USGS
science for a changing world

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

Chapter A6

A COUPLED SURFACE-WATER AND GROUND-WATER FLOW MODEL (MODBRANCH) FOR SIMULATION OF STREAM-AQUIFER INTERACTION

By Eric D. Swain and Eliezer J. Wexler

Book 6
MODELING TECHNIQUES

LIST OF VARIABLES —Continued

(Variables specific to original BRANCH code not included)

Variable	Range	Definition
NCOL	Global	Maximum number of MODFLOW aquifer columns.
NELAP	Package	Number of elapsed MODFLOW time steps since beginning of simulation.
NLAY	Global	Maximum number of MODFLOW aquifer layers.
NROW	Global	Maximum number of MODFLOW aquifer rows.
NSEC	Package	Maximum cross sections per branch.
NTSAQ	Package	Number of BRANCH' time intervals in one MODFLOW time step.
NTSAQ1	Package	Adjustable NTSAQ for varying MODFLOW time-step length.
PERIM	Module	Local variable for channel wetted perimeter.
QLSUM (MAXS)	Package	Average leakage rate out of a river segment over one MODFLOW time step.
QPSAV (MAXS)	Package	Value of discharge at end of first BRANCH' time interval after beginning of a MODFLOW time step.
QSAV (MAXS)	Package	Value of discharge at beginning of first BRANCH' time interval in a MODFLOW time step.
SLKG (MAXS)	Package	Vector of leakage rates.
TFCTR	Module	Number of seconds in one time unit used in MODFLOW.
ZBOT (MAXS)	Module	Elevation of channel bottom.
ZN (MAXS)	Module	Value of stage at end of final BRANCH' time interval in a MODFLOW time step.
ZPL (MAXS)	Module	Value of stage at end of final BRANCH' time interval in a MODFLOW time step for previous trial.
ZPSAV (MAXS)	Module	Value of stage at end of first BRANCH' time interval after beginning of a MODFLOW time step.
ZSAV (MAXS)	Package	Value of stage at beginning of first BRANCH' time interval after beginning of a MODFLOW time step.

MODULE BRC1BD

This module calculates the volumetric budget for leakage for a single MODFLOW time step. It is executed each time the two models have converged and calls BRANCH' with a flag set to allow printout in BRANCH'. The numbered sections in the narrative correspond to the numbered steps in the flowchart for BRC1BD shown in figure 9.

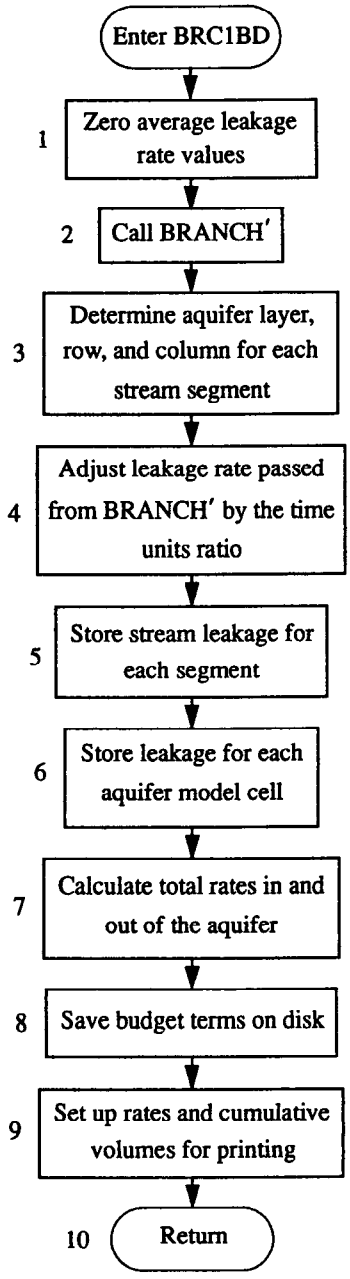

Figure 9. Flowchart of budget module (BRC1BD).

NARRATIVE

1. Zero average leakage rate values. Before BRANCH' is executed, the average leakage rates for each segment (QLSUM [segment number]) are set to zero.
2. Call BRANCH'. BRANCH' is executed with a flag set (IBRPRN=1) so that normal printouts of results are

made (not made when BRANCH' is called from BRC1FM).

3. Determine aquifer layer, row, and column. A loop is begun. For each stream segment, the corresponding aquifer layer, row, and column are located.
4. Adjust leakage rates passed from BRANCH'. The value of the flow rate from river to aquifer (FLOBOT) is calculated as average leakage rate calculated by BRANCH' (QLSUM [segment number]), multiplied by the time units ratio (TFCTR), and converted into units comparable to MODFLOW.
5. Store stream leakage for each segment. The FLOBOT values are stored in the stream leakage rates array (SLKG [segment number]).
6. Store leakage for each aquifer model cell. If the leakage is to be saved, the FLOBOT value is added to the BUFFER value for the corresponding aquifer cell (BUFF [column, row layer]).
7. Calculate total rates in and out of aquifer. If the leakage is out of the aquifer, subtract FLOBOT from the total rate out value (RATOUT); if it is into the aquifer, add FLOBOT to the total rate in (RATIN). The flag that represents the dryness condition of the river segment at a time interval (ITRIAL [segment number, time-interval number]) is set to zero (fully wet). Loop back to step 3 until all river segments have been completed.
8. Save budget terms on disk. If the volumetric budget stored in the buffer in step 6 is to be saved, the subroutine UBUDSV is called (McDonald and Harbaugh, 1988) to write an unformatted record of budget terms.
9. Set up rates and cumulative volumes for printing. The inflow and outflow rates (RATIN and RATOUT) are stored in the VBVL array for printing by BASIOT. In addition, RATIN and RATOUT multiplied by the time-interval length are added to the volume accumulators in the VBVL array, also to be printed by BASIOT. The river budget term labels are moved to the VBNM array for BASIOT printing, and the budget term counter (MSUM) is increased by one.
10. Return to MODFLOW.

PROGRAM LISTING FOR MODULE BRC1BD

```
      SUBROUTINE BRC1BD(INDATA,IZDATA,IQDATA,ITQMAX,ITQMIN,
     1 QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,A,Z,Q,ZP,QP,
     2 AP,BP,RP,B,R,BT,BTP,XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT
     3 ,BETVEL,SUMETA,SUMCZQ,SCZQSQ,SZQETA,ITYPEO,ZA,AA,BB,BS,IPT,
     4 QA,TA,ETA,FUNETA,ROW,AM,BMX,BRNAME,IJF,IJT,NSEC,XSKT,
     5 PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ITYPE,IBJNC,NDATA,IZQBVE,
     6 ISTATN,KTTDBC,ZQ,DTT,DATUM,ZQBVCO,ZQPMIN,LARBPR,ZHIGH,
     7 ZLOW,LINPRT,ARBERR,CLK,ZBOT,
     8 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     9 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,WINDSP,WINDDR,
     & IDX,ICT,W,U,UU,BU,BUU,ZSAV,QSAV,ZPSAV,QPSAV,NELAP,IOUT,ZPL,SLKG,
     & HNEW,HOLD,IBOUND,NCOL,NROW,NLAY,TFCTR,NTSAQ,
     2 DELT,KKSTP,KKPER,ISTRM,
     1 VBVL,VBNM,MSUM,IMPCB1,ICBCFL,IHDDFL,BUFF,IPTFL2,ZN,
     2 ISS,QLSUM,ITRIAL,DCFM)
C-----VERSION  1  01APR1991 BRC1BD                                      C
C     ***************************************************************C
C     CALCULATE VOLUMETRIC BUDGET FOR STREAMS                          C
C     ***************************************************************C
C                                                                      C
C     SPECIFICATIONS:                                                  C
C     -----------------------------------------------------------------C
C                                                                      C
C BEGIN COMMON_COMCON ================================================
C
      CHARACTER*2 IUNIT,OUNIT
      INTEGER*4 NBCH,NJNC,NBND,NSTEPS,IRDGEO,NIT,IPROPT,IPLOPT,IPLDEV,
     1 IPRMSG,IPLMSG,IEXOPT,INHR,INMN,IDTM,IWRTIC,IRDIC,NUMCOM,INWIND,
     2 TYPETA,OTTDDB,ISMOPT,NTDIOF,IRDNXT,IARDEM
      REAL THETA,QQTOL,ZZTOL,WSPEED,WDIREC,WSDRAG,H2ODEN,CHI,QZCONV,
     1 ZDATUM,DT,G,AIRDEN,GLETA,GLBETA,ETAMIN,ETAMAX,TOLERR
      COMMON /COMCON/ NBCH,NJNC,NBND,NSTEPS,IRDGEO,NIT,IPROPT,
     1 IPLOPT,IPLDEV,IPRMSG,IPLMSG,IEXOPT,INHR,INMN,IDTM,IWRTIC,
     2 IRDIC,NUMCOM,INWIND,THETA,QQTOL,ZZTOL,WSPEED,WDIREC,WSDRAG,
     3 H2ODEN,CHI,ZDATUM,IUNIT,OUNIT,TYPETA,OTTDDB,ISMOPT,G,QZCONV,DT,
     4 AIRDEN,IARDEM,NTDIOF,IRDNXT,GLETA,GLBETA,ETAMIN,ETAMAX,TOLERR
C
C  END  COMMON_COMCON ================================================
C BEGIN  COMMON_DTYPES ================================================
C
      CHARACTER*2 DTYPE,ZTYPE,QTYPE,ATYPE,BTYPE,ZPTYPE,QPTYPE,DPTYPE
      COMMON /DTYPES/ DTYPE,ZTYPE,QTYPE,ATYPE,BTYPE,ZPTYPE,QPTYPE,DPTYPE
C
C  END   COMMON_DTYPES ================================================
C BEGIN COMMON_UNITS ==================================================
C
      CHARACTER*2 IBLK,UNIT,EN,ME,MT,FT,TUNIT,DC
      COMMON /UNITS/ IBLK,UNIT,EN,ME,MT,FT,TUNIT,DC
C
C  END  COMMON_UNITS ==================================================
C BEGIN COMMON_LUNUMS ================================================
C
      INTEGER*4 READER,PRINTR,PUNCH,DSREF,TDDATA,LUPTRK,LUIFLO,LUIVOL,
     1         LUGEOM,LUINIT,LUCVOL
      COMMON /LUNUMS/ READER,PRINTR,PUNCH,DSREF,TDDATA,LUPTRK,LUIFLO,
     1 LUIVOL,LUGEOM,LUINIT,LUCVOL
C
C  END  COMMON_LUNUMS ================================================
C BEGIN COMMON_DADCOM ================================================
C
```

Program listing—Continued

```
      INTEGER*2 LISTB,LISTA,STRIP,RTCODE
      COMMON /DADCOM/ LISTB,LISTA,STRIP,RTCODE
C
C  END  COMMON_DADCOM ================================================
C BEGIN COMMON_DAYPMO ================================================
C
      INTEGER*2 DPERM(12)
      COMMON /DAYPMO/ DPERM
C
C  END  COMMON_DAYPMO ================================================
C BEGIN COMMON_LOGICS ================================================
C
      LOGICAL*4 PRTMSG, NOCONV, ERROR, OPLOTS, FOUND, NOEXTP,
     1 NOPRIT, DAYSUM, MOREBD, DTPRT, PTPLT, DAOPEN, STAGES, MODETA
      COMMON /LOGICS/ PRTMSG, NOCONV, ERROR, OPLOTS, FOUND, NOEXTP,
     1 NOPRIT, DAYSUM, MOREBD, DTPRT, PTPLT, DAOPEN, STAGES, MODETA
C
C  END  COMMON_LOGICS ================================================
C BEGIN COMMON_BCTIME ================================================
C
      INTEGER*4 IETIME,NETIME
      INTEGER *2 IRDPDY,IYR,IMO,IDA,IHR,IMN,NYR,NMO,NDA,NHR,NMN
      COMMON /BCTIME/ IETIME,NETIME,IRDPDY,IYR,IMO,IDA,IHR,IMN,
     1                                     NYR,NMO,NDA,NHR,NMN
C
C  END  COMMON_BCTIME ================================================
C BEGIN COMMON_DATIME ================================================
C
      INTEGER*4 KYR,KMO,KDA,KHR,KMN,M,KYRS,KMOS,KDAS,KHRS,KMNS
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,M,KYRS,KMOS,KDAS,KHRS,KMNS
C
C  END  COMMON_DATIME ================================================
C BEGIN COMMON_NETWRK ================================================
C
      CHARACTER*80 NETNAM
      COMMON /NETWRK/ NETNAM
C
C  END  COMMON_NETWRK ================================================
C BEGIN COMMON_MODBRCH ===============================================
C
      COMMON /MODBRCH/ TWOCSQ,IDTPDY,TWOG,CW,II,ONECHI,DCHI,DTHETA,
     1                       IBCH,IJZPBC,IJQPBC,DCFM1,KKITER
C
C  END  COMMON_MODBRCH ================================================
      INTEGER*4 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT
      INTEGER*4 ITQMAX(MAXS), ITQMIN(MAXS),ISTRM(3,MAXS),ITRIAL(MAXS,
     1 MAXCZQ)
      REAL QMAX(MAXS), QMIN(MAXS), QSUM(MAXS), ZQMIN(MAXS), ZQMAX(MAXS),
     1     AQMAX(MAXS), AQMIN(MAXS)
      REAL A(MAXS),Z(MAXS),Q(MAXS),ZP(MAXS),QP(MAXS),AP(MAXS),
     1 BP(MAXS),RP(MAXS),B(MAXS),R(MAXS),BT(MAXS),BTP(MAXS),
     2 ZSAV(MAXS),QSAV(MAXS),ZPSAV(MAXS),QPSAV(MAXS),ZPL(MAXS),
     3 ZN(MAXS)
      INTEGER*4 XSTATN(MAXS)
      REAL DX(MAXS), T(MAXS), RN(4,MAXS), WANGLE(MAXS), GDATUM(MAXS),
     1 ORIENT(MAXS), BETVEL(MAXS), SUMETA(MAXS), SUMCZQ(MAXS),
     2 SCZQSQ(MAXS),SZQETA(MAXS)
      CHARACTER*4 ITYPEO(4,MAXS)
      INTEGER*4 IPT(MAXS)
```

