USGS
science for a changing world

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

Chapter A2

DOCUMENTATION OF A COMPUTER PROGRAM TO SIMULATE AQUIFER-SYSTEM COMPACTION USING THE MODULAR FINITE-DIFFERENCE GROUND-WATER FLOW MODEL

By S.A. Leake and D.E. Prudic

This chapter supersedes U.S. Geological Survey Open-File Report 88-482

Book 6
Chapter A2

IBS1OT

Narrative for Module IBS1OT

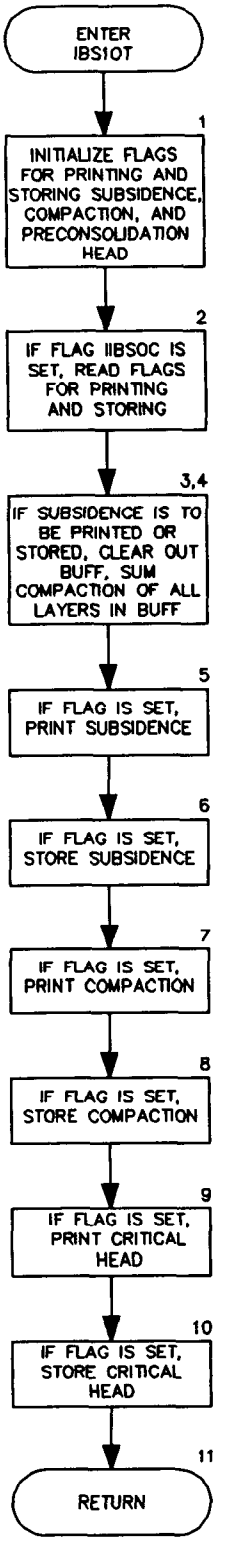
This module prints and stores subsidence, compaction, and preconsolidation head.

1. Initialize flags for printing and saving subsidence, compaction, and preconsolidation head.
2. Read flags for printing and saving subsidence, compaction, and preconsolidation head. Print flags.
3. Prepare to print and store subsidence. First, clear out buffer (BUFF).
4. Sum compaction in all layers to compute total subsidence.
5. If flag is set, print subsidence array with module ULAPRS or ULAPRW.
6. If flag is set, store subsidence array with module ULASAV.
7. If ICOMPR flag is set, print compaction array (all layers) with module ULAPRS or ULAPRW.
8. If ICOMSV flag is set, save compaction array (all layers) with module ULASAV.
9. If IHCPR flag is set, print preconsolidation-head array (all layers) with module ULAPRS or ULAPRW.
10. If IHCSV flag is set, save preconsolidation-head array (all layers) with module ULASAV.
11. RETURN.

Flow chart for module IBS1OT

IIBSOC is a flag.
If IIBSOC > 0, output control flags for printing and recording subsidence, compaction, and preconsolidation head will be read each time step.
If IIBSOC ≤ 0, output control flags will not be read.

BUFF is an array in which values are stored as they are being gathered for printing or recording.



