The Sea Water Intrusion (SWI) Package Manual
Part II. Module Documentation
Version 0.1

Frans Schaars
Artesia Water Research, Unlimited
2817 BP Schoonhoven, The Netherlands
Email: f.schaars@artesia-water.nl

Mark Bakker
Department of Biological and Agricultural Engineering
University of Georgia, Athens, GA 30602, USA
Email: mbakker@engr.uga.edu

September 10, 2004

Contents

License

1 Module documentation for the SWI Package

This section contains a description of the modules of the SWI package, instructions on how to implement the SWI package in MODFLOW2000, flowcharts, listings of variables and FORTRAN code of the primary modules and submodules of the SWI package.

1.1 Modules and submodules of the SWI package

The Sea Water Intrusion package (SWI1) has four primary modules and four submodules. The relationships of the modules to MAIN and to each other are shown in Figure 10. The first three characters of the names identify the modules as being part of the Sea Water Intrusion Package; the next character identifies the version number of the package; the last two characters identify the procedure performed in the module. The procedures used by the Sea Water Intrusion Package are consistent with procedures used by existing packages in the ground-water flow model. The modules are:

- Primary Modules
  - SWI1AL: Allocates space for data arrays.
  - SWI1RP: Reads and prepares all data needed by the package.
  - SWI1FM: Calculates coefficients of the system of equations that are not constant (formulate the finite-difference equations).
  - SWI1BD: Calculates flow terms and moves surfaces (called planes in the code documentation).
- Submodules
  - SWI1SQR: Calculates discharge between two cells below a given plane in row direction.
  - SWI1SQC: Calculates discharge between two cells below a given plane in column direction.
  - SWI1SD: Performs matrix multiplication for double precision arrays.
  - SWI1SR: Performs matrix multiplication for real arrays.

1.2 How to implement the SWI package in MODFLOW2000

The MAIN program of the ground-water flow model must be modified to call the four modules of the Sea Water Intrusion Package. Call statements to the modules of the Sea Water Intrusion Package must be placed in sections of the MAIN program in which the particular procedure is being carried out for other packages. For example, the SWIlAL module must be called within the

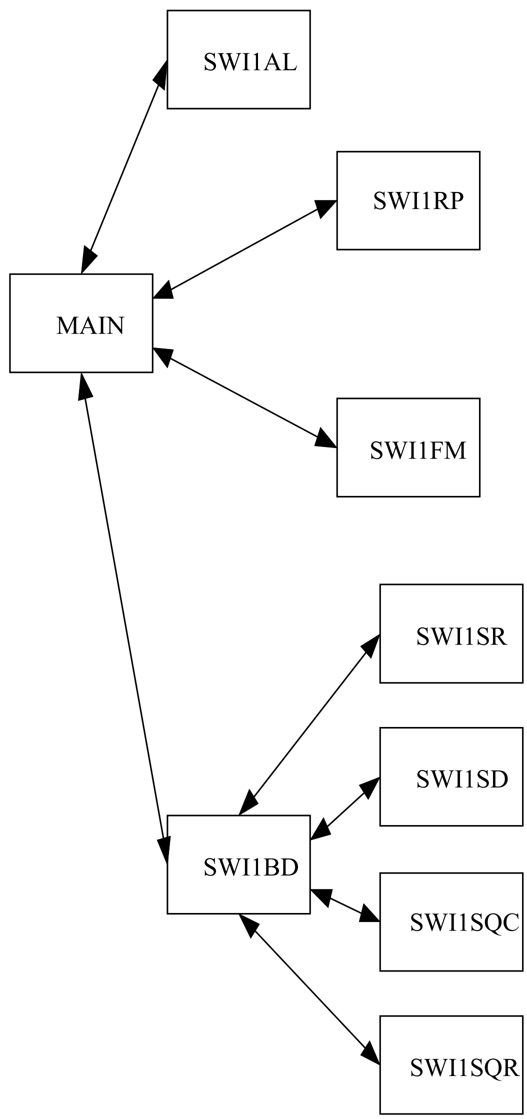

Figure 1: Relationship among the modules in the Sea Water Intrusion package

section of the MAIN program in which other modules that allocate space for arrays are called. In all sections of the MAIN program, the call to the Basic (BAS) Package module (subroutine) must come before any other module call statements. The FORTRAN call statements to be added to the MAIN program are as follows:

```
      IF (IUNIT(44).GT.0)
     &     CALL SWI1AL(ISUMRX,ISUMIR,ISUMZ,NCOL,NROW,NLAY,IUNIT(44),
     & IOUT,
     & LCZETA,LCSSZ,LCIPLPOS,LCSWICR,LCSWICC,LCSWICUMCR,LCSWICUMCC,
     & NZONES,ISTRAT,LCNUTOP,LCNUBOT,LCBRHS,LCQZEXTRA,
     & LCSWIAU,LCSWIIUPP,LCSWIAL,LCSWIIEQP,LCSWID4B,LCSWILRCH,
     & LCSWIHDCG,LCSWIIBO,LCSWIHCOF,LCSWIDE45CR,LCSWIDE45CC,
     & LCSWIDE45CV,LCDUM,ISWIZT,NPRS,LCEPS,LCDELNUS,LCNUS,LCNUPLANE)

      IF (IUNIT(44).GT.0)
     &    CALL SWI1RP(RX(LCNUS),NZONES,TOESLOPE,TIPSLOPE,ZETAMIN,
     & DELZETA,ISTRAT,RX(LCZETA),RX(LCSSZ),IR(LCIPLPOS),NCOL,NROW,NLAY,
     & GX(LCBOTM),NBOTM,IUNIT(44),IOUT,RX(LCDELNUS),RX(LCEPS),
     & RX(LCNUPLANE))

          IF (IUNIT(44).GT.0)
     &     CALL SWI1FM(GX(LCBOTM),NBOTM,RX(LCEPS),RX(LCNUS),
     & RX(LCDELNUS),IG(LCIBOU),GZ(LCHNEW),GX(LCCR),GX(LCCC),GX(LCCV),
     & GX(LCHCOF),GX(LCRHS),
     & NCOL,NROW,NLAY,NZONES,IR(LCIPLPOS),RX(LCZETA),
     & RX(LCSWICR),RX(LCSWICC),RX(LCSWICUMCR),RX(LCSWICUMCC),
     & GX(LCDELR),GX(LCDELC),RX(LCNUTOP),RX(LCNUBOT),RX(LCQZEXTRA))

       IF (IUNIT(44).GT.0)
     &      CALL SWI1BD(DELT,TOESLOPE,TIPSLOPE,ZETAMIN,DELZETA,
     & RX(LCEPS),RX(LCNUS),RX(LCDELNUS),
     & IG(LCIBOU),GZ(LCHNEW),GX(LCCR),GX(LCCC),GX(LCCV),
     & GX(LCHCOF),GX(LCRHS),
     & NCOL,NROW,NLAY,NZONES,IR(LCIPLPOS),RX(LCZETA),RX(LCSSZ),
     & RX(LCSWICR),RX(LCSWICC),RX(LCSWICUMCR),RX(LCSWICUMCC),
     & GX(LCDELR),GX(LCDELC),RX(LCNUTOP),RX(LCNUBOT),RX(LCBRHS),
     & RX(LCSWIAU),IR(LCSWIIUPP),RX(LCSWIAL),IR(LCSWIIEQP),
     & RX(LCSWID4B),IR(LCSWILRCH),
     & RX(LCSWIHDCG),IR(LCSWIIBO),RX(LCSWIHCOF),RX(LCQZEXTRA),
     & RX(LCSWIDE45CR),RX(LCSWIDE45CC),RX(LCSWIDE45CV),RX(LCDUM),
     & IOUT,ISWIZT,NPRS,NSTP(KKPER),KSTP)
```

In this case the SWI-package is assigned to IUNIT(44). This is accomplished by placing the "SWI" keyword in the 44th element of CUNIT in the MAIN program.

1.3 Module SWI1AL

Variables of module SWI1AL

```
Variable    Range   Description
-------------------------------
ID4DIM      package 5 -- for a 2-dimensional grid.7 -- for a 3-dimensional grid.
```