Program listing—Continued

```
      REAL ZA(MXPT,MAXS), AA(MXPT,MAXS), BB(MXPT,MAXS), BS(MXPT,MAXS)
      REAL QA(MXPT,MAXS), TA(MXPT,MAXS), ETA(MXPT,MAXS),
     1 FUNETA(MXPT,MAXS)
      INTEGER*4 ROW(4*MXBH)
      REAL*4 AM((4*MXBH)**2),BMX(4*MXBH)
      CHARACTER*40 BRNAME(MXBH)
      INTEGER*4 IJF(MXBH),IJT(MXBH),NSEC(MXBH),XSKT(MXBH),
     1 PLTBCH(MXBH),PRTXSG(MXBH),PRTBCH(MXBH),PRTSUM(MXBH),PPLTBH(MXBH)
      CHARACTER*4 ITYPE(MXJN)
      INTEGER*4 IBJNC(MXJN),NDATA(MXJN),IZQBVE(MXJN)
      INTEGER*4 ISTATN(MXJN),KTTDBC
      REAL ZQ(MAXZBD,MXTDBC),DTT(MXJN),DATUM(MXJN),ZQBVCO(4,MXJN),
     1 ZQPMIN(MXJN)
      LOGICAL*4 LARBPR(MAXS),ZHIGH(MAXS),ZLOW(MAXS),LINPRT(MAXS),ARBERR
      INTEGER*4 IZDATA(MAXZBD)
      INTEGER*4 IQDATA(MAXQBD),INDATA(MAXQBD)
C     EQUIVALENCE (INDATA(1),IQDATA(1),IZDATA(1))
      REAL CLK(MAXS),ZBOT(MAXS)
      REAL WINDSP( MXWIND),WINDDR( MXWIND)
      INTEGER*4 IDX(MXJN,MXBH), ICT(MXJN)
      REAL W(MXJN)
      REAL*8 U(2*MAXS), UU(4*MAXS), BU(2*MXBH),BUU(4*MXBH)
      REAL*8 C1, C2, C3, C4, UUIJP1, UUIJP2, UUIJP3, UUIJP4
      REAL LAMBDA,MU,SETA,WDTT,TWOCSQ,TWOG,CW,ONECHI,DCHI,DTHETA,TH,WIND
      INTEGER*4 IAR,I,J,K,L,II,IJ,NS,KT,IS,N,NWREAD,NWDATA,INTDBC,IDTPDY
      REAL QTOL,ZTMIN,ZTMAX,ZPMIN,QPMIN,DXMIN,DXMAX
      COMMON /LIMITS/ QTOL,ZTMIN,ZTMAX,ZPMIN,QPMIN,DXMIN,DXMAX
      INTEGER*2 JYR,JMO,JDA,JHR,JMN,MYR,MMO,MDA,MHR,MMN
      INTEGER*4 ND,NDFIRT,NDPART,JETIME,NTSAQ
      COMMON /PARTIM/ ND,NDFIRT,NDPART,JETIME,JYR,JMO,JDA,JHR,JMN,
     1MYR,MMO,MDA,MHR,MMN
      CHARACTER*80 COMENT(9)
      COMMON /CMMNT/ COMENT
      CHARACTER*2 IDETA(7)
      COMMON /ETASYM/ IDETA
      REAL WRATIO,DTZERO,ZTEMP,QTEMP,ZIJ,QIJ,DXIJ,QIJP1,ZIJP1,APZPIJ,
     1     BPZPIJ,BTZPIJ,RPZPIJ,BAVG,BTAVG,AAVG,RAVG,QAVG,ZAVG,BETCOR,
     2     RNIJ,AAVGSQ,AAVGCU,SIGMA,EPSLON,ZETA,OMEGA,GAMMA,DELTA,DET,
     3 DZDT,DQDXC,DQDT,DQDXM,DADX,DZDX,FRIC,ZQPIJ,BIGQ,BIGZ,ZTOL,SOLPDT
     4 ,ALPHA
      INTEGER*4 IBCH,IJZPBC,IJQPBC,KTMATS,LASTN,IJP1,NSM1,JP1,IJ2,IJ4,
     1  IJ4P1,IJ4P2,IJ4P3,IJ4P4,IJ2P1,IJ2P2,I2,I4,I4P1,I4P2,I4P3,I4P4,
     2  I2P1,I2P2,NN,MM,NNN,NBPJ,M0,IBIGZ,JBIGZ,IBIGQ,JBIGQ,IJPNS,ICHK,
     3  IBOUND(NCOL,NROW,NLAY)
      DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY),
     2    QLSUM(MAXS+MXJN)
      CHARACTER*4 VBNM,TEXT,STRTXT
      DIMENSION BUFF(NCOL,NROW,NLAY),VBVL(4,20),VBNM(4,20)
      DIMENSION TEXT(4),STRTXT(4),SLKG(MAXS)
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'  BR','ANCH',' LEA','KAGE'/
      DATA STRTXT(1),STRTXT(2),STRTXT(3),STRTXT(4) /'BRAN','CH F',
     1                                              'LOW ','OUT '/
C     ------------------------------------------------------------------C
        IDBG=0
        IBEGIN=1
C                                                                       C
C       WRITE(*,*) 'DOWN TO BRC1BD'
C                                                                       C
C1 ----ZERO LEAKAGE QUANTITIES
```

Program listing—Continued

```
      DO 80 I=1,NBCH
      NSM1=NSEC(I)-1
      IJ=MAXS-XSKT(I)
      IJFI=MAXS+IJF(I)
      IJTI=MAXS+IJT(I)
      DO 81 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      QLSUM(IJ)=0.0
      QLSUM(IJP1)=0.0
81    CONTINUE
      QLSUM(IJFI)=0.0
      QLSUM(IJTI)=0.0
80    CONTINUE
C2 ------CALL BRANCH.
C
      IBRPRN=1
      CALL BRCH(INDATA,IZDATA,IQDATA,ITQMAX,ITQMIN,
     1 QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,A,Z,Q,ZP,QP,
     2 AP,BP,RP,B,R,BT,BTP,XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT
     3 ,BETVEL,SUMETA,SUMCZQ,SCZQSQ,SZQETA,ITYPEO,ZA,AA,BB,BS,IPT,
     4 QA,TA,ETA,FUNETA,ROW,AM,BMX,BRNAME,IJF,IJT,NSEC,XSKT,
     5 PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ITYPE,IBJNC,NDATA,IZQBVE,
     6 ISTATN,KTTDBC,ZQ,DTT,DATUM,ZQBVCO,ZQPMIN,LARBPR,ZHIGH,
     7 ZLOW,LINPRT,ARBERR,CLK,ZBOT,
     8 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     9 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,WINDSP,WINDDR,
     & IDX,ICT,W,U,UU,BU,BUU,ZSAV,QSAV,ZPSAV,QPSAV,NELAP,IOUT,
     & NTSAQ,HOLD,HNEW,IBOUND,NCOL,NROW,NLAY,ISTRM,ZPL,IBEGIN,
     & IBRPRN,ZN,ISS,QLSUM,ITRIAL,DCFM)
C                                                                    C
C3 ------DETERMINE LAYER, ROW, COLUMN OF EACH REACH.
      DCFM1=DCFM
      RATIN=0
      RATOUT=0
      DO 500 I=1,NBCH
      NSM1=NSEC(I)-1
      L=MAXS-XSKT(I)
      DO 501 J=1,NSM1
      L=L+1
      IF(ISTRM(3,L).LT.0) GO TO 501
      IC=ISTRM(1,L)
      IR=ISTRM(2,L)
      IL=ISTRM(3,L)
C                                                                    C
C4 ----CALCULATE LEAKAGE RATE                                        C
C
C4A ----IF NTSAQ=1, LEAKAGE RATE IS CALCULATED IMPLICITLY
      IF(NTSAQ.EQ.1) THEN
      DXL=DX(L)
      HSTR=ZN(L)+ZDATUM
      PERIM=BP(L)
      CSTR=CLK(L)*PERIM*DXL*TFCTR
      HAQ=HNEW(IC,IR,IL)
      BOT=ZBOT(L)
      IF(HAQ.GT.BOT) THEN
        FLOBOT=0.5*CSTR*(HSTR-HAQ)
      ELSE
        FLOBOT=0.5*CSTR*(HSTR-BOT)
      END IF
```

Program listing—Continued

```
        HSTR=ZN(L+1)+ZDATUM
        PERIM=BP(L+1)
        CSTR=CLK(L)*PERIM*DXL*TFCTR
        BOT=ZBOT(L+1)
        IF(HAQ.GT.BOT) THEN
          FLOBOT=FLOBOT+0.5*CSTR*(HSTR-HAQ)
        ELSE
          FLOBOT=FLOBOT+0.5*CSTR*(HSTR-BOT)
        END IF
        ELSE

C4B ---IF NTSAQ>1, THE LEAKAGE RATE IS TAKEN FROM BRANCH
          FLOBOT=QLSUM(L)*TFCTR
        ENDIF
C                                                                       C
C5 ----STORE STREAM INFLOW, OUTFLOW AND LEAKAGE FOR EACH REACH.         C
        SLKG(L)=FLOBOT
C                                                                       C
C6 ----IF LEAKAGE FROM STREAMS IS TO BE SAVED THEN ADD RATE TO BUFFER.
        IF(IBD.EQ.1) BUFF(IC,IR,IL)=BUFF(IC,IR,IL)+FLOBOT
C                                                                       C
C7 -----SUBTRACT FLOW RATE FROM RATOUT IF AQUIFER DISCHARGES TO STREAM.C
C         OR ADD FLOW RATE TO RATIN IF STREAM DISCHARGES TO AQUIFER.
        IF(FLOBOT.LT. 0.0) RATOUT=RATOUT-FLOBOT
        IF(FLOBOT.GT. 0.0) RATIN=RATIN+FLOBOT
C                                                                       C
        ITRIAL(L,1)=ITRIAL(L,NTSAQ+1)
        DO 1800 ICOT=2,NTSAQ+1
 1800   ITRIAL(L,ICOT)=0
  501   CONTINUE
  500   CONTINUE
C                                                                       C
C8 -----IF BUDGET TERMS WILL BE SAVED THEN WRITE TO DISK.               C
        IF(IBD.EQ.1) CALL UBUDSV(KKSTP,KKPER,TEXT,IMPCB1,BUFF,NCOL,NROW,
     1                           NLAY,IOUT)
C                                                                       C
C9 ------MOVE RATES INTO VBVL FOR PRINTING BY MODULE BAS_OT.            C
    600 VBVL(3,MSUM)=RATIN
        VBVL(4,MSUM)=RATOUT
C                                                                       C
C9A -----MOVE PRODUCT OF RATE AND TIME STEP INTO VBVL ACCUMULATORS.     C
        VBVL(1,MSUM)=VBVL(1,MSUM)+RATIN*DELT
        VBVL(2,MSUM)=VBVL(2,MSUM)+RATOUT*DELT
C                                                                       C
C9B -----MOVE BUDGET TERM LABELS INTO VBNM FOR PRINTING BY BAS_OT.      C
        VBNM(1,MSUM)=TEXT(1)
        VBNM(2,MSUM)=TEXT(2)
        VBNM(3,MSUM)=TEXT(3)
        VBNM(4,MSUM)=TEXT(4)
C                                                                       C
C9C ----INCREASE BUDGET TERM COUNTER BY ONE.                            C
        MSUM=MSUM+1
C                                                                       C
C9D ----RESET IBD COUNTER TO ZERO.                                      C
        IBD=0
        IF((IPTFL2.GT.0).OR.(IHDDFL.LE.0)) GO TO 800
        WRITE(IOUT,710)
  710   FORMAT(1H0,8X,'LAYER',2X,'ROW',1X,'COLUMN',1X,'REACH',1X,
     1'SEGMENT',4X,'AQUIFER HEAD',6X,'STAGE',6X,'LEAKAGE'/)
        DO 750 I=1,NBCH
```

Program listing—Continued

```
      NSM1=NSEC(I)-1
      L=MAXS-XSKT(I)
      DO 751 J=1,NSM1
      L=L+1
      IF(ISTRM(3,L).LT.0) GO TO 751
      IC=ISTRM(1,L)
      IR=ISTRM(2,L)
      IL=ISTRM(3,L)
      HAQ=HNEW(IC,IR,IL)
      STAGE = (ZN(L)+ZN(L+1))/2.+ZDATUM
      WRITE(IOUT,775)IL,IR,IC,I,J,HAQ,STAGE,SLKG(L)
  775 FORMAT(1X,5X,5I6,8X,F11.4,1X,F12.4,4X,F9.2)
  751 CONTINUE
  750 CONTINUE
  800 CONTINUE
      NELAP=NELAP+1
C                                                                    C
C10-----RETURN.                                                      C
      RETURN
      END
```

LIST OF VARIABLES

(Variables specific to original BRANCH code are not referenced)

Variable	Range	Definition
BOT	Module	Local variable for channel bottom elevation.
BUFF (NCOL, NROW, NLAY)	Global	Buffer used to accumulate information before sending it to a disk file.
CLK (MAXS)	Package	Value of K'/b' (leakage coefficient).
CSTR	Module	Local variable for $K'/b' \cdot B \cdot L \cdot$ TFCTR.
DCFM	Package	Multiplier for the friction term in the momentum equation when the channel runs dry.
DELT	Global	Time-step length in MODFLOW.
DH	Module	Difference in stage between the last two MODBRANCH iterations.
DHMAX	Module	Absolute value of DHSIGN.
DHSIGN	Module	Maximum difference in stage between the last two MODBRANCH iterations.
DXL	Module	Local variable for segment length.
FLOBOT	Module	Leakage rate between BRANCH and MODFLOW.
HAQ	Module	Aquifer head at upstream node at beginning of BRANCH time interval.
HNEW (NCOL, NROW, NLAY)	Global	Aquifer head at end of MODFLOW time step.
HOLD (NCOL, NROW, NLAY)	Global	Aquifer head at beginning of MODFLOW time step.
HSTR	Module	Local variable for stage in the channel.
IBEGIN	Package	Flag indicating if BRANCH' is being called for the first time (0, first time; 1, not first time).

LIST OF VARIABLES —Continued

(Variables specific to original BRANCH code are not referenced)

Variable	Range	Definition
IBOUND (NCOL, NROW, NLAY)	Global	Status of each cell (<0 is constant head cell, = 0 is inactive cell, >0 is variable head cell).
IBRPRN	Package	Flag indicating if BRANCH' is being called from the formulation or budget package (0, formulation; 1, budget).
IOUT	Package	FORTRAN unit number for printed output.
ISS	Global	Flag indicating steady-state simulation (0, not steady state; 1, steady state).
ISTRM (3, MAXS)	Package	Vector describing row, column, and layer of aquifer model that corresponds to channel segment.
ITRIAL (MAXS), MAXCZQ)	Package	Flag indicating the wet-dry condition of a channel reach and the relative position of the surrounding aquifer to the riverbed by a two-digit number. For first digit, 0 is reach wet, 1 is upstream node dry, 2 is downstream node dry, and 3 to 8 are both nodes dry with successively high frictional resistance. For second digit, 0 is aquifer above riverbed at both nodes, 1 is aquifer below riverbed at upstream node, 2 is aquifer below riverbed at downstream node, and 3 is aquifer below riverbed at both nodes.
KDAS	Package	Saved value of day.
KHRS	Package	Saved value of hour.
KKITER	Global	MODFLOW iteration number.

LIST OF VARIABLES —Continued

(Variables specific to original BRANCH code are not referenced)

Variable	Range	Definition
KKPER	Global	MODFLOW stress-period number.
KKSTP	Global	MODFLOW time-step number.
KMNS	Package	Saved value of minute.
KMOS	Package	Saved value of month.
KYRS	Package	Saved value of year.
MAXCZQ	Package	Maximum number of daily computed results held in storage for plotting purposes.
MAXMZQ	Package	Maximum number of measured data held in storage for plotting purposes.
MAXQBD	Package	Maximum number of discharge boundary value data held in storage for computation.
MAXS	Package	Maximum number of cross sections in the entire channel network.
MAXZBD	Package	Maximum number of stage boundary value data held in storage for computation.
MSUM	Global	Counter for budget entries and labels into VBVL and VBNM.
MXBH	Package	Maximum number of branches in the network.
MXJN	Package	Maximum number of junctions in the network.
MXMD	Package	Maximum number of measured data locations accommodated in the network.
MXPT	Package	Maximum number of points used to define a cross section.
MXTDBC	Package	Maximum number of boundaries in the network.
MXWIND	Package	Maximum number of wind data points input.
NCOL	Global	Maximum number of MODFLOW aquifer columns.
NELAP	Package	Number of elapsed MODFLOW time steps since beginning of simulation.
NLAY	Global	Maximum number of MODFLOW aquifer layers.
NROW	Global	Maximum number of MODFLOW aquifer rows.
NSEC	Package	Maximum cross sections per branch.
NTSAQ	Package	Number of BRANCH' time intervals in one MODFLOW time step.
PERIM	Module	Local variable for channel wetted perimeter.
QLSUM (MAXS)	Package	Average leakage rate out of a river segment over one MODFLOW time step.
QPSAV (MAXS)	Package	Value of discharge at end of first BRANCH' time interval after beginning of a MODFLOW time step.
QSAV (MAXS)	Package	Value of discharge at beginning of first BRANCH' time interval in a MODFLOW time step.
RATIN	Package	Flow rate from channel to aquifer.
RATOUT	Package	Flow rate from aquifer to channel.
SLKG (MAXS)	Package	Vector of leakage rates.
STAGE	Module	Average stage in channel segment.
TEXT	Module	Label to be recorded with leakage into or out of aquifer model cells from channel.
TFCTR	Module	Number of seconds in one time unit in MODFLOW.
VBNM (4,20)	Global	Labels for entries in volumetric budget.
VBVL (4,20)	Global	Entries for the volumetric budget.
ZBOT (MAXS)	Module	Elevation of channel bottom.
ZN (MAXS)	Module	Value of stage at end of final BRANCH' time interval in a MODFLOW time step.
ZPL (MAXS)	Module	Value of stage at end of final BRANCH' time interval in a MODFLOW time step for previous trial.
ZPSAV (MAXS)	Module	Value of stage at end of first BRANCH' time interval after beginning of a MODFLOW time step.
ZSAV (MAXS)	Module	Value of stage at beginning of first BRANCH' time interval after beginning of a MODFLOW time step.

