1/1

CNWRA INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary Date

Yr.	Mo.	Day
9 2	0 1	0 8

02. Summary prepared by *(Name and phone)*

03. Summary action
New [x] Replacement [ ] Deletion [ ]
Previous Internal Software ID

03. Software title
NEFTRAN II (MKS)

04. Software Date

Yr.	Mo.	Day
0 2	1 2	9 1

06. Short title
NEFTRAN II (MKS)

07. Internal Software ID

08. Software type
- [ ] Automated Data System
- [x] Computer Program
- [ ] Subroutine/Module

09. Processing Mode
- [ ] Interactive
- [x] Batch
- [ ] Combination

10. APPLICATION AREA

General:
- [ ] Computer Systems Support/Utility
- [x] Scientific/Engineering
- [ ] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Flow & Transport Simulation

11. Submitting organization and address
SwRI

12. Technical contact(s) and phone
Ron Janetzke
512-522-3318

13. Narrative
See Document

Used To develop TPA Phase 2. [signature] 12/31/98

5.25-inch diskette

14. Keywords
Metric
Flow
Transport

15. Computer manufacturer and model
DEC/VAX

16. Computer operating system
VMS

17. Programming language(s)
FORTRAN 77

18. Number of source program statements
8773

19. Computer memory requirements
Virtual

20. Tape drives
0

21. Disk/Drum units
0

22. Terminals
0

23. Other operational requirements
None

24. Software availability
Available [x] Limited [ ] In-house only [ ]

25. Documentation availability
Available [ ] Inadequate [x] In-house only [ ]

26. FOR SUBMITTING ORGANIZATION USE

CNWRA Form TOP-4 10/90

1/7

RECEIVED
CENTER FOR NUCLEAR WASTE
REGULATORY ANALYSES

008798 FEB 19 92

FEB 18 1992

SUBJECT CODE 707.4
PROJECT NO. 20-3702-065

NOTE TO: Budhi Sagar

FROM: Task 3 Worker Bees

SUBJECT: COMPLETION OF FLOW MODULE

The liquid pathway flow module is completed to allow implementation in the system code. The module is flexible to adapt to a variety of conditions and is also flexible in the I/O operations which should allow for a straight forward and hopefully easy implementation. Please find attached; 1) a description of the units used within the flow module, 2) modifications necessary within NEFTRAN2 and 3) listings of various input files. In addition, a computer disk is enclosed which contains the source code and the files needed to execute the module.

The exact parametric values to be used in the production runs have not yet been completely determined. It is anticipated that these values will be ready in approximately two weeks. Please contact Tim McCartin (492-3847) to coordinate any further modifications you would like made to the flow module.

Distribution:

CNWRA	NMSS	RES
R. Ababou	J. Bradbury	G. Birchard
B. Gureghian	D. Brooks	R. Cady
R. Green	D. Chery	R. Kornasiewicz
R. Pabalan	R. Codell	L. Kovach
H. Manaktala	S. Coplan	T. Margulies
W. Murphy	W. Ford	T. McCartin
B. Sagar W/DISK	N. Eisenberg	T. Nicholson
J. Walton	J. Pohle	W. Ott
	R. Wescott	J. Randall
	M. Lee	

Please provide a copy to Ron Janetzke! ✓

Attachment 1 (Unit Descriptions)

INPUT

Unit 5 (input to flow module)

The information expected from the system code is read in on Unit 5. A further description is provided within the input file - see Attachment 3. (changing to different unit would be a trivial change)

Unit 20 (interpolation table to spatial distribute flux)

The interpolation table is provided by the Task 3 Group and an example is provided in Attachment 3. (changing to a different unit is trivial)

Unit 21 (interpolation table to distribute flux between matrix and fracture)

The interpolation table is provided by the Task 3 Group and an example is provided in Attachment 3. (changing to a different unit is trivial)

