO,VNCF
      IF(IPNV.EQ.0) WRITE (IOUT,10) L,NBE,NO,VNCF
C *** (2) COMPUTE ELEMENT AND NODE INDICES ***
      NEND=NBE+NO-1
      NDC=4*(NBE-1)+1
      NTE=2*(NBE-1)
C *** FOR EACH ELEMENT IN ZONE I: ***
      DO 30 J=NBE,NEND
      NE=1
C *** (1) CHECK FOR COMBINED INPUT OF NODE NUMBERS FOR TWO ELEMENTS ***
      IF(ND(NDC+3).GT.0) NE=2
C *** (2) COMPUTE LEAKAGE COEFFICIENT, EC, AND STORE IN E-VECTOR
C     BY NODE ***
      DO 20 IE=1,NE
      NA=ND(NDC)
      NB=ND(NDC+IE)
      NC=ND(NDC+IE+1)
      EC=VNCF*AR(NTE+IE)
      E(NA)=E(NA)+EC
      E(NB)=E(NB)+EC
      E(NC)=E(NC)+EC
   20 CONTINUE
      NTE=NTE+2
   30 NDC=NDC+4
   40 CONTINUE
C *** READ AND WRITE CONTROLLING HEAD (OR ELEVATION) ***
```

```
      SUBROUTINE VNPRED(H,A,B,TOP,E,HS,IN)
C        FORMS COEFFICIENTS FOR NONLINEAR STEADY-LEAKAGE FUNCTIONS;
C        ADDS TERMS TO MATRIX EQUATION FOR THE PREDICTOR
      DIMENSION H(1),A(1),B(1),TOP(1),E(1),HS(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      NME=1
C *** FOR EACH NODE I ON THE LEAKAGE BOUNDARY: ***
      DO 40 I=1,NEQ
      K=IN(I)
C *** (1) BYPASS SPECIFIED-HEAD NODES ***
      IF(K.LT.0) GO TO 40
C *** (2) DETERMINE FUNCTION TYPE BY SIGN OF E(I) ***
      IF(E(I)) 5,40,10
C ***
C *** DISCHARGE-ONLY TYPE ***
    5 EI=-E(I)
C *** COMPARE AQUIFER HEAD WITH CONTROLLING ELEVATIONS ***
      IF(H(I).LT.HS(I)) GO TO 35
      IF(H(I).GT.TOP(I)) GO TO 20
      GO TO 15
C ***
C *** STEADY VERTICAL-LEAKAGE TYPE ***
   10 EI=E(I)
C *** COMPARE HEAD WITH AQUIFER TOP ***
      IF(H(I).LT.TOP(I)) GO TO 20
C ***
C *** COMPUTE MAIN DIAGONAL TERM FOR: ***
C *** CASE (4) - DISCHARGE ONLY
C     CASE (1) - STEADY VERTICAL LEAKAGE ***
   15 A(NME)=A(NME)+EI
C *** COMPUTE HEAD DIFFERENCE, TMPA, FOR RIGHT-SIDE TERMS FOR: ***
C *** CASE (4) - DISCHARGE ONLY
C     CASE (1) - STEADY VERTICAL LEAKAGE ***
      TMPA=HS(I)-H(I)
      GO TO 30
C *** CASE (1) - DISCHARGE ONLY ***
C     CASE (4) - STEADY VERTICAL LEAKAGE ***
   20 TMPA=HS(I)-TOP(I)
C *** COMPUTE RIGHT-SIDE LEAKAGE COEFFICIENT AND ADD TO B-VECTOR ***
   30 B(K)=B(K)+EI*TMPA
   35 NME=NME+MBWC
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WTCCHK(H,DH,B,TOP,IN,ISC)
C        CHECKS TO SEE IF A PREDICTED STORAGE CONVERSION DID NOT TAKE
C        PLACE OR IF AN UNPREDICTED CONVERSION DID TAKE PLACE
      DIMENSION H(1),DH(1),B(1),TOP(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      ISC=0
      DO 40 I=1,NEQ
      J=IN(I)
      IF(J.LT.0) GO TO 40
      HP=H(I)+.5*DH(J)
      IF(H(I)-DH(J).LT.TOP(I)) GO TO 20
      IF(HP.GE.TOP(I)) GO TO 30
   10 IF(HP+1.5*B(J).GE.TOP(I)) ISC=1
      GO TO 40
   20 IF(HP.LT.TOP(I)) GO TO 10
   30 IF(HP+1.5*B(J).LT.TOP(I)) ISC=1
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WTFMCO(AR,THK,TOP,ASY,ND)
C        FORMS COEFFICIENTS FOR WATER-TABLE AQUIFER
      DIMENSION AR(1),THK(1),TOP(1),ASY(1)
      DIMENSION ND(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    4 FORMAT (8F10.0)
    5 FORMAT (16I5)
    6 FORMAT (1H0,24X,25HINITIAL AQUIFER THICKNESS/1H ,3(6H  NODE,5X
     1,9HTHICKNESS,5X))
    7 FORMAT (1H0,8X,49HINITIAL AQUIFER THICKNESS READ, OUTPUT SUPPRESSE
     1D)
    8 FORMAT (1H0,25X,24HELEVATION OF AQUIFER TOP/1H ,3(6H  NODE,5X
     1,9HELEVATION,5X))
    9 FORMAT (1H0,8X,45HAQUIFER TOP ELEVATION READ, OUTPUT SUPPRESSED)
   10 FORMAT (1H0,6H  ZONE,5X,3HNO.,5X,8HSPECIFIC/1H ,10X,4HELS.,7X
     1,5HYIELD)
   12 FORMAT (2I5,F10.0)
   14 FORMAT (1H ,I5,3X,I5,4X,G11.5)
C *** READ AND WRITE INITIAL AQUIFER THICKNESS ***
      READ(IIN,5) IPTK,IPTP
      READ(IIN,4) (THK(I),I=1,NNDS)
      IF(IPTK.GT.0) GO TO 16
      WRITE(IOUT,6)
      CALL PRTOA(THK,NNDS)
      GO TO 17
   16 WRITE(IOUT,7)
C *** READ AND WRITE INITIAL ELEVATION OF AQUIFER TOP ***
   17 READ(IIN,4) (TOP(I),I=1,NNDS)
      IF(IPTP.GT.0) GO TO 18
      WRITE(IOUT,8)
      CALL PRTOA(TOP,NNDS)
      GO TO 19
   18 WRITE(IOUT,9)
C *** INITIALIZE NODAL SPECIFIC YIELD COEFFICIENTS ***
   19 DO 20 I=1,NNDS
   20 ASY(I)=0.
C *** BEGIN ZONAL LOOP ***
      NDC=1
      NTE=0
      WRITE(IOUT,10)
      DO 50 IZ=1,NZNS
C *** READ AND WRITE ZONAL SPECIFIC YIELD DATA ***
      READ(IIN,12) KZ,NO,SY
      WRITE(IOUT,14) KZ,NO,SY
C *** BEGIN ELEMENT LOOP WITHIN ZONE ***
      DO 40 I=1,NO
      NE=1
      IF(ND(NDC+3).GT.0) NE=2
      DO 30 IE=1,NE
      NA=ND(NDC)
      NB=ND(NDC+IE)
      NC=ND(NDC+IE+1)
C *** COMPUTE NODAL SPECIFIC YIELD COEFFICIENTS ***
      TESY=SY*AR(NTE+IE)
```