MODEL BRANCH'

The BRANCH' model (or submodule called by a MODBRANCH module) simulates the unsteady flow in networks of reaches composed of interconnecting channels with leakage to the aquifer. It is the computation section of the BRANCH model. The BRANCH' model was modified to (1) equate the BRANCH' calculations with the time sequence in MODFLOW, (2) allow an iterative solution between MODFLOW and BRANCH', (3) model the effects of riverbed leakage to and from the aquifer, (4) allow the channel to run dry, (5) implement an alternate steady-state solution, and (6) pass leakage information back to MODFLOW.

This module is called either by BRC1FM when working out the iterative solution between MODFLOW and BRANCH' or by BRC1BD when an iterative solution has been completed for a MODFLOW time step and a printout

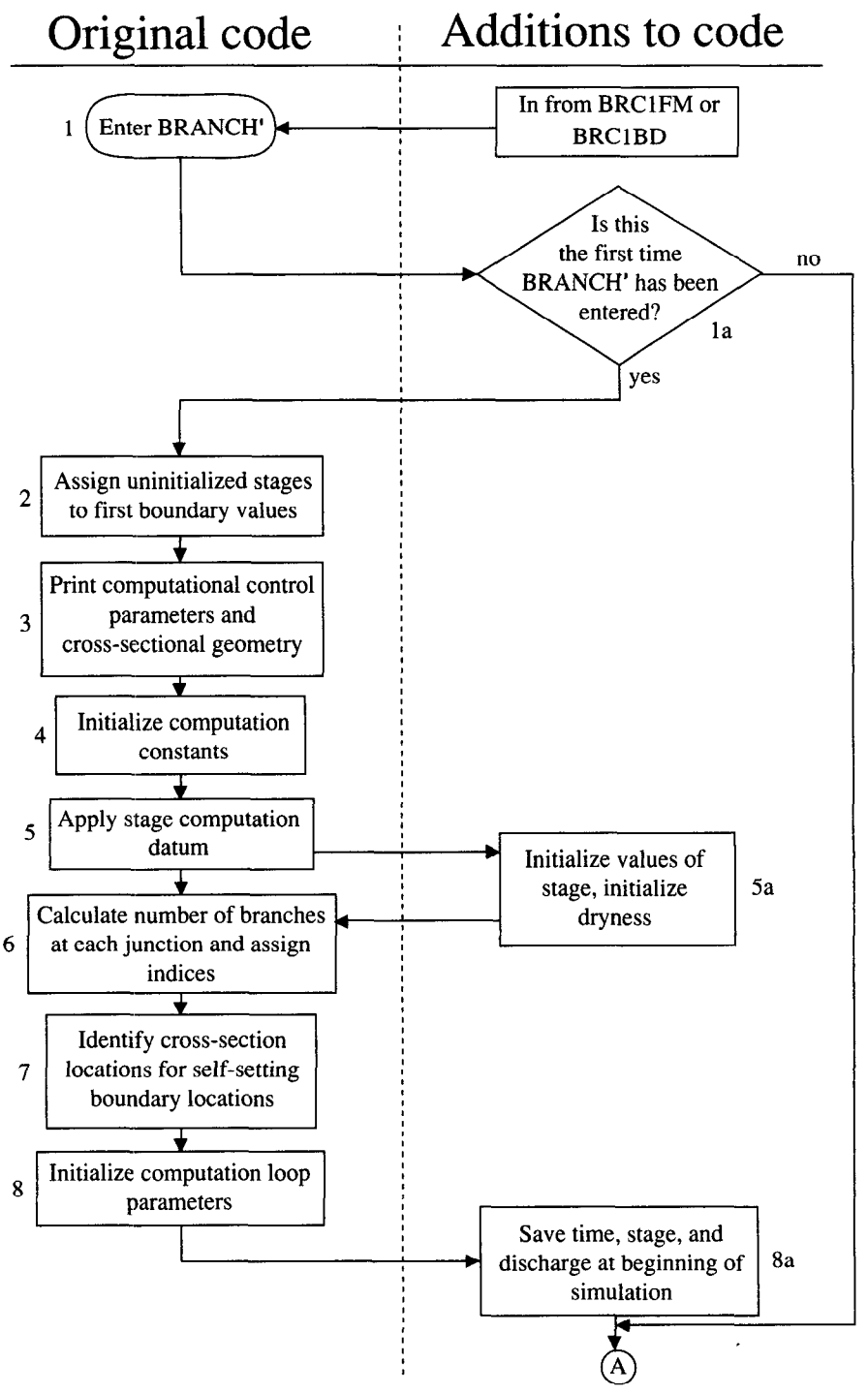

Figure 10. Flowchart of modified BRANCH'.

and budget calculation are needed. The numbered sections in the narrative correspond to the numbered steps in the flowchart for BRANCH' in figure 10. The flowchart is divided with the original code to the left of the dotted line and the modifications made to interface with MODFLOW to the right. The modifications are signified by a number and letter (the original code by a number only). More details are presented in this report for the modifications than for the original code, which is described by Schaffranek and others (1981).

NARRATIVE

1. Enter BRANCH'. Upon entry, the statement function that locates elements in the coefficient matrix is set, the

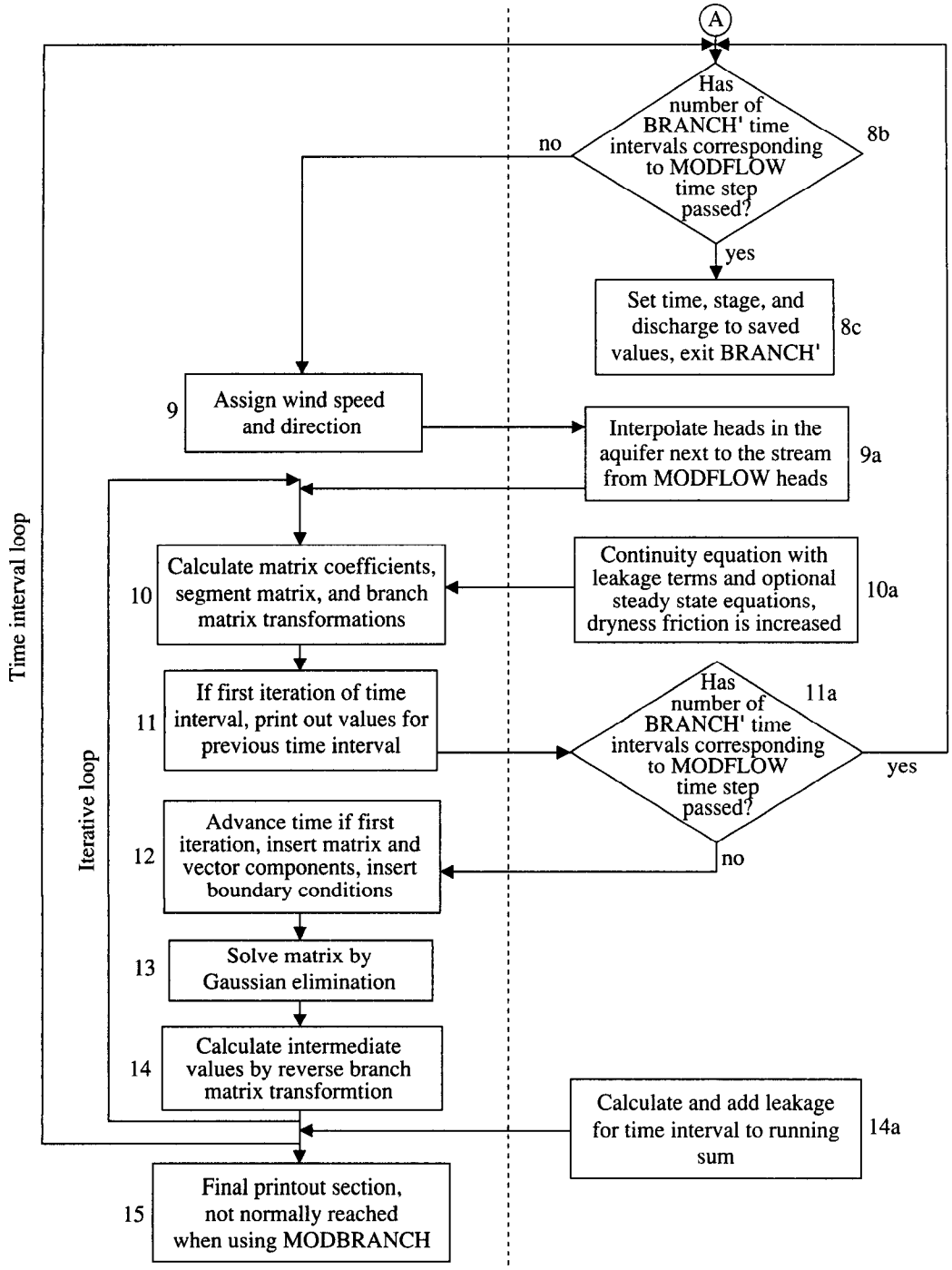

Figure 10. Continued.

MODFLOW and BRANCH' print devices are made equivalent, and the counter of BRANCH' time intervals elapsed within the MODFLOW time step (one count) is initialized.

1a. If this is not the first time interval, skip initialization. The following initialization steps (2–8a) are necessary once at the beginning of the simulation. Thus, if the initial flag is set (IBEGIN=1), steps 2 through 8a are skipped.

2. Assign uninitialized stage values to first boundary values. If the initial stages at the boundary nodes are not given values, their values are set to the first boundary value data point for the corresponding node.

3. Print computational-control parameter table and cross-section geometry data. The input data are written to the output file, and the Courant numbers and computational datum are calculated.
4. Initialize computation constants. The wind speed and Manning's n are adjusted for inch-pound or metric units. The number of BRANCH' time intervals per day are calculated, and various multiples of the weighting factors and gravitational acceleration commonly used are calculated.
5. Apply stage computation datum. The stage computation datum (ZDATUM), which adjusts stages so they all have lower absolute values in the computations, is applied to all stages and to the cross-sectional geometry data.

5a. Initialize stage values. Initialize dryness array (ITRIAL). Within the loop in step 5, the values of stages in BRANCH' for the end of the previous MODFLOW time step (ZPL [node number]) and this MODFLOW time step (ZN [node number]) are initialized to the initial values of stage in BRANCH' at the beginning of the simulation. The array of time-specified channel dryness flags (ITRIAL [node number, time-step number]) is also set to zero values (totally wet).

6. Calculate number of branches at each junction and assign indices. Indices are assigned to properly define the continuity of flows from all branches that meet at a junction.
7. Identify cross-section locations for self-setting boundary conditions. The self-setting boundary conditions (approximations of free outflows) are located.
8. Initialize computation loop parameters. The initial time counter, number of days in February (depends on leap year) matrix solution counter, and several other counters are initialized.

8a. Save time, stage, and discharge at beginning of simulation. The initial values of year, month, day, hour, minute, stages, and discharges are saved in the variables KYRS, KMOS, KDAS, KHRS, KMNS, ZSAV (node number), ZPSAV (node number), QSAV (node number), and QPSAV (node number).

8b. Check if the number of BRANCH' time intervals corresponding to the single MODFLOW time step has passed. If so, step 8c is executed. If not, the algorithm proceeds to step 9.

8c. If the solution has not converged between MODFLOW and BRANCH', the time, stage, and discharge values are reset to saved values, and BRANCH' is exited. If the solution has converged, BRANCH' is exited without resetting values. An expansion of sections 8b and 8c is shown in figure 11. The following lettered steps (a–i) correspond to the flowchart shown in figure 11.

(a) Advance the time-interval counter by one.

(b) If the time-interval counter is equal to the number of elapsed MODFLOW time steps (NELAP), multiplied by the number of BRANCH' time intervals (NTSAQ), plus 2, the preparations for exiting the BRANCH' model (steps c–i) are made. The term "plus 2" occurs because the time-interval loop must pass one extra time to print the last time-interval data and the counter adds one more in step a, making the counter equal to 2 more than

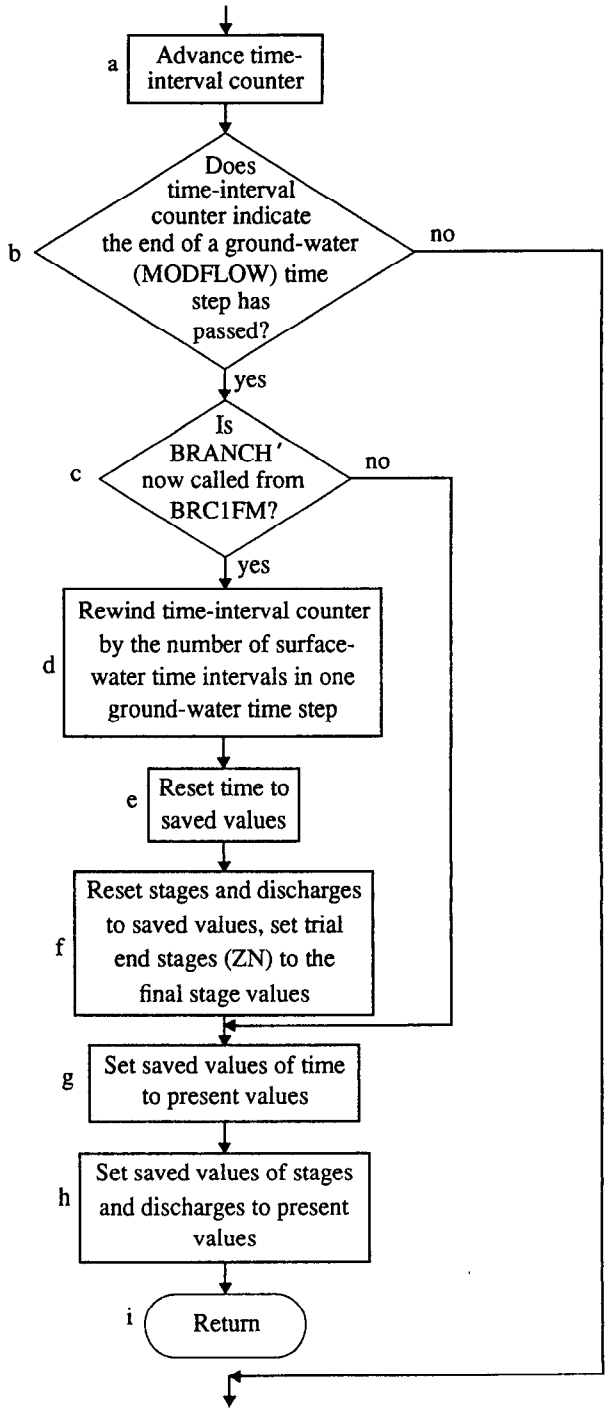

Figure 11. Expanded flowchart of step 8b in BRANCH'.

NELAP times NTSAQ. If the time-interval counter is not equal to NELAP×NTSAQ+2, the rest of section 8b (steps c–i) is skipped and step 9 is executed.

(c) If BRANCH' is being called from the formulation module (IBRPRN=0), it is assumed that the models have not converged for the MODFLOW time steps, and steps d to f are implemented to rewind the clock and repeat the calculations for the next trial. If BRANCH' is being called from the budget module (IBRPRN=1), the models have converged and steps d to f are skipped.