OUTPUT

Unit 35 (NEFTRAN2 input files)

The output of the flow module is NEFTRAN2 input files - these are written to unit 35. (changing to a different is trivial)

Attachment 2 - NEFTRAN2 Modifications

Modification to allow input of and external source term

The current version of NEFTRAN2 does not allow an external source term to be read. A small modification within Subroutine SOURCE will allow this flexibility. The suggested changes are:

1) insert the following at the beginning of SUBROUTINE SOURCE

```
C  ROUTINE TO READ AN EXTERNAL SOURCE BASED ON IOPT(14) .NE. 0
C
      IF (IOPT(14) .NE. 0) THEN
C
C
C   READ THE NUMBER OF ISOTOPES(ITOT) AND TIME STEPS(NUM) TO BE READ
C
      READ(14,*) ITOT,NUM
C
C
C  NOW LOOP OVER THE VALUES WHERE IT IS ASSUMED THAT THE INPUT
C  CONVENTION IS A DUMMY DESCRIPTOR IS READ WHICH CAN NAME THE
C  THE ISOTOPE BEING READ, FOLLOWED BY THE TIMES IN YEARS AND
C  THE AND SOURCES FOR THIS NUCLIDE IN (CI/YR)
C
C
      DO 5 J=1,ITOT
C
      READ(14,'(A80)') (DUMMY STRING VARIABLE)
C
      DO 5 I=1,NUM
C
C  READ THE EXTERNAL TIME (TEXT) AND THE SOURCE (EXTSRC)
C
      READ(14,*) TEXT(I),SRCEXT(I,J)
C
C
C NOW CHANGE (CI/YR) TO WHAT NEFTRAN2 WANTS (ATOMS/YR)
C
      SRCEXT(I,J)=SRCEXT(I,J)*1.6834E18*HLFALL(J)
C
    5 CONTINUE
C
C
C
C  NOW WRITE THE INFORMATION TO UNIT 24 AS NEFTRAN2 WOULD
C
      WRITE(24) ITOT
C
      DO 20 I=1,20
```

```
C
      WRITE(24) TEXT(I),(SRCEXT(I,J),J=1,ITOT)
C
   20 CONTINUE
C
C  NOW SKIP THE REST OF THE SUBROUTINE
C
      GOTO 3000
C
C
      ENDIF
C
C
```

2) The variable types and array dimension should be declared at the beginning of the subroutine. I would suggest that the "J" index be dimensioned to the "mxiso" the maximum number of isotopes. The "I" index is for the number of time steps used which could be 100 - or a number that the source term people could provide.

<u>Modification to write concentrations to an external file</u>

The current version of NEFTRAN2 has the ability to write concentrations to the standard output (unit 6). However, it would be useful to write the concentrations to an external file for later use by the dose module. The suggested change to NEFTRAN2 is:

insert the following in SUBROUTINE TPPRT after the line which reads (icount=icount + 1):

```
C
C  ROUTINE TO WRITE THE CONCENTRATIONS TO AN EXTERNAL FILE
C  BASED ON IOPT(29) .NE. 0 INDICATING TO READ
C
C
         IF (IOPT(29) .NE. 0) THEN
C
         IF (ICOUNT .EQ. IFREQ) THEN
C
         WRITE(29,9002) T,(CUROUT(J)/QDIS,J=NOISO)
C
         IF (IOPT(10) .EQ. 0) ICOUNT=0
C
         ENDIF
C
         ENDIF
C
C
```