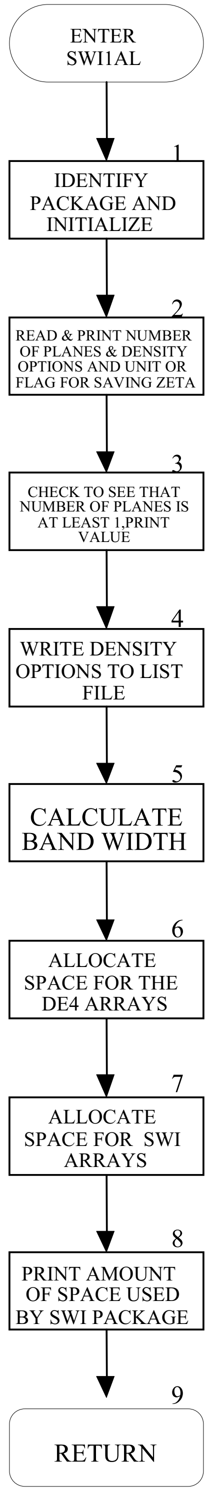

Figure 2: Flowchart of module SWI1AL

ID4DIR	package	Flag that indicates the relative size of NCOL, NROW, and NLAY
IN	package	unit number for the SWI- inputfile
IOUT	global	Primary unit number for all printed output.
ISOLD	module	Value of ISUM upon entry to this module.
ISOLDI	module	Value of ISUMI upon entry to this module.
ISOLDZ	module	Value of ISUMZ upon entry to this module.
ISP	module	Amount of space used by this package
ISTRAT	package	Flag indicating density distribution (0 = lineair between planes, 1= constant between planes)
ISUM	global	Element number of the lowest element in the X array that has not yet been allocated
ISUMI	global	Element number of the lowest element in the IX array that has not yet been allocated
ISUMZ	global	Element number of the lowest element in the GZ array that has not yet been allocated
ISWIZT	package	Flag and Unit number, if ISWIZT>0 unit number for ZETA else ZETA will not be recorded
LCBRHS	package	Location in the X array of the first element of array BRHS
LCDELNUS	package	Location in the X array of the first element of array DELNUS
LCDUM	package	Location in the GZ array of the first element of array DUM
LCEPS	package	Location in the X array of the first element of array EPS
LCIPLPOS	package	Location in the IX array of the first element of array IPLPOS
LCNUBOT	package	Location in the X array of the first element of array NUBOT
LCNUPLANE	package	Location in the X array of the first element of array NUPLANE
LCNUS	package	Location in the X array of the first element of array NUS
LCNUTOP	package	Location in the X array of the first element of array NUTOP
LCQZEXTRA	package	Location in the X array of the first element of array QZEXTRA
LCSSZ	package	Location in the X array of the first element of array SSZ
LCSWIAL	package	Location in the X array of the first element of array AL
LCSWIAU	package	Location in the X array of the first element of array AU
LCSWICC	package	Location in the X array of the first element of array SWICC
LCSWICR	package	Location in the X array of the first element of array SWICR
LCSWICUMCC	package	Location in the X array of the first element of array SWICUMCC
LCSWICUMCR	package	Location in the X array of the first element of array SWICUMCR
LCSWID4B	package	Location in the X array of the first element of array SWID4B
LCSWIDE45CC	package	Location in the X array of the first element of array SWIDE45CC
LCSWIDE45CR	package	Location in the X array of the first element of array SWIDE45CR
LCSWIDE45CV	package	Location in the X array of the first element of array SWIDE45CV
LCSWIHCOF	package	Location in the X array of the first element of array SWIHCOF
LCSWIHDCG	package	Location in the X array of the first element of array SWIHDCG
LCSWIIBO	package	Location in the IX array of the first element of array SWIIBO
LCSWIIEQP	package	Location in the IX array of the first element of array SWIEQP
LCSWIIUPP	package	Location in the IX array of the first element of array SWIIUPP
LCSWILRCH	package	Location in the IX array of the first element of array LRCH
LCZETA	package	Location in the X array of the first element of array ZETA
MXBW	package	the maximum band width plus 1 of the [AL] matrix
MXEQ	package	MXUP+MXLOW
MXLOW	package	The maximum allowed number of equations in the lower part of [A].
MXUP	package	The maximum allowed number of equations in the upper part of [A].
NBWGRD	package	One plus the product of the two smallest grid dimensions.
NCOL	global	Number of columns
NHALFL	package	the maximum possible number of equations in the lower part of [A]
NHALFU	package	the maximum possible number of equations in the upper part of [A]
NLAY	global	Number of layers
NLAYSWI	module	One
NODES	global	Number of nodes
NPLN	module	Number of planes
NPRS	package	Print out interval

```
NROW        global  Number of rows
NZONES      package Number of zones
```

FORTRAN code of SWI1AL

```
      SUBROUTINE SWI1AL(ISUM,ISUMI,ISUMZ,NCOL,NROW,NLAY,IN,IOUT,
    1 LCZETA,LCSSZ,LCIPLPOS,LCSWICR,LCSWICC,LCSWICUMCR,LCSWICUMCC,
     2 NZONES,ISTRAT,LCNUTOP,LCNUBOT,LCBRHS,LCQZEXTRA,
     3 LCSWIAU,LCSWIIUPP,LCSWIAL,LCSWIIEQP,LCSWID4B,LCSWILRCH,
     4 LCSWIHDCG,LCSWIIBO,LCSWIHCOF,LCSWIDE45CR,LCSWIDE45CC,
     5 LCSWIDE45CV,LCDUM,ISWIZT,NPRS,LCEPS,LCDELNUS,LCNUS,LCNUPLANE)

C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR SEA WATER INTRUSION PACKAGE
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------

C1------IDENTIFY PACKAGE AND INITIALIZE
      WRITE(IOUT,1)IN
    1 FORMAT(1X,/1X,'SWI1 -- SWI PACKAGE, VERSION 0.1, 29-Nov-2002',
     1' INPUT READ FROM UNIT',I3)
C
C2------READ & PRINT NUMBER OF PLANES
C2------ & DENSITY OPTIONS AND UNIT OR FLAG FOR SAVING ZETA
      READ(IN,2) NPLN,ISTRAT,ISWIZT,NPRS
    2 FORMAT(4I10)

      IF (ISWIZT.GT.0) WRITE(IOUT,3) ISWIZT
    3 FORMAT(1X,'ZETA WILL BE SAVED ON UNIT',I3)
C
C3-----CHECK TO SEE THAT NUMBER OF PLANES IS AT LEAST 1,
C3-----PRINT VALUE
      IF(NPLN.GT.0) THEN
       WRITE(IOUT,6) NPLN
    6 FORMAT(1X,'TOTAL NUMBER OF PLANES: ',I3)
      ELSE
       WRITE (IOUT,7)
    7 FORMAT(1X,'ABORTING, NUMBER OF PLANES LESS THAN 1...')
      STOP
      ENDIF
C
C4-----WRITE DENSITY OPTIONS TO LIST FILE
    IF(ISTRAT.EQ.1) WRITE(IOUT,8)
    8 FORMAT(1X,'ISTRAT 1 -- STRATIFIED FLOW')
    IF(ISTRAT.EQ.0) WRITE(IOUT,9)
    9 FORMAT(1X,'ISTRAT 0 -- VARIABLE DENSITY FLOW')

      NZONES=NPLN+1;
C
C5------COPIED FROM DE45 PACKAGE SUBSTITUTE NLAY BY 1
C5------CALCULATE BAND WIDTH
```

```
      NLAYSWI=1
      NODES=NCOL*NROW*NLAYSWI
      NHALFU=(NODES-1)/2 +1
      NHALFL=NODES-NHALFU
      ID4DIM=7
      IF(NLAYSWI.LE.NCOL .AND. NLAYSWI.LE.NROW) THEN
         IF(NLAYSWI.EQ.1) ID4DIM=5
         IF(NCOL.GE.NROW) THEN
            ID4DIR=1
            NBWGRD=NROW*NLAYSWI+1
         ELSE
            ID4DIR=2
            NBWGRD=NCOL*NLAYSWI+1
         END IF
      ELSE IF(NROW.LE.NCOL .AND. NROW.LE.NLAYSWI) THEN
         IF(NROW.EQ.1) ID4DIM=5
         IF(NCOL.GE.NLAYSWI) THEN
            ID4DIR=3
            NBWGRD=NROW*NLAYSWI+1
         ELSE
            ID4DIR=4
            NBWGRD=NROW*NCOL+1
         END IF
      ELSE
         IF(NCOL.EQ.1) ID4DIM=5
         IF(NROW.GE.NLAYSWI) THEN
            ID4DIR=5
            NBWGRD=NCOL*NLAYSWI+1
         ELSE
            ID4DIR=6
            NBWGRD=NCOL*NROW+1
         END IF
      END IF
      MXUP=NHALFU
      MXLOW=NHALFL
    MXBW=NBWGRD + 4
      MXEQ=MXUP+MXLOW
C
C6------ALLOCATE SPACE FOR THE DE4 ARRAYS.
      ISOLD=ISUM
      ISOLDI=ISUMI
    ISOLDZ=ISUMZ

      LCSWIAU=ISUM
      ISUM=ISUM+MXUP*ID4DIM
      LCSWIIUPP=ISUMI
      ISUMI=ISUMI+MXUP*ID4DIM
      LCSWIAL=ISUM
      ISUM=ISUM+MXLOW*MXBW
      LCSWIIEQP=ISUMI
      ISUMI=ISUMI+NODES
      LCSWID4B=ISUM
      ISUM=ISUM+MXEQ
```

```
      LCSWILRCH=ISUMI
      ISUMI=ISUMI+ITMX*3
      LCSWIHDCG=ISUM
      ISUM=ISUM+ITMX
      LCSWIIBO=ISUMI
      ISUMI=ISUMI+NROW*NCOL
      LCSWIHCOF=ISUM
      ISUM=ISUM+NCOL*NROW
      LCSWIDE45CR=ISUM
      ISUM=ISUM+NCOL*NROW
      LCSWIDE45CC=ISUM
      ISUM=ISUM+NCOL*NROW
      LCSWIDE45CV=ISUM
      ISUM=ISUM+NCOL*NROW
C
C7------ALLOCATE SPACE FOR  SWI ARRAYS.
       LCZETA=ISUM
       ISUM=ISUM+(NZONES+1)*NROW*NCOL*NLAY
       LCSSZ=ISUM
       ISUM=ISUM+NCOL*NROW*NLAY

       LCEPS=ISUM
       ISUM=ISUM+NZONES
       LCNUS=ISUM
       ISUM=ISUM+NZONES
       LCDELNUS=ISUM
       ISUM=ISUM+NZONES
       LCNUPLANE=ISUM
       ISUM=ISUM+NZONES+1

     LCSWICR=ISUM
       ISUM=ISUM+NZONES*NROW*NCOL*NLAY
     LCSWICC=ISUM
       ISUM=ISUM+NZONES*NROW*NCOL*NLAY
     LCSWICUMCR=ISUM
       ISUM=ISUM+NZONES*NROW*NCOL*NLAY
     LCSWICUMCC=ISUM
       ISUM=ISUM+NZONES*NROW*NCOL*NLAY
     LCNUTOP=ISUM
       ISUM=ISUM+NROW*NCOL*NLAY
     LCNUBOT=ISUM
       ISUM=ISUM+NROW*NCOL*NLAY
     LCBRHS=ISUM
       ISUM=ISUM+NROW*NCOL*NZONES
     LCQZEXTRA=ISUM
       ISUM=ISUM+NROW*NCOL*NLAY
     LCDUM=ISUM
       ISUM=ISUM+2*NROW*NCOL
     LCIPLPOS=ISUMI
     ISUMI=ISUMI+NZONES*NROW*NCOL*NLAY
C
C8------PRINT AMOUNT OF SPACE USED BY SWI PACKAGE.
        ISP=ISUM-ISOLD
```

```
        WRITE (IOUT,30)ISP
   30   FORMAT(1X,I10,' ELEMENTS IN RX ARRAY ARE USED FOR SWI')
        ISP=ISUMI-ISOLDI
        WRITE (IOUT,40)ISP
   40   FORMAT(1X,I10,' ELEMENTS IN IR ARRAY ARE USED FOR SWI')

C9------RETURN
      RETURN
      END
```