(d) The time-interval counter (M) is reduced by NTSAQ.

(e) The date and time (KYR, KMO, KDA, KHR, and KMN) are assigned to the saved values (KYRS, KMOS, KDAS, KHRS, and KMNS).

(f) The values of stage at the end of the trial (ZN [node number]) are assigned to the final stages (ZP [node number]). The stages and discharges are assigned to the saved values.

(g) The saved values of date and time are assigned to the present values.

(h) The saved values of stages and discharges are assigned to the present values.

(i) Return to calling module (BRC1FM or BRC1BD).

9. Assign wind speed and direction. After advancing the counter of BRANCH' time intervals within the MODFLOW time step (ICOUNT) and printing date and time if call is made from the budget module, the wind speed and direction are linearly interpolated from the time-dependent wind data. In preparation for the next time interval, the starting values of stage and time are set to the final values of the previous time interval. Average values of topwidth, area, hydraulic radius, discharge, stage, and momentum coefficient are calculated.

9a. Interpolate head in aquifer next to stream from MODFLOW heads. The heads in the aquifer at the beginning and end of the BRANCH' time interval (HAQ and HAQP) are linearly interpolated, using ICOUNT, from the aquifer heads at the beginning and end of the MODFLOW time step (HOLD and HNEW). If the river segment crosses an inactive aquifer model cell (IBOUND=0), the leakage coefficient for the segment is set to zero. If the aquifer at a river node at a specific time drops below the river bottom, the aquifer head, for the purpose of leakage, is set to the river bottom. If both aquifer and river stage drop below the river bottom at the present time (beginning of time interval), the leakage coefficient is temporarily set to zero (dry channel).

10. Calculate matrix coefficients, segment matrix, and branch matrix coefficients. This procedure is described by Schaffranek and others (1981). In short, the coefficients in the matrix form of the channel flow equation 7 are calculated, the branch-transformation process is implemented, and the coefficient positions are located.

10a. Continuity equation is arranged with leakage terms, and if steady-state conditions are simulated, without time-dependent terms. Dryness friction is increased. These modifications are placed directly into step 10 and basically implement the three major options added to the model. The inclusion of leakage terms in the continuity equation and the steady-state form of the equations are described in the formulation section of this report. The steady-state flag (ISS) is checked, and if MODFLOW is running in steady state, the steady-state forms of the continuity and momentum equations are used. The flag that signifies the dryness of a channel in previous trials (ITRIAL [segment number, time-interval number]) is checked. Based on the information in ITRIAL, the friction term (σ from Schaffranek and others, 1981) is multiplied by the appropriate fraction of DCFM to increase the dryness friction. If ITRIAL indicates a channel node dry and the aquifer below the riverbed, the leakage coefficient is set to zero at the node.

11. If first iteration of time interval, print values for previous time interval. If the iteration loop is on its first iteration and IPRPRN=1, the subroutine DTOUT is called to print discharges, velocities, stages, areas, topwidths, and Manning's n values at all nodes. This position for the printout of results from the last time interval was retained from the original code. (Setting the value of IPROPT in the computation control record to 3 will cause the values only to be printed for the last BRANCH time interval in a MODFLOW time step.)

11a. If number of BRANCH' time intervals corresponding to the MODFLOW time step has passed, go to beginning of the computation loop (step 8b). This allows the printout of the last time interval (step 11), but the next set of calculations (steps 12–14) does not occur unless further BRANCH' time intervals occur before the end of the MODFLOW time step.

12. Advance time if first BRANCH' iteration, insert matrix and vector components, and insert boundary conditions. The time is advanced by one BRANCH' time interval. The BRANCH' matrices calculated in step 10 are placed in the solution matrix. The right-hand side vector components are also put in position. For junctions, discharge continuity and stage compatibility conditions are created. Boundary conditions are calculated from the boundary-condition equation, parabolically interpolated from boundary-value data, or established by the self-setting stage or discharge boundary conditions.

13. Solve matrix by Gaussian elimination. The BRANCH' subroutine GEMXP solves the matrix of linear equations.

14. Calculate intermediate values by reverse branch matrix transformation. The values of stage and discharge at cross sections in the branch are calculated by reversing the branch matrix transformation in step 10. If solution for the BRANCH' time interval has not converged, go back to the beginning of the iterative loop. The largest changes in stage and discharge are compared to the convergence criterion to see if the BRANCH' solution has been completed. If not, return to step 10.

14a. Calculate and add leakage rate for time interval to running sum. The leakage rate for the last time interval is divided by NTSAQ and added to the sum (QLSUM [segment number]). Thus, after NTSAQ time intervals have passed (equal to the passage of one MODFLOW time step), QLSUM equals average leakage rate over the MODFLOW time step. After this, return to step 8a. Based on the positions of the aquifer heads and channel stages relative to the riverbed and the previous values of ITRIAL, and the variable expressing channel dryness, the values of ITRIAL (segment number, time-interval number) are reset. To avoid dual solution oscillations, the following criteria are followed: for the first five iterations, between BRANCH' and MODFLOW, all solutions of ITRIAL are accepted; from iterations 6 through 30, preference is given to solutions where the aquifer head and river stage are on the same side of the riverbed (above or below) or the river is completely wet; no changes in ITRIAL are made beyond iteration 30.

15. Final printout section is not normally reached when using MODBRANCH. This section from the original BRANCH prints cumulative flows and the final time interval. It is reached when the BRANCH boundary value data are exhausted or the number of time intervals specified in BRANCH is finished. Normally MODFLOW, the driver for BRANCH', controls when the simulation is ended and if step 15 is reached, an error message is printed.

PROGRAM LISTING FOR MODEL BRANCH'

```
      SUBROUTINE BRCH(INDATA,IZDATA,IQDATA,ITQMAX,ITQMIN,
     1 QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,A,Z,Q,ZP,QP,
     2 AP,BP,RP,B,R,BT,BTP,XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT
     3 ,BETVEL,SUMETA,SUMCZQ,SCZQSQ,SZQETA,ITYPEO,ZA,AA,BB,BS,IPT,
     4 QA,TA,ETA,FUNETA,ROW,AM,BMX,BRNAME,IJF,IJT,NSEC,XSKT,
     5 PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ITYPE,IBJNC,NDATA,IZQBVE,
     6 ISTATN,KTTDBC,ZQ,DTT,DATUM,ZQBVCO,ZQPMIN,LARBPR,ZHIGH,
     7 ZLOW,LINPRT,ARBERR,CLK,ZBOT,
     8 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     9 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,WINDSP,WINDDR,
     & IDX,ICT,W,U,UU,BU,BUU,ZSAV,QSAV,ZPSAV,QPSAV,NELAP,IOUT,
     & NTSAQ,HOLD,HNEW,IBOUND,NCOL,NROW,NLAY,ISTRM,ZPL,IBEGIN,
     & IBRPRN,ZN,ISS,QLSUM,ITRIAL,DCFM)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #    BRANCH-NETWORK FLOW MODEL USING A 4-POINT IMPLICIT TECHNIQUE    #
C #      BY R. W. SCHAFFRANEK, R. A. BALTZER, AND D. E. GOLDBERG       #
C #                        VERSION 91/04/01                            #
C #                BY R. W. SCHAFFRANEK AND R. S. REGAN                #
C #              MODIFIED BY E. D. SWAIN AND E. J. WEXLER              #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                    #
C #   THIS PROGRAM CALCULATES TRANSIENT FLOW IN A NETWORK OF           #
C #   INTERCONNECTED OPEN CHANNELS.  TIME DERIVATIVES ARE APPROXIMATED#
C #   AS CENTERED IN SPACE AND TIME; SPATIAL DERIVATIVES ARE TREATED   #
C #   AS CENTERED IN SPACE AND WEIGHTED IN TIME ACCORDING TO A USER    #
C #   DEFINED WEIGHTING FACTOR.  A LINEAR MATRIX SOLUTION IS EFFECTED  #
C #   WITH ITERATIVE IMPROVEMENT OF RESULTS OPTIONALLY SPECIFIABLE.    #
C #   THE 4*N BY 4*N MATRIX (N IS THE NUMBER OF BRANCHES) IS SOLVED    #
C #   BY GAUSS ELIMINATION USING MAXIMUM PIVOT STRATEGY. THIS VERSION  #
C #   IS MODIFIED TO INTERFACE WITH THE USGS GROUNDWATER FLOW MODEL    #
C #   MODFLOW, IN ORDER TO SIMULATE STREAM/AQUIFER RELATIONS.          #
C #                                                                    #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                          ARRAY SPECIFICATIONS                      #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                    #
C #   MAXIMUM BRANCHES:   MXBH          MAXIMUM JUNCTIONS:   MXJN      #
C #   X-SECTIONS PER BRANCH: NSEC       SEGMENTS PER BRANCH: NSEG      #
C #   MAX CROSS SECTIONS:   MAXS        MAX POINTS PER X-SECT:  MXPT   #
C #   MAX T.D. BDY LOCATIONS: MXTDBC    MAX MEASURED LOCATIONS: MXMD   #
C #   MAXIMUM Z(T) B.V.D.: MAXZBD       MAXIMUM Q(T) B.V.D.: MAXQBD    #
C #   MAX COMPUTED PER DAY: MAXCZQ      MAX MEASURED DATA: MAXMZQ      #
C #   MAX WIND DATA INPUT: MXWIND                                      #
C #                                                                    #
C #   DIMENSION MAXQBD ONE-HALF OF MAXZBD                              #
C #                                                                    #
C #   DIMENSION DATAI4(MAXCZQ*MAXS/4,4) TO COINCIDE WITH ALLOCATION    #
C #   OF ZQCOMP(MAXCZQ,100) TO WHICH IT IS EQUIVALENCED                #
C #                                                                    #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C BEGIN COMMON_COMCON ==============================================
C
      CHARACTER*2 IUNIT,OUNIT
      INTEGER*4 NBCH,NJNC,NBND,NSTEPS,IRDGEO,NIT,IPROPT,IPLOPT,IPLDEV,
     1 IPRMSG,IPLMSG,IEXOPT,INHR,INMN,IDTM,IWRTIC,IRDIC,NUMCOM,INWIND,
     2 TYPETA,OTTDDB,ISMOPT,NTDIOF,IRDNXT,IARDEM
      REAL THETA,QQTOL,ZZTOL,WSPEED,WDIREC,WSDRAG,H2ODEN,CHI,QZCONV,
     1 ZDATUM,DT,G,AIRDEN,GLETA,GLBETA,ETAMIN,ETAMAX,TOLERR
      COMMON /COMCON/ NBCH,NJNC,NBND,NSTEPS,IRDGEO,NIT,IPROPT,
     1 IPLOPT,IPLDEV,IPRMSG,IPLMSG,IEXOPT,INHR,INMN,IDTM,IWRTIC,
```

Program listing—Continued

```
     2 IRDIC,NUMCOM,INWIND,THETA,QQTOL,ZZTOL,WSPEED,WDIREC,WSDRAG,
     3 H2ODEN,CHI,ZDATUM,IUNIT,OUNIT,TYPETA,OTTDDB,ISMOPT,G,QZCONV,DT,
     4 AIRDEN,IARDEM,NTDIOF,IRDNXT,GLETA,GLBETA,ETAMIN,ETAMAX,TOLERR
C
C  END  COMMON_COMCON ==============================================
C BEGIN  COMMON_DTYPES ==============================================
C
      CHARACTER*2 DTYPE,ZTYPE,QTYPE,ATYPE,BTYPE,ZPTYPE,QPTYPE,DPTYPE
      COMMON /DTYPES/ DTYPE,ZTYPE,QTYPE,ATYPE,BTYPE,ZPTYPE,QPTYPE,
     1 DPTYPE
C
C  END   COMMON_DTYPES ==============================================
C BEGIN COMMON_UNITS ==============================================
C
      CHARACTER*2 IBLK,UNIT,EN,ME,MT,FT,TUNIT,DC
      COMMON /UNITS/ IBLK,UNIT,EN,ME,MT,FT,TUNIT,DC
C
C  END  COMMON_UNITS ==============================================
C BEGIN COMMON_LUNUMS ==============================================
C
      INTEGER*4 READER,PRINTR,PUNCH,DSREF,TDDATA,LUPTRK,LUIFLO,LUIVOL,
     1         LUGEOM,LUINIT,LUCVOL
      COMMON /LUNUMS/ READER,PRINTR,PUNCH,DSREF,TDDATA,LUPTRK,LUIFLO,
     1 LUIVOL,LUGEOM,LUINIT,LUCVOL
C
C  END  COMMON_LUNUMS ==============================================
C BEGIN COMMON_DADCOM ==============================================
C
      INTEGER*2 LISTB,LISTA,STRIP,RTCODE
      COMMON /DADCOM/ LISTB,LISTA,STRIP,RTCODE
C
C  END  COMMON_DADCOM ==============================================
C BEGIN COMMON_DAYPMO ==============================================
C
      INTEGER*2 DPERM(12)
      COMMON /DAYPMO/ DPERM
C
C  END  COMMON_DAYPMO ==============================================
C BEGIN COMMON_LOGICS ==============================================
C
      LOGICAL*4 PRTMSG, NOCONV, ERROR, OPLOTS, FOUND, NOEXTP,
     1 NOPRIT, DAYSUM, MOREBD, DTPRT, PTPLT, DAOPEN, STAGES, MODETA
      COMMON /LOGICS/ PRTMSG, NOCONV, ERROR, OPLOTS, FOUND, NOEXTP,
     1 NOPRIT, DAYSUM, MOREBD, DTPRT, PTPLT, DAOPEN, STAGES, MODETA
C
C  END  COMMON_LOGICS ==============================================
C BEGIN COMMON_BCTIME ==============================================
C
      INTEGER*4 IETIME,NETIME
      INTEGER *2 IRDPDY,IYR,IMO,IDA,IHR,IMN,NYR,NMO,NDA,NHR,NMN
      COMMON /BCTIME/ IETIME,NETIME,IRDPDY,IYR,IMO,IDA,IHR,IMN,
     1                                     NYR,NMO,NDA,NHR,NMN
C
C  END  COMMON_BCTIME ==============================================
C BEGIN COMMON_DATIME ==============================================
C
      INTEGER*4 KYR,KMO,KDA,KHR,KMN,M,KYRS,KMOS,KDAS,KHRS,KMNS
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,M,KYRS,KMOS,KDAS,KHRS,KMNS
C
C  END  COMMON_DATIME ==============================================
```