Program Listing for Module IBS1OT

```
      SUBROUTINE IBS1OT(NCOL,NROW,NLAY,PERTIM,TOTIM,KSTP,KPER,NSTP,
     1           BUFF,SUB,HC,IIBSOC,ISUBFM,ICOMFM,IHCFM,ISUBUN,
     2           ICOMUN,IHCUN,IN,IOUT)
C-----VERSION 1227 02JUN1988 IBS1OT
C     ******************************************************************
C     PRINT AND STORE SUBSIDENCE, COMPACTION AND CRITICAL HEAD.
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 TEXT
      DIMENSION HC(NCOL,NROW,NLAY),SUB(NCOL,NROW,NLAY),
     1          BUFF(NCOL,NROW,NLAY),TEXT(4,3)
      COMMON /IBSCOM/ IBQ(80)
      DATA TEXT(1,1),TEXT(2,1),TEXT(3,1),TEXT(4,1) /'    ','  SU',
     1     'BSID','ENCE'/,TEXT(1,2),TEXT(2,2),TEXT(3,2),TEXT(4,2)
     2    /'    ','  CO','MPAC','TION'/,TEXT(1,3),TEXT(2,3),
     3    TEXT(3,3),TEXT(4,3) /'   C','RITI','CAL ','HEAD'/
C     ------------------------------------------------------------------
C
C1------INITIALIZE FLAGS FOR PRINTING AND SAVING SUBSIDENCE, COMPACTION,
C1------AND CRITICAL HEAD
      ISUBPR=0
      ICOMPR=0
      IHCPR=0
      ISUBSV=0
      ICOMSV=0
      IHCSV=0
      IF(KSTP.EQ.NSTP) ISUBPR=1
C2------READ FLAGS FOR PRINTING AND SAVING.
      IF(IIBSOC.LE.0) GO TO 28
      READ(IN,10) ISUBPR,ICOMPR,IHCPR,ISUBSV,ICOMSV,IHCSV
   10 FORMAT(6I10)
      WRITE(IOUT,15) ISUBPR,ICOMPR,IHCPR,ISUBSV,ICOMSV,IHCSV
   15 FORMAT(1H0,'FLAGS FOR PRINTING AND STORING SUBSIDENCE, '
     1 'COMPACTION, AND CRITICAL HEAD:'/
     2 '   ISUBPR    ICOMPR    IHCPR     ISUBSV    ICOMSV    IHCSV   '/
     3 ' ------------------------------------------------------------'/
     4 I6,5I10)
C
C3------PRINT AND STORE SUBSIDENCE, FIRST, CLEAR OUT BUFF.
   28 IF(ISUBPR.LE.0.AND.ISUBSV.LE.0) GO TO 100
      DO 30 IR=1,NROW
      DO 30 IC=1,NCOL
      BUFF(IC,IR,1)=0.
   30 CONTINUE
C
C4------SUM COMPACTION IN ALL LAYERS TO GET SUBSIDENCE.
      KQ=0
      DO 50 K=1,NLAY
      IF(IBQ(K).EQ.0) GO TO 50
```

```
      KQ=KQ+1
      DO 40 I=1,NROW
      DO 40 J=1,NCOL
      BUFF(J,I,1)=BUFF(J,I,1)+SUB(J,I,KQ)
   40 CONTINUE
   50 CONTINUE
C
C5------PRINT SUBSIDENCE.
      IF(ISUBPR.LE.0) GO TO 60
      IF(ISUBFM.LT.0) CALL ULAPRS(BUFF,TEXT(1,1),KSTP,KPER,NCOL,NROW,1,
     1          -ISUBFM,IOUT)
      IF(ISUBFM.GE.0) CALL ULAPRW(BUFF,TEXT(1,1),KSTP,KPER,NCOL,NROW,1,
     1           ISUBFM,IOUT)
C
C6------STORE SUBSIDENCE.
   60 IF(ISUBSV.LE.0) GO TO 100
      CALL ULASAV(BUFF,TEXT(1,1),KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,1,
     1             ISUBUN)
C
C7------PRINT COMPACTION FOR ALL LAYERS WITH INTERBED STORAGE.
  100 IF(ICOMPR.LE.0) GO TO 140
      KQ=0
      DO 130 K=1,NLAY
      IF(IBQ(K).LE.0) GO TO 130
      KQ=KQ+1
      IF(ICOMFM.LT.0) CALL ULAPRS(SUB(1,1,KQ),TEXT(1,2),KSTP,KPER,NCOL,
     1          NROW,K,-ICOMFM,IOUT)
      IF(ICOMFM.GE.0) CALL ULAPRW(SUB(1,1,KQ),TEXT(1,2),KSTP,KPER,NCOL,
     1           NROW,K,ICOMFM,IOUT)
  130 CONTINUE
C
C8------SAVE COMPACTION FOR ALL LAYERS WITH INTERBED STORAGE.
  140 IF(ICOMSV.LE.0) GO TO 200
      KQ=0
      DO 160 K=1,NLAY
      IF(IBQ(K).LE.0) GO TO 160
      KQ=KQ+1
      CALL ULASAV(SUB(1,1,KQ),TEXT(1,2),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1            NROW,K,ICOMUN)
  160 CONTINUE
C
C9------PRINT CRITICAL HEAD FOR ALL LAYERS WITH INTERBED STORAGE.
  200 IF(IHCPR.LE.0) GO TO 240
      KQ=0
      DO 230 K=1,NLAY
      IF(IBQ(K).LE.0) GO TO 230
      KQ=KQ+1
      IF(IHCFM.LT.0) CALL ULAPRS(HC(1,1,KQ),TEXT(1,3),KSTP,KPER,NCOL,
     1          NROW,K,-IHCFM,IOUT)
      IF(IHCFM.GE.0) CALL ULAPRW(HC(1,1,KQ),TEXT(1,3),KSTP,KPER,NCOL,
     1           NROW,K,IHCFM,IOUT)
  230 CONTINUE
C
C10-----SAVE CRITICAL HEAD FOR ALL LAYERS WITH INTERBED STORAGE.
```

```
  240 IF(IHCSV.LE.0) GO TO 300
      KQ=0
      DO 260 K=1,NLAY
      KQ=KQ+1
      IF(IBQ(K).LE.0) GO TO 260
      CALL ULASAV(HC(1,1,KQ),TEXT(1,3),KSTP,KPER,PERTIM,TOTIM,NCOL,
     1            NROW,K,IHCUN)
  260 CONTINUE
C
C11-----RETURN
  300 RETURN
      END
```

List of Variables for Module IBS1OT

Variable	Range	Definition
BUFF	Global	DIMENSION (NCOL,NROW,NLAY), Buffer used to accumulate information before printing or recording it.
HC	Package	DIMENSION (NCOL,NROW,NAQL), Preconsolidation head in each cell with interbed storage. NAQL is number of layers for which IBQ > 0.
I	Module	Index for rows.
IBQ	Package	DIMENSION (80), Layer flag for interbed storage: > 0, Layer has interbed storage. ≤ 0, Layer does not have interbed storage.
IC	Module	Index for columns.
ICOMFM	Package	Code for format in which compaction will be printed.
ICOMPR	Module	Output flag for printing entire compaction array for current time step. ≤ 0, compaction is not printed. > 0, compaction is printed.
ICOMSV	Module	Output flag for recording entire compaction array on disk for current time step. ≤ 0, compaction is not recorded. > 0, compaction is recorded.
ICOMUN	Package	Unit number on which an unformatted record containing compaction should be recorded.
IHCFM	Package	Code for format in which preconsolidation head will be printed.