1.4 Module SWI1RP

Variables used in module SWI1RP

```
Variable    Range   Description
-------------------------------
BOTM        global  Bottom elevation of a model layer or quasi 3D confining bed
DELNUS      package difference in NUS between zones
DELZETA     package Elevation for a plane when it is moved into an adjacent empty cell
EPS         package Epsilon, variation of density within zone
IN          package unit number for the SWI- inputfile
IOUT        global  Primary unit number for all printed output.
IPLPOS      package Flag indicating plane position (1=top, 2=bottom, 0=in between)
ISTRAT      package Flag indicating density distribution ( 0 = lineair between planes, 1= constant between planes)
IZ          module  Loop variable for zones
K           module  Loop variable for layers
KK          module  Loop variable for layers
NBOTM       global  Number of layers and quasi-3d layers
NCOL        global  Number of columns
NLAY        global  Number of layers
NROW        global  Number of rows
NUPLANE     module  Value of NU for the planes
NUS         module  The values of NUS
NZONES      package Number of zones
SSZ         package Effective porosity
TIPSLOPE    package Maximum slope of tipcells
TOESLOPE    package Maximum slope of toecells
ZETA        package Elevation of the planes
ZETAMIN     package Minimum elevation of a plane before it is removed from a cell
ZETANAME    module  Identification for printing to list-file
```

FORTRAN code of module SWI1RP

```
    SUBROUTINE SWI1RP(NUS,NZONES,TOESLOPE,TIPSLOPE,ZETAMIN,DELZETA,
    1 ISTRAT,ZETA,SSZ,IPLPOS,NCOL,NROW,NLAY,BOTM,NBOTM,IN,IOUT,
     2 DELNUS,EPS,NUPLANE)

C
C-----VERSION 1 19-7-2002 SWI1RP
C     ******************************************************************

C     ******************************************************************

      COMMON /DISCOM/LBOTM(200)
```

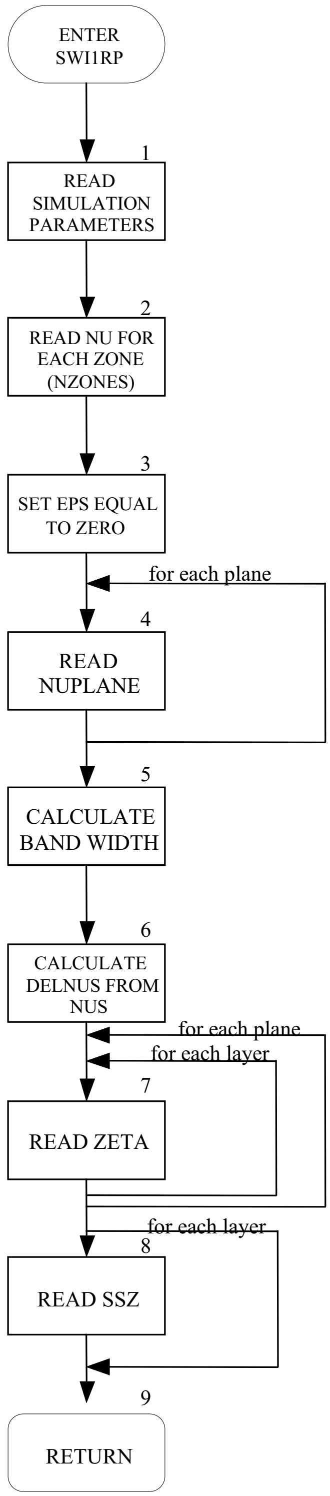

Figure 3: Flowchart of module SWI1RP

```
      DIMENSION ZETA(NCOL,NROW,NLAY,NZONES+1), BOTM(NCOL,NROW,0:NBOTM)
      DIMENSION SSZ(NCOL,NROW,NLAY)
      REAL NUS(NZONES)
    REAL NUPLANE(NZONES+1)
    REAL DELNUS(NZONES)
    REAL EPS(NZONES)
    CHARACTER*24 ZETANAME

C     -----------------------------------------------------------------

C
C1------READ SIMULATION PARAMETERS
      READ(IN,'(F10.0,F10.0,F10.0,F10.0)')
    1     TOESLOPE,TIPSLOPE,ZETAMIN,DELZETA


    IF (ISTRAT.EQ.1) THEN
C
C2------READ NU FOR EACH ZONE (NZONES)
       CALL U1DREL(NUS,'                 NUZONE',NZONES,IN,IOUT)

C3------SET EPS EQUAL TO ZERO
     DO IZ=1,NZONES
      EPS(IZ)=0.0
     END DO

    ELSEIF (ISTRAT.EQ.0) THEN
C
C4------READ NUPLANE FOR EACH PLANE (NZONES+1)
       CALL U1DREL(NUPLANE,'                NUPLANE',NZONES+1,IN,IOUT)
C
C5------CALCULATE NU AND EPS FROM NUPLANE
     DO IZ=1,NZONES
      NUS(IZ)=0.5*(NUPLANE(IZ)+NUPLANE(IZ+1))
      EPS(IZ)=(NUPLANE(IZ+1)-NUPLANE(IZ))/6
     END DO

    END IF
C
C6------CALCULATE DELNUS FROM NUS
    DELNUS(1)=NUS(1);
    DO IZ=2,NZONES
     DELNUS(IZ)=(NUS(IZ)-NUS(IZ-1))
    END DO
C
C7------READ ZETA FOR EACH PLANE
    DO 55 IZ=2,NZONES
       DO 55 K=1,NLAY
          KK=K
         WRITE(ZETANAME,111) IZ-1
  111      FORMAT('           ZETA PLANE ',I2)
```

```
          CALL U2DREL(ZETA(1,1,K,IZ),ZETANAME,
    1     NROW,NCOL,KK,IN,IOUT)
   55 CONTINUE
C
C8------READ SSZ FOR EACH LAYER
      DO 110 K=1,NLAY
          KK=K
          CALL U2DREL(SSZ(1,1,K),'                   SSZ ',
    1     NROW,NCOL,KK,IN,IOUT)
  110 CONTINUE
C
C9-----RETURN
      RETURN
      END
```

1.5 Module SWI1FM

Variables used in module SWI1FM

```
Variable    Range   Description
------------------------------
BBOT        global  Bottom of cell in loop
BOTM        global  Bottom elevation of a model layer or quasi 3D confining bed
CC          global  Branch conductance in column direction
CR          global  Branch conductance in row direction
CV          global  Branch conductance in vertical direction
DELC        global  DIMENSION (NROW) cell width
DELNUS      package difference in NUS between zones
DELR        global  DIMENSION (NCOL) cell width
DZETA1      module  Thickness of zone in current cell
DZETA2      module  Thickness of zone in adjacent cell
EPS         package Epsilon, variation of density within zone
HCOF        global  DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the finite-difference
HEADDIFF    module  Head difference between layers caused by density differences
HHD         module  Height of the watertabel
HNEW        global  DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell
I           module  Loop variable for rows
IBOUND      global  DIMENSION (NCOL,NROW,NLAY), Status of each cell
IOUT        global  Primary unit number for all printed output.
IPLPOS      package Flag indicating plane position (1=top, 2=bottom, 0=in between)
IZ          module  Loop variable for zones
IZREV       module  Reverse loop variable for zones
J           module  Loop variable for columns
K           module  Loop variable for layers
NBOTM       global  Number of layers and quasi-3d layers
NCOL        global  Number of columns
NLAY        global  Number of layers
NROW        global  Number of rows
NUBOT       module  Value of NU below the lowest zone
NUS         module  The values of NUS
NUTOP       module  Value of NU of the upper zone
NZONES      package Number of zones
QZEXTRA     module  Extra discharge due to headdiff
RHS         global  DIMENSION (NCOL,NROW,NLAY), Right-hand side of the finite-difference equation.
```

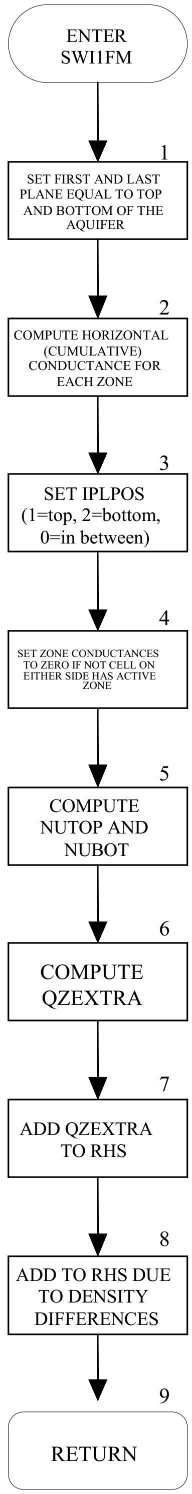

Figure 4: Flowchart for module SWI1FM

```
SUMSWICC    module  Sum of CC for current cell
SUMSWICR    module  Sum of CR for current cell
SUMTHICKFF  module  Sum of thicknesses of zones between this cell ande the one to the right
SUMTHICKRF  module  Sum of thicknesses of zones between this cell ande the one to the front
SWICC       package Conductance in column direction for each zone
SWICR       package Conductance in row direction for each zone
SWICUMCC    package Cumulative conductance in column direction for each zone
SWICUMCR    package Cumulative conductance in row direction for each zone
THICKFF     module  Thickness of zones between this cell and the one to the right
THICKRF     module  Thickness of zones between this cell and the one to the front
TTOP        module  Top of cell in loop
ZETA        package Elevation of the planes
```

