รายการอ้างอิง

1. Aranyanark, C. Major Problems in Conservation of Monuments in Thailand. Proc. of the First Seminar on Thai-Japanese Cooperation in Conservation of Monuments in Thailand., Bangkok, March 8-9, 1999.

2. Tsongas, G. A. A field test for correlation of air leakage and high moisture content sites in tightly built walls. ASHRAE Transactions. 97 (1991) : 1-8.

3. Salonvaara, M. H. Prediction of hygrothermal performance of building envelope parts coupled with indoor climate. ASHRAE Transactions. 104 (1998) : 908-918.

4. Kuchitsu, N., Ishizaki, T., and Nishiura, T. Salt Weathering of The Brick Monuments in Ayutthaya, Thailand., Proc. of the First Seminar on Thai-Japanese Cooperation in Conservation of Monuments in Thailand., Bangkok, March 8-9, 1999.

5. Nishiura, T., Ishizaki, T., Aranyanark, C., and Yupho, K. Conservation Threatment fot the Giant Buddha of Wat Sri Chum in Sukhothai, Thailand., Proc. of the First Seminar on Thai-Japanese Cooperation in Conservation of Monuments in Thailand., Bangkok, March 8-9, 1999.

6. Ishizaki, T., Kuchitsu, N., Nishiura, T., Simunek J., and Van Genuchten, M. Th. Analysis of Water Regime in Historical Brick Building and Monuments in Thailand., Proc. of the First Seminar on Thai-Japanese Cooperation in Conservation of Monuments in Thailand., Bangkok, March 8-9, 1999.

7. Rubin, J. Theoretical Analysis of Two-Dimensional, Transient Flow of Water in Unsaturated and Partly Unsaturated Soils. Soil Science Society of America Proceedings. No.5, 32 (1968) : 607-615.

8. Hornberger, G. M., and Remson, I. Numerical Studies of a Composite Soil Moisture Ground-Water System. Water Resource Research. No.4, 5 (1969) : 797-802.

9. Neuman, S. P. Saturated-Unsaturated Seepage by Finite Elements. Journal of the Hydraulics Division. HY12 (1973) : 2233-2250.

10. Todd D. K., Ground Water Hydrology. 2nd ed., New York : John Wiley & Sons, Inc., 1980.

11. Schwaetz, F. W., and Zhang, H. B. Fundamentals of Groundwater., New York : John Wiley & Sons, Inc., 2003.

12. Richards, L. A. Capillary Conduction of Liquids through Porous Mediums. Physics. No.5, 1 (1931) : 318-333.

13. ปราโมทย์ เดชะอำไพ, ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. พิมพ์ครั้งที่ 2. กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2542.

14. Huyakorn, P. S., and Pinder, G. F. Computational methods in subsurface flow., New York : Academic Press, 1983.

15. ปราโมทย์ เดชะอำไพ, ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการคำนวณพลศาสตร์ของไหล. กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2545.

16. Huebner, K. H., Thornton, E. A. and Byron, T. G. The Finite Element Method for Engineers. 3rd ed., New York : John Wiley & Sons, Inc., 1995.

17. ปราโมทย์ เดชะอำไพ, ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม. พิมพ์ครั้งที่ 2. กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2541.

18. Phongthanapanich, S. and Dechaumphai, P. Evaluation of Combined Delaunay Triangulation and Adaptive Finite Elements for Heat Transfer Problems. Transactions of Canadian Society for Mechanical Engineering. No.4, 27 (2003).

19. Poulikakos, D. Conduction Heat Transfer., New Jersey : Prentice-Hall, 1994.

20. Dechaumphai, P. Transient Finite Element Thermal Analysis Using Adaptive Mesh Movement. Journal of Energy, Heat and Mass Transfer. 19 (1997) : 153-158.

21. Huyakorn, P. S., and Thomas, S. D. Techniques for Making Finite Elements Competitive in Modeling Flow in Variably Saturated Porous Media. Water Resource Research. No.8, 20 (1984) : 1099-1115.

22. Haverkamp, R., Vauclin, M., Tauma, J., Wierenga, P., and Vachaud, G. Comparison of Numerical Simulation Models for One-Dimensional Infiltration. Soil Science Society of America Journal. 41 (1977) : 285-294.

23. Celia, M. A., and Bouloutas, E. T. A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation. Water Resource Research. No.7, 26 (1990) : 1483-1496.

24. Antonopoulos, V. Z., and Papazafiriou, Z. G. Solutions of One-Dimensional Water Flow and Mass Transport Equations in Variably Saturated Porous Media by the Finite Element Method. Journal of Hydrology. 119 (1990) : 151-167.

25. Li, C. W. A Simplified Newton Iteation Method with Linear Finite Elements for Transient Unsaturated Flow. Water Resource Research. No.4, 29 (1993) : 965-971.

26. Celia, M. A., Ahuja, L. R., and Pinder, G. F. Orthogonal Collocation and Alternating-Direction Procedures for Unsaturated Flow Problems. Advances in Water Resources. 10 (1987) : 178-187.

27. Warrick, A. W. Simulataneous Solute and Water Transfer for an Unsaturated Soil. Water Resource Research. No.5, 7 (1971) : 1216-1225.

28. Chatkul Na Ayudhya, W. Brick Architechture: A Case Study of Restoration of the Chedi Wat Pansat. Proc. of the First Seminar on Thai-Japanese Cooperation in Conservation of Monuments in Thailand., Bangkok, March 8-9, 1999.