Program Listing—Continued

```
C BEGIN COMMON_NETWRK ==========================================================
C
      CHARACTER*80 NETNAM
      COMMON /NETWRK/ NETNAM
C
C  END  COMMON_NETWRK ============================================================
C BEGIN COMMON_MODBRCH ==========================================================
C
      COMMON /MODBRCH/ TWOCSQ,IDTPDY,TWOG,CW,II,ONECHI,DCHI,DTHETA,
     1                          IBCH,IJZPBC,IJQPBC,DCFM1,KKITER
C
C  END  COMMON_MODBRCH ============================================================
      INTEGER*4 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT
      INTEGER*4 ITQMAX(MAXS), ITQMIN(MAXS),ISTRM(3,MAXS),ITRIAL(MAXS,
     1 MAXCZQ)
      REAL QMAX(MAXS), QMIN(MAXS), QSUM(MAXS), ZQMIN(MAXS), ZQMAX(MAXS),
     1     AQMAX(MAXS), AQMIN(MAXS)
      REAL A(MAXS),Z(MAXS),Q(MAXS),ZP(MAXS),QP(MAXS),AP(MAXS),
     1 BP(MAXS),RP(MAXS),B(MAXS),R(MAXS),BT(MAXS),BTP(MAXS),
     2 ZSAV(MAXS),QSAV(MAXS),ZPSAV(MAXS),QPSAV(MAXS),ZPL(MAXS),
     3 ZN(MAXS)
      INTEGER*4 XSTATN(MAXS)
      REAL DX(MAXS), T(MAXS), RN(4,MAXS), WANGLE(MAXS), GDATUM(MAXS),
     1 ORIENT(MAXS), BETVEL(MAXS), SUMETA(MAXS), SUMCZQ(MAXS),
     2 SCZQSQ(MAXS),SZQETA(MAXS)
      CHARACTER*4 ITYPEO(4,MAXS)
      INTEGER*4 IPT(MAXS)
      REAL ZA(MXPT,MAXS), AA(MXPT,MAXS), BB(MXPT,MAXS), BS(MXPT,MAXS)
      REAL QA(MXPT,MAXS), TA(MXPT,MAXS), ETA(MXPT,MAXS),
     1 FUNETA(MXPT,MAXS)
      INTEGER*4 ROW(4*MXBH)
      REAL*4 AM((4*MXBH)**2),BMX(4*MXBH)
      CHARACTER*40 BRNAME(MXBH)
      INTEGER*4 IJF(MXBH),IJT(MXBH),NSEC(MXBH),XSKT(MXBH),
     1 PLTBCH(MXBH),PRTXSG(MXBH),PRTBCH(MXBH),PRTSUM(MXBH),PPLTBH(MXBH)
      CHARACTER*4 ITYPE(MXJN)
      INTEGER*4 IBJNC(MXJN),NDATA(MXJN),IZQBVE(MXJN)
      INTEGER*4 ISTATN(MXJN),KTTDBC
      REAL ZQ(MAXZBD,MXTDBC),DTT(MXJN),DATUM(MXJN),ZQBVCO(4,MXJN),
     1 ZQPMIN(MXJN)
      LOGICAL*4 LARBPR(MAXS),ZHIGH(MAXS),ZLOW(MAXS),LINPRT(MAXS),ARBERR
      INTEGER*4 IZDATA(MAXZBD)
      INTEGER*4 IQDATA(MAXQBD),INDATA(MAXQBD)
C     EQUIVALENCE (INDATA(1),IQDATA(1),IZDATA(1))
      REAL CLK(MAXS),ZBOT(MAXS)
      REAL WINDSP( MXWIND),WINDDR( MXWIND)
      INTEGER*4 IDX(MXJN,MXBH), ICT(MXJN)
      REAL W(MXJN)
      REAL*8 U(2*MAXS), UU(4*MAXS), BU(2*MXBH),BUU(4*MXBH)
      REAL*8 C1, C2, C3, C4, UUIJP1, UUIJP2, UUIJP3, UUIJP4
      REAL LAMBDA,MU,SETA,WDTT,TWOCSQ,TWOG,CW,ONECHI,DCHI,DTHETA,TH,WIND
      INTEGER*4 IAR,I,J,K,L,II,IJ,NS,KT,IS,N,NWREAD,NWDATA,INTDBC,IDTPDY
      REAL QTOL,ZTMIN,ZTMAX,ZPMIN,QPMIN,DXMIN,DXMAX
      COMMON /LIMITS/ QTOL,ZTMIN,ZTMAX,ZPMIN,QPMIN,DXMIN,DXMAX
      INTEGER*2 JYR,JMO,JDA,JHR,JMN,MYR,MMO,MDA,MHR,MMN
      INTEGER*4 ND,NDFIRT,NDPART,JETIME,NTSAQ
      COMMON /PARTIM/ ND,NDFIRT,NDPART,JETIME,JYR,JMO,JDA,JHR,JMN,
     1MYR,MMO,MDA,MHR,MMN
      CHARACTER*80 COMENT(9)
```

Program Listing—Continued

```
      COMMON /CMMNT/ COMENT
      CHARACTER*2 IDETA(7)
      COMMON /ETASYM/ IDETA
      REAL WRATIO,DTZERO,ZTEMP,QTEMP,ZIJ,QIJ,DXIJ,QIJP1,ZIJP1,APZPIJ,
     1     BPZPIJ,BTZPIJ,RPZPIJ,BAVG,BTAVG,AAVG,RAVG,QAVG,ZAVG,BETCOR,
     2     RNIJ,AAVGSQ,AAVGCU,SIGMA,EPSLON,ZETA,OMEGA,GAMMA,DELTA,DET,
     3 DZDT,DQDXC,DQDT,DQDXM,DADX,DZDX,FRIC,ZQPIJ,BIGQ,BIGZ,ZTOL,SOLPDT
     4 ,ALPHA
      INTEGER*4 IBCH,IJZPBC,IJQPBC,KTMATS,LASTN,IJP1,NSM1,JP1,IJ2,IJ4,
     1  IJ4P1,IJ4P2,IJ4P3,IJ4P4,IJ2P1,IJ2P2,I2,I4,I4P1,I4P2,I4P3,I4P4,
     2  I2P1,I2P2,NN,MM,NNN,NBPJ,M0,IBIGZ,JBIGZ,IBIGQ,JBIGQ,IJPNS,ICHK
      DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY),
     2          IBOUND(NCOL,NROW,NLAY),QLSUM(MAXS+MXJN)
C1----STATEMENT FUNCTION FOR LOCATING ELEMENTS IN COEFFICIENT MATRIX
      IAR(I,J,II)=I+II*(J-1)
      PRINTR=IOUT
      ICOUNT=0
C
C1A---IF THIS IS NOT THE FIRST TIMESTEP, SKIP INITIALIZATIONS.
C
      IF(IBEGIN.EQ.1) GO TO 1700
C
C2----ASSIGN UNINITIALIZED STAGE VALUE AT BOUNDARY-VALUE-DATA LOCATION
C     TO FIRST STAGE VALUE OF BOUNDARY-VALUE-DATA INPUT.
C
  640 DO 695 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 690 J=1,NS
      IJ=IJ+1
      IF (Z(IJ).NE.0.0) GO TO 690
      IF (J.NE.1.AND.J.NE.NS) GO TO 680
      FOUND=.FALSE.
      INTDBC=0
      DO 670 L=1,NBND
      IF (IBJNC(L).LE.0) GO TO 670
      IF (IZQBVE(L).EQ.1) GO TO 670
      IF (ITYPE(L).EQ.ZPTYPE.OR.ITYPE(L).EQ.QPTYPE) GO TO 670
      INTDBC=INTDBC+1
      IF (ITYPE(L).NE.ZTYPE) GO TO 670
      IF (J.EQ.NS) GO TO 650
      IF (IBJNC(L).EQ.IJF(I)) GO TO 660
      GO TO 670
  650 IF (IBJNC(L).NE.IJT(I)) GO TO 670
  660 FOUND=.TRUE.
      Z(IJ)=ZQ(1,INTDBC)+ZDATUM
      ZP(IJ)=Z(IJ)
      CALL ARB(ZP(IJ),I,J,M,AP(IJ),BP(IJ),BTP(IJ),RP(IJ),
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) THEN
         WRITE (*,2050) I,J
         RETURN
      ENDIF
  670 CONTINUE
```

Program Listing—Continued

```
      IF (FOUND) GO TO 690
  680 ERROR=.TRUE.
      WRITE (*,1920) I,J
  690 CONTINUE
  695 CONTINUE
      IF (ERROR) RETURN
      DCFM1=DCFM
C
C3----PRINT COMPUTATION-CONTROL PARAMETER TABLE AND CROSS-SECTION
C     GEOMETRY DATA
C
      DT=IDTM*60.
      CALL PRTBXS(NWREAD,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     3 A,Z,Q,ZP,QP,AP,BP,RP,B,R,BT,BTP,
     & XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT,BETVEL,SUMETA,SUMCZQ,
     & SCZQSQ,SZQETA,ITYPEO,ZA,AA,BB,BS,IPT,
     & QA,TA,ETA,FUNETA,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR,
     & ITYPE,IBJNC,NDATA,IZQBVE,ISTATN,
     & ZQ,DTT,DATUM,ZQBVCO,ZQPMIN,KTTDBC,IOUT)
C
C4----INITIALIZE COMPUTATION CONSTANTS
C
      IF (IUNIT.NE.EN) THEN
         WSPEED=WSPEED*1000./3600.
         TWOCSQ=2.0
      ELSE
         WSPEED=WSPEED*5280./3600.
         TWOCSQ=2.0*1.486*1.486
      ENDIF
      IF(IDTM.LE.1440) THEN
      IDTPDY=1440/IDTM
      ELSE
      IDTPDY=1
      ENDIF
      TWOG=2.0*G
      CW=WSDRAG*AIRDEN/(H2ODEN*G)*WSPEED*WSPEED
      II=4*NBCH
      ONECHI=1.0-CHI
      DCHI=ONECHI/CHI
      DTHETA=(1.0-THETA)/THETA
C
C5----APPLY STAGE COMPUTATION DATUM
C
C     IF (ZDATUM.EQ.0.0) GO TO 790
      DO 785 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 784 J=1,NS
      IJ=IJ+1
      Z(IJ)=Z(IJ)-ZDATUM
      ZP(IJ)=Z(IJ)
C
C5A---INITIALIZE VALUES OF STAGE AT THE END OF LAST TRIAL
C     AND THIS TIMESTEP, INITIALIZE ITRIAL
C
```

Program Listing—Continued

```
      ZPL(IJ)=ZP(IJ)
      ZN(IJ)=ZP(IJ)
      ND=IPT(IJ)
      DO 780 K=1,ND
  780 ZA(K,IJ)=ZA(K,IJ)-ZDATUM
      DO 1800 ICOT=1,NTSAQ+1
 1800 ITRIAL(IJ,ICOT)=0
  784 CONTINUE
  785 CONTINUE
C
C6----CALCULATE NUMBER OF BRANCHES AT EACH JUNCTION AND ASSIGN INDICES
C
  790 DO 815 J=1,NJNC
      ICT(J)=0
      DO 810 I=1,NBCH
      IF (IJF(I).NE.J) GO TO 800
      ICT(J)=ICT(J)+1
      IDX(J,ICT(J))=-I
  800 IF (IJT(I).NE.J) GO TO 810
      ICT(J)=ICT(J)+1
      IDX(J,ICT(J))=I
  810 CONTINUE
  815 CONTINUE
C
C7----IDENTIFY CROSS-SECTION LOCATIONS FOR SELF-SETTING BOUNDARY COND.
C
C     IN THIS APPROXIMATION, THE WATER DEPTH (OR DISCHARGE) AT THE
C     SPECIFIED EXTERNAL BOUNDARY-CONDITION JUNCTION FOR THE PRESENT
C     TIME STEP WILL BE EQUATED TO THE WATER DEPTH (OR DISCHARGE)
C     COMPUTED DURING THE PREVIOUS TIME STEP AT THE NEAREST NEIGHBORING
C     UPSTREAM (IF EXTERNAL JUNCTION ENDS THE BRANCH) OR DOWNSTREAM
C     (IF EXTERNAL JUNCTION BEGINS THE BRANCH) CROSS SECTION.  AS A
C     SPECIAL CASE, A MINIMUM WATER DEPTH (OR DISCHARGE) CAN BE
C     SPECIFIED BELOW WHICH THE WATER DEPTH (OR DISCHARGE) WILL NOT BE
C     PERMITTED TO DROP, THEREBY SIMULATING A SUBMERGED WIER (OR GATE)
C     CONDITION.
C
      DO 820 L=1,NBND
      IBCH=IDX(IBJNC(L),1)
      IF (ITYPE(L).NE.ZPTYPE.AND.ITYPE(L).NE.DPTYPE) GO TO 816
      IF (IBCH.LT.0) THEN
         IJZPBC=MAXS-XSKT(IABS(IBCH))+2
      ELSE
         IJZPBC=MAXS-XSKT(IBCH)+NSEC(IBCH)-1
      ENDIF
      GO TO 820
  816 IF (ITYPE(L).NE.QPTYPE) GO TO 820
      IF (IBCH.LT.0) THEN
         IJQPBC=MAXS-XSKT(IABS(IBCH))+2
      ELSE
         IJQPBC=MAXS-XSKT(IBCH)+NSEC(IBCH)-1
      ENDIF
  820 CONTINUE
      IBEGIN=1
C
C8----INITIIALIZE COMPUTATION LOOP PARAMETERS
C
      LETIME=IETIME
      IF (PTPLT) THEN
         KT=0
```

Program Listing—Continued

```
      ELSE
         KT=(INHR*60+INMN-1)/IDTM
      ENDIF
      CALL GEMXP(II,IS,ICHK,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ROW,AM,BMX,XSKT)
      DPERM(2)=28+(4-MOD(KYR,4))/4
      KTMATS=0
      ND=1
      N=0
C8A---THE TIME, STAGE, AND DISCHARGE VALUES AT THE BEGINNING
C     OF THE SET OF TIMESTEPS ARE SAVED FOR REITERATION.
C
      M=0
      KYRS=KYR
      KMOS=KMO
      KDAS=KDA
      KHRS=KHR
      KMNS=KMN
      DO 1985 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 1984 J=1,NS
      IJ=IJ+1
      ZSAV(IJ)=Z(IJ)
      ZPSAV(IJ)=ZP(IJ)
      QSAV(IJ)=Q(IJ)
      QPSAV(IJ)=QP(IJ)
 1984 CONTINUE
 1985 CONTINUE
C
C     BEGIN COMPUTATION LOOP
C
 1700 CONTINUE
      M=M+1
C
C8B---IF REITERATION IS NECESSARY, TIME, STAGE, AND DISCHARGE VALUES
C     ARE REINITIALIZED TO THE PREVIOUS VALUES
C
      IF(M.EQ.NELAP*NTSAQ+2) THEN
      IF (IBRPRN.EQ.0) THEN
      M=M-NTSAQ
      KYR=KYRS
      KMO=KMOS
      KDA=KDAS
      KHR=KHRS
      KMN=KMNS
      DO 1885 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 1884 J=1,NS
      IJ=IJ+1
      ZN(IJ)=ZP(IJ)
      Z(IJ)=ZSAV(IJ)
      ZP(IJ)=ZPSAV(IJ)
      Q(IJ)=QSAV(IJ)
      QP(IJ)=QPSAV(IJ)
 1884 CONTINUE
 1885 CONTINUE
```

Program Listing—Continued

```
      ENDIF
      KYRS=KYR
      KMOS=KMO
      KDAS=KDA
      KHRS=KHR
      KMNS=KMN
      DO 1785 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 1784 J=1,NS
      IJ=IJ+1
      ZSAV(IJ)=Z(IJ)
      ZPSAV(IJ)=ZP(IJ)
      QSAV(IJ)=Q(IJ)
      QPSAV(IJ)=QP(IJ)
 1784 CONTINUE
 1785 CONTINUE
C
C8B---WHEN THE SET OF BRANCH TIMESTEPS ARE FINISHED, RETURN TO MODFLOW
C
      M=M-2
      RETURN
      ENDIF
      ICOUNT=ICOUNT+1
      IF(IBRPRN.GT.0.AND.ISS.EQ.0) WRITE(*,1410) KYR,KMO,KDA,KHR,KMN
 1410 FORMAT(' SIMULATION TIME = ',I2,'/',I2,'/',I2,' ',I2,':',I2)
      LASTN=N
      KT=KT+1
C
C     PREPARATION FOR NEXT TIME STEP
C
C9----ASSIGN WIND SPEED AND DIRECTION FROM TIME-DEPENDENT WIND INPUT
      IF (INWIND.NE.0) THEN
         K=(M-1)*DT/WDTT+1.
         WRATIO=((M-1)*DT-(K-1)*WDTT)/WDTT
         WSPEED=WINDSP(K)+WRATIO*(WINDSP(K+1)-WINDSP(K))
         WDIREC=WINDDR(K)+WRATIO*(WINDDR(K+1)-WINDDR(K))
         IF (IUNIT.NE.EN) THEN
            WSPEED=WSPEED*1000./3600.
         ELSE
            WSPEED=WSPEED*5280./3600.
         ENDIF
         CW=WSDRAG*AIRDEN/(H2ODEN*G)*WSPEED*WSPEED
      ENDIF
      DO 935 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 930 J=1,NS
      IJ=IJ+1
      IF (IPROPT.NE.4) GO TO 880
  880 ZTEMP=Z(IJ)
      QTEMP=Q(IJ)
      Z(IJ)=ZP(IJ)
      Q(IJ)=QP(IJ)
      ZIJ=Z(IJ)+ZDATUM
      QIJ=Q(IJ)
      IF (INWIND.NE.0.AND.J.NE.NS) WANGLE(IJ)=COS(0.01745329*ABS(WDIREC-
     1ORIENT(IJ)))
      IF (OTTDDB.EQ.1) WRITE (LUIFLO) QIJ,ZIJ,AP(IJ),BP(IJ)
      QSUM(IJ)=QSUM(IJ)+QIJ
```