FORTRAN code of module SWI1FM

```
      SUBROUTINE SWI1FM(BOTM,NBOTM,EPS,NUS,DELNUS,IBOUND,HNEW,CR,CC,
     &  CV,HCOF,RHS,NCOL,NROW,NLAY,NZONES,IPLPOS,ZETA,SWICR,SWICC,
     &  SWICUMCR,SWICUMCC,DELR,DELC,NUTOP,NUBOT,QZEXTRA)


C-----VERSION 1 9AUG2002 SWI1FM
C     ******************************************************************
C     ADD SWI TERMS TO RHS AND HCOF
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------

      COMMON /DISCOM/LBOTM(200)
      COMMON /LPFCOM/LAYTYP(200),LAYAVG(200),CHANI(200),LAYVKA(200),
     1               LAYWET(200)
    DIMENSION CR(NCOL,NROW,NLAY),CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY)
    DIMENSION ZETA(NCOL,NROW,NLAY,NZONES+1),BOTM(NCOL,NROW,0:NBOTM)
    DIMENSION IPLPOS(NCOL,NROW,NLAY,NZONES)
      DOUBLE PRECISION HNEW
      DIMENSION IBOUND(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     1 HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY)
    DIMENSION SWICR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICC(NCOL,NROW,NLAY,NZONES)
      DIMENSION SWICUMCR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICUMCC(NCOL,NROW,NLAY,NZONES),DELR(NCOL),DELC(NROW)
    REAL NUS(NZONES),EPS(NZONES),DELNUS(NZONES)
    REAL NUTOP(NCOL,NROW,NLAY),NUBOT(NCOL,NROW,NLAY)
    REAL QZEXTRA(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1-------SET FIRST AND LAST PLANE EQUAL TO TOP AND BOTTOM OF THE AQUIFER
      DO 90 K=1,NLAY
       DO 90 I=1,NROW
        DO 90 J=1,NCOL
         BBOT=BOTM(J,I,LBOTM(K))
         TTOP=BOTM(J,I,LBOTM(K)-1)
         IF(BBOT.GT.TTOP) THEN
            WRITE(IOUT,35) K,I,J
```

```
   35       FORMAT(1X,/1X,'Negative cell thickness at (layer,row,col)',
    1       I4,',',I4,',',I4)
            WRITE(IOUT,36) TTOP,BBOT
   36       FORMAT(1X,'Top elevation, bottom elevation:',1P,2G13.5)
            STOP
         END IF
         IF(LAYTYP(K).NE.0) THEN
            HHD=HNEW(J,I,K)
            IF(HHD.LT.TTOP) TTOP=HHD
         END IF
         ZETA(J,I,K,1)=TTOP
       ZETA(J,I,K,NZONES+1)=BBOT
   90  CONTINUE
C
C2----COMPUTE HORIZONTAL (CUMULATIVE) CONDUCTANCE FOR EACH ZONE
      DO 120 K=1,NLAY
       DO 120 I=1,NROW
        DO 120 J=1,NCOL
        SUMTHICKRF=0
          SUMTHICKFF=0
         DO IZ=1,NZONES
        IF(J.NE.NCOL) THEN
         DZETA1=ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1);
         DZETA2=ZETA(J+1,I,K,IZ)-ZETA(J+1,I,K,IZ+1);
         IF (DZETA1.LE.0 .OR. DZETA2.LE.0) THEN
            THICKRF=0
         ELSE
          THICKRF=(DZETA1*DELR(J+1)+DZETA2*DELR(J))/
     &                    (DELR(J)+DELR(J+1))
         END IF
         SWICR(J,I,K,IZ)=CR(J,I,K)*THICKRF
         SUMTHICKRF=SUMTHICKRF+THICKRF
        ELSE
         SWICR(J,I,K,IZ)=0
           SUMTHICKRF=1
        END IF
        IF(I.NE.NROW) THEN
         DZETA1=ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)
           DZETA2=ZETA(J,I+1,K,IZ)-ZETA(J,I+1,K,IZ+1)
         IF (DZETA1.LE.0 .OR. DZETA2.LE.0) THEN
            THICKFF=0
         ELSE
          THICKFF=(DZETA1*DELC(I+1)+DZETA2*DELC(I))/
     &                    (DELC(I)+DELC(I+1))
         END IF
         SWICC(J,I,K,IZ)=CC(J,I,K)*THICKFF
         SUMTHICKFF=SUMTHICKFF+THICKFF
        ELSE
           SWICC(J,I,K,IZ)=0
         SUMTHICKFF=1
        END IF
       END DO
         DO IZ=1,NZONES
```

```
          SWICR(J,I,K,IZ)=SWICR(J,I,K,IZ)/SUMTHICKRF
          SWICC(J,I,K,IZ)=SWICC(J,I,K,IZ)/SUMTHICKFF
       END DO
        SUMSWICR=0
          SUMSWICC=0
         DO IZ=1,NZONES
          SUMSWICR=SUMSWICR+SWICR(J,I,K,NZONES+1-IZ)
        SUMSWICC=SUMSWICC+SWICC(J,I,K,NZONES+1-IZ)
          SWICUMCR(J,I,K,NZONES+1-IZ)=SUMSWICR
        SWICUMCC(J,I,K,NZONES+1-IZ)=SUMSWICC
         END DO
  120 CONTINUE
C
C3------SET IPLPOS (1=top, 2=bottom, 0=in between)
      DO 130 K=1,NLAY
       DO 130 I=1,NROW
        DO 130 J=1,NCOL
       IPLPOS(J,I,K,1)=0
         DO 130 IZ=2,NZONES
        IF (ZETA(J,I,K,IZ).GE.ZETA(J,I,K,1)) THEN
         IPLPOS(J,I,K,IZ)=1
          ELSEIF (ZETA(J,I,K,IZ).LE.ZETA(J,I,K,NZONES+1)) THEN
           IPLPOS(J,I,K,IZ)=2
        ELSE
         IPLPOS(J,I,K,IZ)=0
        END IF
  130 CONTINUE
C
C4------SET ZONE CONDUCTANCES TO ZERO IF NOT CELL ON EITHER SIDE HAS ACTIVE ZONE
      DO 140 K=1,NLAY
       DO 140 I=1,NROW
        DO 140 J=1,NCOL
         DO 140 IZ=2,NZONES-1
        IF (J.NE.NCOL) THEN
           IF ((IPLPOS(J,I,K,IZ)+IPLPOS(J+1,I,K,IZ)+
    1    IPLPOS(J,I,K,IZ+1)+IPLPOS(J+1,I,K,IZ+1)).NE.0) THEN
            SWICR(J,I,K,IZ)=0
         END IF
        END IF
        IF (I.NE.NROW) THEN
           IF ((IPLPOS(J,I,K,IZ)+IPLPOS(J,I+1,K,IZ)+
    1    IPLPOS(J,I,K,IZ+1)+IPLPOS(J,I+1,K,IZ+1)).NE.0) THEN
            SWICC(J,I,K,IZ)=0
         END IF
        END IF
  140 CONTINUE
C
C5-------COMPUTE NUTOP AND NUBOT
      DO 150 K=1,NLAY
       DO 150 I=1,NROW
        DO 150 J=1,NCOL
       NUTOP(J,I,K)=NUS(1)
         NUBOT(J,I,K)=NUS(NZONES)
```

```
      DO 150 IZ=2,NZONES
       IF (IPLPOS(J,I,K,IZ).EQ.1) THEN
        NUTOP(J,I,K)=NUTOP(J,I,K)+DELNUS(IZ)
       END IF
        IZREV=NZONES-IZ+2
       IF (IPLPOS(J,I,K,IZREV).EQ.2) THEN
        NUBOT(J,I,K)=NUBOT(J,I,K)-DELNUS(IZREV)
       END IF
  150 CONTINUE
C
C6-------COMPUTE QZEXTRA
    DO 160 K=1,NLAY
      DO 160 I=1,NROW
       DO 160 J=1,NCOL
      QZEXTRA(J,I,K)=0
      HEADDIFF=0
      IF (K.NE.1) THEN
       DO IZ=2,NZONES+1
        HEADDIFF=HEADDIFF-
     &  NUS(IZ-1)*(ZETA(J,I,K-1,IZ)-ZETA(J,I,K-1,IZ-1))
         END DO
        QZEXTRA(J,I,K)=-CV(J,I,K-1)*
     &   (HEADDIFF+0.5*(ZETA(J,I,K-1,NZONES+1)-ZETA(J,I,K,1))
     &                      *(NUBOT(J,I,K-1)+NUTOP(J,I,K)))
      END IF
  160 CONTINUE
C
C7------ADD QZEXTRA TO RHS
      DO 170 K=1,NLAY
      DO 170 I=1,NROW
       DO 170 J=1,NCOL
       RHS(J,I,K)=RHS(J,I,K)+QZEXTRA(J,I,K)
      IF (K.NE.NLAY) THEN
       RHS(J,I,K)=RHS(J,I,K)-QZEXTRA(J,I,K+1)
        END IF
  170 CONTINUE
C
C8-----ADD TO RHS DUE TO DENSITY DIFFERENCES
      DO 180 K=1,NLAY
      DO 180 I=1,NROW
       DO 180 J=1,NCOL
      DO 180 IZ=1,NZONES

C -------LEFT FACE--------------------
       IF (J.NE.1) THEN
         RHS(J,I,K)=RHS(J,I,K)-DELNUS(IZ)*
     &       SWICUMCR(J-1,I,K,IZ)*(ZETA(J-1,I,K,IZ)-ZETA(J,I,K,IZ))+
     &         EPS(IZ)*SWICR(J-1,I,K,IZ)*(
     &       (ZETA(J-1,I,K,IZ)-ZETA(J-1,I,K,IZ+1))-
     &          (ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)))
       END IF

C -------RIGHT FACE--------------------
```

```
        IF (J.NE.NCOL) THEN
          RHS(J,I,K)=RHS(J,I,K)-DELNUS(IZ)*
     &        SWICUMCR(J,I,K,IZ)*(ZETA(J+1,I,K,IZ)-ZETA(J,I,K,IZ))+
     &          EPS(IZ)*SWICR(J,I,K,IZ)*(
     &        (ZETA(J+1,I,K,IZ)-ZETA(J+1,I,K,IZ+1))-
     &           (ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)))
        END IF

C -------BACK FACE--------------------
        IF (I.NE.1) THEN
          RHS(J,I,K)=RHS(J,I,K)-DELNUS(IZ)*
     &        SWICUMCC(J,I-1,K,IZ)*(ZETA(J,I-1,K,IZ)-ZETA(J,I,K,IZ))+
     &          EPS(IZ)*SWICC(J,I-1,K,IZ)*(
     &        (ZETA(J,I-1,K,IZ)-ZETA(J,I-1,K,IZ+1))-
     &           (ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)))
        END IF

C -------FRONT FACE--------------------
        IF (I.NE.NROW) THEN
          RHS(J,I,K)=RHS(J,I,K)-DELNUS(IZ)*
     &        SWICUMCC(J,I,K,IZ)*(ZETA(J,I+1,K,IZ)-ZETA(J,I,K,IZ))+
     &          EPS(IZ)*SWICC(J,I,K,IZ)*(
     &        (ZETA(J,I+1,K,IZ)-ZETA(J,I+1,K,IZ+1))-
     &           (ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)))

        END IF
  180 CONTINUE
C9------RETURN
      RETURN
      END
```