ภาคผนวก

ภาคผนวก ก

รายละเอียดของโปรแกรม RHBFE ในระบบแกนพิกัดฉาก

โปรแกรมคอมพิวเตอร์ RHBFE ที่ประดิษฐ์ขึ้นดังที่ได้กล่าวไว้ในบทที่ 4 มีรายละเอียดดังนี้

```
C
C    PROGRAM  RHBFE_XZ
C    A FINITE ELEMENT GROUNDWATER ANALYSIS PROGRAM FOR TRANSIENT
C    TWO-DIMENSIONAL MOISTURE MOVEMENT IN SATURATED-UNSATURATED SOILS
C    HEAD-BASED RICHARDS' EQUATION
C
C    MXPOI  = MAXIMUM NUMBER OF NODES IN THE MODEL
C    MXELE  = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C    MXFLUX = MAXIMUM NUMBER OF SPECIFIED VELOCITIES BOUNDARY
C
     PARAMETER (MXPOI=1809, MXELE=3200, MXFLUX=2)
C
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION  COORD(MXPOI,2), FLUX(MXPOI), TEXT(20)
     DIMENSION  PHEAD(MXPOI)
     DIMENSION  PHEAD_P(MXPOI), PHEAD_N(MXPOI), PHEAD_O(MXPOI)
     DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
     DIMENSION  QQQ(MXPOI)
     DIMENSION  QELE(MXELE), HK(MXELE), SLOPE(MXELE)
     DIMENSION  EXTMAT(MXELE,2), XVELO(MXELE), ZVELO(MXELE)
C
     CHARACTER*30  NAME1, NAME2
     CHARACTER*10  WAN, WAELA
     CHARACTER*1   CH1, CH2, CH3, CM2, CM3, CS2, CS3
C
     INTEGER  INTMAT(MXELE,3), IBC(MXPOI), INTFLUX(MXFLUX,3)
C
C    OPEN INPUT FILE:
C
 10  WRITE(6,20)
 20  FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
     READ(5, '(A)', ERR=10)  NAME1
     OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C    OPEN CHECKING FILE:
C
     OPEN(UNIT=9, FILE='CHECK.OUT', STATUS='UNKNOWN')
     OPEN(UNIT=10, FILE='CONV.OUT', STATUS='UNKNOWN')
C
C*************************************************************
C                      READ INPUT DATA
C*************************************************************
C
     READ(7,*)  NLINES
     DO 50  ILINE=1,NLINES
     READ(7,1)  TEXT
  1  FORMAT(20A4)
 50  CONTINUE
     READ(7,*)  NPOIN, NELEM   ,NFLUX, TOL
     IF(NPOIN.GT.MXPOI)   WRITE(6,100)  NPOIN
100  FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
     IF(NPOIN.GT.MXPOI)   STOP
```

```
      IF(NELEM.GT.MXELE)   WRITE(6,105)  NELEM
  105 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)   STOP
      IF(NFLUX.GT.MXFLUX)  WRITE(6,110)  NFLUX
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFLUX TO ', I5)
      IF(NFLUX.GT.MXFLUX)  STOP
C
C     READ MATERIAL PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  THICK, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R
      READ(7,1)  TEXT
      READ(7,*)  ST_PH, MITER, NSAVE
C
C     READ TRANSIENT PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  DT, NSTEP, TWF
C
C     READ NODAL COORDINATIONS, BOUNDARY CONDITIONS:
C
      READ(7,1)  TEXT
      DO 120  IP=1,NPOIN
      READ(7,*)  I, IBC(I), (COORD(I,K), K=1,2), PHEAD(I), FLUX(I)
      IF(I.NE.IP)  WRITE(6,115)  IP
  115 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
  120 CONTINUE
C
      IQ = 0
C
C     READ ELEMENT NODAL CONNECTIONS:
C
      READ(7,1)  TEXT
      DO 130  IE=1,NELEM
      READ(7,*)  I, (INTMAT(I,J), J=1,3), QELE(I)
      IF(I.NE.IE)  WRITE(6,125)  IE
  125 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IE)  STOP
      IF(QELE(I).NE.0)  IQ = 1
  130 CONTINUE
C
C     READ FLUX-BOUNDARY CONDITION ON NODES:
C
      IF(NFLUX.NE.0) THEN
      READ(7,1)TEXT
      DO 140 IB=1,NFLUX
      READ(7,*)  (INTFLUX(IB,J), J=1,3)
      IF(INTFLUX(IB,1).NE.IB)  WRITE(6,135)  IB
  135 FORMAT(/, ' SURFACE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(INTFLUX(IB,1).NE.IB)  STOP
  140 CONTINUE
      ENDIF
C
      WRITE(6,150)
  150 FORMAT(/,' THE F.E. MODEL INCLUDES THE FOLLOWING',
     *          ' MOISTURE TRANSFER MODE(S):',
     *        /,'      --  MOISTURE CONDUCTION                ')
      IF(IQ.EQ.1)  WRITE(6,160)
  160 FORMAT(  '      --  INTERNAL MOISTURE GENERATION        ')
      WRITE(6,170)  NPOIN, NELEM
```

```
170  FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I6,
     *          ' NODES AND', I6,' ELEMENTS ***                 ')
C
C    INITIAL NODE PRESSURE HEAD:
C
     CALL INITIAL (NPOIN, NELEM, IBC, PHEAD, INTMAT, HK, SLOPE,
     *              COORD, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R,
     *              ST_PH, MXPOI,MXELE                          )
C
     LENG = LEN_TRIM(NAME1) - 3
     OPEN(UNIT=8,FILE=NAME1(1:LENG)//'.out',STATUS='UNKNOWN')
C
C    ************************************************************
C                           TRANSIENT LOOP
C    ************************************************************
C
     RR = 0.
     SS = 0.
     SM = 0.
     SH = 0.
C
     DO 190 J=1,NPOIN
     PHEAD_P(J) = PHEAD(J)
190  CONTINUE
C
     DO 5000 NCOUNT=1,NSTEP
C
     RR = RR + DT
C
     DO 200 K=1,NPOIN
     PHEAD_O(K) = 0.
     QQQ(K) = 0.
200  CONTINUE
C
C    ************************************************************
C                           MAIN ITERATION
C    ************************************************************
C
     DO 700 LOOP=1,MITER
     WRITE(6,380) LOOP
380  FORMAT(/,10X,' *** ITERATION NO.',I5,' ***')
C
     NON   = NPOIN
     NOBP  = NFLUX
     NOE   = NELEM
     NEQ   = NPOIN
C
     DO 400 L=1,NON
     SYSC(L)    = 0.
     PHEAD_N(L) = 0.
     DO 390 M=1,NON
     SYSQ(L,M)  = 0.
     SYSK(L,M)  = 0.
390  CONTINUE
400  CONTINUE
C
     DO 410 N=1,NOE
     XVELO(N) = 0.
     ZVELO(N) = 0.
410  CONTINUE
C
```

```
      SS = MOD(RR,60.)
      SH = INT(RR/3600.)
      SM = INT(RR/60.) - SH*60
C
C     ESTABLISH ALL ELEMENT MATRICES ASSOCIATED WITH THE SPECIFIED
C     MOISTURE TRANSFER MODES AND ASSEMBLE THEM FOR SYSTEM MATRICES
C     IN THE FORM NEEDED FOR MINIMUM MEMORY REQUIREMENT:
C
      WRITE(6,420) SH, SM, SS
420   FORMAT(/,' *** TIME =',X,F3.0,X,'Hr.',X,F3.0,X,'Min.'
     *         ,X,F8.3,2X,'Sec.',X,' ***'                    )
C
      WRITE(6,430)
430   FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *          ' ASSEMBLING ELEMENT EQUATIONS ***'      )

      CALL TRI(NELEM, INTMAT, COORD,    HK,  QELE, THICK,
     *           SYSK,   SYSC,  SYSQ, SLOPE, MXPOI, MXELE)
C
      IF (NOBP.NE.0) THEN
      WRITE(6,440)
440   FORMAT(/,' *** APPLYING SURFACE BOUNDARY CONDITIONS',
     *          ' OF NODAL VELOCITIES ***'                  )
      CALL VELBOUND(NOBP, INTFLUX,  COORD,   FLUX, THICK,
     *              SYSQ,   MXPOI, MXFLUX, NCOUNT       )
      ENDIF
C
      WRITE(6,450)
450   FORMAT(/,' *** GENERATE TRANSIENT MATRICES ***')
      CALL TRANSIENT(NON, SYSC,  SYSK,  SYSQ, PHEAD_P,
     *               QQQ,  TWF,    DT, MXPOI         )
C
      WRITE(6,460)
460   FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS OF NODAL',
     *          ' PRESSURE HEAD ***'                       )
      CALL APPLYBC(NPOIN, IBC, PHEAD, SYSK, SYSQ, MXPOI)
C
      WRITE(6,470)
470   FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
     *          ' FOR PRESSURE HEAD SOLUTIONS ***'          )
      WRITE(6,480)  NEQ
480   FORMAT(5X,'( TOTAL OF', I6,' EQUATIONS TO BE SOLVED )')
      CALL SOLVEPDP(NEQ, SYSK, SYSQ, PHEAD_N, MXPOI)
C
      WRITE(6,490)
490   FORMAT(/,' *** USE DARCY LAW TO FIND OUT VELOCITIES ***')
      CALL DARCY(  NOE, INTMAT, COORD,    HK, PHEAD_N,
     *           EXTMAT,  XVELO, ZVELO, MXPOI, MXELE  )
C
C     COMPUTE  NEW  HYDRAULIC  CONDUCTITY  AND  MOISTURE-PRESSURE  HEAD
SLOPE:
C
      WRITE(6,500)
500   FORMAT(/,' *** SOLVING FOR THE NEW SET OF HYDRAULIC ',
     *          'CONDUCTIVITY AND MOISTURE-PRESSURE HEAD SLOPE ***')
      CALL ELEHK (NOE, INTMAT,  PHEAD_N,     HK, SLOPE, HKSAT, ALPHA,
     *              EM,     EN,   ZETA_S, ZETA_R, MXPOI, MXELE       )
C
C     CHECK FOR CONVERGENCE:
C
      IF (LOOP.EQ.1)  GOTO 600
```

```
      WRITE(6,510)
  510 FORMAT(/,' *** CHECKING FOR CONVERGENCE ***')
C
      CALL CONVERGE(NON, PHEAD_N, PHEAD_O, TOLPH, MXPOI)
C
C     PRINT OUT TOLERANCE:
C
      IF (LOOP.EQ.2) THEN
      WRITE(10,520)
  520 FORMAT(/,16X,'*****************************',/,
     *          16X,'*        SEEP V.I           *',/,