Attachment 3 - Unit 5 Input

```
THE TIME TO END THE SIMUALTION IN YEARS
   1.0E4
NUMBER OF AREAS,  MAX # OF LAYERS, AND MAX # OF UNSATURATED LAYERS
    2    4    4
INFILTRATION (MM/YR)  AND SATURATED GRADIENT FOR THIS REALIZATION
       0.3          0.1
DISPERSION LENGTH (M) USED FOR ALL NEFTRAN LEGS
     10.0
THE NUMBER OF CHAINS AND THE NUMBER OF ISOTOPES
      2    5
THE NUMBER OF ISOTOPES PER CHAIN
   3    2
ISOTOPE INFORMATION - 3 LINES PER ISOTOPE
LN 1 NAME, AMAS, P#1, P#1, FRC F1, FRC F2, INV(CI), HLIFE(Y), EPA WGT
LINE 2 MATRIX RD'S PER LAYER
LINE 3 FRACTURE RD'S PER LAYER
'NP237'     237   0   0     0.0        0.0        1000.    2.14E6    1.
   20.   30.    40.   50.
   5.    6.     7.     8.
'U233'      233   1   0   1.0    0.0            1000.     1.62E5     1.
   120.  130.  140.   150.
   15.   16.  17.  18.
'TH229'     229   2   0    1.0    0.0            1000.     7.3E3      1.
  220.   230.   240.  150.
  25.    26.    27.  28.
'MC230'  230    0    0    0.0    0.0    500.   5.            1.
  90.    91.  92.   93
  9.     8.   7.    6.
'MC232'  232    1    0    1.0    0.0    500.   20.           1.
 80.   81.   82.  83.
 18.   17.   16.  15.
TOPOPAH SPRING  CONDUCTIVITY(MM/YR), POROSITY, AND BETA
        .6         .11        1.8
       30.    4.1E-5        4.2
CALICO HILLS VITRIC CONDUCTIVITY(MM/YR), POROSITY, AND BETA
     8500.         .46        2.5
      300.    4.6E-5        4.2
CALICO HILLS ZEOLITIC CONDUCTIVITY(MM/YR), POROSITY, AND BETA
        .6         .28        1.6
      300.    4.6E-5        4.2
PROW PASS CONDUCTIVITY(MM/YR), POROSITY, AND BETA
       60.         .25        2.2
      300.    4.6E-5        4.2
********THIS INFORMATION IS REPEATED FOR EACH PART *****PART #1
INLET AND OUTLET AREAS (SQUARE METERS) FOR AREA #1
      100.      200.
AREA #1 LAYERS (M)
     50.   100.    10.   200.
SATURATED LENGTHS (M)
    55.  105.  15. 0.0
********THIS INFORMATION IS REPEATED FOR EACH PART *****PART #2
INLET AND OUTLET AREAS (SQUARE METERS) FOR AREA #2
      110.      220.
AREA #1 LAYERS (M)
     55.    110.    11.    220.
SATURATED LENGTHS (M)
    60.  0.0  0.0   0.0
```

Attachment 4 - Unit 20 Input

```
THE NUMBER OF VALUES IN THE TABLE FOR AREA PART #1
    8
NOW INPUT THE VALUES (ONE SET/LINE) TOTAL INFIL., INFIL. FOR THIS PART
  GOING FROM LOW TO HIGH VALUES (MM/YR)
   .01       .01
   .03       .03
    .1        .1
    .2        .18
    .3        .28
    .4        .3
    .5        .4
    10.       .6
THE NUMBER OF VALUES FOR AREA PART #2
    6
 TOTAL INFILTRATION,    INFILTRATION FOR THIS PART
  LOW TO HIGH VALUES   (MM/YR)
    .01     .01
     .1      .1
     .6       .5
     1.      .52
     4.      .53
     10.     .54
```

Attachment 5 - Unit 21 Input

```
TOPOPAH SPRING MATRIX INTERPOLATION TABLE # OF ENTRIES
    6
  VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
  FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION
   0.1       0.1
   0.4       0.4
    0.7    0.7
   0.8       0.75
    1.0    .9
    3.0    .95
CAILICO HILLS  MATRIX INTERPOLATION TABLE # OF ENTRIES
    5
 VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
 FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION
0.2     .2
0.5      0.5
.8       .7
1.2      .8
5.       .81
ME AND YOU     MATRIX INTERPOLATION TABLE # OF ENTRIES
   4
 VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
 FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION
 0.3     0.3
  1.0     .8
  4.0      .85
  9.0      .87
JUST YOU       MATRIX INTERPOLATION TABLE # OF ENTRIES
   6
 VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
 FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION
0.3       0.3
0.5        0.5
0.9        0.9
1.0       .93
2.0       .94
8.0       .96
```