1.6 Module SWI1BD

Variables used in module SWI1BD

```
Variable    Range   Description
-------------------------------
B           module  DIMENSION (MXEQ), the vector {b}.
BRHS        module  Right Hand Side values for one zone.
CC          global  Branch conductance in column direction
CR          global  Branch conductance in row direction
CV          global  Branch conductance in vertical direction
DELC        global  DIMENSION (NROW) cell width
DELNUS      package difference in NUS between zones
DELR        global  DIMENSION (NCOL) cell width
DELT        global  Length of the current time step.
DELZETA     package Elevation for a plane when it is moved into an adjacent empty cell
EPS         package Epsilon, variation of density within zone
HCLOSE      global  head change closure criterion
HCOF        global  DIMENSION (NCOL,NROW,NLAY), Coefficient of head in the finite-difference
HDCG        module  DIMENSION (MXITER), Maximum head change for each iteration.
HNEW        global  DIMENSION (NCOL,NROW,NLAY), Most recent estimate of head in each cell
I           module  Loop variable for rows
IBO         module  Logical variable IBO equals one for active cells zero otherwise
```

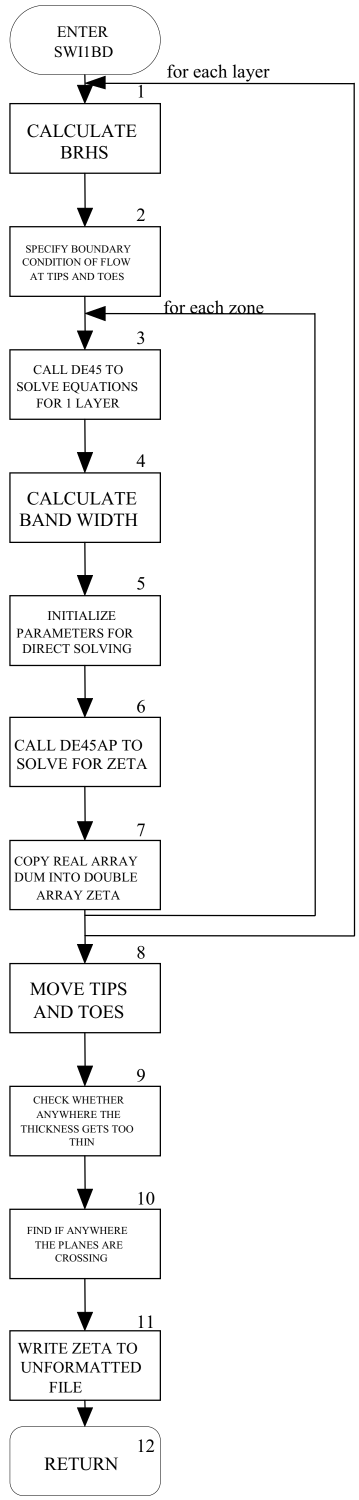

Figure 5: Flowchart for module SWI1BD

IBOUND	global	DIMENSION (NCOL,NROW,NLAY), Status of each cell
ICNVG	global	Flag that is set to one when convergence has occurred.
ICOUNT	module	Counter to calculate average
ID4DIM	package	5 -- for a 2-dimensional grid.7 -- for a 3-dimensional grid.
ID4DIR	package	Flag that indicates the relative size of NCOL, NROW, and NLAY
IEQP	package	DIMENSION (NCOL,NROW,NLAY), D4 order number for model cells.
IFREQ	package	indicates the frequency at which stress and conductance terms in [A] change.
IOUT	global	Primary unit number for all printed output.
IPLPOS	package	Flag indicating plane position (1=top, 2=bottom, 0=in between)
ISWIZT	package	Flag and Unit number, if ISWIZT>0 unit number for ZETA else ZETA will not be recorded
IZ	module	Loop variable for zones
IZ2	module	Loop variable for zones
IZ3	module	Loop variable for zones
IZREV	module	Reverse loop variable for zones
J	module	Loop variable for columns
K	module	Loop variable for layers
KKITER	global	Current iteration step
KKPER	global	Current stress period
KKSTP	global	Loop variable for timesteps
KSTP	global	Loop variable for timesteps
LRCH	package	Layer, row, and column of the cell containing the maximum head change (HDCG) for each iteration.
MUTD4	package	Print flag
MXBW	package	the maximum band width plus 1 of the [AL] matrix
MXEQ	package	MXUP+MXLOW
MXLOW	package	The maximum allowed number of equations in the lower part of [A].
MXUP	package	The maximum allowed number of equations in the upper part of [A].
NBWGRD	package	One plus the product of the two smallest grid dimensions.
NCOL	global	Number of columns
NHALFL	package	the maximum possible number of equations in the lower part of [A]
NHALFU	package	the maximum possible number of equations in the upper part of [A]
NITER	global	the maximum number of internal iterations
NLAY	global	Number of layers
NLAYSWI	module	One
NODES	global	Number of nodes
NPRS	package	Print out interval
NROW	global	Number of rows
NSTP	global	Number of timesteps
NUBELBOT	module	Value of NU below the current zone
NUBOT	module	Value of NU below the lowest zone
NUONTOP	module	Value of NU above the upper zone
NUS	module	The values of NUS
NUTOP	module	Value of NU of the upper zone
NZONES	package	Number of zones
Q	module	Discharge between two adjacent cells
QZBOT	module	Density flow through bottom of current cell
QZEXTRA	module	Extra discharge due to headdiff
QZTOP	module	Density flow through top of current cell
RHS	global	DIMENSION (NCOL,NROW,NLAY), Right-hand side of the finite-difference equation.
SSZ	package	Effective porosity
SWICC	package	Conductance in column direction for each zone
SWICR	package	Conductance in row direction for each zone
SWICUMCC	package	Cumulative conductance in column direction for each zone
SWICUMCR	package	Cumulative conductance in row direction for each zone

```
SWIDE45CC   package Conductance in column direction for one layer
SWIDE45CR   package Conductance in row direction for one layer
SWIDELT     module  one
SWIHCOF     package HCOF for one layer
TIPSLOPE    package Maximum slope of tipcells
TOESLOPE    package Maximum slope of toecells
ZB          module  Zeta of the first zone
ZETA        package Elevation of the planes
ZETAAVG     module  Average zeta of crossing planes
ZETAC       module  Zeta of the current cell
ZETAMIN     package Minimum elevation of a plane before it is removed from a cell
ZT          module  Zeta of the last zone
```