Variable	Range	Definition
IHCPR	Module	Output flag for printing entire preconsolidation-head array for current time step. ≤ 0, preconsolidation head is not printed. > 0, preconsolidation head is printed.
IHCSV	Module	Output flag for recording entire preconsolidation-head array on disk for current time step. ≤ 0, preconsolidation head is not recorded. > 0, preconsolidation head is recorded.
IHCUN	Package	Unit number on which an unformatted record containing preconsolidation head should be recorded.
IIBSOC	Package	Flag. > 0, output control will be read each time step for printing and recording subsidence, compaction, and preconsolidation head. ≤ 0, subsidence will be printed at the end of each stress period.
IN	Package	Primary unit number from which input for this package will be read.
IOUT	Global	Primary unit number for all printed output. IOUT = 6.
IR	Module	Index for rows.
ISUBFM	Package	Code for format in which subsidence will be printed.
ISUBPR	Module	Output flag for printing entire subsidence array for current time step. ≤ 0, subsidence is not printed. > 0, subsidence is printed.
ISUBSV	Module	Output flag for recording entire subsidence array on disk for current time step. ≤ 0, subsidence is not recorded. > 0, subsidence is recorded.
ISUBUN	Package	Unit number on which an unformatted record containing subsidence should be recorded.
J	Module	Index for columns.
K	Module	Index for layers.

Variable	Range	Definition
KPER	Global	Stress period counter.
KQ	Module	Index of model layers with interbed storage.
KSTP	Global	Time step counter. Reset at the start of each stress period.
MSUM	Global	Counter for budget entries and labels in VBVL and VBNM.
NCOL	Global	Number of columns in the grid.
NLAY	Global	Number of layers in the grid.
NROW	Global	Number of rows in the grid.
NSTP	Global	Number of time steps in the current stress period.
PERTIM	Global	Elapsed time during the current stress period.
SUB	Package	DIMENSION (NCOL,NROW,NAQL), Computed compaction of each cell with interbed storage. NAQL is number of layers for which IBQ > 0.
TEXT	Module	DIMENSION(4,3), Labels to be printed or recorded with array data.
TOTIM	Global	Elapsed time in the simulation.

References Cited

Bull, W.B., 1975, Land subsidence due to ground-water withdrawal in the Los Banos Kettleman City area, California—Part 2, subsidence and compaction of deposits: U.S. Geological Survey Professional Paper 437-F, 90 p.

Chi, S.C., and Reilinger, R.E., 1984, Geodetic evidence for land subsidence due to groundwater withdrawal in many parts of the United States of America: Journal of Hydrology, v. 67, p. 155-182.

Helm, D.C., 1975, One-dimensional simulation of aquifer-system compaction near Pixley, California. 1, Constant parameters: Water Resources Research, v. 11, no. 3, p. 465-478.

———1984, Latrobe Valley subsidence predictions: The modeling of time-dependent ground movement due to groundwater withdrawal: State Electricity Commission of Victoria, Australia, Fuel Department and Design Engineering and Environment Department, 2 volumes, v.p.

Holzer, T.L., 1981, Preconsolidation stress of aquifer systems in areas of induced land subsidence: Water Resources Research, v. 17, no. 3, p. 693-704.

Jacob, C.E., 1940, On the flow of water in an elastic artesian aquifer: American Geophysical Union Transmittal 21st Annual Meeting, part 2, p. 574-586.

Jorgensen, D.G., 1980, Relationships between basic soils-engineering equations and basic ground-water flow equations: U.S. Geological Survey Water-Supply Paper 2064, 40 p.

Lofgren, B.E., 1968, Analysis of stresses causing land subsidence, in Geological Survey Research 1968: U.S. Geological Survey Professional Paper 600-B, p. 219-225.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 6, Chapter A1, 586 p.

Meade, R.H., 1967, Petrology of sediments underlying areas of land subsidence in Central California: U.S. Geological Survey Professional Paper 497-C, 83 p.

Meyer, W.R., and Carr, J.E., 1979, A digital model for simulation of ground-water hydrology in the Houston area, Texas: Texas Department of Water Resources Report LP-103, 131 p.

Poland, J.F., and Davis, G.H., 1969, Land subsidence due to withdrawals of fluids, *in* Varnes, D.J., and Kiersch, George, eds., Reviews in Engineering Geology: Volume 2: Boulder, Colorado, Geological Society of America, p. 187-269.

Poland, J.F., 1981, Subsidence in United States due to ground-water withdrawal, *in* Legal, socio-economic, and environmental significance of land subsidence: Session on Land Subsidence, American Society of Civil Engineers Annual Convention, October 26, 1979, Proceedings, p. 115-135.

Poland, J.F., ed., 1984, Guidebook to studies of land subsidence due to ground-water withdrawal: Paris, France, United Nations Educational, Scientific and Cultural Organization Studies and Reports in Hydrology, number 40, 305 p.

Riley, F.S., 1969, Analysis of borehole extensometer data from central California, *in* Tison, L.J., ed., Land subsidence, v. 2: International Association of Scientific Hydrology Publication 89, p. 423-431.