1/30

```
@@@@@@@@@@@@          @@@@@@@@      @@@@        @@@@      @@@@
  @@@@    @@@@          @@@@      @@@@@@@@      @@@@@@    @@@@
  @@@@    @@@@          @@@@    @@@@    @@@@    @@@@@@@@  @@@@
  @@@@@@@@@@            @@@@    @@@@    @@@@    @@@@  @@@@@@@@
  @@@@  @@@@    @@@@    @@@@    @@@@@@@@@@@@    @@@@    @@@@@@
  @@@@    @@@@  @@@@    @@@@    @@@@    @@@@    @@@@      @@@@
@@@@@@    @@@@    @@@@@@@@      @@@@    @@@@    @@@@      @@@@
```

```
          @@@@  @@   @@ @@   @@ @@@@@@    @@
         @@  @@ @@@  @@ @@   @@  @@  @@  @@@@
        @@      @@@@ @@ @@   @@  @@  @@ @@  @@
        @@      @@ @@@@ @@ @ @@  @@@@@  @@  @@
        @@      @@  @@@ @@@@@@@  @@ @@  @@@@@@
         @@  @@ @@   @@ @@@ @@@  @@  @@ @@  @@
          @@@@  @@   @@ @@   @@ @@@  @@ @@  @@
```