     *          16X,'* PRESSURE HEAD TOLERANCE *',/,
     *          16X,'*****************************'  )
      WRITE(10,530)
  530 FORMAT(/,5X,'LOOP',5X,'PRESSURE HEAD TOLERANCE')
      ENDIF
C
      IF ((MOD(NCOUNT,NSAVE).EQ.0).OR.(NCOUNT.EQ.NSTEP)) THEN
      WRITE(10,540)  LOOP, TOLPH
  540 FORMAT(5X,I3,8X,F12.4)
      ENDIF
      WRITE(6,550)  TOLPH
  550 FORMAT(/,' PRESSURE HEAD TOLERANCE  =',F12.4,X,'%')
C
C     SET TOLERANCE IN PERCENT:
C
      IF (TOLPH.LE.TOL) THEN
      WRITE(10,560)  LOOP, TOLPH
  560 FORMAT(5X,I3,8X,F12.4)
      GOTO 750
      ENDIF
C
  600 CONTINUE
C
C     NSAVE RESULTS HISTORY:
C
      DO 650 IH=1,NPOIN
      PHEAD_O(IH) = PHEAD_N(IH)
  650 CONTINUE
C
  700 CONTINUE
C
      WRITE(6,710)
  710 FORMAT(/,'***** SOLUTIONS DO NOT CONVERGE *****')
C
C     ************************************************************
C                        END OF MAIN ITERATION
C     ************************************************************
C
  750 CONTINUE
C
      DO 800 K=1,NPOIN
      PHEAD_P(K) = PHEAD_N(K)
  800 CONTINUE
C
      WRITE(6,810) SH, SM, SS
  810 FORMAT(/,' COMPLETE! TIME STEP AT #',X,F3.0,X,'Hr.',X,F3.0,X,
     *          'Min.',X,F8.3,2X,'Sec.',X,' ***')
C
      IF (NCOUNT.EQ.1) THEN
      WRITE(8,830)
```

```
830  FORMAT(/,12X,'************************************',/,
  *          12X,'*            SEEP V.1              *',/,
  *          12X,'* PRESSURE HEAD IN EACH TIME STEP *',/,
  *          12X,'************************************'  )
     ENDIF
C
     IF ((MOD(NCOUNT,NSAVE).EQ.0).OR.(NCOUNT.EQ.1).OR.
  *      (NCOUNT.EQ.NSTEP)) THEN
     CALL DATE(WAN)
     CALL TIME(WAELA)
     WRITE(8,840) WAN, WAELA
840  FORMAT(/,8X,A10,3X,A10)
     WRITE(8,850) SH, SM, SS
850  FORMAT(/,' *** TIME =',X,F3.0,X,'Hr.',X,F3.0,X,'Min.'
  *         ,X,F8.3,2X,'Sec.',X,' ***',/,3X,'NODE',9X,'X-COORD'
  *         ,12X,'Y-COORD',9X,'PRESSURE HEAD'                  )
C
     DO 870 IP=1,NPOIN
     WRITE(8,860) IP, COORD(IP,1), COORD(IP,2), PHEAD_P(IP)
860  FORMAT(I6,3X,E16.6,3X,E16.6,3X,E16.6)
870  CONTINUE
C
     ENDIF
C
C    PRINT OUT NODAL PRESSURE HEAD SOLUTIONS IN NASTRAN FORM:
C
     FREQ = MOD(NCOUNT,NSAVE)
C
     IF ((FREQ.EQ.0).OR.(NCOUNT.EQ.1).OR.(NCOUNT.EQ.NSTEP)) THEN
C
     TYME = NCOUNT*DT
     NH   = INT(TYME/3600)
     NM   = INT(TYME/60) - NH*60
     NS   = INT(MOD(TYME,60.))
C
     NH1 = INT(NH/100)
     CH1 = CHAR(NH1 + 48)
     NH  = NH - NH1*100
     NH2 = INT(NH/10)
     CH2 = CHAR(NH2 + 48)
     NM2 = INT(NM/10)
     CM2 = CHAR(NM2 + 48)
     NS2 = INT(NS/10)
     CS2 = CHAR(NS2 + 48)
     NH  = NH - NH2*10
     NM  = NM - NM2*10
     NS  = NS - NS2*10
     NH3 = MOD(NH,10)
     CH3 = CHAR(NH3 + 48)
     NM3 = MOD(NM,10)
     CM3 = CHAR(NM3 + 48)
     NS3 = MOD(NS,10)
     CS3 = CHAR(NS3 + 48)
C
C    NASTRAN OUTPUT:
C
     OPEN(UNIT=11,FILE=NAME1(1:LENG)//'_'//CH1//CH2//CH3//'h'//CM2//
  *       CM3//'m'//CS2//CS3//'s'//'.F06',STATUS='UNKNOWN'          )
     WRITE(11,900)
900  FORMAT('1MSC/NASTRAN PAGE')
     WRITE(11,910)
```

```
910  FORMAT(/,'0')
     WRITE(11,920)
920  FORMAT('  D I S P L A C E M E N T')
     DO 940  IP=1,NPOIN
     WRITE(11,930) IP, PHEAD_P(IP)
930  FORMAT(I6,2X,'G',3X,E16.6,3X,'.000000E+00',3X,'.000000E+00',
   *        3X,'.000000E+00',3X,'.000000E+00',3X,'.000000E+00'  )
940  CONTINUE
     WRITE(11,950)
950  FORMAT(' 0')
C
C    TECPLOT OUTPUT:
C
     OPEN(UNIT=12,FILE=NAME1(1:LENG)//'_'//CH1//CH2//CH3//'h'//CM2//
   *       CM3//'m'//CS2//CS3//'s'//'.PLT',STATUS='UNKNOWN'          )
     WRITE(12,952) NPOIN, NELEM
952  FORMAT('VARIABLES = "NODE","X-COOR","Y-COOR","PHEAD"', /
   *        ,'ZONE N=',I6,',E=',I6,',F=FEPOINT,ET=TRIANGLE'  )
     DO 960 IP=1,NPOIN
     WRITE(12,955) IP, COORD(IP,1), COORD(IP,2), PHEAD_P(IP)
955  FORMAT(I6,2F12.6,E16.6)
960  CONTINUE
     DO 965 IE=1,NELEM
     WRITE(12,962) (INTMAT(IE,J),J=1,3)
962  FORMAT(3I6)
965  CONTINUE
C
     ENDIF
C
C    ************************************************************
C                        END OF TRANSIENT LOOP
C    ************************************************************
C
5000 CONTINUE
C
970  WRITE(6,975)
975  FORMAT(/, ' PLEASE ENTER FILE NAME FOR VELOCITIES'
   *           ' OR TYPE -PASS- TO IGNORE'             )
     READ(5, '(A)', ERR=970)  NAME2
     IF (NAME2.EQ.'p')  STOP
     IF (NAME2.EQ.'P')  STOP
     IF (NAME2.EQ.'pass')  STOP
     IF (NAME2.EQ.'PASS')  STOP
     OPEN(UNIT=14, FILE=NAME2, STATUS='NEW', ERR=970)
     WRITE(14,980)
980  FORMAT(/, ' ELE-NO.',6X,'XXX',13X,'YYY',12X,'X-VEL',11X,'Y-VEL')
     DO 990 IE=1,NELEM
     WRITE(14,985) IE,EXTMAT(IE,1),EXTMAT(IE,2),XVELO(IE),ZVELO(IE)
985  FORMAT(I5,4E16.6)
990  CONTINUE
C
     STOP
     END
!
!---------------------------------------------------------------------
!
     SUBROUTINE APPLYBC(NPOIN, IBC, PHEAD, SYSK, SYSQ, MXPOI)
C
C    APPLY PRESSURE HEAD BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C      0 = FREE TO CHANGE (TO BE COMPUTED)
C      1 = FIXED AS SPECIFIED
```

```
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), PHEAD(MXPOI)
C
      INTEGER    IBC(MXPOI)
C
      DO 100  IEQ=1,NPOIN
      IF(IBC(IEQ).EQ.0)  GO TO 100
C
      DO 200  IR=1,NPOIN
      IF(IR.EQ.IEQ)  GO TO 200
      SYSQ(IR) = SYSQ(IR) - SYSK(IR,IEQ)*PHEAD(IEQ)
      SYSK(IR,IEQ) = 0.
200   CONTINUE
C
      DO 300  IC=1,NPOIN
      SYSK(IEQ,IC) = 0.
300   CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSQ(IEQ) = PHEAD(IEQ)
C
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE ASSMBLE(   IE, INTMAT,    AKC,     QQ, SYSK,
     *                     SYSC,   SYSQ,    ACC, MXPOI, MXELE)
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  AKC(3,3), QQ(3), ACC(3,3)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
C
      INTEGER    INTMAT(MXELE,3)
C
      NNODE = 3
C
      DO 100  IR=1,NNODE
      DO 200  IC=1,NNODE
      IROW = INTMAT(IE,IR)
      ICOL = INTMAT(IE,IC)
      SYSK(IROW,ICOL) = SYSK(IROW,ICOL) + AKC(IR,IC)
      SYSC(IROW,ICOL) = SYSC(IROW,ICOL) + ACC(IR,IC)
200   CONTINUE
      SYSQ(IROW) = SYSQ(IROW) + QQ(IR)
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE SOLVEPDP(N, A, B, X, MXPOI, NHBW)
      IMPLICIT   REAL*8(A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI), X(MXPOI)
C
      CALL SCALE(N, A, B, MXPOI)
```

```
C
C    FORWARD ELEMINATION:
C
     DO 100 IP=1,N-1
C
     CALL PIVOT(N, A, B, MXPOI, IP)
C
     DO 200 IE=IP+1,N
     RATIO = A(IE,IP)/A(IP,IP)
C
     DO 300 IC=IP+1,N
     A(IE,IC) = A(IE,IC) - RATIO*A(IP,IC)
300  CONTINUE
     B(IE) = B(IE) - RATIO*B(IP)
200  CONTINUE
C
     DO 400 IE=IP+1,N
     A(IE,IP) = 0.
400  CONTINUE
100  CONTINUE
C
     X(N) = B(N)/A(N,N)
C
     DO 500 IE=N-1,1,-1
     SUM = 0.
     DO 600 IC=IE+1,N
     SUM = SUM + A(IE,IC)*X(IC)
600  CONTINUE
     X(IE) = (B(IE) - SUM)/A(IE,IE)
500  CONTINUE
C
     RETURN
     END
!
!---------------------------------------------------------------------
!
     SUBROUTINE PIVOT(N, A, B, MXPOI, IP)
     IMPLICIT   REAL*8(A-H,O-Z)
     DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
     JP = IP
     BIG = ABS(A(IP,IP))
     DO 10 I=IP+1,N
     IF(AMAX.GT.BIG) THEN
     BIG = AMAX
     JP = I
     ENDIF
 10  CONTINUE
     IF(JP.NE.IP) THEN
     DO 20 J=IP,N
     DUMY    = A(JP,J)
     A(JP,J) = A(IP,J)
     A(IP,J) = DUMY
20   CONTINUE
     DUMY  = B(JP)
     B(JP) = B(IP)
     B(IP) = DUMY
     ENDIF
     RETURN
     END
C
```

```
C-----------------------------------------------------------------------
C
      SUBROUTINE SCALE(N, A, B, MXPOI)
      IMPLICIT   REAL*8(A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
      DO 10 IE=1,N
      BIG = ABS(A(IE,1))
      DO 20 IC=2,N
      AMAX = ABS(A(IE,IC))
      IF (AMAX.GT.BIG) BIG = AMAX
 20   CONTINUE
      DO 30 IC=1,N
      A(IE,IC) = A(IE,IC)/BIG
 30   CONTINUE
      B(IE) = B(IE)/BIG
 10   CONTINUE
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE TRI(NELEM, INTMAT, COORD,    HK,  QELE, THICK,