Terzaghi, Karl, 1925, Erdbaumechanik auf Bodenphysikalischer Grundlage: Wein Leipzig, Deuticke, 399 p.

Williamson, A.K., Prudic, D.E., and Swain, L.A., 1989, Ground-water flow in the Central Valley, California: U.S. Geological Survey Professional Paper 1401-D, 127 p.

APPENDIXES A–D

APPENDIX A

INPUT DATA SETS FOR STORAGE-DEPLETION TEST PROBLEM

The storage-depletion test problem tests the ability of the Interbed-Storage Package to compute compaction and storage changes from a specified head decline. Details of the test problem and results are discussed in the section entitled "Test Problems."

Basic Package Input Data Set

Input for the Basic Package follows the column numbers. The input consists of 53 records, all of which are shown below and on the following page. Input for this package is read from the unit number for Basic Package input specified in the MAIN program.

```
                                        Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
MODULAR MODEL STORAGE-DEPLETION TEST PROBLEM--TESTS CALCULATION OF ULTIMATE
COMPACTION OF INTERBEDS
         2        10        12         3         4
 7  0  0  0  0  0  0  0 10  0  0 11  0  0  0  0  0  0 13
         0         1    IAPART,ISTRT
         5         1(20I3)
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
         5         1(20I3)
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
-1  1  1  1  1  1  1  1  1  1  1 -1
```

Input for the Basic Package is continued following the column numbers.

```
                                        Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
    444.44     HNOFLO
         5        1.(18F3.0)
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
         5        1.(18F3.0)
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
  0 11 12 13 14 15 16 17 18 19 20 11
     1000.          10        1.5
     1000.          10        1.5
     1000.          10        1.5
```

Block-Centered Flow Package Input Data Set

Input for the Block-Centered Flow Package follows the column numbers. The input consists of 10 records, all of which are shown below. Input for this package is read from unit number 7 as specified in the data set for the Basic Package.

```
                                        Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         0         0   ISS  IBCFCB
 0 0
         0        1.                                          TRPY
         0     1000.                                          DELR
         0     1000.                                          DELC
         0     .0001                                          sf1 -layer 1
         0     1000.                                          tran -layer 1
         0  0.000001                                          VCONT -layer 1
         0     .0001                                          sf1 -layer 2
         0     1000.                                          tran -layer2
```

Strongly Implicit Procedure Package Input Data Set

Input for the Strongly-Implicit Procedure Package follows the column numbers. The input consists of two records, both of which are shown below. Input for this package is read from unit number 8 as specified in the data set for the Basic Package.

```
                                        Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         50        5
         1.    .00001         1         0         1
```

Interbed-Storage Package Input Data Set

Input for the Interbed-Storage Package follows the column numbers. The input consists of 16 records, all of which are shown below. Input for this package is read from unit number 13 as specified in the data set for the Basic Package.

```
                                        Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         0         0
 1 0
        13          1.(12F3.0)
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
  5  6  7  8  9 10 11 12 13 14 15 16
         0       .0001
         0        .001
         0         0.0
```

Output Control Package Input Data Set

Input for the Output Control Package follows the column numbers. The input consists of 61 records, 21 of which are shown below. Input for this package is read from unit number 11 as specified in the data set for the Basic Package.

```
                              Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         5         5         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         1         0         0
         1         0         0         0
    (include 2 additional sets of the above 20 records)
```

APPENDIX B

INPUT DATA SETS FOR RAMP-LOAD TEST PROBLEM

The ramp-load test problem tests the ability of the Interbed-Storage Package to compute compaction consistent with a one-dimensional compaction model by Helm (1975, 1984). Details of the test problem and results are discussed in the section entitled "Ramp-Load Test Problem."

Basic Package Input Data Set

Input for the Basic Package follows the column numbers. The input consists of 109 records, 10 of which are shown below. Input for this package is read from the unit number for Basic Package input specified in the MAIN program.

```
                                    Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
Test of IBS1 package--Ramp loading problem for comparison with model by
Helm (1975, 1984)
         1         1        21       100         4
 07 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00 00 00 12 11
         0         0
         5         1(40I2)
-1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      999.
         0       10.
       18.        18       1.0
```

(include 99 additional records identical to the above record)

Block-Centered Flow Package Input Data Set

Input for the Block-Centered Flow Package follows the column numbers. The input consists of eight records, all of which are shown below. Input for this package is read from unit number 7 as specified in the data set for the Basic Package.

```
                                    Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         0         0
 0
         0         1
         0      0.25
         0      1.00
         0      0.00
         7    1.E-06(16F5.0)
1.E10 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78 2.78
```

Strongly Implicit Procedure Package Input Data Set

Input for the Strongly Implicit Procedure Package follows the column numbers. The input consists of two records, both of which are shown below. Input for this package is read from unit number 8 as specified in the data set for the Basic Package.

```
                                  Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        50         5
        1.      .001         1         0       999
```

Interbed-Storage Package Input Data Set

Input for the Interbed-Storage Package follows the column numbers. The input consists of eight records, all of which are shown below. Input for this package is read from unit number 12 as specified in the data set for the Basic Package.

```
                                  Column Numbers
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        0         0
 1
        0        10.
       12    1.0E-6(40F2.0)
 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
       12    1.0E-4(40F2.0)
 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        0        0.
```