Created at.........: 09:42:26 on: 03-10-92

Job Number.........: 78

2/30

```
//TUIPATH  JOB (WDCC,136,E),MCCARTIN
//START  EXEC   FORVCOMP,REGION=5000K
//COMP.SYSIN    DD  *
C
C
C
C  SET A DUMMY STRING VARIABLE TO USE IN READING INPUT FILE
C
      CHARACTER*80 STRING
C
C
C
C  SET THE IDBUG TO 1 TO GET BEDUG PRINTS
C
      PARAMETER (IDBUG=0)
C
C
C
C    DECLARE THE VARIABLE TYPES
C
C   FIRST THE INTEGER VALUES
C   NLYERS = NUMBER OF LAYERS BEING ANALYSED
C   NPART  = NUMBER OF AREAL SECTIONS BEING ANALYSED
C   NISO   = NUMBER OF ISOTOPES BEING ANALYSED
C   NCHAIN = NUMBER OF CHAINS BEING ANALYSED
C   MEMPCH(MAXCHN)  NUMBER OF ISOTOPE MEMBERS PER CHAIN
C   NLYERU = NUMBER OF UNSATURATED LAYERS
C   PAR1(NISO) = INDEX OF FIRST PARENT
C   PAR2(NISO) = INDEX OF SECOND PARENT
C   IZERO(NCHAIN)  -  PLACES 0 IN GROUP 2 OF NEFTRAN TO INDICATE
C                     THAT THIS CHAIN IS TRANSPORTED
C   INDEX(NPART) - NUMBER OF VALUES IN THE INFILTRATION INTERPOLATION
C                  YMTABLE FOR EACH AREAL PART
C   MTINDX(NLYERS) - NUMBER OF ENTRIES IN THE MATRIX VERSUS FRACTURE
C                    FLUX INTERPOLATION TABLE (VARIABLE BY LAYER)
C
C
      INTEGER NLYERS,NPART,NISO,NCHAIN,MEMPCH,NLYERU,PAR1,PAR2,
     1   IZERO,INDEX,MTINDX
C
C   PROBLEM DIMENSIONS
C   MAXPTH  =  MAXIMUM NUMBER OF PATHS
C   MAXPRT  =  MAXIMUM NUMBER OF AREAL PATHS
C   MAXLYR  =  MAXIMUM NUMBER OF LAYERS TO BE ANALYSED
C   MAXCHN  =  MAXIMUM NUMBER OF CHAINS TO BE ANALYSED
C   MAXISO  =  MAXIMUM NUMBER OF ISOTOPES TO BE ANALYSED
C   MAXDEX  -  MAXIMUM NUMBER OF ENTRIES IN THE AREAL INFIL TABLE PER
C              AREA
C
      INTEGER MAXPTH,MAXPRT,MAXLYR,MAXCHN,MAXISO
C
C
C  DUMMY INTEGER VALUES USED FOR INCREMENTING AND STORING
C  ARRAY LOCATIONS
C  NP   INCREMENT FOR AREAL SECTION BEING ANALYSED
C  NSTEP  ARRAY INDEX COUNTER USED FOR SATURATION CALCULATION
C  NEFLEG  INDICATES THE NUMBER OF NEFTRAN LEGS TO USE
C  NPATH   INCREMENTS THROUGH THE PATH NUMBERS
C  INC     INDICATES THE NUMBER OF NEFTRAN LEGS BEING USED
C  LAYUNS  INDEX FOR LAYERS IN THE UNSATURATED PATH
```

```
C  MEDIA  INDICATES EITHER 1(MATRIX) OR 2(FRACTURE)
C  NELM    NUMBER OF ISOTOPES BEING ANALYSED
C   NDEX     INCREMENT COUNTER FOR INTERPOLATION TABLES
C
      INTEGER NP,NSTEP,NEFLEG,NPATH,INC,LAYUNS,MEDIA,
     1 NELM,NDEX
C
C
C
C
C
C
C
C  REAL VARIABLES
C
C  FIRST THE REAL HYDROLOGIC INPUT VALUES
C  COND      CONDUCTIVITY ARRAY (I=NLAYERS, J=1(MATRIX) OR 2(FRC)
C  POR       POROSITY ARRAY (I=NLAYERS, J=1(MATRIX) OR 2(FRC)
C  BETA      VAN GENUCHTEN ARRAY; I=NLYERS, J=1(MATRIX) OR 2(FRC)
C  LEGLEN    LAYER THICKNESS ARRAY; I=NPART, J=NLYERS