     *                SYSK,   SYSC,  SYSQ, SLOPE, MXPOI, MXELE)
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     UP SYSTEM EQUATIONS
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), SYSK(MXPOI,MXPOI), SYSQ(MXPOI)
      DIMENSION  SYSC(MXPOI,MXPOI)
      DIMENSION  QELE(MXELE), HK(MXELE) ,SLOPE(MXELE)
      DIMENSION  QG(3)
      DIMENSION  AKC(3,3), QQ(3), B(2,3), BT(3,2), ACC(3,3)
C
      INTEGER    INTMAT(MXELE,3)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II  = INTMAT(IE,1)
      JJ  = INTMAT(IE,2)
      KK  = INTMAT(IE,3)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      AREA= 0.5*(XG2*(YG3-YG1) + XG1*(YG2-YG3) + XG3*(YG1-YG2))
C
      IF(AREA.LE.0.)  WRITE(6,5)  IE
   5  FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *          ' HAS NEGATIVE OR ZERO AREA ', /,
     *          '  --- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *          ' AND ELEMENT NODAL CONNECTIONS ---'          )
      IF(AREA.LE.0.)  STOP
```

```
C
      B1 = YG2 - YG3
      B2 = YG3 - YG1
      B3 = YG1 - YG2
      C1 = XG3 - XG2
      C2 = XG1 - XG3
      C3 = XG2 - XG1
C
      DO 10  I=1,2
      DO 10  J=1,3
      B(I,J)  = 0.
 10   CONTINUE
C
      B(1,1)  = B1
      B(1,2)  = B2
      B(1,3)  = B3
      B(2,1)  = C1
      B(2,2)  = C2
      B(2,3)  = C3
C
      DO 20  I=1,2
      DO 30  J=1,3
      B(I,J)  = B(I,J)/(2.*AREA)
      BT(J,I) = B(I,J)
 30   CONTINUE
 20   CONTINUE
C
C     ELEMENT CAPACITANCE MATRIX (LUMP):
C
      DO 50 I=1,3
      DO 50 J=1,3
      IF(I.EQ.J)  ACC(I,J) = 1.
      IF(I.NE.J)  ACC(I,J) = 0.
      ACC(I,J) = SLOPE(IE)*AREA*THICK*ACC(I,J)/3.
 50   CONTINUE
C
C     ELEMENT CONDUCTION MATRIX:
C
      DO 100  I=1,3
      DO 100  J=1,3
      AKC(I,J) = 0.
      DO 90   K=1,2
      AKC(I,J) = AKC(I,J) + BT(I,K)*B(K,J)
 90   CONTINUE
      AKC(I,J) = HK(IE)*AREA*THICK*AKC(I,J)
100   CONTINUE
C
C     ELEVATION TERM POSSITIVE IN UPWARD VERTICAL DIRECTION
C
      QQ(1) = -1*HK(IE)*THICK*C1/2.
      QQ(2) = -1*HK(IE)*THICK*C2/2.
      QQ(3) = -1*HK(IE)*THICK*C3/2.
C
C     ELEMENT HEAT LOAD DUE TO INTERNAL MOISTURE GENERATION:
C     <QELE> POSSITIVE FOR SOURCE TERM
C
      FAC = QELE(IE)*AREA*THICK/3.
      DO 110  I=1,3
      QG(I) = FAC
110   CONTINUE
      DO 200  I=1,3
```

```
      QQ(I) = QQ(I) + QG(I)
200   CONTINUE
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT,    AKC,     QQ,   SYSK,
     *               SYSC,   SYSQ,    ACC, MXPOI, MXELE)
C
500   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE DARCY (NOE, INTMAT, COORD,     HK, PHEAD_N,
     *                  EXTMAT,  XVELO, ZVELO, MXPOI, MXELE  )
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), XVELO(MXPOI), ZVELO(MXPOI)
      DIMENSION  PHEAD_N(MXPOI)
      DIMENSION  HK(MXELE)
      DIMENSION  EXTMAT(MXELE,2)
C
      INTEGER    INTMAT(MXELE,3)
C
      DO 100 IE=1,NOE
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
C
      XN1  = COORD(II,1)
      XN2  = COORD(JJ,1)
      XN3  = COORD(KK,1)
      ZN1  = COORD(II,2)
      ZN2  = COORD(JJ,2)
      ZN3  = COORD(KK,2)
      AREA = 0.5*(XN2*(ZN3-ZN1) + XN1*(ZN2-ZN3) + XN3*(ZN1-ZN2))
C
      XELE = (XN1 + XN2 + XN3)/3.
      ZELE = (ZN1 + ZN2 + ZN3)/3.
      EXTMAT(IE,1) = XELE
      EXTMAT(IE,2) = ZELE
C
      BT1 = (ZN2 - ZN3)*PHEAD_N(II)
      BT2 = (ZN3 - ZN1)*PHEAD_N(JJ)
      BT3 = (ZN1 - ZN2)*PHEAD_N(KK)
      CT1 = (XN3 - XN2)*PHEAD_N(II)
      CT2 = (XN1 - XN3)*PHEAD_N(JJ)
      CT3 = (XN2 - XN1)*PHEAD_N(KK)
C
      XVELO(IE) = -1*HK(IE)*(BT1 + BT2 + BT3)/(2.*AREA)
      ZVELO(IE) = -1*HK(IE)*(CT1 + CT2 + CT3)/(2.*AREA) - HK(IE)
C
100   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
```

```
      SUBROUTINE VELBOUND (NOBP, INTFLUX, COORD, FLUX, THICK,
     *                     SYSQ,  MXPOI, MXFLUX, NCOUNT     )
C
C     <FLUX> POSSITIVE FOR SOURCE TERM
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), FLUX(MXPOI), SYSQ(MXPOI)
C
      INTEGER    INTFLUX(MXFLUX,3)
C
C     *************************
C     FOR RUNNING BOUNDARY FLUX
C     *************************
C
      IF (MOD(NCOUNT,250).EQ.0) THEN
C
      IA = INTFLUX(1,2)
C
      DO 50 II=1,NOBP
      JI = INTFLUX(II,2)
      KI = INTFLUX(II,3)
C
      INTFLUX(II,2) = JI + 1
      INTFLUX(II,3) = KI + 1
C
      FLUX(JI + 1)  = FLUX(JI)
      FLUX(KI + 1)  = FLUX(KI)
C
 50   CONTINUE
      FLUX(IA) = 0.
C
      ENDIF
C
      DO 100 IE=1,NOBP
      JJ = INTFLUX(IE,2)
      KK = INTFLUX(IE,3)
C
      X1 = COORD(JJ,1)
      X2 = COORD(KK,1)
      Y1 = COORD(JJ,2)
      Y2 = COORD(KK,2)
      DX = X2 - X1
      DY = Y2 - Y1
      DL = SQRT(DX*DX + DY*DY)
C
      Q1 = FLUX(JJ)*THICK*DL/2.
      Q2 = FLUX(KK)*THICK*DL/2.
C
      SYSQ(JJ) = SYSQ(JJ) + Q1
      SYSQ(KK) = SYSQ(KK) + Q2
C
100   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------
!
      SUBROUTINE ELEHK (NOE, INTMAT, PHEAD_N, HK, SLOPE, HKSAT, ALPHA,
     *                       EM,      EN,   ZETA_S,     ZETA_R, MXPOI, MXELE)
C
```

```
      IMPLICIT    REAL*8 (A-H,O-Z)
      DIMENSION   PHEAD_N(MXPOI)
      DIMENSION   HK(MXELE), SLOPE(MXELE)
C
      INTEGER     INTMAT(MXELE,3)
C
      DO 100 I=1,NOE
      II = INTMAT (I,1)
      JJ = INTMAT (I,2)
      KK = INTMAT (I,3)
C
C     AVERAGE ELEMENT PRESSURE HEAD
C
      AVEPH = (PHEAD_N(II) + PHEAD_N(JJ) + PHEAD_N(KK))/3.
C
C     ELEMENT HYDRAULIC CONDUCTIVITY
C     OR MEANS HEAT CONDUCTION COEFF.IN HEAT TRANFER PROBLEM
C
      HK(I) = HKSAT*1.175E6/(1.175E6 + ABS(AVEPH)**4.74)
C
C     ELEMENT MOISTURE-PRESSURE HEAD SLOPE
C     OR MEANS DENSITY*SPECIFIC HEAT(P) IN HEAT TRANFER PROBLEM
C
      SLOPE(I) = -1*1.611E6*(0.287-0.075)*3.96*AVEPH*
C
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE CONVERGE (NON, PHEAD_N, PHEAD_O, TOLPH, MXPOI)
C
      IMPLICIT    REAL*8 (A-H,O-Z)
      DIMENSION   PHEAD_N(MXPOI), PHEAD_O(MXPOI)
C
      AAA = 0.
      BBB = 0.
C
      DO 100 I=1,NON
      AA  = PHEAD_N(I)
      AAA = AAA + AA
      BB  = PHEAD_O(I)
      BBB = BBB + BB
C
100   CONTINUE
C
      WRITE(9,50) BBB
 50   FORMAT(/,F16.6)
C
      TOLPH = ABS((BBB - AAA)/AAA)*100
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE INITIAL (NPOIN, NELEM, IBC, PHEAD, INTMAT, HK, SLOPE,
     *                    COORD, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R,
     *                    ST_PH, MXPOI, MXELE                        )
```

```
C
      IMPLICIT    REAL*8 (A-H,O-Z)
      DIMENSION   COORD(MXPOI,2), PHEAD(MXPOI)
      DIMENSION   HK(MXELE), SLOPE(MXELE)
C
      INTEGER     INTMAT(MXELE,3)
      INTEGER     IBC(MXPOI)
C
      DO 100 IP=1,NPOIN
      IF (IBC(IP).EQ.1) GOTO 100
C
      PHEAD(IP) = ST_PH
C
      CALL ELEHK (NOE, INTMAT,    PHEAD, HK, SLOPE, HKSAT, ALPHA,
     *              EM,     EN,   ZETA_S,    ZETA_R, MXPOI, MXELE )
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE TRANSIENT (NON, SYSC, SYSK,  SYSQ, PHEAD_P,
     *                       QQQ,  TWF,   DT, MXPOI          )
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
      DIMENSION PHEAD_P(MXPOI), QQQ(MXPOI)
C
C     ************************************************************
C                          FOR STEADY STATE
C     ************************************************************
C
CC    WRITE(6,50)
CC    50      FORMAT(/,' *** STEADY STATE ***')
C
C     ************************************************************
C                           FOR TRANSIENT
C     ************************************************************
C
      WRITE(6,100)
100   FORMAT(/,' *** TRANSIENT ***')
C
      DO 200 K=1,NON
      QQQ(K) = SYSQ(K)
200   CONTINUE
C
      DO 500 I=1,NON
      SYSQ(I) = (1-TWF)*SYSQ(I) + TWF*QQQ(I)
      DO 500 J=1,NON
      SYSQ(I) = SYSQ(I) +
     *          (SYSC(I,J)/DT - (1-TWF)*SYSK(I,J))*PHEAD_P(J)
      SYSK(I,J) = SYSC(I,J)/DT + SYSK(I,J)*TWF
500   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
```