Time-Variant Specified-Head Package Input Data Set

Input for the Time-Variant Specified-Head Package follows the column numbers. The input consists of 201 records, 41 of which are shown below. Input for this package is read from unit number 11 as specified in the data set for the Basic Package. Input instructions for the Time-Variant Specified-Head Package are given in appendix C.

```
                                  Column Numbers
        1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
        1
        1
        1         1         1      10.0       9.0
        1
        1         1         1       9.0       8.0
        1
        1         1         1       8.0       7.0
        1
        1         1         1       7.0       6.0
        1
        1         1         1       6.0       5.0
        1
        1         1         1       5.0       4.0
        1
        1         1         1       4.0       3.0
        1
        1         1         1       3.0       2.0
        1
        1         1         1       2.0       1.0
        1
        1         1         1       1.0       0.0
        1
        1         1         1       0.0       1.0
        1
        1         1         1       1.0       2.0
        1
        1         1         1       2.0       3.0
        1
        1         1         1       3.0       4.0
        1
        1         1         1       4.0       5.0
        1
        1         1         1       5.0       6.0
        1
        1         1         1       6.0       7.0
        1
        1         1         1       7.0       8.0
        1
        1         1         1       8.0       9.0
        1
        1         1         1       9.0      10.0
(include 4 additional sets of the above 40 records)
```

APPENDIX C

TIME-VARIANT SPECIFIED-HEAD PACKAGE

The Time-Variant Specified-Head Package was developed to allow constant-head cells to take on different values for each time step. It provides a means of simulating head variations with time that can be approximated with piecewise linear segments. The Time-Variant Specified-Head Package does not alter the way constant-head boundaries are formulated in the finite-difference equations of the model by McDonald and Harbaugh (1988). Because the package is not a complex modification of the flow model, documentation in this appendix is the minimum amount required to implement the package. Additional information regarding the Time-Variant Specified-Head Package is given in the section entitled "Ramp-Load Test Problem." Modifications to the main program to implement the package are given in Appendix D.

As the package is presented here, a specified number of cells are defined for each stress period. The package sets the element in the IBOUND array to a negative value for all cells where a time-variant specified-head boundary is selected. Because layer, row, and column locations for cells having a time-variant specified head are read for each stress period, the opportunity exists to specify different sets of cells for each stress period. If a cell is specified as a time-variant specified-head boundary for a stress period and omitted in the specification for a subsequent period, it remains a constant-head boundary with a head equal to the end head of the last stress period for which it was specified.

The Time-Variant Specified-Head Package includes three modules. The first module, CHD1AL, allocates space for an array that is needed for the package. The second module, CHD1RP, reads boundary-cell indices and head values at the start of each stress period and sets corresponding elements in the IBOUND array to negative values. The third module, CHD1FM, linearly interpolates boundary head for each time-variant specified-head boundary cell and sets the corresponding elements in the HNEW and HOLD arrays to the interpolated values. As defined by McDonald and Harbaugh (1988), the Formulate Procedure calculates conductances and coefficients required by the finite-difference flow equation. Module CHD1FM is considered to be part of the Formulate Procedure because its function is to calculate values for the known heads, which are components of the flow equation. The flowchart for the model (McDonald and Harbaugh, 1988, chap. 3, p. 2) shows that modules in the Formulate Procedure are called each iteration during the iterative solution process; however, module CHD1FM is called outside the iterative program loop. CHD1FM is called only once each time step because the specified heads are constant during a time step.

Input Instructions

Input for the Time-Variant Specified-Head Package is read from the unit in IUNIT(20) specified in the Basic Package input (see McDonald and Harbaugh, 1988, chap. 4, p. 9-11).

FOR EACH SIMULATION

CHD1AL

1. Data: MXCHD
   Format: I10

FOR EACH STRESS PERIOD

CHD1RP

2. Data: ITMP
   Format: I10

					Start	End
3. Data:	Layer	Row	Column	Start Head	End Head
Format:	I10	I10	I10	F10.0	F10.0

(Input item 3 normally consists of one record for each constant-head boundary cell. If ITMP is zero, item 3 is not read)

Explanation of Fields Used in Input Instructions

MXCHD is the maximum number of specified-head cells to be specified each stress period.

ITMP is a flag.

If ITMP < 0, specified-head boundary data from the previous stress period will be reused and input item 3 will not be read. Reusing data from a previous stress period means that the head values at the start and end of the current stress period will be the same as they were at the start and end of the previous stress period.

If ITMP ≥ 0, it is the number of records of specified-head boundary data that will be read for the current stress period.

Layer is the layer number of the cell affected by the specified-head boundary.

Row is the row number of the cell affected by the specified-head boundary.

Column is the column number of the cell affected by the specified-head boundary.

Start head is the head value at the cell at the start of the stress period. Because the package assigns head values on the basis of linear interpolation to the end of a time step, the assigned specified-head value will never equal Start head unless Start head and End head are equal.

End head is the head value at the cell at the end of the stress period. It is the value that will be assigned to the cell for the last time step in the stress period.