Program Listing—Continued

```
      IF (QIJ.GT.QMAX(IJ)) THEN
         QMAX(IJ)=QIJ
         ZQMAX(IJ)=ZIJ
         AQMAX(IJ)=AP(IJ)
         ITQMAX(IJ)=KHR*100+KMN
      ENDIF
      IF (QIJ.LT.QMIN(IJ)) THEN
         QMIN(IJ)=QIJ
         ZQMIN(IJ)=ZIJ
         AQMIN(IJ)=AP(IJ)
         ITQMIN(IJ)=KHR*100+KMN
      ENDIF
      IF (NOEXTP) GO TO 930
C     (IEXOPT=0) USE CURRENT VALUES AS INITIAL VALUES FOR UNKNOWNS
C     (IEXOPT=1) EXTRAPOLATE INITIAL VALUES FOR UNKNOWNS FROM CURRENT
      ZP(IJ)=2.*ZP(IJ)-ZTEMP
      QP(IJ)=2.*QP(IJ)-QTEMP
  930 CONTINUE
  935 CONTINUE
C
C     BEGIN ITERATIVE IMPROVEMENT LOOP
C
  936 DO 1360 N=1,NIT
C
C     CALCULATE BRANCH MATRICES
C
      DO 975 I=1,NBCH
      IJ=MAXS-XSKT(I)
      IJP1=IJ+1
      NSM1=NSEC(I)-1
      CALL ARB(Z(IJP1),I,1,M,A(IJP1),B(IJP1),BT(IJP1),R(IJP1),
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) GO TO 1530
      IF (M.EQ.NSTEPS) GO TO 940
      CALL ARB(ZP(IJP1),I,1,M,AP(IJP1),BP(IJP1),BTP(IJP1),RP(IJP1),
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) GO TO 1530
  940 DO 970 J=1,NSM1
      IJ=IJ+1
      DXIJ=DX(IJ)
      QIJ=Q(IJ)
      ZIJ=Z(IJ)
      JP1=J+1
      IJP1=IJ+1
      QIJP1=Q(IJP1)
      ZIJP1=Z(IJP1)
      CALL ARB(ZIJP1,I,JP1,M,A(IJP1),B(IJP1),BT(IJP1),R(IJP1),
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
```

Program Listing—Continued

```
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) GO TO 1530
      IF (M.EQ.NSTEPS) GO TO 970
      CALL ARB(ZP(IJ),I,JP1,M,APZPIJ,BPZPIJ,BTZPIJ,RPZPIJ,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) GO TO 1529
      CALL ARB(ZP(IJP1),I,JP1,M,AP(IJP1),BP(IJP1),BTP(IJP1),RP(IJP1),
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ZA,AA,BB,BS,IPT,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,
     & LARBPR,ZHIGH,ZLOW,LINPRT,ARBERR)
      IF (ARBERR) GO TO 1530
      BAVG=CHI*((BP(IJ)+BP(IJP1))*0.5)+ONECHI*((B(IJ)+B(IJP1))*0.5)
      BTAVG=CHI*((BTP(IJ)+BTP(IJP1))*0.5)+ONECHI*((BT(IJ)+BT(IJP1))*0.5)
      AAVG=CHI*((AP(IJ)+AP(IJP1))*0.5)+ONECHI*((A(IJ)+A(IJP1))*0.5)
      RAVG=CHI*((RP(IJ)+RP(IJP1))*0.5)+ONECHI*((R(IJ)+R(IJP1))*0.5)
      QAVG=CHI*((QP(IJ)+QP(IJP1))*0.5)+ONECHI*((QIJ+QIJP1)*0.5)
      ZAVG=CHI*((ZP(IJ)+ZP(IJP1))*0.5)+ONEGHI*((ZIJ+ZIJP1)*0.5)+ZDATUM
      BETCOR=(BETVEL(IJ)+BETVEL(IJP1))*0.5
C
C9A---THE NEW AND OLD AQUIFER HEADS ARE USED TO INTERPOLATE THE
C     HEAD AT THE PRESENT BRANCH TIMESTEP (HAQ).
C
      HNEW1=HNEW(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HOLD1=HOLD(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HAQ=HOLD1+FLOAT(ICOUNT-1)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
      HAQP=HOLD1+FLOAT(ICOUNT)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
      IF(IBOUND(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ)).LE.0) CLK(IJ)=0.
      HAQP1=HAQ
      HAQPP1=HAQP
      CLKIJ=CLK(IJ)
      CLKP1=CLK(IJ)
      CLKP=CLK(IJ)
      CLKPP1=CLK(IJ)
C
C     IF THE AQUIFER IS BELOW THE STREAMBED, HEAD IN STREAM IS USED
C     TO CALCULATE LEAKAGE.
C
      IF(HAQP1.LT.ZBOT(IJP1)) HAQP1=ZBOT(IJP1)
      IF(HAQ.LT.ZBOT(IJ)) HAQ=ZBOT(IJ)
      IF(HAQPP1.LT.ZBOT(IJP1)) HAQPP1=ZBOT(IJP1)
      IF(HAQP.LT.ZBOT(IJ)) HAQP=ZBOT(IJ)
      HAQ=HAQ-ZDATUM
      HAQP1=HAQP1-ZDATUM
      HAQP=HAQP-ZDATUM
      HAQPP1=HAQPP1-ZDATUM
      IF (ETA(1,IJ).NE.0.0) THEN
         CALL SETAB(2,IJ,ZAVG,QAVG,AAVG,RAVG,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT,BETVEL,SUMETA,SUMCZQ,
```

Program Listing—Continued

```
     & SCZQSQ,SZQETA,ITYPEO,ZA,AA,BB,BS,IPT,
     & QA,TA,ETA,FUNETA)
      ELSE
         IF (TYPETA.NE.1) RN(4,IJ)=SETA(IJ,ZAVG,QAVG,AAVG,RAVG)
      ENDIF
      RNIJ=RN(4,IJ)
      AAVGSQ=AAVG*AAVG
      AAVGCU=AAVGSQ*AAVG
C
C10---THE MATRIX COEFFICIENTS LAMBDA, SIGMA, MU, ALPHA, EPSLON,
C     OMEGA, AND DELTA ARE COMPUTED.
C
      LAMBDA=DXIJ/(TWOG*AAVG*DT*THETA)
      IF (ISS.NE.0) LAMBDA=0.
      SIGMA=ABS(QAVG)*RNIJ*RNIJ*DXIJ*CHI/(TWOCSQ*AAVGSQ*RAVG**1.3333333*
     1THETA)
C
C10A--THE DRYNESS CONDITION IS CHECKED FOR FRICTION AND LEAKAGE
      IF(ITRIAL(IJ,ICOUNT+1).GE.10) SIGMA=SIGMA*DCFM1/64.
      DO 2121 L=3,8
      IF(ITRIAL(IJ,ICOUNT+1).GE.L*10) SIGMA=SIGMA*2.
2121  CONTINUE
      IF(ITRIAL(IJ,ICOUNT+1).EQ.11) CLKP=0.0
      IF(ITRIAL(IJ,ICOUNT+1).EQ.22) CLKPP1=0.0
      IF(ITRIAL(IJ,ICOUNT+1).GE.33) THEN
      CLKP=0.0
      CLKPP1=0.0
      ENDIF
      IF(ITRIAL(IJ,ICOUNT).EQ.11) CLKIJ=0.0
      IF(ITRIAL(IJ,ICOUNT).EQ.22) CLKP1=0.0
      IF(ITRIAL(IJ,ICOUNT).GE.33) THEN
      CLKIJ=0.0
      CLKP1=0.0
      ENDIF
      MU=2.0*BETCOR*QAVG/(G*AAVGSQ)
      ALPHA=DXIJ*BTAVG/(2.*DT*THETA)+DXIJ*CHI*CLKP*BP(IJ)
     1 /(2.*THETA)
       IF(ISS.NE.0) ALPHA=DXIJ*CHI*CLKP*BP(IJ)/(2.*THETA)
      EPSLON=(LAMBDA-DCHI*SIGMA)*(QIJ+QIJP1)-MU*DTHETA*(QIJP1-QIJ)-
     1 DTHETA*(ZIJP1-ZIJ)+BETCOR*QAVG*QAVG/(G*THETA*AAVGCU)*(BAVG*
     2 (ZP(IJP1)-ZP(IJ))+(APZPIJ-AP(IJ)))+CW*BAVG*WANGLE(IJ)*DXIJ/
     3 (THETA*AAVG)
      ZETA=LAMBDA+SIGMA+MU
      OMEGA=LAMBDA+SIGMA-MU
      GAMMA=DXIJ*BTAVG/(2.*DT*THETA)+CHI*DXIJ*CLKPP1*BP(IJP1)
     1 /(2.*THETA)
      IF(ISS.NE.0) GAMMA=CHI*DXIJ*CLKPP1*BP(IJP1)/(2.*THETA)
      DELTA=-DTHETA*(Q(IJP1)-Q(IJ))+(BAVG*DXIJ/(2.*DT*THETA)
     &-ONECHI*DXIJ*CLKP1*B(IJP1)/(2.*THETA))*Z(IJP1)+(BAVG*DXIJ
     &/(2.*DT*THETA)-ONECHI*DXIJ*CLKIJ*B(IJ)/(2.*THETA))*Z(IJ)
     &+DXIJ*(CHI*(CLKP*HAQP*BP(IJ)+CLKPP1*HAQPP1*BP(IJP1))+
     &ONECHI*(CLKIJ*HAQ*B(IJ)+CLKP1*HAQP1*B(IJP1)))/(2*THETA)
      IF(ISS.NE.0) DELTA=-DTHETA*(Q(IJP1)-Q(IJ))-
     &(ONECHI*DXIJ*CLKP1*B(IJP1)/(2.*THETA))*Z(IJP1)-
     &(ONECHI*DXIJ*CLKIJ*B(IJ)/(2.*THETA))*Z(IJ)
     &+DXIJ*(CHI*(CLKP*HAQP*BP(IJ)+CLKPP1*HAQPP1*BP(IJP1))+
     &ONECHI*(CLKIJ*HAQ*B(IJ)+CLKP1*HAQP1*B(IJP1)))/(2*THETA)
C
      DET=1./(1.-ZETA*GAMMA)
      IF (IPROPT.LT.5.OR.IPROPT.GT.8) GO TO 950
```

Program Listing—Continued

```
      DZDT=BTAVG/(2.*DT)*(ZP(IJP1)+ZP(IJ)-ZIJP1-ZIJ)
      DQDXC=THETA/DXIJ*(QP(IJP1)-QP(IJ)+DTHETA*(QIJP1-QIJ))
      DQDT=LAMBDA*(THETA/DXIJ)*(QP(IJP1)+QP(IJ)-QIJP1-QIJ)
      DQDXM=MU*(THETA*(QP(IJP1)-QP(IJ))+(1.0-THETA)*(QIJP1-QIJ))/DXIJ
      DADX=BETCOR*QAVG*QAVG/(G*DXIJ*AAVGCU)*(BAVG*(ZP(IJP1)-ZP(IJ))+
     1 (APZPIJ-AP(IJ)))
      DZDX=(THETA*(ZP(IJP1)-ZP(IJ))+(1.0-THETA)*(ZIJP1-ZIJ))/DXIJ
      FRIC=SIGMA*(THETA/DXIJ)*(QP(IJP1)+QP(IJ)+DCHI*(QIJP1+QIJ))
      WIND=CW*BAVG*WANGLE(IJ)/AAVG
      WRITE (PRINTR,2150) DZDT,DQDXC,DQDT,DQDXM,DADX,DZDX,FRIC,WIND
      IF (IPROPT.LT.7) GO TO 950
      WRITE (PRINTR,2160) N,I,J,IJ,JP1,IJP1
      WRITE (PRINTR,2170) ZIJ,QIJ,ZIJP1,QIJP1,ZP(IJ),QP(IJ),ZP(IJP1),
     1 QP(IJP1)
      WRITE (PRINTR,2170) APZPIJ,BPZPIJ,A(IJ),B(IJ),R(IJ),A(IJP1),
     1 B(IJP1),R(IJP1)
      WRITE (PRINTR,2170) AP(IJ),BP(IJ),RP(IJ),AP(IJP1),BP(IJP1),
     1 RP(IJP1)
      WRITE (PRINTR,2170) DXIJ,BAVG,AAVG,RAVG,QAVG,BETCOR,RNIJ,
     1 WANGLE(IJ)
      WRITE (PRINTR,2170) LAMBDA,SIGMA,MU,EPSLON,ZETA,OMEGA,GAMMA,DELTA
  950 CONTINUE
C
C     SEGMENT MATRIX COMPUTATION
C
      IJ2=(IJ-1)*2
      IJ4=IJ2*2
      IJ4P1=IJ4+1
      IJ4P2=IJ4+2
      IJ4P3=IJ4+3
      IJ4P4=IJ4+4
      UU(IJ4P1)=(1.+ZETA*ALPHA)*DET
      UU(IJ4P2)=(-OMEGA-ZETA)*DET
      UU(IJ4P3)=(-ALPHA-GAMMA)*DET
      UU(IJ4P4)=(1.+OMEGA*GAMMA)*DET
      IJ2P1=IJ2+1
      IJ2P2=IJ2+2
      U(IJ2P1)=(EPSLON-ZETA*DELTA)*DET
      U(IJ2P2)=(DELTA-EPSLON*GAMMA)*DET
      I2=(I-1)*2
      I4=I2*2
      I4P1=I4+1
      I4P2=I4+2
      I4P3=I4+3
      I4P4=I4+4
      I2P1=I2+1
      I2P2=I2+2
C
C     BRANCH MATRIX COMPUTATION
C
      IF (J.GT.1) THEN
         C1=BUU(I4P1)
         C2=BUU(I4P2)
         C3=BUU(I4P3)
         C4=BUU(I4P4)
         UUIJP1=UU(IJ4P1)
         UUIJP2=UU(IJ4P2)
         UUIJP3=UU(IJ4P3)
         UUIJP4=UU(IJ4P4)
         BUU(I4P1)=UUIJP1*C1+UUIJP2*C3
```