ภาคผนวก ข

รายละเอียดของโปรแกรม RHBFE ในระบบแกนพิกัดทรงกระบอก

โปรแกรมคอมพิวเตอร์ RHBFE ที่ประดิษฐ์ขึ้นดังที่ได้กล่าวไว้ในบทที่ 4 มีรายละเอียดดังนี้

```
C
C    PROGRAM  RHBFE_RZ
C    A FINITE ELEMENT GROUNDWATER ANALYSIS PROGRAM FOR TRANSIENT
C    TWO-DIMENSIONAL MOISTURE MOVEMENT IN SATURATED-UNSATURATED SOILS
C    HEAD-BASED RICHARDS' EQUATION
C
C    MXPOI  = MAXIMUM NUMBER OF NODES IN THE MODEL
C    MXELE  = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C    MXFLUX = MAXIMUM NUMBER OF SPECIFIED VELOCITIES BOUNDARY
C
     PARAMETER (MXPOI=1809, MXELE=3200, MXFLUX=2)
C
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION  COORD(MXPOI,2), FLUX(MXPOI), TEXT(20)
     DIMENSION  PHEAD(MXPOI)
     DIMENSION  PHEAD_P(MXPOI), PHEAD_N(MXPOI), PHEAD_O(MXPOI)
     DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
     DIMENSION  QQQ(MXPOI)
     DIMENSION  QELE(MXELE), HK(MXELE), SLOPE(MXELE)
     DIMENSION  EXTMAT(MXELE,2), XVELO(MXELE), ZVELO(MXELE)
C
     CHARACTER*30  NAME1, NAME2
     CHARACTER*10  WAN, WAELA
     CHARACTER*1   CH1, CH2, CH3, CM2, CM3, CS2, CS3
C
     INTEGER  INTMAT(MXELE,3), IBC(MXPOI), INTFLUX(MXFLUX,3)
C
C    OPEN INPUT FILE:
C
 10  WRITE(6,20)
 20  FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
     READ(5, '(A)', ERR=10)  NAME1
     OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C    OPEN CHECKING FILE:
C
     OPEN(UNIT=9, FILE='CHECK.OUT', STATUS='UNKNOWN')
     OPEN(UNIT=10, FILE='CONV.OUT', STATUS='UNKNOWN')
C
C*************************************************************
C                     READ INPUT DATA
C*************************************************************
C
     READ(7,*)  NLINES
     DO 50  ILINE=1,NLINES
     READ(7,1)  TEXT
  1  FORMAT(20A4)
 50  CONTINUE
     READ(7,*)  NPOIN, NELEM   ,NFLUX, TOL
     IF(NPOIN.GT.MXPOI)   WRITE(6,100)  NPOIN
100  FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
     IF(NPOIN.GT.MXPOI)   STOP
```

```
      IF(NELEM.GT.MXELE)   WRITE(6,105)  NELEM
  105 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)   STOP
      IF(NFLUX.GT.MXFLUX)  WRITE(6,110)  NFLUX
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFLUX TO ', I5)
      IF(NFLUX.GT.MXFLUX)  STOP
C
C     READ MATERIAL PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  THICK, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R
      READ(7,1)  TEXT
      READ(7,*)  ST_PH, MITER, NSAVE
C
C     READ TRANSIENT PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  DT, NSTEP, TWF
C
C     READ NODAL COORDINATIONS, BOUNDARY CONDITIONS:
C
      READ(7,1)  TEXT
      DO 120  IP=1,NPOIN
      READ(7,*)  I, IBC(I), (COORD(I,K), K=1,2), PHEAD(I), FLUX(I)
      IF(I.NE.IP)  WRITE(6,115)  IP
  115 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
  120 CONTINUE
C
      IQ = 0
C
C     READ ELEMENT NODAL CONNECTIONS:
C
      READ(7,1)  TEXT
      DO 130  IE=1,NELEM
      READ(7,*)  I, (INTMAT(I,J), J=1,3), QELE(I)
      IF(I.NE.IE)  WRITE(6,125)  IE
  125 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IE)  STOP
      IF(QELE(I).NE.0)  IQ = 1
  130 CONTINUE
C
C     READ FLUX-BOUNDARY CONDITION ON NODES:
C
      IF(NFLUX.NE.0) THEN
      READ(7,1)TEXT
      DO 140 IB=1,NFLUX
      READ(7,*)  (INTFLUX(IB,J), J=1,3)
      IF(INTFLUX(IB,1).NE.IB)  WRITE(6,135)  IB
  135 FORMAT(/, ' SURFACE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(INTFLUX(IB,1).NE.IB)  STOP
  140 CONTINUE
      ENDIF
C
      WRITE(6,150)
  150 FORMAT(/,' THE F.E. MODEL INCLUDES THE FOLLOWING',
     *          ' MOISTURE TRANSFER MODE(S):',
     *        /,'      --  MOISTURE CONDUCTION                ')
      IF(IQ.EQ.1)  WRITE(6,160)
  160 FORMAT(  '      --  INTERNAL MOISTURE GENERATION        ')
      WRITE(6,170)  NPOIN, NELEM
```

```
170  FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I6,
     *          ' NODES AND', I6,' ELEMENTS ***                 ')
C
C    INITIAL NODE PRESSURE HEAD:
C
     CALL INITIAL (NPOIN, NELEM, IBC, PHEAD, INTMAT, HK, SLOPE,
     *              COORD, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R,
     *              ST_PH, MXPOI,MXELE                          )
C
     LENG = LEN_TRIM(NAME1) - 3
     OPEN(UNIT=8,FILE=NAME1(1:LENG)//'.out',STATUS='UNKNOWN')
C
C    ************************************************************
C                           TRANSIENT LOOP
C    ************************************************************
C
     RR = 0.
     SS = 0.
     SM = 0.
     SH = 0.
C
     DO 190 J=1,NPOIN
     PHEAD_P(J) = PHEAD(J)
190  CONTINUE
C
     DO 5000 NCOUNT=1,NSTEP
C
     RR = RR + DT

     DO 200 K=1,NPOIN
     PHEAD_O(K) = 0.
     QQQ(K) = 0.
200  CONTINUE
C
C    ************************************************************
C                           MAIN ITERATION
C    ************************************************************
C
     DO 700 LOOP=1,MITER
     WRITE(6,380) LOOP
380  FORMAT(/,10X,' *** ITERATION NO.',I5,' ***')
C
     NON   = NPOIN
     NOBP  = NFLUX
     NOE   = NELEM
     NEQ   = NPOIN
C
     DO 400 L=1,NON
     SYSC(L)    = 0.
     PHEAD_N(L) = 0.
     DO 390 M=1,NON
     SYSQ(L,M)  = 0.
     SYSK(L,M)  = 0.
390  CONTINUE
400  CONTINUE
C
     DO 410 N=1,NOE
     XVELO(N) = 0.
     ZVELO(N) = 0.
410  CONTINUE
C
```

```
      SS = MOD(RR,60.)
      SH = INT(RR/3600.)
      SM = INT(RR/60.) - SH*60
C
C     ESTABLISH ALL ELEMENT MATRICES ASSOCIATED WITH THE SPECIFIED
C     MOISTURE TRANSFER MODES AND ASSEMBLE THEM FOR SYSTEM MATRICES
C     IN THE FORM NEEDED FOR MINIMUM MEMORY REQUIREMENT:
C
      WRITE(6,420) SH, SM, SS
420   FORMAT(/,' *** TIME =',X,F3.0,X,'Hr.',X,F3.0,X,'Min.'
     *         ,X,F8.3,2X,'Sec.',X,' ***'                    )
C
      WRITE(6,430)
430   FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *          ' ASSEMBLING ELEMENT EQUATIONS ***'     )

      CALL TRI(NELEM, INTMAT, COORD,    HK,  QELE, THICK,
     *           SYSK,   SYSC,  SYSQ, SLOPE, MXPOI, MXELE)
C
      IF (NOBP.NE.0) THEN
      WRITE(6,440)
440   FORMAT(/,' *** APPLYING SURFACE BOUNDARY CONDITIONS',