Program Listing for Module CHD1AL

```
      SUBROUTINE CHD1AL(ISUM,LENX,LCCHDS,NCHDS,MXCHD,IN,IOUT)
C
C-----VERSION 0000 23SEP1987 CHD1AL
C   ******************************************************************************
C     ALLOCATE ARRAY STORAGE FOR CONSTANT-HEAD CELLS
C   ******************************************************************************
C
C     SPECIFICATIONS:
C   ------------------------------------------------------------------------------
C   ------------------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND INITIALIZE # OF CONSTANT HEAD CELLS
      WRITE(IOUT,1)IN
    1 FORMAT(1H0,'CHD1 -- CHD PACKAGE, VERSION 1, 09/23/87',
     2' INPUT READ FROM UNIT',I3)
      NCHDS=0
C
C2------READ AND PRINT MXCHD (NUMBER OF CONSTANT-HEAD
C2------CELLS TO BE SPECIFIED EACH STRESS PERIOD)
      READ(IN,2) MXCHD
    2 FORMAT(I10)
      WRITE(IOUT,3) MXCHD
    3 FORMAT(1H ,'A TOTAL OF',I5,' CONSTANT-HEAD CELLS MUST BE',
     2 ' SPECIFIED EACH STRESS PERIOD.')
C
C3------SET LCCHDS EQUAL TO ADDRESS OF FIRST UNUSED SPACE IN X.
      LCCHDS=ISUM
C
C4------CALCULATE AMOUNT OF SPACE USED BY THE CONSTANT-HEAD LIST.
      ISP=5*MXCHD
      ISUM=ISUM+ISP
C
C5------PRINT AMOUNT OF SPACE USED BY THE CHD PACKAGE.
      WRITE(IOUT,4) ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR CONSTANT',
     1      '-HEAD CELLS')
      ISUM1=ISUM-1
      WRITE(IOUT,5) ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C6------RETURN
      RETURN
      END
```

Program Listing for Module CHD1RP

```
      SUBROUTINE CHD1RP(CHDS,NCHDS,MXCHD,IBOUND,NCOL,NROW,NLAY,
     2  PERLEN,DELT,NSTP,TSMULT,IN,IOUT)
C
C
C-----VERSION 0000 23SEP1987 CHD1RP
C   ******************************************************************************
C     READ DATA FOR CHD
C   ******************************************************************************
C
C     SPECIFICATIONS:
C   ------------------------------------------------------------------------------
      DIMENSION CHDS(5,MXCHD),IBOUND(NCOL,NROW,NLAY)
C   ------------------------------------------------------------------------------
C
C1------READ ITMP(FLAG TO REUSE DATA.)
      READ(IN,8) ITMP
    8 FORMAT(I10)
C
C2------TEST ITMP
      IF(ITMP.GE.0) GO TO 50
C
C2A-----IF ITMP<0 THEN REUSE DATA FROM LAST STRESS PERIOD.
      WRITE(IOUT,7)
    7 FORMAT(1HO,'REUSING CONSTANT-HEAD CELLS FROM LAST STRESS',
     1      ' PERIOD')
      GO TO 260
C
C3------IF ITMP=>0 THEN IT IS THE # OF CONSTANT-HEAD CELLS.
   50 NCHDS=MXCHD
C
C4------PRINT # OF CONSTANT-HEAD CELLS THIS STRESS PERIOD.
  100 WRITE(IOUT,1) NCHDS
    1 FORMAT(1HO,//1X,I5,' CONSTANT-HEAD CELLS')
C
C5------IF THERE ARE NO CONSTANT-HEAD CELLS THEN RETURN.
      IF(NCHDS.EQ.0) GO TO 260
C
C6------READ & PRINT DATA FOR EACH CONSTANT-HEAD CELL.
      WRITE(IOUT,3)
    3 FORMAT(1HO,15X,'LAYER',5X,'ROW',5X
     1,'COL   STRT HEAD   ENDING HEAD'/1X,15X,48('-'))
      DO 250 II=1,NCHDS
      READ (IN,4) K,I,J,CHDS(4,II),CHDS(5,II)
    4 FORMAT(3I10,2F10.0)
      WRITE (IOUT,5) K,I,J,CHDS(4,II),CHDS(5,II)
    5 FORMAT(1X,15X,I4,I9,I8,G13.4,G14.4)
      CHDS(1,II)=K
      CHDS(2,II)=I
      CHDS(3,II)=J
      IF(IBOUND(J,I,K).NE.0) IBOUND(J,I,K)=-IABS(IBOUND(J,I,K))
  250 CONTINUE
```

```
C
C7------RECOMPUTE LENGTH OF PERIOD, PERLEN, A LOCAL VARIABLE IN
C7------SUBROUTINE BAS1AD
      PERLEN=DELT*FLOAT(NSTP)
      IF(TSMULT.NE.1.) PERLEN=DELT*(1.-TSMULT**NSTP)/(1.-TSMULT)
C8------RETURN
  260 RETURN
      END
```