Program Listing—Continued

```
            BUU(I4P2)=UUIJP1*C2+UUIJP2*C4
            BUU(I4P3)=UUIJP3*C1+UUIJP4*C3
            BUU(I4P4)=UUIJP3*C2+UUIJP4*C4
            C1=BU(I2P1)
            C2=BU(I2P2)
            BU(I2P1)=UUIJP1*C1+UUIJP2*C2+U(IJ2P1)
            BU(I2P2)=UUIJP3*C1+UUIJP4*C2+U(IJ2P2)
         ELSE
            BUU(I4P1)=UU(IJ4P1)
            BUU(I4P2)=UU(IJ4P2)
            BUU(I4P3)=UU(IJ4P3)
            BUU(I4P4)=UU(IJ4P4)
            BU(I2P1)=U(IJ2P1)
            BU(I2P2)=U(IJ2P2)
         ENDIF
  970 CONTINUE
  975 CONTINUE
C
C11---IS THIS THE FIRST ITERATION (N=1) OF THIS TIME STEP (M) ?
C
      IF(N.NE.1) GO TO 1030
      IF(IPROPT.NE.3.OR.M.EQ.NELAP*NTSAQ+1) THEN
      IF (IBRPRN.GT.0) CALL DTOUT(LASTN,Q,Z,A,B,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     2 ITQMAX,ITQMIN,QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,
     & XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT,BETVEL,SUMETA,SUMCZQ,
     & SCZQSQ,SZQETA,ITYPEO,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ZBOT,ITRIAL,ICOUNT)
      ENDIF
C
C11A--IF END OF GROUNDWATER TIMESTEP, GO TO COMPUTATION LOOP START
      IF(M.EQ.NELAP*NTSAQ+1) GO TO 1700
      IF (PTPLT) THEN
         IF (KT.LT.MAXCZQ.AND.KHR.NE.24) GO TO 990
         KETIME=IETIME+(M-1)*IDTM
         KT=0
      ELSE
         IF (KT.LT.IDTPDY) GO TO 990
         KT=0
         IF (OPLOTS) THEN
            KETIME=IETIME+(M-1)*IDTM
         ENDIF
      ENDIF
  990 IF (M.EQ.NSTEPS) GO TO 1610
      KMN=KMN+IDTM
      IF (KMN.LT.60) GO TO 1000
      KHR=KHR+KMN/60
      KMN=MOD(KMN,60)
 1000 IF (KHR.LT.24.OR.(KHR.EQ.24.AND.KMN.EQ.0)) GO TO 1010
      KHR=KHR-24
      KDA=KDA+1
 1010 IF (KDA.GT.DPERM(KMO)) THEN
         KDA=1
         KMO=KMO+1
      ENDIF
      IF (KMO.LT.13) GO TO 1030
      KMO=1
      KYR=KYR+1
```

Program Listing—Continued

```
      IF (KYR.GT.99) KYR=0
      DPERM(2)=28+(4-MOD(KYR,4))/4
 1030 IF (NOPRIT) GO TO 1040
      LASTN=N-1
C     IF(IBRPRN.GT.0)
      CALL DTOUT(LASTN,QP,ZP,AP,BP,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     2 ITQMAX,ITQMIN,QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,
     & XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT,BETVEL,SUMETA,SUMCZQ,
     & SCZQSQ,SZQETA,ITYPEO,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ZBOT,ITRIAL,ICOUNT)
 1040 CONTINUE
C
C12---SET UP NETWORK MATRIX AND VECTOR
C
      NN=1
      MM=1
      DO 1050 I=1,NBCH
C     INSERT BRANCH MATRICES
      NNN=IAR(NN,MM,II)
      I2=(I-1)*2
      I4=I2*2
      AM(NNN)=BUU(I4+1)
      NNN=NNN+II
      AM(NNN)=BUU(I4+2)
      NNN=NNN+II
      AM(NNN)=-1.0
      NNN=IAR(NN+1,MM,II)
      AM(NNN)=BUU(I4+3)
      NNN=NNN+II
      AM(NNN)=BUU(I4+4)
      NNN=NNN+II+II
      AM(NNN)=-1.0
C     CONSTRUCT RIGHT SIDE VECTOR
      BMX(NN)=-BU(I2+1)
      BMX(NN+1)=-BU(I2+2)
      NN=NN+2
      MM=MM+4
 1050 CONTINUE
C
C     INSERT BOUNDARY CONDITIONS FOR INTERNAL JUNCTIONS
C
      DO 1080 J=1,NJNC
      IF (ICT(J).EQ.1) GO TO 1080
      NBPJ=ICT(J)
C     INSERT DISCHARGE CONTINUITY
      DO 1060 I=1,NBPJ
      IBCH=IDX(J,I)
      MM=4+(IABS(IBCH)-1)*4
      IF (IBCH.LT.0) MM=MM-2
      NNN=IAR(NN,MM,II)
      AM(NNN)=IBCH/IABS(IBCH)
 1060 CONTINUE
      BMX(NN)=-W(J)
      NN=NN+1
C     INSERT STAGE COMPATIBILITY
      IBCH=IDX(J,1)
      M0=3+(IABS(IBCH)-1)*4
```

Program Listing—Continued

```
      IF (IBCH.LT.0) M0=M0-2
      DO 1070 I=2,NBPJ
      IBCH=IDX(J,I)
      MM=3+(IABS(IBCH)-1)*4
      NNN=IAR(NN,M0,II)
      AM(NNN)=1.0
      IF (IBCH.LT.0) MM=MM-2
      NNN=IAR(NN,MM,II)
      AM(NNN)=-1.0
      BMX(NN)=0.0
      NN=NN+1
 1070 CONTINUE
 1080 CONTINUE
C
C     RETRIEVE ADDITIONAL BOUNDARY-VALUE DATA FROM DIRECT-ACCESS STORAGE
C
      IF (N.NE.1.OR.MODETA) GO TO 1230
      IF (MOREBD) CALL GETBVD
     1 (MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     & ITYPE,IBJNC,NDATA,IZQBVE,ISTATN,
     & ZQ,DTT,DATUM,ZQBVCO,ZQPMIN,KTTDBC,
     & INDATA,IZDATA,IQDATA)
C
C     INSERT BOUNDARY CONDITIONS FOR EXTERNAL JUNCTIONS
C
 1230 INTDBC=0
      DO 1310 L=1,NBND
      IF (IBJNC(L).LE.0) GO TO 1310
      IBCH=IDX(IBJNC(L),1)
      MM=1+(IABS(IBCH)-1)*4
      IF (ITYPE(L).EQ.QTYPE.OR.ITYPE(L).EQ.QPTYPE) MM=MM+1
      IF (IBCH.GT.0) MM=MM+2
      NNN=IAR(NN,MM,II)
      AM(NNN)=1.0
C     DETERMINE BOUNDARY VALUE FROM STAGE (OR DISCHARGE) BOUNDARY-
C     CONDITION EQUATION
      IF (IZQBVE(L).EQ.1) THEN
      IF (IBCH.LT.0) THEN
         IJ=MAXS-XSKT(IABS(IBCH))+1
      ELSE
         IJ=MAXS-XSKT(IBCH)+NSEC(IBCH)
      ENDIF
      IF (ITYPE(L).EQ.QTYPE) THEN
         ZQPIJ=QP(IJ)
      ELSE
         ZQPIJ=ZP(IJ)
      ENDIF
      BMX(NN)=ZQBVCO(1,L)+(ZQBVCO(2,L)+(ZQBVCO(3,L)+ZQBVCO(4,L)*ZQPIJ)*Z
     1QPIJ)*ZQPIJ
      ELSE
C     PARABOLIC INTERPOLATION FOR BOUNDARY CONDITION FROM BOUNDARY-VALUE
C     DATA
      IF (ITYPE(L).NE.ZPTYPE.AND.ITYPE(L).NE.QPTYPE) THEN
         INTDBC=INTDBC+1
         IF (NDATA(L).EQ.0) THEN
            K=ND*DT/DTT(L)+1.
            TH=(ND*DT-(K-1)*DTT(L))/DTT(L)
         ELSE
            K=M*DT/DTT(L)+1.
```

Program Listing—Continued

```
               TH=(M*DT-(K-1)*DTT(L))/DTT(L)
            ENDIF
            IF (K.NE.1) GO TO 1280
            K=2
            TH=TH-1.
 1280       BMX(NN)=.5*TH*(ZQ(K+1,INTDBC)-ZQ(K-1,INTDBC)+TH*(ZQ(K+1,INTDBC)
     1       +ZQ(K-1,INTDBC)-2.0*ZQ(K,INTDBC)))+ZQ(K,INTDBC)
          ELSE
          IF (ITYPE(L).EQ.ZPTYPE) THEN
C         SELF-SETTING STAGE BOUNDARY CONDITION
            BMX(NN)=Z(IJZPBC)
           IF (ISS.NE.0)
     1     BMX(NN)=ZP(IJZPBC)
            IF (Z(IJZPBC).LT.ZQPMIN(L)) BMX(NN)=ZQPMIN(L)
          ENDIF
          IF (ITYPE(L).EQ.DPTYPE) THEN
C         SELF-SETTING DEPTH OF FLOW BOUNDARY CONDITION
            BMX(NN)=Z(IJZPBC)-ZBOT(IJZPBC)+ZBOT(IJZPBC+1)
           IF (ISS.NE.0)
     1     BMX(NN)=ZP(IJZPBC)-ZBOT(IJZPBC)+ZBOT(IJZPBC+1)
            IF (Z(IJZPBC).LT.ZQPMIN(L)) BMX(NN)=ZQPMIN(L)
          ENDIF
          IF (ITYPE(L).EQ.QPTYPE) THEN
C         SELF-SETTING DISCHARGE BOUNDARY CONDITION
            BMX(NN)=Q(IJQPBC)
            IF (Q(IJQPBC).LT.ZQPMIN(L)) BMX(NN)=ZQPMIN(L)
          ENDIF
          ENDIF
          ENDIF
          NN=NN+1
 1310   CONTINUE
C
C13---SOLVE MATRIX OF LINEAR EQUATIONS
C
          IF (II.NE.NN-1) GO TO 1430
          CALL GEMXP(II,IS,ICHK,
     1   MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1   MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     &   ROW,AM,BMX,XSKT)
          KTMATS=KTMATS+1
          IF (IS.EQ.1) THEN
            WRITE (PRINTR,1990)
            GO TO 1610
          ENDIF
C
C14---CALCULATE INTERMEDIATE VALUES
C
          NN=1
          BIGQ=0.0
          BIGZ=0.0
          BIGQP=0.0
          BIGZP=0.0
          DO 1355 I=1,NBCH
          IJ=MAXS-XSKT(I)
          IJP1=IJ+1
          ZTEMP=ZP(IJP1)
          QTEMP=QP(IJP1)
          ZP(IJP1)=BMX(NN)
          QP(IJP1)=BMX(NN+1)
          ZTOL=ABS(ZTEMP-ZP(IJP1))
```

Program Listing—Continued

```
      QTOL=ABS(QTEMP-QP(IJP1))
      ZPTOL=ABS(ZP(IJP1)-Z(IJP1))
      QPTOL=ABS(QP(IJP1)-Q(IJP1))
      IF (ZTOL.GT.BIGZ) THEN
         BIGZ=ZTOL
         IBIGZ=I
         JBIGZ=1
      ENDIF
      IF (QTOL.GT.BIGQ) THEN
         BIGQ=QTOL
         IBIGQ=I
         JBIGQ=1
      ENDIF
      IF(ZPTOL.GT.BIGZP) BIGZP=ZPTOL
      IF(QPTOL.GT.BIGQP) BIGQP=QPTOL
      NN=NN+4
      NSM1=NSEC(I)-1
      DO 1350 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      IJ2=(IJ-1)*2
      IJ4=(IJ-1)*4
      ZTEMP=ZP(IJP1)
      QTEMP=QP(IJP1)
      ZP(IJP1)=UU(IJ4+1)*ZP(IJ)+UU(IJ4+2)*QP(IJ)+U(IJ2+1)
      QP(IJP1)=UU(IJ4+3)*ZP(IJ)+UU(IJ4+4)*QP(IJ)+U(IJ2+2)
      ZTOL=ABS(ZTEMP-ZP(IJP1))
      QTOL=ABS(QTEMP-QP(IJP1))
      ZPTOL=ABS(ZP(IJP1)-Z(IJP1))
      QPTOL=ABS(QP(IJP1)-Q(IJP1))
      IF (ZTOL.GT.BIGZ) THEN
         BIGZ=ZTOL
         IBIGZ=I
         JBIGZ=J+1
      ENDIF
      IF (QTOL.GT.BIGQ) THEN
         BIGQ=QTOL
         IBIGQ=I
         JBIGQ=J+1
      ENDIF
      IF(ZPTOL.GT.BIGZP) BIGZP=ZPTOL
      IF(QPTOL.GT.BIGQP) BIGQP=QPTOL
 1350 CONTINUE
 1355 CONTINUE
      IF (BIGZ.LE.ZZTOL.AND.BIGQ.LE.QQTOL) GO TO 1390
 1360 CONTINUE
C
C     END ITERATIVE IMPROVEMENT LOOP
C
      IF (NOCONV) GO TO 1380
      IF (IUNIT.EQ.EN) THEN
         BIGZ=BIGZ*0.3048
         BIGQ=BIGQ*0.02832
      ELSE
         BIGZ=BIGZ*3.281
         BIGQ=BIGQ*35.31
      ENDIF
 1380 WRITE (PRINTR,2110) KHR,KMN,KYR,KMO,KDA,IBIGZ,JBIGZ,BIGZ,IBIGQ,
     1 JBIGQ,BIGQ
 1390 IF (TOLERR.EQ.0.0) GO TO 1395
```

Program Listing—Continued

```
C     IF (MODETA) GO TO 936
 1395 ND=ND+1
C
C14A--THE NEW AND OLD AQUIFER HEADS ARE USED TO INTERPOLATE THE
C     HEAD AT THE PRESENT BRANCH TIMESTEP (HAQ) FOR CALCULATING
C     THE TOTAL VOLUME OF LEAKAGE.
C
      DO 2185 I=1,NBCH
      NSM1=NSEC(I)-1
      IJ=MAXS-XSKT(I)
      IJFI=MAXS+IJF(I)
      IJP1=IJ+1
      QLSUM(IJFI)=QLSUM(IJFI)-QP(IJP1)/NTSAQ
      DO 2184 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      IF (ISTRM(3,IJ).LT.0) GO TO 2184
      HNEW1=HNEW(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HOLD1=HOLD(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HAQ=HOLD1+FLOAT(ICOUNT-1)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
      HAQP=HOLD1+FLOAT(ICOUNT)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
      IF(IBOUND(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ)).LE.0) CLK(IJ)=0.
      HAQP1=HAQ
      HAQPP1=HAQP
      CLKIJ=CLK(IJ)
      CLKP1=CLK(IJ)
      CLKP=CLK(IJ)
      CLKPP1=CLK(IJ)
C
C     IF THE AQUIFER IS BELOW THE STREAMBED, HEAD IN STREAM IS USED
C     TO CALCULATE LEAKAGE.
C
      IF(HAQP1.LT.ZBOT(IJP1)) HAQP1=ZBOT(IJP1)
      IF(HAQ.LT.ZBOT(IJ)) HAQ=ZBOT(IJ)
      IF(HAQPP1.LT.ZBOT(IJP1)) HAQPP1=ZBOT(IJP1)
      IF(HAQP.LT.ZBOT(IJ)) HAQP=ZBOT(IJ)
      HAQ=HAQ-ZDATUM
      HAQP1=HAQP1-ZDATUM
      HAQP=HAQP-ZDATUM
      HAQPP1=HAQPP1-ZDATUM
      IF(ITRIAL(IJ,ICOUNT+1).EQ.11) CLKP=0.0
      IF(ITRIAL(IJ,ICOUNT+1).EQ.22) CLKPP1=0.0
      IF(ITRIAL(IJ,ICOUNT+1).GE.33) THEN
      CLKP=0.0
      CLKPP1=0.0
      ENDIF
      IF(ITRIAL(IJ,ICOUNT).EQ.11) CLKIJ=0.0
      IF(ITRIAL(IJ,ICOUNT).EQ.22) CLKP1=0.0
      IF(ITRIAL(IJ,ICOUNT).GE.33) THEN
      CLKIJ=0.0
      CLKP1=0.0
      ENDIF
      QLSUM(IJ)=QLSUM(IJ)+(CHI*(CLKPP1*BP(IJP1)*(ZP(IJP1)-HAQPP1)+
     1 CLKP*BP(IJ)*(ZP(IJ)-HAQP))+ONECHI*(CLKP1*B(IJP1)*(Z(IJP1)-
     2 HAQP1)+CLKIJ*B(IJ)*(Z(IJ)-HAQ)))*DX(IJ)/(2.*NTSAQ)
      IF(KKITER.LE.50) THEN
      HNEW1=HNEW(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HOLD1=HOLD(ISTRM(1,IJ),ISTRM(2,IJ),ISTRM(3,IJ))
      HAQ=HOLD1+FLOAT(ICOUNT-1)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
      HAQP=HOLD1+FLOAT(ICOUNT)*(HNEW1-HOLD1)/FLOAT(NTSAQ)
```