FORTRAN code of module SWI1BD

```
      SUBROUTINE SWI1BD(DELT,TOESLOPE,TIPSLOPE,ZETAMIN,DELZETA,
     &  EPS,NUS,DELNUS,IBOUND,HNEW,CR,CC,
     &  CV,HCOF,RHS,NCOL,NROW,NLAY,NZONES,IPLPOS,ZETA,SSZ,SWICR,SWICC,
     &  SWICUMCR,SWICUMCC,DELR,DELC,NUTOP,NUBOT,BRHS,
     &  AU,IUPP,AL,IEQP,
     &  B,LRCH,HDCG,IBO,SWIHCOF,QZEXTRA,
     &  SWIDE45CR,SWIDE45CC,SWIDE45CV,DUM,IOUT,ISWIZT,NPRS,NSTP,KSTP)

C-----VERSION 1 9AUG2002 SWI1BD
C     ******************************************************************
C     CALCULATE AND SAVE ZETA FOR SWI PACKAGE
C     ******************************************************************
C
C     SPECIFICATIONS:
      DIMENSION BRHS(NCOL,NROW,NZONES),SSZ(NCOL,NROW,NLAY)
    DIMENSION CR(NCOL,NROW,NLAY),CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY)
    DIMENSION ZETA(NCOL,NROW,NLAY,NZONES+1)
    DIMENSION IPLPOS(NCOL,NROW,NLAY,NZONES)
      DOUBLE PRECISION HNEW
      DIMENSION IBOUND(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     1 HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY)
    DIMENSION SWICR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICC(NCOL,NROW,NLAY,NZONES)
      DIMENSION SWICUMCR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICUMCC(NCOL,NROW,NLAY,NZONES),DELR(NCOL),DELC(NROW)
    REAL NUTOP(NCOL,NROW,NLAY)
    REAL NUBOT(NCOL,NROW,NLAY),QZEXTRA(NCOL,NROW,NLAY)
    REAL SSZ DELT
    DIMENSION IBO(NCOL,NROW),SWIHCOF(NCOL,NROW)
    REAL SWIDE45CR(NCOL,NROW),SWIDE45CC(NCOL,NROW)
    REAL SWIDE45CV(NCOL,NROW)
    DOUBLE PRECISION DUM(NCOL,NROW)
      REAL NUS(NZONES),DELNUS(NZONES),EPS(NZONES)
    REAL NUBELBOT, NUONTOP
C
C1-------CALCULATE BRHS
      DO 90 K=1,NLAY
       DO 100 I=1,NROW
        DO 100 J=1,NCOL
```

```
       DO 100 IZ=1,NZONES
        BRHS(J,I,IZ)=0.
        IF (IZ.NE.1) THEN
           IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
            BRHS(J,I,IZ)= - SSZ(J,I,K)*DELR(J)*DELC(I)*ZETA(J,I,K,IZ)
     &                     /DELT
            CALL SWI1SD(BRHS(1,1,IZ),IPLPOS(1,1,K,IZ),
     &              SWICUMCR(1,1,K,IZ),SWICUMCC(1,1,K,IZ),1.,
     &              HNEW(1,1,K),J,I,NCOL,NROW)
            DO IZ2=1,IZ-1
             CALL SWI1SR(BRHS(1,1,IZ),IPLPOS(1,1,K,IZ),
     &               SWICUMCR(1,1,K,IZ),SWICUMCC(1,1,K,IZ),DELNUS(IZ2),
     &               ZETA(1,1,K,IZ2),J,I,NCOL,NROW)
            END DO
           END IF
            CALL SWI1SR(BRHS(1,1,IZ),IPLPOS(1,1,K,IZ),SWICR(1,1,K,IZ),
     &         SWICC(1,1,K,IZ),EPS(IZ),ZETA(1,1,K,IZ+1),J,I,NCOL,NROW)


C-----------Zones IZ+1 through bottom
            DO IZ2=IZ+1,NZONES
             IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
              CALL SWI1SR(BRHS(1,1,IZ),IPLPOS(1,1,K,IZ),
     &              SWICUMCR(1,1,K,IZ2),SWICUMCC(1,1,K,IZ2),DELNUS(IZ2),
     &              ZETA(1,1,K,IZ2),J,I,NCOL,NROW)

             END IF

C -------LEFT FACE--------------------
             IF (J.NE.1) THEN
              BRHS(J,I,IZ) = BRHS(J,I,IZ)+EPS(IZ2)*
     &       SWICR(J-1,I,K,IZ2)*((ZETA(J-1,I,K,IZ2)-ZETA(J,I,K,IZ2))-
     &                         (ZETA(J-1,I,K,IZ2+1)-ZETA(J,I,K,IZ2+1)))
           END IF

C -------RIGHT FACE--------------------
           IF (J.NE.NCOL) THEN
              BRHS(J,I,IZ) = BRHS(J,I,IZ)+EPS(IZ2)*
     &       SWICR(J,I,K,IZ2)*((ZETA(J+1,I,K,IZ2)-ZETA(J,I,K,IZ2))-
     &                         (ZETA(J+1,I,K,IZ2+1)-ZETA(J,I,K,IZ2+1)))
           END IF

C -------BACK FACE--------------------
           IF (I.NE.1) THEN
              BRHS(J,I,IZ) = BRHS(J,I,IZ)+EPS(IZ2)*
     &       SWICC(J,I-1,K,IZ2)*((ZETA(J,I-1,K,IZ2)-ZETA(J,I,K,IZ2))-
     &                         (ZETA(J,I-1,K,IZ2+1)-ZETA(J,I,K,IZ2+1)))
           END IF

C -------FRONT FACE--------------------
           IF (I.NE.NROW) THEN
              BRHS(J,I,IZ) = BRHS(J,I,IZ)+EPS(IZ2)*
     &       SWICC(J,I,K,IZ2)*((ZETA(J,I+1,K,IZ2)-ZETA(J,I,K,IZ2))-
```

```
     &                        (ZETA(J,I+1,K,IZ2+1)-ZETA(J,I,K,IZ2+1)))
           END IF
          END DO
C
C2--------SPECIFY BOUNDARY CONDITION OF FLOW AT TIPS AND TOES

          IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
C -------LEFT FACE--------------------
             IF (J.NE.1) THEN
            IF (IPLPOS(J-1,I,K,IZ).NE.0) THEN
               CALL SWI1SQR (Q,HNEW,ZETA,NZONES,DELNUS,EPS,
     &                 SWICUMCR,SWICR,J,I,K,IZ,NCOL,NROW,NLAY)
             BRHS(J,I,IZ)=BRHS(J,I,IZ)-Q
                END IF
           END IF

C -------RIGHT FACE--------------------
           IF (J.NE.NCOL) THEN
            IF (IPLPOS(J+1,I,K,IZ).NE.0) THEN
               CALL SWI1SQR (Q,HNEW,ZETA,NZONES,DELNUS,EPS,
     &                 SWICUMCR,SWICR,J+1,I,K,IZ,NCOL,NROW,NLAY)
             BRHS(J,I,IZ)=BRHS(J,I,IZ)+Q
            END IF
           END IF

C -------BACK FACE--------------------
               IF (I.NE.1) THEN
            IF (IPLPOS(J,I-1,K,IZ).NE.0) THEN
               CALL SWI1SQC (Q,HNEW,ZETA,NZONES,DELNUS,EPS,
     &                 SWICUMCC,SWICC,J,I,K,IZ,NCOL,NROW,NLAY)
             BRHS(J,I,IZ)=BRHS(J,I,IZ)-Q
            END IF
           END IF

C -------FRONT FACE--------------------
           IF (I.NE.NROW) THEN
            IF (IPLPOS(J,I+1,K,IZ).NE.0) THEN
               CALL SWI1SQC (Q,HNEW,ZETA,NZONES,DELNUS,EPS,
     &                 SWICUMCC,SWICC,J,I+1,K,IZ,NCOL,NROW,NLAY)
             BRHS(J,I,IZ)=BRHS(J,I,IZ)+Q
            END IF
           END IF

C -------UPPER FACE--------------------
           IF (K.NE.1) THEN
              QZTOP=CV(J,I,K-1)*(HNEW(J,I,K)-HNEW(J,I,K-1))+
     &                                QZEXTRA(J,I,K)
            NUONTOP=NUBOT(J,I,K-1)
            IF ((QZTOP.LT.0).AND.(NUONTOP.GE.NUS(IZ))) THEN
               BRHS(J,I,IZ)=BRHS(J,I,IZ)+QZTOP
              END IF
           END IF
```

```
C -------LOWER FACE--------------------
           IF (K.NE.NLAY) THEN
              QZBOT=CV(J,I,K)*(HNEW(J,I,K)-HNEW(J,I,K+1))+
     &                             QZEXTRA(J,I,K+1)
            NUBELBOT=NUTOP(J,I,K+1)
            IF ((QZBOT.GT.0).AND.(NUBELBOT.GE.NUS(IZ))) THEN
               BRHS(J,I,IZ)=BRHS(J,I,IZ)-QZBOT
              END IF
           END IF
         END IF
        END IF
  100 CONTINUE
C
C3----- CALL DE45 TO SOLVE EQUATIONS FOR 1 LAYER
    DO 115 IZ=2,NZONES
       DO 110 I=1,NROW
        DO 110 J=1,NCOL
       IF (J.NE.NCOL) THEN
         IF ((IPLPOS(J,I,K,IZ).EQ.0).AND.(IPLPOS(J+1,I,K,IZ).EQ.0)) THEN
          SWIDE45CR(J,I)=DELNUS(IZ)*SWICUMCR(J,I,K,IZ)
         ELSE
          SWIDE45CR(J,I)=0.
       END IF
       END IF
         SWIDE45CR(J,I)=SWIDE45CR(J,I)-EPS(IZ)*SWICR(J,I,K,IZ)

       IF (I.NE.NROW) THEN
         IF ((IPLPOS(J,I,K,IZ).EQ.0).AND.(IPLPOS(J,I+1,K,IZ).EQ.0)) THEN
          SWIDE45CC(J,I)=DELNUS(IZ)*SWICUMCC(J,I,K,IZ)
         ELSE
          SWIDE45CC(J,I)=0.
       END IF
       END IF
         SWIDE45CC(J,I)=SWIDE45CC(J,I)-EPS(IZ)*SWICC(J,I,K,IZ)

       SWIDE45CV(J,I)=0
       SWIHCOF(J,I)=-SSZ(J,I,K)*DELR(J)*DELC(I)/DELT
       DUM(J,I)=ZETA(J,I,K,IZ)

         IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
        IBO(J,I)=1
       ELSE
          IBO(J,I)=0
       END IF
  110 CONTINUE
C
C4------COPIED FROM DE45 PACKAGE SUBSTITUTE NLAY BY 1
C4------CALCULATE BAND WIDTH
      NLAYSWI=1
      NODES=NCOL*NROW*NLAYSWI