     *          ' OF NODAL VELOCITIES ***'                  )
      CALL VELBOUND(NOBP, INTFLUX,  COORD,   FLUX, THICK,
     *               SYSQ,   MXPOI, MXFLUX, NCOUNT       )
      ENDIF
C
      WRITE(6,450)
450   FORMAT(/,' *** GENERATE TRANSIENT MATRICES ***')
      CALL TRANSIENT(NON, SYSC,  SYSK,  SYSQ, PHEAD_P,
     *                QQQ,  TWF,    DT, MXPOI         )
C
      WRITE(6,460)
460   FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS OF NODAL',
     *          ' PRESSURE HEAD ***'                       )
      CALL APPLYBC(NPOIN, IBC, PHEAD, SYSK, SYSQ, MXPOI)
C
      WRITE(6,470)
470   FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
     *          ' FOR PRESSURE HEAD SOLUTIONS ***'          )
      WRITE(6,480)  NEQ
480   FORMAT(5X,'( TOTAL OF', I6,' EQUATIONS TO BE SOLVED )')
      CALL SOLVEPDP(NEQ, SYSK, SYSQ, PHEAD_N, MXPOI)
C
      WRITE(6,490)
490   FORMAT(/,' *** USE DARCY LAW TO FIND OUT VELOCITIES ***')
      CALL DARCY(  NOE, INTMAT, COORD,    HK, PHEAD_N,
     *           EXTMAT,  XVELO, ZVELO, MXPOI, MXELE  )
C
C     COMPUTE  NEW  HYDRAULIC  CONDUCTITY  AND  MOISTURE-PRESSURE  HEAD
SLOPE:
C
      WRITE(6,500)
500   FORMAT(/,' *** SOLVING FOR THE NEW SET OF HYDRAULIC ',
     *          'CONDUCTIVITY AND MOISTURE-PRESSURE HEAD SLOPE ***')
      CALL ELEHK (NOE, INTMAT,  PHEAD_N,     HK, SLOPE, HKSAT, ALPHA,
     *              EM,     EN,   ZETA_S, ZETA_R, MXPOI, MXELE       )
C
C     CHECK FOR CONVERGENCE:
C
      IF (LOOP.EQ.1)  GOTO 600
```

```
      WRITE(6,510)
  510 FORMAT(/,' *** CHECKING FOR CONVERGENCE ***')
C
      CALL CONVERGE(NON, PHEAD_N, PHEAD_O, TOLPH, MXPOI)
C
C     PRINT OUT TOLERANCE:
C
      IF (LOOP.EQ.2) THEN
      WRITE(10,520)
  520 FORMAT(/,16X,'*************************',/,
     *          16X,'*        SEEP V.I         *',/,
     *          16X,'* PRESSURE HEAD TOLERANCE *',/,
     *          16X,'*************************'  )
      WRITE(10,530)
  530 FORMAT(/,5X,'LOOP',5X,'PRESSURE HEAD TOLERANCE')
      ENDIF
C
      IF ((MOD(NCOUNT,NSAVE).EQ.0).OR.(NCOUNT.EQ.NSTEP)) THEN
      WRITE(10,540)  LOOP, TOLPH
  540 FORMAT(5X,I3,8X,F12.4)
      ENDIF
      WRITE(6,550)  TOLPH
  550 FORMAT(/,' PRESSURE HEAD TOLERANCE  =',F12.4,X,'%')
C
C     SET TOLERANCE IN PERCENT:
C
      IF (TOLPH.LE.TOL) THEN
      WRITE(10,560)  LOOP, TOLPH
  560 FORMAT(5X,I3,8X,F12.4)
      GOTO 750
      ENDIF
C
  600 CONTINUE
C
C     NSAVE RESULTS HISTORY:
C
      DO 650 IH=1,NPOIN
      PHEAD_O(IH) = PHEAD_N(IH)
  650 CONTINUE
C
  700 CONTINUE
C
      WRITE(6,710)
  710 FORMAT(/,'***** SOLUTIONS DO NOT CONVERGE *****')
C
C     ************************************************************
C                        END OF MAIN ITERATION
C     ************************************************************
C
  750 CONTINUE
C
      DO 800 K=1,NPOIN
      PHEAD_P(K) = PHEAD_N(K)
  800 CONTINUE
C
      WRITE(6,810) SH, SM, SS
  810 FORMAT(/,' COMPLETE! TIME STEP AT #',X,F3.0,X,'Hr.',X,F3.0,X,
     *          'Min.',X,F8.3,2X,'Sec.',X,' ***')
C
      IF (NCOUNT.EQ.1) THEN
      WRITE(8,830)
```

```
830  FORMAT(/,12X,'************************************',/,
  *          12X,'*            SEEP V.1              *',/,
  *          12X,'* PRESSURE HEAD IN EACH TIME STEP *',/,
  *          12X,'************************************'  )
     ENDIF
C
     IF ((MOD(NCOUNT,NSAVE).EQ.0).OR.(NCOUNT.EQ.1).OR.
  *      (NCOUNT.EQ.NSTEP)) THEN
     CALL DATE(WAN)
     CALL TIME(WAELA)
     WRITE(8,840) WAN, WAELA
840  FORMAT(/,8X,A10,3X,A10)
     WRITE(8,850) SH, SM, SS
850  FORMAT(/,' *** TIME =',X,F3.0,X,'Hr.',X,F3.0,X,'Min.'
  *         ,X,F8.3,2X,'Sec.',X,' ***',/,3X,'NODE',9X,'X-COORD'
  *         ,12X,'Y-COORD',9X,'PRESSURE HEAD'                  )
C
     DO 870 IP=1,NPOIN
     WRITE(8,860) IP, COORD(IP,1), COORD(IP,2), PHEAD_P(IP)
860  FORMAT(I6,3X,E16.6,3X,E16.6,3X,E16.6)
870  CONTINUE
C
     ENDIF
C
C    PRINT OUT NODAL PRESSURE HEAD SOLUTIONS IN NASTRAN FORM:
C
     FREQ = MOD(NCOUNT,NSAVE)
C
     IF ((FREQ.EQ.0).OR.(NCOUNT.EQ.1).OR.(NCOUNT.EQ.NSTEP)) THEN
C
     TYME = NCOUNT*DT
     NH   = INT(TYME/3600)
     NM   = INT(TYME/60) - NH*60
     NS   = INT(MOD(TYME,60.))
C
     NH1 = INT(NH/100)
     CH1 = CHAR(NH1 + 48)
     NH  = NH - NH1*100
     NH2 = INT(NH/10)
     CH2 = CHAR(NH2 + 48)
     NM2 = INT(NM/10)
     CM2 = CHAR(NM2 + 48)
     NS2 = INT(NS/10)
     CS2 = CHAR(NS2 + 48)
     NH  = NH - NH2*10
     NM  = NM - NM2*10
     NS  = NS - NS2*10
     NH3 = MOD(NH,10)
     CH3 = CHAR(NH3 + 48)
     NM3 = MOD(NM,10)
     CM3 = CHAR(NM3 + 48)
     NS3 = MOD(NS,10)
     CS3 = CHAR(NS3 + 48)
C
C    NASTRAN OUTPUT:
C
     OPEN(UNIT=11,FILE=NAME1(1:LENG)//'_'//CH1//CH2//CH3//'h'//CM2//
  *       CM3//'m'//CS2//CS3//'s'//'.F06',STATUS='UNKNOWN'          )
     WRITE(11,900)
900  FORMAT('1MSC/NASTRAN PAGE')
     WRITE(11,910)
```

```
910  FORMAT(/,'0')
     WRITE(11,920)
920  FORMAT('  D I S P L A C E M E N T')
     DO 940  IP=1,NPOIN
     WRITE(11,930) IP, PHEAD_P(IP)
930  FORMAT(I6,2X,'G',3X,E16.6,3X,'.000000E+00',3X,'.000000E+00',
   *       3X,'.000000E+00',3X,'.000000E+00',3X,'.000000E+00'  )
940  CONTINUE
     WRITE(11,950)
950  FORMAT(' 0')
C
C    TECPLOT OUTPUT:
C
     OPEN(UNIT=12,FILE=NAME1(1:LENG)//'_'//CH1//CH2//CH3//'h'//CM2//
   *      CM3//'m'//CS2//CS3//'s'//'.PLT',STATUS='UNKNOWN'          )
     WRITE(12,952) NPOIN, NELEM
952  FORMAT('VARIABLES = "NODE","X-COOR","Y-COOR","PHEAD"', /
   *       ,'ZONE N=',I6,',E=',I6,',F=FEPOINT,ET=TRIANGLE'  )
     DO 960 IP=1,NPOIN
     WRITE(12,955) IP, COORD(IP,1), COORD(IP,2), PHEAD_P(IP)
955  FORMAT(I6,2F12.6,E16.6)
960  CONTINUE
     DO 965 IE=1,NELEM
     WRITE(12,962) (INTMAT(IE,J),J=1,3)
962  FORMAT(3I6)
965  CONTINUE
C
     ENDIF
C
C    ************************************************************
C                       END OF TRANSIENT LOOP
C    ************************************************************
C
5000 CONTINUE
C
970  WRITE(6,975)
975  FORMAT(/, ' PLEASE ENTER FILE NAME FOR VELOCITIES'