C  SATLEN    SATURATED LAYER THICKNESS ARRAY; I=NPART, J=NLYERS
C  AREA      AREA ARRAY; I=NPART, J=1(INLET) OR 2(OUTLET)
C  INFIL     INFILTRATION ARRAY; I=NPART
C  TINFIL  TOTAL INFILTRATION FOR A REALIZATION
C  GRAD      GRADIENT TO BE USED IN THE SATURATAED ZONE
C  DISPER  DISPERSION LENGTH TO BE USED FOR ALL LEGS
C
C  INFINT(NPART,INDEX(NPART),2) - TABLE TO INTERPOLATE INFIL. AMONG THE
C          AREAL PARTS,  I=AREA INDEX, J=JTH ENTRY, K=1 (TOTAL INFIL.)
C          AND =2 (INFILTRATION FOR THIS AREA)
C
C   MATINT(NLYERS,(MTINDX(NLYERS)),2)   -  INTERPOLATION TABLE FOR
C                     EVALUATING MATRIX VERSUS FRACTURE FLOW FOR EACH
C                     LAYER; I=# OF LAYERS, J=# OF ENTRIES,
C                     K=1 (INFILTRATION/KSAT) AND
C                     K=2 (MATRX FLUX/KSAT)
C
C
C
      REAL COND,POR,BETA,LEGLEN,AREA,INFIL,SATLEN,TINFIL,GRAD,
     1  DISPER,INFINT,MATINT
C
C
C
C  REAL VARIABLES USED TO CALCULATE THE MASS FLUX FOR A PATH
C
C  MASFRC   MASS FRACTION ARRAY; I=NLYERS, J =1(MATRIX) OR 2(FRC)
C  PTHMAS   MASS FRACTION ARRAY; I=NPART, J=MAXPTH
C  INFLO(NLYERS)  INFILTRATION FRACTION LOWER CUT-OFF FOR EACH LAYER
C  INFHI(NLYERS)  INFILTRATION FRACTION HIGH CUT-OFF FOR EACH LAYER
C  FLUXLO(NLYERS) FLUX FRACTION LOWER CUT-OFF FOR EACH LAYER
C  FLUXHI(NLYERS) FLUX FRACTION HIGH CUT-OFF FOR EACH LAYER
C  SLOPE(NLYERS)  SLOPE OF THE CUT-OFF CURVE FOR EACH LAYER
C  LYFAC   VARIABLE USED TO CALCULATE THE MASS FLUX OF A PATH
C
      REAL MASFRC,PTHMAS,INFLO,INFHI,FLUXLO,FLUXHI,SLOPE,LYFAC
C
C
C  NOW  THE REAL VARIABLES USED TO CALCULATE SATURATION AND VELOCITY
C
```

```
C
C  SAT      ARRAY OF SATURATION USED TO INTERPOLATE FROM
C  KMAT     ARRAY OF MATRIX CONDUCTIVITY USED TO INTERPOLATE FROM
C  KFRC     ARRAY OF FRACTURE CONDUCTIVITY USED TO INTERPOLATE FROM
C  KREL     DUMMY VARIABLE TO CALCULATE RELATIVE CONDUCTIVITY
C  LAMBDA   DUMMY VARIABLE USE IN VAN GENUCHTEN CALCULATION
C  SATM(NLYERS)  ARRAY OF MATRIX SATURATION FOR EACH LAYER
C  SATF(NLYERS)  ARRAY OF FRACTURE SATURATION FOR EACH LAYER
C  VEL(NLYERS,2) VELOCITY ARRAY; J=1(MATRIX) OR 2(FRACTURE)
C  MC(NLYERS,2) MOISTURE CONTENT ARRAY; J=1(MATRIX) OR 2(FRC)
C
      REAL SAT,KMAT,KFRC,KREL,LAMBDA,SATM,SATF,VEL,MC
C
C
C
C
C   THE REAL VARIABLE USED FRO THE ISOTOPE INFORMATION
C
C  NAME(MAXISO)   -  NAME OF ISOTOPE 'A6'
C  AMASS(MAXISO) -  ATOMIC MASS OF EACH ISOTOPE
C  FRCFM1(MAXISO) - FRACTION DECAYS FROM ISOTOPE 1 IN THIS CHAIN
C  FRMFM2(MAXISO) - FRACTION DECAYS FROM ISOTOPE 2 IN THIS CHAIN
C  INV(MAXISO)     - INVENTORY IN CURIES FOR EACH ISOTOPE