      NHALFU=(NODES-1)/2 +1
      NHALFL=NODES-NHALFU
      ID4DIM=7
```

```
      IF(NLAYSWI.LE.NCOL .AND. NLAYSWI.LE.NROW) THEN
         IF(NLAYSWI.EQ.1) ID4DIM=5
         IF(NCOL.GE.NROW) THEN
            ID4DIR=1
            NBWGRD=NROW*NLAYSWI+1
         ELSE
            ID4DIR=2
            NBWGRD=NCOL*NLAYSWI+1
         END IF
      ELSE IF(NROW.LE.NCOL .AND. NROW.LE.NLAYSWI) THEN
         IF(NROW.EQ.1) ID4DIM=5
         IF(NCOL.GE.NLAYSWI) THEN
            ID4DIR=3
            NBWGRD=NROW*NLAYSWI+1
         ELSE
            ID4DIR=4
            NBWGRD=NROW*NCOL+1
         END IF
      ELSE
         IF(NCOL.EQ.1) ID4DIM=5
         IF(NROW.GE.NLAYSWI) THEN
            ID4DIR=5
            NBWGRD=NCOL*NLAYSWI+1
         ELSE
            ID4DIR=6
            NBWGRD=NCOL*NROW+1
         END IF
      END IF
      MXUP=NHALFU
      MXLOW=NHALFL
C5------INITIALIZE PARAMETERS FOR DIRECT SOLVING
    MXBW=NBWGRD + 4
      MXEQ=MXUP+MXLOW
      ITMX=1
    MXITER=1
    NITER=1
    ACCL=1.0
    KKITER=1
    HCLOSE=0.00001
    IPRD4=0
    IFREQ=3
    SWIDELT=1.
C



C6-------CALL DE45AP TO SOLVE FOR ZETA
      CALL DE45AP(DUM,IBO,AU,AL,
     &           IUPP,IEQP,B,MXUP,
     &           MXLOW,MXEQ,MXBW,SWIDE45CR,SWIDE45CC,
     &           SWIDE45CV,SWIHCOF,BRHS(1,1,IZ),ACCL,
     &           KKITER,ITMX,MXITER,NITER,HCLOSE,IPRD4,
     &           ICNVG,NCOL,NROW,NLAYSWI,IOUT,LRCH,
```

```
     &            HDCG,IFREQ,KKSTP,KKPER,SWIDELT,
     &            NSTP,ID4DIR,ID4DIM,MUTD4,IERR,
     &            IERRU)
C
C7-------COPY REAL ARRAY DUM INTO DOUBLE ARRAY ZETA
       DO 95 I=1,NROW
        DO 95 J=1,NCOL
       ZETA(J,I,K,IZ)=DUM(J,I)
   95 CONTINUE
  115 CONTINUE
   90 CONTINUE
C
C8-----MOVE TIPS AND TOES
C8-----NOTE: MAY DIFFER FROM MATLAB PROTOTYPE BECAUSE ZETA IS
C8-----ALTERED DURING THE SEARCH PROCESS
      DO 120 K=1,NLAY
     DO 120 IZ=2,NZONES
        DO 120 I=1,NROW
         DO 120 J=1,NCOL
        ZT=ZETA(J,I,K,1)
        ZB=ZETA(J,I,K,NZONES+1)
        ZETAC=ZETA(J,I,K,IZ)

       IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
          IF ((J.NE.1).AND.(J.NE.NCOL)) THEN
C -------LEFT FACE--------------------
         IF (IPLPOS(J-1,I,K,IZ).EQ.1) THEN
            IF ((ZT-ZETAC).GT.(TIPSLOPE*0.5*(DELR(J)+DELR(J-1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC+DELZETA*DELR(J-1)/DELR(J)
             ZETA(J-1,I,K,IZ)=ZETA(J-1,I,K,1)-DELZETA
          ELSEIF ((ZT-ZETAC).LT.ZETAMIN) THEN
             ZETA(J+1,I,K,IZ)=ZETA(J+1,I,K,IZ)-(ZT-ZETAC)
             ZETA(J,I,K,IZ)=ZT
            END IF
           ELSEIF (IPLPOS(J-1,I,K,IZ).EQ.2) THEN
            IF (ZETAC.GT.(ZB+TOESLOPE*0.5*(DELR(J)+DELR(J-1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC-DELZETA*DELR(J-1)/DELR(J)
             ZETA(J-1,I,K,IZ)=ZETA(J-1,I,K,NZONES+1)+DELZETA
          ELSEIF ((ZETAC-ZB).LT.ZETAMIN) THEN
             ZETA(J+1,I,K,IZ)=ZETA(J+1,I,K,IZ)+(ZETAC-ZB)
             ZETA(J,I,K,IZ)=ZB
            END IF
             END IF
C -------RIGHT FACE--------------------
         IF (IPLPOS(J+1,I,K,IZ).EQ.1) THEN
            IF ((ZT-ZETAC).GT.(TIPSLOPE*0.5*(DELR(J)+DELR(J+1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC+DELZETA*DELR(J+1)/DELR(J)
             ZETA(J+1,I,K,IZ)=ZETA(J+1,I,K,1)-DELZETA
          ELSEIF ((ZT-ZETAC).LT.ZETAMIN) THEN
             ZETA(J-1,I,K,IZ)=ZETA(J-1,I,K,IZ)-(ZT-ZETAC)
             ZETA(J,I,K,IZ)=ZT
            END IF
           ELSEIF (IPLPOS(J+1,I,K,IZ).EQ.2) THEN
```

```
            IF (ZETAC.GT.(ZB+TOESLOPE*0.5*(DELR(J)+DELR(J+1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC-DELZETA*DELR(J+1)/DELR(J)
             ZETA(J+1,I,K,IZ)=ZETA(J+1,I,K,NZONES+1)+DELZETA
          ELSEIF ((ZETAC-ZB).LT.ZETAMIN) THEN
             ZETA(J-1,I,K,IZ)=ZETA(J-1,I,K,IZ)+(ZETAC-ZB)
             ZETA(J,I,K,IZ)=ZB
            END IF
             END IF

        END IF

          IF ((I.NE.1).AND.(I.NE.NROW)) THEN
C -------BACK FACE--------------------
         IF (IPLPOS(J,I-1,K,IZ).EQ.1) THEN
            IF ((ZT-ZETAC).GT.(TIPSLOPE*0.5*(DELC(I)+DELC(I-1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC+DELZETA*DELC(I-1)/DELC(I)
             ZETA(J,I-1,K,IZ)=ZETA(J,I-1,K,1)-DELZETA
          ELSEIF ((ZT-ZETAC).LT.ZETAMIN) THEN
             ZETA(J,I+1,K,IZ)=ZETA(J,I+1,K,IZ)-(ZT-ZETAC)
             ZETA(J,I,K,IZ)=ZT
            END IF
           ELSEIF (IPLPOS(J,I-1,K,IZ).EQ.2) THEN
            IF (ZETAC.GT.(ZB+TOESLOPE*0.5*(DELC(I)+DELC(I-1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC-DELZETA*DELC(I-1)/DELC(I)
             ZETA(J,I-1,K,IZ)=ZETA(J,I-1,K,NZONES+1)+DELZETA
          ELSEIF ((ZETAC-ZB).LT.ZETAMIN) THEN
             ZETA(J,I+1,K,IZ)=ZETA(J,I+1,K,IZ)+(ZETAC-ZB)
             ZETA(J,I,K,IZ)=ZB
            END IF
             END IF
C -------FRONT FACE--------------------
         IF (IPLPOS(J,I+1,K,IZ).EQ.1) THEN
            IF ((ZT-ZETAC).GT.(TIPSLOPE*0.5*(DELC(I)+DELC(I+1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC+DELZETA*DELC(I+1)/DELC(I)
             ZETA(J,I+1,K,IZ)=ZETA(J,I+1,K,1)-DELZETA
          ELSEIF ((ZT-ZETAC).LT.ZETAMIN) THEN
             ZETA(J,I-1,K,IZ)=ZETA(J,I-1,K,IZ)-(ZT-ZETAC)
             ZETA(J,I,K,IZ)=ZT
            END IF
           ELSEIF (IPLPOS(J,I+1,K,IZ).EQ.2) THEN
            IF (ZETAC.GT.(ZB+TOESLOPE*0.5*(DELC(I)+DELC(I+1)))) THEN
             ZETA(J,I,K,IZ)=ZETAC-DELZETA*DELC(I+1)/DELC(I)
             ZETA(J,I+1,K,IZ)=ZETA(J,I+1,K,NZONES+1)+DELZETA
          ELSEIF ((ZETAC-ZB).LT.ZETAMIN) THEN
             ZETA(J,I-1,K,IZ)=ZETA(J,I-1,K,IZ)+(ZETAC-ZB)
             ZETA(J,I,K,IZ)=ZB
            END IF
             END IF

        END IF

         END IF
  120 CONTINUE
```

```
C
C9------CHECK WHETHER ANYWHERE THE THICKNESS GETS TOO THIN

      DO 130 K=1,NLAY
       DO 130 I=1,NROW
        DO 130 J=1,NCOL
       DO 130 IZ=2,NZONES
        IF (IPLPOS(J,I,K,IZ).EQ.0) THEN
           IF ((ZETA(J,I,K,IZ)-ZETA(J,I,K,NZONES+1)).LT.ZETAMIN) THEN
              ZETA(J,I,K,IZ)=ZETA(J,I,K,NZONES+1)
           END IF
           IF ((ZETA(J,I,K,1)-ZETA(J,I,K,IZ)).LT.ZETAMIN) THEN
              ZETA(J,I,K,IZ)=ZETA(J,I,K,1)
           END IF
        END IF
  130 CONTINUE
C
C10------FIND IF ANYWHERE THE PLANES ARE CROSSING
      DO 150 K=1,NLAY
       DO 150 I=1,NROW
        DO 150 J=1,NCOL
       DO 150 IZ=2,NZONES-1
          IF ((ZETA(J,I,K,IZ)-ZETA(J,I,K,IZ+1)).LT.0.001) THEN
             ZETAAVG=0.5*(ZETA(J,I,K,IZ)+ZETA(J,I,K,IZ+1))
             ZETA(J,I,K,IZ)=ZETAAVG
             ZETA(J,I,K,IZ+1)=ZETAAVG
           DO IZ2=2,IZ-1
              IZREV=IZ+1-IZ2
              IF ((ZETA(J,I,K,IZREV)-ZETA(J,I,K,IZ+1)).LT.0.001) THEN
               ZETAAVG=0.
             ICOUNT=0
               DO IZ3=IZREV,IZ+1
                ICOUNT=ICOUNT+1
                ZETAAVG=ZETAAVG + ZETA(J,I,K,IZ3)
             END DO
               IF (ICOUNT.NE.0) THEN
                ZETAAVG=ZETAAVG/ICOUNT
                DO IZ3=IZREV,IZ+1
                 ZETA(J,I,K,IZ3)=ZETAAVG
                END DO
             END IF
            END IF
           END DO
          END IF
  150 CONTINUE
C
C11------WRITE ZETA TO UNFORMATTED FILE
    IF (ISWIZT.NE.0) THEN
     IF (NPRS.GE.0) THEN
      IF(MOD(KSTP,NPRS).EQ.0) THEN
       WRITE (ISWIZT) ZETA
        ELSEIF ((NPRS.EQ.0).AND.(KSTP.EQ.NSTP)) THEN
       WRITE (ISWIZT) ZETA
```