   *           ' OR TYPE -PASS- TO IGNORE'            )
     READ(5, '(A)', ERR=970)  NAME2
     IF (NAME2.EQ.'p')  STOP
     IF (NAME2.EQ.'P')  STOP
     IF (NAME2.EQ.'pass')  STOP
     IF (NAME2.EQ.'PASS')  STOP
     OPEN(UNIT=14, FILE=NAME2, STATUS='NEW', ERR=970)
     WRITE(14,980)
980  FORMAT(/, ' ELE-NO.',6X,'XXX',13X,'YYY',12X,'X-VEL',11X,'Y-VEL')
     DO 990 IE=1,NELEM
     WRITE(14,985) IE,EXTMAT(IE,1),EXTMAT(IE,2),XVELO(IE),ZVELO(IE)
985  FORMAT(I5,4E16.6)
990  CONTINUE
C
     STOP
     END
!
!---------------------------------------------------------------------
!
     SUBROUTINE APPLYBC(NPOIN, IBC, PHEAD, SYSK, SYSQ, MXPOI)
C
C    APPLY PRESSURE HEAD BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C      0 = FREE TO CHANGE (TO BE COMPUTED)
C      1 = FIXED AS SPECIFIED
```

```
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), PHEAD(MXPOI)
C
      INTEGER    IBC(MXPOI)
C
      DO 100  IEQ=1,NPOIN
      IF(IBC(IEQ).EQ.0)  GO TO 100
C
      DO 200  IR=1,NPOIN
      IF(IR.EQ.IEQ)  GO TO 200
      SYSQ(IR) = SYSQ(IR) - SYSK(IR,IEQ)*PHEAD(IEQ)
      SYSK(IR,IEQ) = 0.
200   CONTINUE
C
      DO 300  IC=1,NPOIN
      SYSK(IEQ,IC) = 0.
300   CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSQ(IEQ) = PHEAD(IEQ)
C
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE ASSMBLE(   IE, INTMAT,   AKC,    QQ, SYSK,
     *                    SYSC,   SYSQ,   ACC, MXPOI, MXELE)
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  AKC(3,3), QQ(3), ACC(3,3)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
C
      INTEGER    INTMAT(MXELE,3)
C
      NNODE = 3
C
      DO 100  IR=1,NNODE
      DO 200  IC=1,NNODE
      IROW = INTMAT(IE,IR)
      ICOL = INTMAT(IE,IC)
      SYSK(IROW,ICOL) = SYSK(IROW,ICOL) + AKC(IR,IC)
      SYSC(IROW,ICOL) = SYSC(IROW,ICOL) + ACC(IR,IC)
200   CONTINUE
      SYSQ(IROW) = SYSQ(IROW) + QQ(IR)
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE SOLVEPDP(N, A, B, X, MXPOI, NHBW)
      IMPLICIT   REAL*8(A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI), X(MXPOI)
C
      CALL SCALE(N, A, B, MXPOI)
```

```
C
C    FORWARD ELEMINATION:
C
     DO 100 IP=1,N-1
C
     CALL PIVOT(N, A, B, MXPOI, IP)
C
     DO 200 IE=IP+1,N
     RATIO = A(IE,IP)/A(IP,IP)
C
     DO 300 IC=IP+1,N
     A(IE,IC) = A(IE,IC) - RATIO*A(IP,IC)
300  CONTINUE
     B(IE) = B(IE) - RATIO*B(IP)
200  CONTINUE
C
     DO 400 IE=IP+1,N
     A(IE,IP) = 0.
400  CONTINUE
100  CONTINUE
C
     X(N) = B(N)/A(N,N)
C
     DO 500 IE=N-1,1,-1
     SUM = 0.
     DO 600 IC=IE+1,N
     SUM = SUM + A(IE,IC)*X(IC)
600  CONTINUE
     X(IE) = (B(IE) - SUM)/A(IE,IE)
500  CONTINUE
C
     RETURN
     END
!
!---------------------------------------------------------------------
!
     SUBROUTINE PIVOT(N, A, B, MXPOI, IP)
     IMPLICIT   REAL*8(A-H,O-Z)
     DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
     JP = IP
     BIG = ABS(A(IP,IP))
     DO 10 I=IP+1,N
     IF(AMAX.GT.BIG) THEN
     BIG = AMAX
     JP = I
     ENDIF
 10  CONTINUE
     IF(JP.NE.IP) THEN
     DO 20 J=IP,N
     DUMY    = A(JP,J)
     A(JP,J) = A(IP,J)
     A(IP,J) = DUMY
20   CONTINUE
     DUMY  = B(JP)
     B(JP) = B(IP)
     B(IP) = DUMY
     ENDIF
     RETURN
     END
C
```

```
C-----------------------------------------------------------------------
C
      SUBROUTINE SCALE(N, A, B, MXPOI)
      IMPLICIT   REAL*8(A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
      DO 10 IE=1,N
      BIG = ABS(A(IE,1))
      DO 20 IC=2,N
      AMAX = ABS(A(IE,IC))
      IF (AMAX.GT.BIG) BIG = AMAX
 20   CONTINUE
      DO 30 IC=1,N
      A(IE,IC) = A(IE,IC)/BIG
 30   CONTINUE
      B(IE) = B(IE)/BIG
 10   CONTINUE
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE TRI(NELEM, INTMAT, COORD,    HK,  QELE, THICK,
     *                SYSK,   SYSC,  SYSQ, SLOPE, MXPOI, MXELE)
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     UP SYSTEM EQUATIONS
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), SYSK(MXPOI,MXPOI), SYSQ(MXPOI)
      DIMENSION  SYSC(MXPOI,MXPOI)
      DIMENSION  QELE(MXELE), HK(MXELE) ,SLOPE(MXELE)
      DIMENSION  QG(3)
      DIMENSION  AKC(3,3), QQ(3), B(2,3), BT(3,2), ACC(3,3)
C
      INTEGER    INTMAT(MXELE,3)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II  = INTMAT(IE,1)
      JJ  = INTMAT(IE,2)
      KK  = INTMAT(IE,3)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      AREA= 0.5*(XG2*(YG3-YG1) + XG1*(YG2-YG3) + XG3*(YG1-YG2))
C
      IF(AREA.LE.0.)  WRITE(6,5)  IE
   5  FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *          ' HAS NEGATIVE OR ZERO AREA ', /,
     *          '  --- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *          ' AND ELEMENT NODAL CONNECTIONS ---'          )
      IF(AREA.LE.0.)  STOP
```

```
C
      B1 = YG2 - YG3
      B2 = YG3 - YG1
      B3 = YG1 - YG2
      C1 = XG3 - XG2
      C2 = XG1 - XG3
      C3 = XG2 - XG1
C
      DO 10  I=1,2
      DO 10  J=1,3
      B(I,J)  = 0.
 10   CONTINUE
C
      B(1,1)  = B1
      B(1,2)  = B2
      B(1,3)  = B3
      B(2,1)  = C1
      B(2,2)  = C2
      B(2,3)  = C3
C
      DO 20  I=1,2
      DO 30  J=1,3
      B(I,J)  = B(I,J)/(2.*AREA)
      BT(J,I) = B(I,J)
 30   CONTINUE
 20   CONTINUE
C
C     ELEMENT CAPACITANCE MATRIX (LUMP):
C
      DO 50 I=1,3
      DO 50 J=1,3
      IF(I.EQ.J)  ACC(I,J) = 1.
      IF(I.NE.J)  ACC(I,J) = 0.
      ACC(I,J) = SLOPE(IE)*AREA*ACC(I,J)/3.
 50   CONTINUE
C
C     ELEMENT CONDUCTION MATRIX:
C
      DO 100  I=1,3
      DO 100  J=1,3
      AKC(I,J) = 0.
      DO 90   K=1,2
      AKC(I,J) = AKC(I,J) + BT(I,K)*B(K,J)
 90   CONTINUE
      AKC(I,J) = HK(IE)*AREA*(XG1+XG2+XG3)*AKC(I,J)/3.
100   CONTINUE
C
C     ELEVATION TERM POSSITIVE IN UPWARD VERTICAL DIRECTION
C
      QQ(1) = -1*HK(IE)*(XG1+XG2+XG3)*C1/6.
      QQ(2) = -1*HK(IE)*(XG1+XG2+XG3)*C2/6.
      QQ(3) = -1*HK(IE)*(XG1+XG2+XG3)*C3/6.
C
C     ELEMENT HEAT LOAD DUE TO INTERNAL MOISTURE GENERATION:
C     <QELE> POSSITIVE FOR SOURCE TERM
C
      FAC = QELE(IE)*AREA*THICK/3.
      QG(1) = FAC*(2.*XG1+XG2+XG3)/12.
      QG(2) = FAC*(XG1+2.*XG2+XG3)/12.