Program Listing—Continued

```
      HAQP1=HAQ
      HAQPP1=HAQP
C
C     THE DRYNESS CONDITIONS ARE CHECKED TO SET VALUES OF ITRIAL
      ITRIA1=0
      IF(ZP(IJ).LT.ZBOT(IJ)-ZDATUM) ITRIA1=ITRIA1+10
      IF(ZP(IJP1).LT.ZBOT(IJP1)-ZDATUM) ITRIA1=ITRIA1+20
       IF (HAQP.LT.ZBOT(IJ)) ITRIA1=ITRIA1+1
       IF (HAQPP1.LT.ZBOT(IJP1)) ITRIA1=ITRIA1+2
       ITRIA2=ITRIAL(IJ,ICOUNT+1)
       ITRIA3=ITRIAL(IJ,ICOUNT)
       MOD1=MOD(ITRIA1,10)
       MOD2=MOD(ITRIA2,10)
       MOD3=MOD(ITRIA3,10)
       IF(ITRIA3.GE.30) THEN
       IF(ITRIA3.LT.80) THEN
       IF(ITRIA1.GE.30) THEN
       ITRIA1=MOD1+ITRIA3-MOD3+10
       ELSE
       ITRIA1=MOD1+ITRIA3-MOD3-10
       ENDIF
       ELSE
       IF(ITRIA1.GE.30)
     1 ITRIA1=MOD1+ITRIA3-MOD3
       ENDIF
       ENDIF
       IF(KKITER.LE.5.OR.(ITRIA1-MOD1)/10.EQ.MOD2.OR.ITRIA1-MOD1.LT.
     1 ITRIA3-MOD3) ITRIAL(IJ,ICOUNT+1)=ITRIA1
       ITRIAL(IJ,ICOUNT+1)=ITRIAL(IJ,ICOUNT+1)-MOD(ITRIAL(IJ,ICOUNT+1),
     1 10)+MOD1
       ENDIF
 2184 CONTINUE
      IJTI=MAXS+IJT(I)
      QLSUM(IJTI)=QLSUM(IJTI)+QP(IJP1)/NTSAQ
 2185 CONTINUE
      IF(M.LT.NSTEPS) GO TO 1700
 1400 CONTINUE
C
C     END COMPUTATION LOOP
C
      GO TO 1610
C
C15---TERMINATE EXECUTION
C
 1430 WRITE (PRINTR,1960)
      GO TO 1610
 1450 WRITE (PRINTR,1980)
      RETURN
 1529 WRITE (PRINTR,2075)
 1530 DO 1600 K=1,2
      WRITE (PRINTR,2080) ZTYPE
      DO 1590 I=1,NBCH
      IJ=MAXS-XSKT(I)
      IJP1=IJ+1
      IJPNS=IJ+NSEC(I)
      IF (K.EQ.1) THEN
      DO 1540 IJ=IJP1,IJPNS
 1540 Z(IJ)=Z(IJ)+ZDATUM
      WRITE (PRINTR,2090) ZTYPE,I,(Z(IJ),IJ=IJP1,IJPNS)
      DO 1550 IJ=IJP1,IJPNS
```

Program Listing—Continued

```
 1550 Z(IJ)=Z(IJ)-ZDATUM
      ELSE
      DO 1570 IJ=IJP1,IJPNS
 1570 ZP(IJ)=ZP(IJ)+ZDATUM
      WRITE (PRINTR,2090) ZTYPE,I,(ZP(IJ),IJ=IJP1,IJPNS)
      DO 1580 IJ=IJP1,IJPNS
 1580 ZP(IJ)=ZP(IJ)-ZDATUM
      ENDIF
 1590 CONTINUE
      ZTYPE=ZPTYPE
 1600 CONTINUE
 1610 CONTINUE
      IF (DTPRT.AND.N.NE.1) CALL DTOUT(LASTN,QP,ZP,AP,BP,
     1 MXBH,MXJN,MAXS,MXPT,MXTDBC,MXMD,MAXZBD,
     1 MAXQBD,MAXCZQ,MAXMZQ,MXWIND,MAXBD,MXOTDT,
     2 ITQMAX,ITQMIN,QMAX,QMIN,QSUM,ZQMIN,ZQMAX,AQMAX,AQMIN,
     & XSTATN,DX,T,RN,WANGLE,GDATUM,ORIENT,BETVEL,SUMETA,SUMCZQ,
     & SCZQSQ,SZQETA,ITYPEO,
     & BRNAME,IJF,IJT,NSEC,XSKT,
     & PLTBCH,PRTXSG,PRTBCH,PRTSUM,PPLTBH,ZBOT,ITRIAL,ICOUNT)
      KETIME=NETIME
      WRITE (PRINTR,2140) KTMATS
      SOLPDT=FLOAT(KTMATS)/(NSTEPS-1)
      WRITE (PRINTR,2145) SOLPDT
      WRITE(*,*) ' ERROR, BRANCH SIMULATION FINISHED BEFORE MODFLOW'
      STOP
C
C     INPUT/OUTPUT FORMAT STATEMENTS
C
 1740 FORMAT (2(A2,3X,2(I2,1X)))
 1750 FORMAT (10F8.2)
 1760 FORMAT (37X,I2,6X,I2)
 1780 FORMAT (4X,I3,F2.0,52X,I4)
 1790 FORMAT (8F10.3)
 1920 FORMAT (' ***** ERROR,     INITIAL STAGE VALUE UNSPECIFIED IN BRAN
     1CH ',I2,' SECTION ',I2,' *****')
 1960 FORMAT (' ***** ERROR,     MATRIX NOT SQUARE : REVIEW SCHEMATIZATI
     1ON AND EXTERNAL BOUNDARY-CONDITION SPECIFICATIONS *****')
 1980 FORMAT (' ***** ERROR,     WIND DATA FREQUENCY UNSPECIFIED OR INCO
     1NSISTENT *****')
 1990 FORMAT (' ***** ERROR,     MATRIX IS SINGULAR *****')
 2020 FORMAT (' ***** ERROR,     IMPROPER NUMBER OF WIND DATA SPECIFIED
     1(1<=NWDATA<=',I5,') *****')
 2050 FORMAT (' ***** ERROR,     INITIAL VALUE(S) OUT OF DEFINED RANGE O
     1F CHANNEL GEOMETRY FOR BRANCH ',I2,' SECTION ',I2,' *****')
 2075 FORMAT ('+',T69,'(STEEP SLOPE CHANNEL NEEDS ADDITIONAL A(Z) FOR DA
     1/DX TERM)')
 2080 FORMAT (/' ',11X,A2,'(J,I),J=1,NSEC(I)')
 2090 FORMAT (' ',1X,A2,'( J,',I2,')',12F10.4/11X,12F10.4)
 2110 FORMAT (' WARNING,   MAXIMUM ITERATIONS EXCEEDED AT ',I2,':',I2,'
     1ON ',I2,'/',I2,'/',I2,'     Z-ZP(',I2,',',I2,')=',F7.4,' Q-QP(',I2
     2,',',I2,')=',F7.1)
 2140 FORMAT (/' NUMBER OF SOLUTIONS = ',I5/)
 2145 FORMAT (' SOLUTIONS/TIME STEP = ',F5.1)
 2150 FORMAT (' Z=',E13.6,' Q=',E13.6,' QT=',E13.6,' QX=',E13.6,' AX=',E
     113.6,' ZX=',E13.6,' F=',E13.6,' W=',E13.6)
 2160 FORMAT (1X,8I16)
 2170 FORMAT (1X,8E16.8)
      END
```

LIST OF VARIABLES

(Value in parentheses is array dimension; only variables necessary to convert BRANCH to BRANCH' are included)

Variable	Range	Definition
CLK (MAXS)	Package	Value of K'/b' (leakage coefficient).
CLKIJ	Module	K'/b' at upstream cross section at beginning of BRANCH' time interval.
CLKP	Module	K'/b' at upstream cross section at end of BRANCH' time interval.
CLKP1	Module	K'/b' at downstream cross section at beginning of BRANCH' time interval.
CLKPP1	Module	K'/b' at downstream cross section at end of BRANCH' time interval.
DCFM	Package	Maximum value of the multiplier for the friction term in the momentum equation when the channel runs dry.
DCFM1	Package	Multiplier for the friction term in the momentum equation when the channel runs dry; maximum value is DCFM.
HAQ	Module	Aquifer head at upstream node at beginning of BRANCH time interval.
HAQP	Module	Aquifer head at upstream node at end of BRANCH time interval.
HAQP1	Module	Aquifer head at downstream node at beginning of BRANCH time interval.
HAQPP1	Module	Aquifer head at downstream node at end of BRANCH time interval.
HNEW (NCOL, NROW, NLAY)	Global	Value of aquifer head at end of MODFLOW time step.
HNEW1	Module	Local variable for aquifer head at end of MODFLOW time step.
HOLD (NCOL, NROW, NLAY)	Global	Value of aquifer head at beginning of MODFLOW time step.
HOLD1	Module	Local variable for aquifer head at beginning of MODFLOW time step.
IBEGIN	Package	Flag indicating if BRANCH' is being called for the first time (0, first; 1, not first time).
IBOUND (NCOL, NROW, NLAY)	Global	Status of each cell (<0 is constant head cell, =0 is inactive cell, >0 is variable head cell).
IBRPRN	Package	Flag indicating if BRANCH' is being called from the formulation or budget package (0, formulation; 1, budget).
ICOUNT	Module	Counter of number of BRANCH time intervals passed within the MODFLOW time step.
ISS	Global	Flag indicating steady-state simulation (0, not steady state; 1, steady state).

LIST OF VARIABLES—Continued

(Value in parentheses is array dimension, only variables necessary to convert BRANCH to BRANCH' are included)

Variable	Range	Definition
ISTRM (3, MAXS)	Package	Vector describing row, layer, and column of aquifer model cell that corresponds to channel segment.
ITRIA1	Module	Temporary value for ITRIAL vector for comparison.
ITRIA2	Module	Temporary value for ITRIAL vector for comparison.
ITRIA3	Module	Temporary value for ITRIAL vector for comparison.
ITRIAL (MAXS, MAXCZQ)	Package	Flag indicating the wet-dry condition of a channel reach and the relative position of the surrounding aquifer to the riverbed by a two-digit number. For first digit, 0 is reach wet, 1 is upstream node dry, 2 is downstream node dry, and 3 to 8 are both nodes dry with successively high frictional resistance. For second digit, 0 is aquifer above riverbed at both nodes, 1 is aquifer below riverbed at upstream node, 2 is aquifer below riverbed at downstream node, and 3 is aquifer below riverbed at both nodes.
KDAS	Package	Saved value of day.
KHRS	Package	Saved value of hour.
KKITER	Global	MODFLOW iteration number.
KMNS	Package	Saved value of minute.
KMOS	Package	Saved value of month.
KYRS	Package	Saved value of year.
MAXCZQ	Package	Maximum number of daily computed results held in storage for plotting purposes.
MAXMZQ	Package	Maximum number of measured data held in storage for plotting purposes.
MAXQBD	Package	Maximum number of discharge boundary value data held in storage for computation.
MAXS	Package	Maximum number of cross sections in the entire channel network.
MAXZBD	Package	Maximum number of stage boundary value data held in storage for computation.
MXBH	Package	Maximum number of branches in the network.
MXJN	Package	Maximum number of junctions in the network.
MXMD	Package	Maximum number of measured data locations accommodated in the network.
MXPT	Package	Maximum number of points used to define a cross section.

LIST OF VARIABLES—Continued

(Value in parentheses is array dimension; only variables necessary to convert BRANCH to BRANCH' are included)

Variable	Range	Definition
MXTDBC	Package	Maximum number of boundaries in the network.
MXWIND	Package	Maximum number of wind data points input.
NCOL	Global	Maximum number of MODFLOW aquifer columns.
NELAP	Package	Number of elapsed MODFLOW time steps since beginning of simulation.
NLAY	Global	Maximum number of MODFLOW aquifer layers.
NROW	Global	Maximum number of MODFLOW aquifer rows.
NTSAQ	Package	Number of BRANCH time intervals in one MODFLOW time step.
QLSUM (MAXS)	Package	Average leakage rate out of a river segment over one MODFLOW time step.
QPSAV (MAXS)	Package	Value of discharge at end of first BRANCH' time interval in a MODFLOW time step.
QSAV (MAXS)	Package	Value of discharge at beginning of first BRANCH' time interval in a MODFLOW time step.
ZBOT (MAXS)	Package	Elevation of channel bottom.
ZN (MAXS)	Package	Value of stage at end of final BRANCH' time interval in a MODFLOW time step.
ZPL (MAXS)	Package	Value of stage at end of final BRANCH' time interval in a MODFLOW time step for previous trial.
ZPSAV (MAXS)	Package	Value of stage at end of first BRANCH' time interval after beginning of a MODFLOW time step.
ZSAV (MAXS)	Package	Value of stage at beginning of first BRANCH' time interval after beginning of a MODFLOW time step.

SIMULATIONS OF STREAM-AQUIFER INTERACTION

The verification procedure for MODBRANCH was developed with the following criteria to be satisfied: (1) compare results with previously existing models; (2) simulate events that cannot be modeled with existing models; (3) demonstrate the use of special options, drying and rewetting of channels and steady-state simulation; and (4) compare simulation results with field data collected at a site in southern Florida.

SCHEMES FOR COMPARISON

In order to verify the MODBRANCH solution scheme, MODBRANCH results were compared with results from three other solution schemes: (1) the one-dimensional, unsteady, constant cross-section model described by Pinder and Sauer (1971); (2) a simple, four-point implicit scheme for a rectangular channel attached to MODFLOW in the same manner that BRANCH was attached; and (3) the flow-routing Stream package for MODFLOW (Prudic, 1989).

The one-dimensional, unsteady, constant cross-section streamflow model (referred to as the Pinder model) solved the continuity and momentum equations by an explicit finite-difference, staggered-net method. The two-dimensional ground-water flow equation is solved by the iterative, alternating direction implicit technique. The streamflow and ground-water equations are coupled by a leakage equation similar to equation 3a. This coupled model was used to demonstrate the modification of a floodwave because of bank storage (Pinder and Sauer, 1971). Thus, the Pinder model results can be reproduced by MODBRANCH for comparison.

The four-point implicit scheme (referred to as the four-point model) was created by Lewis Delong, Jon Lee, and David Thompson of the USGS as a training supplement in surface-water modeling. It solves the continuity and momentum equations in integral form for a unit width of channel. As in the case of the Pinder model, its use is confined to single rectangular channels of constant width. As a prelude to creating MODBRANCH, the four-point model was coupled with MODFLOW. The same format was used as in MODBRANCH; allocation, data input, formulation, and budget subroutines were created for the four-point model. This coupled four-point model can be used for comparison with MODBRANCH.

The stream module in MODFLOW (Prudic, 1989) can route flow from more than one tributary into a channel, so it is not limited to single channels as are the Pinder and four-point models. However, it is restricted to rectangular cross sections; only routes flow downstream, and backwater effects cannot be simulated. Flows into diversions and forks must be user defined, and the depth in each reach is calculated as steady uniform flow. This makes it comparable to MODBRANCH especially in steady-state simulation.

PROBLEM 1—FLOODWAVE PROPAGATION WITH BANK STORAGE

This verification involves duplicating the results from Pinder and Sauer (1971) with MODBRANCH and the four-point model. The hypothetical aquifer used extends 130,000 ft along the length of the channel and is 1,400 ft across. The hydraulic conductivity of the aquifer is 0.01 ft/s (864 ft/d), and the initial saturated thickness ranges from 220 ft at the