Program Listing for Module CHD1FM

```
      SUBROUTINE CHD1FM(NCHDS,MXCHD,CHDS,IBOUND,HNEW,
     1          HOLD,PERLEN,PERTIM,DELT,NCOL,NROW,NLAY)
C
C-----VERSION 0000 23SEP1987 CHD1FM
C   ******************************************************************************
C     COMPUTE HEAD FOR TIME STEP AT EACH CONSTANT HEAD CELL
C   ******************************************************************************
C
C     SPECIFICATIONS:
C   ------------------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION CHDS(5,MXCHD),IBOUND(NCOL,NROW,NLAY),
     2          HNEW(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY)
C   ------------------------------------------------------------------------------
C
C1------IF NCHDS<=0 THEN THERE ARE NO CONSTANT-HEAD CELLS. RETURN.
      IF(NCHDS.LE.0) RETURN
C
C2------COMPUTE PROPORTION OF STRESS PERIOD TO END OF THIS TIME STEP
      FRAC=PERTIM/PERLEN
C
C2------PROCESS EACH ENTRY IN THE CONSTANT-HEAD CELL LIST (CHDS)
      DO 100 L=1,NCHDS
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY
      IL=CHDS(1,L)
      IR=CHDS(2,L)
      IC=CHDS(3,L)
C
C5------COMPUTE HEAD AT CELL BY LINEAR INTERPOLATION.
      HB=CHDS(4,L)+(CHDS(5,L)-CHDS(4,L))*FRAC
C
C6------UPDATE THE APPROPRIATE HNEW VALUE
      HNEW(IC,IR,IL)=HB
      HOLD(IC,IR,IL)=HB
  100 CONTINUE
C
C7------RETURN
      RETURN
      END
```

APPENDIX D

EXAMPLE OF MAIN PROGRAM TO USE WITH INTERBED-STORAGE AND TIME-VARIANT SPECIFIED-HEAD PACKAGES

The following main program has been modified to allow use of the Interbed-Storage Package and the Time-Variant Specified-Head Package. Either one of the packages can be run independently of the other. The additions to the code specify that input data for the Interbed-Storage Package are to be read from the unit number stored in element 19 of the IUNIT array. Input data for the Time-Variant Specified-Head Package are to be read from the unit number stored in element 20 of the IUNIT array.

In the following program listing, records added to access the Interbed-Storage Package are identified with the characters "IBS" in rightmost columns. Records added to access the Time-Variant Specified-Head Package are identified with the characters "CHD" in rightmost columns. The identifiers "IBS" and "CHD" are not necessary parts of the added records and if they are included in the FORTRAN program, they must be placed within columns 73-80.