```
      ASY(NA)=ASY(NA)+TESY
      ASY(NB)=ASY(NB)+TESY
   30 ASY(NC)=ASY(NC)+TESY
      NTE=NTE+2
   40 NDC=NDC+4
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WTINIT(G,IDHA,ITKA,ITPA,ISYA)
C        DEFINES AND INITIALIZES VARIABLES FOR A WATER-TABLE AQUIFER
      DIMENSION G(1)
      COMMON/GDIM/ISUM
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      COMMON/ITP/IIN,IOUT,ITA,ITB
C *** FORMAT LIST ***
    2 FORMAT (1H0,20HWATER-TABLE AQUIFER:/1H ,38HNOW G MUST BE DIMENSION
     1ED TO AT LEAST ,I6)
C *** DEFINE ADDRESSES OF ARRAYS WITHIN G ***
      IDHA=ISUM
      ISUM=ISUM+NNDS-NHDS
      ITKA=ISUM
      ISUM=ISUM+NNDS
      ITPA=ISUM
      ISUM=ISUM+NNDS
      ISYA=ISUM
      ISUM=ISUM+NNDS
C *** WRITE NEW SIZE OF G ***
      WRITE(IOUT,2) ISUM
C *** INITIALIZE NEW ELEMENTS OF G ***
      DO 10 I=ITKA,ISUM
   10 G(I)=0.
      RETURN
      END
```

```
      SUBROUTINE XTRPWT(H,DH,DHB,THK,TOP,IN,ISTP)
C        EXTRAPOLATES HEADS AND THICKNESSES AT ALL NODES TO VALUES AT
C        THE END OF THE TIME STEP AND ZEROES HEAD CHANGES
      DIMENSION H(1),DH(1),DHB(1),THK(1),TOP(1)
      DIMENSION IN(1)
      COMMON/NO/NELS,NNDS,NSTEPS,NPER,NZNS,NWELS,NQBND,NHDS,NEQ,MBWC,MBW
      DO 60 I=1,NNDS
      J=IN(I)
      IF(J.GT.0) GO TO 20
      J=-J
C *** DEFINE HEAD CHANGE, DHC, AND ZERO HEAD CHANGE VECTORS DHB
C     AND DH ***
      DHC=DHB(J)
      DHB(J)=0.
      GO TO 30
   20 DHC=DH(J)
      DH(J)=0.
   30 HO=H(I)-DHC
C *** EXTRAPOLATE HEAD FROM THE MEAN POINT IN THE TIME STEP TO THE END
C     OF THE TIME STEP ***
      H(I)=.5*DHC+H(I)
C *** EXTRAPOLATE THICKNESS FROM THE BEGINNING TO THE END OF THE TIME
C     STEP ***
      IF(HO.LT.TOP(I)) GO TO 40
      IF(H(I).GT.TOP(I)) GO TO 60
      THK(I)=THK(I)+H(I)-TOP(I)
      GO TO 60
   40 IF(H(I).LT.TOP(I)) GO TO 50
      THK(I)=THK(I)-HO+TOP(I)
      GO TO 60
   50 THK(I)=THK(I)+1.5*DHC
   60 CONTINUE
      RETURN
      END
```