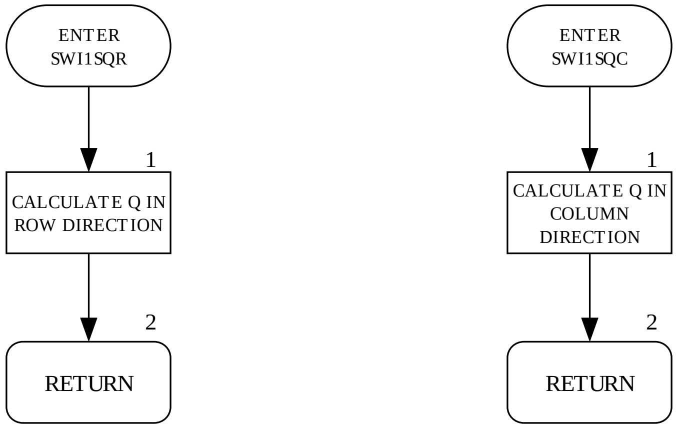

Figure 6: Flowchart of submodules SWI1SQR (left) and SWI1SQC (right)

```
        END IF
     END IF
    END IF
C
C12-----RETURN
      RETURN
      END
```

1.7 Submodules SWI1SQR and SWI1SQC

Fortran code of submodules SWI1SQR and SWI1SQC

```
      SUBROUTINE SWI1SQR (Q,HNEW,ZETA,NZONES,DELNUS,EPS,SWICUMCR,SWICR,
     &                                        J,I,K,IZ,NCOL,NROW,NLAY)


C     ****************************************************************
C     SWI
C     ****************************************************************
C
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
    DIMENSION ZETA(NCOL,NROW,NLAY,NZONES+1)
    DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY)
    DIMENSION SWICR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICC(NCOL,NROW,NLAY,NZONES)
      DIMENSION SWICUMCR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICUMCC(NCOL,NROW,NLAY,NZONES)
    REAL NUS(NZONES),EPS(NZONES),DELNUS(NZONES)
C
C1-----CALCULATE Q IN ROW DIRECTION
      Q=SWICUMCR(J-1,I,K,IZ)*(HNEW(J-1,I,K)-HNEW(J,I,K))

    DO IZ2=1,IZ
```

```
       Q=Q+SWICUMCR(J-1,I,K,IZ)*DELNUS(IZ)*
     &                 (ZETA(J-1,I,K,IZ2)-ZETA(J,I,K,IZ2))
    END DO

      Q=Q-SWICR(J-1,I,K,IZ)*EPS(IZ)*
     &                 (ZETA(J-1,I,K,IZ)-ZETA(J,I,K,IZ))
      Q=Q+SWICR(J-1,I,K,IZ)*EPS(IZ)*
     &                 (ZETA(J-1,I,K,IZ+1)-ZETA(J,I,K,IZ+1))


      DO IZ2=IZ+1,NZONES
       Q=Q+SWICUMCR(J-1,I,K,IZ2)*DELNUS(IZ2)*
     &                 (ZETA(J-1,I,K,IZ2)-ZETA(J,I,K,IZ2))
       Q=Q-SWICR(J-1,I,K,IZ2)*EPS(IZ2)*
     &                 (ZETA(J-1,I,K,IZ2)-ZETA(J,I,K,IZ2))
       Q=Q+SWICR(J-1,I,K,IZ2)*EPS(IZ2)*
     &                 (ZETA(J-1,I,K,IZ2+1)-ZETA(J,I,K,IZ2+1))
    END DO
C     -----------------------------------------------------------------
C
C2------RETURN
      RETURN
      END



      SUBROUTINE SWI1SQC (Q,HNEW,ZETA,NZONES,DELNUS,EPS,SWICUMCC,SWICC,
     &                                       J,I,K,IZ,NCOL,NROW,NLAY)


C     *****************************************************************
C     SWI
C     *****************************************************************
C
C     SPECIFICATIONS:
C     -----------------------------------------------------------------
    DIMENSION ZETA(NCOL,NROW,NLAY,NZONES+1)
    DOUBLE PRECISION HNEW
      DIMENSION HNEW(NCOL,NROW,NLAY)
    DIMENSION SWICR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICC(NCOL,NROW,NLAY,NZONES)
      DIMENSION SWICUMCR(NCOL,NROW,NLAY,NZONES)
    DIMENSION SWICUMCC(NCOL,NROW,NLAY,NZONES)
    REAL NUS(NZONES),EPS(NZONES),DELNUS(NZONES)
C
C1-----CALCULATE Q IN COLUMN DIRECTION
      Q=SWICUMCC(J,I-1,K,IZ)*(HNEW(J,I-1,K)-HNEW(J,I,K))

    DO IZ2=1,IZ
       Q=Q+SWICUMCC(J,I-1,K,IZ)*DELNUS(IZ)*
     &                 (ZETA(J,I-1,K,IZ2)-ZETA(J,I,K,IZ2))
    END DO
```

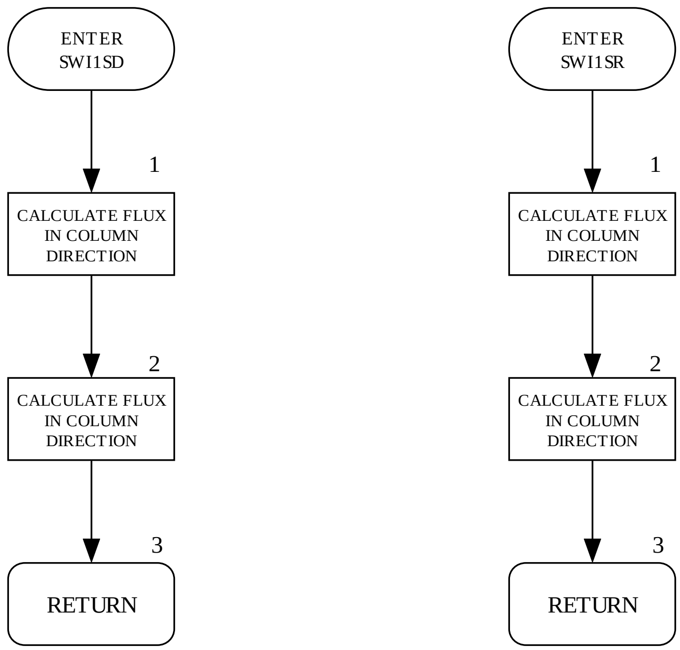

Figure 7: Flowchart of submodules SSWI1D (left) and SSWI1R (right)

```
      Q=Q-SWICC(J,I-1,K,IZ)*EPS(IZ)*
     &                 (ZETA(J,I-1,K,IZ)-ZETA(J,I,K,IZ))
      Q=Q+SWICC(J,I-1,K,IZ)*EPS(IZ)*
     &                 (ZETA(J,I-1,K,IZ+1)-ZETA(J,I,K,IZ+1))


      DO IZ2=IZ+1,NZONES
       Q=Q+SWICUMCC(J,I-1,K,IZ2)*DELNUS(IZ2)*
     &                 (ZETA(J,I-1,K,IZ2)-ZETA(J,I,K,IZ2))
       Q=Q-SWICC(J,I-1,K,IZ2)*EPS(IZ2)*
     &                 (ZETA(J,I-1,K,IZ2)-ZETA(J,I,K,IZ2))
       Q=Q+SWICC(J,I-1,K,IZ2)*EPS(IZ2)*
     &                 (ZETA(J,I-1,K,IZ2+1)-ZETA(J,I,K,IZ2+1))
    END DO
C     -----------------------------------------------------------------
C
C2------RETURN
      RETURN
      END
```

1.8 Submodules SSWI1D and SSWI1R

FORTRAN code of submodules SSWI1D and SSWI1R

```
      SUBROUTINE SWI1SD(B,IPOS,CRLAY,CCLAY,FAC,VAR,J,I,NCOL,NROW)

C     *****************************************************************
C     SWI
```

```
C     ****************************************************************
C
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
    DIMENSION B(NCOL,NROW),CRLAY(NCOL,NROW),CCLAY(NCOL,NROW)
    DIMENSION IPOS(NCOL,NROW)
    DOUBLE PRECISION VAR(NCOL,NROW)
C
C1-----CALCULATE FLUX IN COLUMN DIRECTION
C -------LEFT FACE--------------------
          IF (J.NE.1) THEN
         IF (IPOS(J-1,I).EQ.0) THEN
            B(J,I) = B(J,I)- FAC*CRLAY(J-1,I)*(VAR(J-1,I)-VAR(J,I))
         END IF
        END IF

C -------RIGHT FACE--------------------
        IF (J.NE.NCOL) THEN
         IF (IPOS(J+1,I).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CRLAY(J,I)*(VAR(J+1,I)-VAR(J,I))
         END IF
        END IF

C
C2-----CALCULATE FLUX IN ROW DIRECTION

C -------BACK FACE--------------------
        IF (I.NE.1) THEN
         IF (IPOS(J,I-1).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CCLAY(J,I-1)*(VAR(J,I-1)-VAR(J,I))
         END IF
        END IF

C -------FRONT FACE--------------------
        IF (I.NE.NROW) THEN
         IF (IPOS(J,I+1).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CCLAY(J,I)*(VAR(J,I+1)-VAR(J,I))
         END IF
        END IF

C3------RETURN
      RETURN
      END


      SUBROUTINE SWI1SR(B,IPOS,CRLAY,CCLAY,FAC,VAR,J,I,NCOL,NROW)


C     ****************************************************************
C     SWI
C     ****************************************************************
C
C     SPECIFICATIONS:
```

```
C     -----------------------------------------------------------------
    DIMENSION B(NCOL,NROW),CRLAY(NCOL,NROW),CCLAY(NCOL,NROW)
    DIMENSION IPOS(NCOL,NROW)
    REAL VAR(NCOL,NROW)
C
C1-----CALCULATE FLUX IN COLUMN DIRECTION
C -------LEFT FACE--------------------
          IF (J.NE.1) THEN
         IF (IPOS(J-1,I).EQ.0) THEN
            B(J,I) = B(J,I)- FAC*CRLAY(J-1,I)*(VAR(J-1,I)-VAR(J,I))
         END IF
        END IF

C -------RIGHT FACE--------------------
        IF (J.NE.NCOL) THEN
         IF (IPOS(J+1,I).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CRLAY(J,I)*(VAR(J+1,I)-VAR(J,I))
         END IF
        END IF

C
C2-----CALCULATE FLUX IN ROW DIRECTION

C -------BACK FACE--------------------
        IF (I.NE.1) THEN
         IF (IPOS(J,I-1).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CCLAY(J,I-1)*(VAR(J,I-1)-VAR(J,I))
         END IF
        END IF

C -------FRONT FACE--------------------
        IF (I.NE.NROW) THEN
         IF (IPOS(J,I+1).EQ.0) THEN
            B(J,I) = B(J,I)-FAC*CCLAY(J,I)*(VAR(J,I+1)-VAR(J,I))
         END IF
        END IF

C3------RETURN
      RETURN
      END
```