Program Listing

```
C   ******************************************************************************
C     MAIN CODE FOR MODULAR MODEL --    9/1/87
C          BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1638 24JUL1987 MAIN1
C   ******************************************************************************
C
C        SPECIFICATIONS:
C   ------------------------------------------------------------------------------
      COMMON X(120000)
      COMMON /FLWCOM/LAYCON(80)
      CHARACTER*4 HEADNG,VBNM
      DIMENSION HEADNG(32),VBNM(4,20),VBVL(4,20),IUNIT(24)
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C   ------------------------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=120000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=5
      IOUT=6
C
C3------DEFINE PROBLEM ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1            NODES,INBAS,IOUT,IUNIT)
C
```

```
C4------ALLOCATE SPACE IN "X" ARRAY.
      CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1              LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2              INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1     LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
     2     NCOL,NROW,NLAY,IOUT,IBCFCB)
      IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
     1                 IUNIT(2),IOUT,IWELCB)
      IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
     1                 IUNIT(3),IOUT,IDRNCB)
      IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1            NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
      IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
     1            LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
      IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
     1            IUNIT(4),IOUT,IRIVCB)
      IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
     1            IUNIT(7),IOUT,IGHBCB)
      IF(IUNIT(9).GT.0) CALL SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
     1          LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
     2          IUNIT(9),IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
     1          LCIEQP,MXITER,NCOL,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
      IF (IUNIT(19).GT.0) CALL IBS1AL(ISUM,LENX,LCHC,LCSCE,LCSCV,        IBS
     1          LCSUB,NCOL,NROW,NLAY,IIBSCB,IIBSOC,ISS,IUNIT(19),IOUT)  IBS
      IF(IUNIT(20).GT.0) CALL CHD1AL(ISUM,LENX,LCCHDS,NCHDS,MXCHD,       CHD
     1          IUNIT(20),IOUT)                                          CHD
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1       ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2       IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1          X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2           X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3           IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT)
      IF(IUNIT(9).GT.0) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
     1          IUNIT(9),IPCALC,IPRSIP,IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1RP(MXITER,ACCL,HCLOSE,IUNIT(11),
     1         IPRSOR,IOUT)
      IF(IUNIT(19).GT.0) CALL IBS1RP(X(LCDELR),X(LCDELC),X(LCHNEW),      IBS
     1      X(LCHC),X(LCSCE),X(LCSCV),X(LCSUB),NCOL,NROW,NLAY,           IBS
     2      NODES,IIBSOC,ISUBFM,ICOMFM,IHCFM,ISUBUN,ICOMUN,IHCUN,        IBS
     3      IUNIT(19),IOUT)                                              IBS
C
C7------SIMULATE EACH STRESS PERIOD.
      DO 300 KPER=1,NPER
      KKPER=KPER
C
```

```
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KKPER,INBAS,IOUT)
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
      IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     1             IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
     1                  IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCRECH),
     1            X(LCDELR),X(LCDELC),NROW,NCOL,IUNIT(8),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
     1            IUNIT(5),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
     1            IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
     1              IOUT)
      IF(IUNIT(20).GT.0) CALL CHD1RP(X(LCCHDS),NCHDS,MXCHD,X(LCIBOU),    CHD
     1         NCOL,NROW,NLAY,PERLEN,DELT,NSTP,TSMULT,IUNIT(20),IOUT)    CHD
C
C7C-----SIMULATE EACH TIME STEP.
      DO 200 KSTP=1,NSTP
      KKSTP=KSTP
C
C7C1----CALCULATE TIME STEP LENGTH. SET HOLD=HNEW..
      CALL BAS1AD(DELT,TSMULT,TOTIM,PERTIM,X(LCHNEW),X(LCHOLD),KKSTP,
     1             NCOL,NROW,NLAY)
      IF(IUNIT(20).GT.0) CALL CHD1FM(NCHDS,MXCHD,X(LCCHDS),X(LCIBOU),    CHD
     1          X(LCHNEW),X(LCHOLD),PERLEN,PERTIM,DELT,NCOL,NROW,NLAY)   CHD
C
C7C2----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
      DO 100 KITER=1,MXITER
      KKITER=KITER
C
C7C2A---FORMULATE THE FINITE DIFFERENCE EQUATIONS.
      CALL BAS1FM(X(LCHCOF),X(LCRHS),NODES)
      IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
     1          X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
     2          X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
     3          X(LCDELR),X(LCDELC),DELT,ISS,KKITER,KKSTP,KKPER,NCOL,
     4          NROW,NLAY,IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1FM(NWELLS,MXWELL,X(LCRHS),X(LCWELL),
     1           X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
     1         X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(8).GT.0) CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
     1            X(LCHNEW),NCOL,NROW,NLAY)
      IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
     1            X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(7).GT.0) CALL GHB1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
```

```
      IF(IUNIT(19).GT.0) CALL IBS1FM(X(LCRHS),X(LCHCOF),X(LCHNEW),      IBS
     1       X(LCHOLD),X(LCHC),X(LCSCE),X(LCSCV),X(LCIBOU),             IBS
     2       NCOL,NROW,NLAY,DELT)                                       IBS
C
C7C2B---MAKE ONE CUT AT AN APPROXIMATE SOLUTION.
      IF(IUNIT(9).GT.0) CALL SIP1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
     1     X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
     2     X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KKITER,HCLOSE,ACCL,ICNVG,
     3     KKSTP,KKPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
     4     IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1     X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
     2     X(LCHDCG),X(LCLRCH),KKITER,HCLOSE,ACCL,ICNVG,KKSTP,KKPER,
     3     IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C
C7C2C---IF CONVERGENCE CRITERION HAS BEEN MET STOP ITERATING.
      IF(ICNVG.EQ.1) GO TO 110
  100 CONTINUE
      KITER=MXITER
  110 CONTINUE
C
C7C3----DETERMINE WHICH OUTPUT IS NEEDED.
      CALL BAS1OC(NSTP,KKSTP,ICNVG,X(LCIOFL),NLAY,
     1  IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
      MSUM=1
      IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
     1     X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     2     X(LCTOP),X(LCSC2),DELT,ISS,NCOL,NROW,NLAY,KKSTP,KKPER,
     3     IBCFCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,
     1     X(LCWELL),X(LCIBOU),DELT,NCOL,NROW,NLAY,KKSTP,KKPER,IWELCB,
     1     ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
     1     X(LCDRAI),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IDRNCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCIBOU),NROW,NCOL,NLAY,DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,
     2     IRCHCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
     1     X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,
     2     DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,IEVTCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1BD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
     1     X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2     KKSTP,KKPER,IRIVCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,
     1     X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IGHBCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(19).GT.0) CALL IBS1BD(X(LCIBOU),X(LCHNEW),X(LCHOLD),      IBS
     1      X(LCHC),X(LCSCE),X(LCSCV),X(LCSUB),X(LCDELR),X(LCDELC),      IBS
     2      NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,IIBSCB,       IBS
     3      ICBCFL,X(LCBUFF),IOUT)                                       IBS
C
```

```
C7C5---PRINT AND OR SAVE HEADS AND DRAWDOWNS. PRINT OVERALL BUDGET.
C7C5A--FIRST PRINT AND OR SAVE SUBSIDENCE, COMPACTION,                  IBS
C7C5A-- AND CRITICAL HEAD.                                              IBS
      IF(IUNIT(19).GT.0) CALL IBS1OT(NCOL,NROW,NLAY,PERTIM,TOTIM,KKSTP, IBS
     1      KKPER,NSTP,X(LCBUFF),X(LCSUB),X(LCHC),IIBSOC,ISUBFM,ICOMFM, IBS
     2      IHCFM,ISUBUN,ICOMUN,IHCUN,IUNIT(19),IOUT)                   IBS
      CALL BAS1OT(X(LCHNEW),X(LCSTRT),ISTRT,X(LCBUFF),X(LCIOFL),
     1        MSUM,X(LCIBOU),VBNM,VBVL,KKSTP,KKPER,DELT,
     2        PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     3     IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT)
C
C7C6----IF ITERATION FAILED TO CONVERGE THEN STOP.
      IF(ICNVG.EQ.0) STOP
  200 CONTINUE
  300 CONTINUE
C
C8------END PROGRAM
      STOP
C
      END
```