C  HLIFE(MAXISO)   - HALF LIVES IN YEARS FOR EACH ISOTOPE
C  EPAWGT(MAXISO) - EPA WEIGHT FOR EACH ISOTOPE
C  RD(MAXISO,MAXLYR,2) - RD'D BY ISOPTOPE, LAYER AND 1(MAT) OR 2(FRC)
C
C
      REAL AMASS,FRCFM1,FRCFM2,INV,HLIFE,EPAWGT,RD,NEFRD,NEFRDI
C
      CHARACTER*6 NAME
C
C
C   REAL VARIABLES USED FOR NEFTRAN INPUT CALCULATIONS
C
C  MOISTC(NLYERS)  MOISTURE CONTENTS FOR THE NON-ZERO LEGS (NEFTRAN POROSITY)
C  NEFVEL(NLYERS)  VELOCITIES FOR THE NON-ZERO LEGS
C  NEFLEN(NLYERS)  LENGTHS OF THE NON-ZERO LEGS
C  AREAI   PORE AREA OF THE INLET
C  AREAO   PORE AREA OF THE OUTLET
C  NEFRD(NISO,MAXLYR) - RD'S FOR EACH LEG AND ISOTOPE
C  NEFRDI(NISO,MAXLYR) - IMMOBILE WATER RD'S FOR EACH LEG & ISOTOPE
C  ZERO(NCHAIN)   - PLACES 0.0 IN GROUP 3 OF NEFTRAN TO INDICATE
C                   AUTOMATIC TIME STEP
C  ETIME      -  END TIME OF SIMULATION
C
C
      REAL MOISTC,NEFVEL,NEFLEN,AREAI,AREAO,ZERO,ETIME
C
C
C
C
C
C
C
C  SET THE PARAMETERS FOR THE ARRAY DIMENSIONS
C
      PARAMETER (MAXPTH=32,MAXLYR=8,MAXPRT=5,MAXISO=10,MAXCHN=10,
     1   MAXDEX=100)
C
```

```
      DIMENSION NAME(MAXISO)
C
      DIMENSION INDEX(MAXPRT),MTINDX(MAXLYR)
C
C
C   NOW LET'S DIMENSION THE ARRAYS
C
      DIMENSION COND(MAXLYR,2),POR(MAXLYR,2),BETA(MAXLYR,2)
C
      DIMENSION LEGLEN(MAXPRT,MAXLYR),SATLEN(MAXPRT,MAXLYR)
C
      DIMENSION AREA(MAXPRT,2)
C
      DIMENSION INFIL(MAXPRT),INFINT(MAXPRT,MAXDEX,2)
      DIMENSION MATINT(MAXLYR,MAXDEX,2)
C
C
      DIMENSION MASFRC(MAXLYR,2)
C
      DIMENSION PTHMAS(MAXPRT,MAXPTH)
C
      DIMENSION INFLO(MAXLYR),INFHI(MAXLYR),FLUXLO(MAXLYR)
      DIMENSION FLUXHI(MAXLYR),SLOPE(MAXLYR)
C
C
C
      DIMENSION SAT(200),KMAT(200),KFRC(200)
C
      DIMENSION SATM(MAXLYR),SATF(MAXLYR)
C
      DIMENSION VEL(MAXLYR,2),MC(MAXLYR,2)
C
C
C
      DIMENSION MOISTC(MAXLYR),NEFVEL(MAXLYR),NEFLEN(MAXLYR)
      DIMENSION NEFRD(MAXISO,MAXLYR),NEFRDI(MAXISO,MAXLYR)
C
C
C
      DIMENSION AMASS(MAXISO),FRCFM1(MAXISO),FRCFM2(MAXISO),
     1  INV(MAXISO),HLIFE(MAXISO),EPAWGT(MAXISO),PAR1(MAXISO),
     2   PAR2(MAXISO)
C
      DIMENSION RD(MAXISO,MAXLYR,2)
C
C
      DIMENSION MEMPCH(MAXCHN)
C
      DIMENSION IZERO(MAXCHN),ZERO(MAXCHN)
C
C
C
C
C
C   READ IN THE INFORMATION FOR THIS REALIZATION (DATA FOR ALL AREAS)
C
C
C
      READ(5,'(A80)') STRING
      READ(5,*) ETIME
```

```
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(A)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(5X,E10.3)') ETIME
C
C
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,*) NPART,NLYERS,NLYERU
      IF (IDBUG .NE. 0) WRITE(6,'(3I5)') NPART,NLYERS,NLYERU
C
C
C  READ THE INFILTRATION FOR THIS AREA