      QG(3) = FAC*(XG1+XG2+2.*XG3)/12.
C
```

```
      DO 200  I=1,3
      QQ(I) = QQ(I) + QG(I)
200   CONTINUE
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT,    AKC,     QQ,  SYSK,
     *               SYSC,   SYSQ,    ACC, MXPOI, MXELE)
C
500   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE DARCY (NOE, INTMAT, COORD,     HK, PHEAD_N,
     *                  EXTMAT,  XVELO, ZVELO, MXPOI, MXELE  )
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), XVELO(MXPOI), ZVELO(MXPOI)
      DIMENSION  PHEAD_N(MXPOI)
      DIMENSION  HK(MXELE)
      DIMENSION  EXTMAT(MXELE,2)
C
      INTEGER    INTMAT(MXELE,3)
C
      DO 100 IE=1,NOE
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
C
      XN1  = COORD(II,1)
      XN2  = COORD(JJ,1)
      XN3  = COORD(KK,1)
      ZN1  = COORD(II,2)
      ZN2  = COORD(JJ,2)
      ZN3  = COORD(KK,2)
      AREA = 0.5*(XN2*(ZN3-ZN1) + XN1*(ZN2-ZN3) + XN3*(ZN1-ZN2))
C
      XELE = (XN1 + XN2 + XN3)/3.
      ZELE = (ZN1 + ZN2 + ZN3)/3.
      EXTMAT(IE,1) = XELE
      EXTMAT(IE,2) = ZELE
C
      BT1 = (ZN2 - ZN3)*PHEAD_N(II)
      BT2 = (ZN3 - ZN1)*PHEAD_N(JJ)
      BT3 = (ZN1 - ZN2)*PHEAD_N(KK)
      CT1 = (XN3 - XN2)*PHEAD_N(II)
      CT2 = (XN1 - XN3)*PHEAD_N(JJ)
      CT3 = (XN2 - XN1)*PHEAD_N(KK)
C
      XVELO(IE) = -1*HK(IE)*(BT1 + BT2 + BT3)/(2.*AREA)
      ZVELO(IE) = -1*HK(IE)*(CT1 + CT2 + CT3)/(2.*AREA) - HK(IE)
C
100   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
```

```
!
      SUBROUTINE VELBOUND (NOBP, INTFLUX, COORD, FLUX, THICK,
     *                     SYSQ,  MXPOI, MXFLUX, NCOUNT     )
C
C     <FLUX> POSSITIVE FOR SOURCE TERM
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), FLUX(MXPOI), SYSQ(MXPOI)
C
      INTEGER    INTFLUX(MXFLUX,3)
C
C     ************************
C     FOR RUNNING BOUNDARY FLUX
C     ************************
C
      IF (MOD(NCOUNT,250).EQ.0) THEN
C
      IA = INTFLUX(1,2)
C
      DO 50 II=1,NOBP
      JI = INTFLUX(II,2)
      KI = INTFLUX(II,3)
C
      INTFLUX(II,2) = JI + 1
      INTFLUX(II,3) = KI + 1
C
      FLUX(JI + 1)  = FLUX(JI)
      FLUX(KI + 1)  = FLUX(KI)
C
 50   CONTINUE
      FLUX(IA) = 0.
C
      ENDIF
C
      DO 100 IE=1,NOBP
      JJ = INTFLUX(IE,2)
      KK = INTFLUX(IE,3)
C
      X1 = COORD(JJ,1)
      X2 = COORD(KK,1)
      Y1 = COORD(JJ,2)
      Y2 = COORD(KK,2)
      DX = X2 - X1
      DY = Y2 - Y1
      DL = SQRT(DX*DX + DY*DY)
C
      Q1 = FLUX(JJ)*THICK*DL/2.
      Q2 = FLUX(KK)*THICK*DL/2.
C
      SYSQ(JJ) = SYSQ(JJ) + Q1
      SYSQ(KK) = SYSQ(KK) + Q2
C
100   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE ELEHK (NOE, INTMAT, PHEAD_N, HK, SLOPE, HKSAT, ALPHA,
     *                  EM,     EN,   ZETA_S,     ZETA_R, MXPOI, MXELE)
```

```
C
      IMPLICIT    REAL*8 (A-H,O-Z)
      DIMENSION   PHEAD_N(MXPOI)
      DIMENSION   HK(MXELE), SLOPE(MXELE)
C
      INTEGER     INTMAT(MXELE,3)
C
      DO 100 I=1,NOE
      II = INTMAT (I,1)
      JJ = INTMAT (I,2)
      KK = INTMAT (I,3)
C
C     AVERAGE ELEMENT PRESSURE HEAD
C
      AVEPH = (PHEAD_N(II) + PHEAD_N(JJ) + PHEAD_N(KK))/3.
C
C     ELEMENT HYDRAULIC CONDUCTIVITY
C     OR MEANS HEAT CONDUCTION COEFF.IN HEAT TRANFER PROBLEM
C
      HK(I) = HKSAT*1.175E6/(1.175E6 + ABS(AVEPH)**4.74)
C
C     ELEMENT MOISTURE-PRESSURE HEAD SLOPE
C     OR MEANS DENSITY*SPECIFIC HEAT(P) IN HEAT TRANFER PROBLEM
C
      SLOPE(I) = -1*1.611E6*(0.287-0.075)*3.96*AVEPH*
C
100   CONTINUE
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE CONVERGE (NON, PHEAD_N, PHEAD_O, TOLPH, MXPOI)
C
      IMPLICIT    REAL*8 (A-H,O-Z)
      DIMENSION   PHEAD_N(MXPOI), PHEAD_O(MXPOI)
C
      AAA = 0.
      BBB = 0.
C
      DO 100 I=1,NON
      AA  = PHEAD_N(I)
      AAA = AAA + AA
      BB  = PHEAD_O(I)
      BBB = BBB + BB
C
100   CONTINUE
C
      WRITE(9,50) BBB
 50   FORMAT(/,F16.6)
C
      TOLPH = ABS((BBB - AAA)/AAA)*100
C
      RETURN
      END
!
!---------------------------------------------------------------------
!
      SUBROUTINE INITIAL (NPOIN, NELEM, IBC, PHEAD, INTMAT, HK, SLOPE,
     *                    COORD, HKSAT, ALPHA, EM, EN, ZETA_S, ZETA_R,
```

```
     *                     ST_PH, MXPOI, MXELE                          )
C
      IMPLICIT   REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), PHEAD(MXPOI)
      DIMENSION  HK(MXELE), SLOPE(MXELE)
C
      INTEGER    INTMAT(MXELE,3)
      INTEGER    IBC(MXPOI)
C
      DO 100 IP=1,NPOIN
      IF (IBC(IP).EQ.1) GOTO 100
C
      PHEAD(IP) = ST_PH
C
      CALL ELEHK (NOE, INTMAT,   PHEAD, HK, SLOPE, HKSAT, ALPHA,
     *             EM,     EN,  ZETA_S,    ZETA_R, MXPOI, MXELE )
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
      SUBROUTINE TRANSIENT (NON, SYSC, SYSK,  SYSQ, PHEAD_P,
     *                      QQQ,  TWF,   DT, MXPOI          )
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION SYSK(MXPOI,MXPOI), SYSQ(MXPOI), SYSC(MXPOI,MXPOI)
      DIMENSION PHEAD_P(MXPOI), QQQ(MXPOI)
C
C     ************************************************************
C                         FOR STEADY STATE
C     ************************************************************
C
CC    WRITE(6,50)
CC    50      FORMAT(/,' *** STEADY STATE ***')
C
C     ************************************************************
C                          FOR TRANSIENT
C     ************************************************************
C
      WRITE(6,100)
100   FORMAT(/,' *** TRANSIENT ***')
C
      DO 200 K=1,NON
      QQQ(K) = SYSQ(K)
200   CONTINUE
C
      DO 500 I=1,NON
      SYSQ(I) = (1-TWF)*SYSQ(I) + TWF*QQQ(I)
      DO 500 J=1,NON
      SYSQ(I) = SYSQ(I) +
     *          (SYSC(I,J)/DT - (1-TWF)*SYSK(I,J))*PHEAD_P(J)
      SYSK(I,J) = SYSC(I,J)/DT + SYSK(I,J)*TWF
500   CONTINUE
C
      RETURN
      END
!
!-----------------------------------------------------------------------
!
```

ประวัติผู้เขียนวิทยานิพนธ์

นายสุธี ไตรวิวัฒนา เกิดเมื่อวันที่ 20 เดือนมกราคม พุทธศักราช 2521 จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิตจากภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2543