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,*) TINFIL,GRAD
      IF (IDBUG .NE. 0) WRITE(6,'(2(E10.3,5X))') TINFIL,GRAD
C
C
C
C   READ THE DISPERSION LENGTH (M) - ONE VALUE USED FOR ALL LEGS
C
      READ(5,'(A80)') STRING
      READ(5,*) DISPER
C
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(5X,F10.3)') DISPER
C
C
C
C
C
C READ THE INVENTORY INFORMATION
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
C
      READ(5,*) NCHAIN,NISO
C
      IF (IDBUG .NE. 0) WRITE(6,'(2I5)') NCHAIN,NISO
C
C
C
C  SET THE ZERO ARRAYS
C
      DO 71 I=1,NCHAIN
C
      ZERO(I)=0.0
      IZERO(I)=0
C
   71 CONTINUE
C
C
C
C
C
C  READ THE NUMBER OF MEMBERS PER CHAIN
C
      READ(5,'(A80)') STRING
```

```
      READ(5,*) (MEMPCH(I),I=1,NCHAIN)
C
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(16I5)') (MEMPCH(I),I=1,NCHAIN)
C
C
C
C
C  READ IN THE ISOTOPE DATA ON ONE LINE AND THE TWO SUBSEQUENT LINES
C  READ FIRST THE MATRIX THEN THE FRACTURE RD'S PER LAYER
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
C
      NELM=0
C
      DO 72 I=1,NCHAIN
C
      DO 72 J=1,MEMPCH(I)
C
      NELM=NELM+1
C
C
      READ(5,*) NAME(NELM),AMASS(NELM),PAR1(NELM),PAR2(NELM),
     1 FRCFM1(NELM),FRCFM2(NELM),INV(NELM),HLIFE(NELM),EPAWGT(NELM)
C
      IF (IDBUG .NE. 0)
     1 WRITE(6,'(A1,A6,A1,2X,F6.2,2X,2I5,2X,2(F6.4,2X),3(E10.3,1X))')
     2 '''',NAME(NELM),'''',AMASS(NELM),PAR1(NELM),PAR2(NELM),
     3 FRCFM1(NELM),FRCFM2(NELM),INV(NELM),HLIFE(NELM),EPAWGT(NELM)
C   READ THE MATRIX RD'S BY LAYER
C
      READ(5,*) (RD(NELM,K,1),K=1,NLYERS)
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(8E10.3)') (RD(NELM,K,1),K=1,NLYERS)
C
C
C  READ THE FRACTURE RD'S BY LAYER
C
C
      READ(5,*) (RD(NELM,K,2),K=1,NLYERS)
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(8E10.3)') (RD(NELM,K,2),K=1,NLYERS)
C
C
   72 CONTINUE
C
C
C
C
C
      IF (NELM .NE. NISO) THEN
```

8/30

```
C
C
      WRITE(6,'(A)') 'ERROR IN NUMBER OF ISOTOPES'
      STOP
C
      ENDIF
C
C
C   READ THE LAYER PROPERTIES TO BE USED FOR ALL SECTIONS
C
      DO 70 I=1,NLYERS
C
C   READ THE MATRIX PROPERTIES
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(3F10.0)') COND(I,1),POR(I,1),BETA(I,1)
      IF (IDBUG .NE. 0)
     1 WRITE(6,'(3(E10.3,5X))') COND(I,1),POR(I,1),BETA(I,1)
C
C   READ THE FRACTURE PROPERTIES
C
      READ(5,'(3F10.0)') COND(I,2),POR(I,2),BETA(I,2)
      IF (IDBUG .NE. 0)
     1 WRITE(6,'(3(E10.3,5X))') COND(I,2),POR(I,2),BETA(I,2)
C
C
   70 CONTINUE
C
C
C
C
C
C
C
C
C
C
C   READ IN THE INFILTRATION TABLES FOR INTERPOLATION OF INFILTRATION
C   WHERE INFINT(I,J,K), I=# OF PARTS, J=# OF VALUES (INDEX(NP)
C   READ , K =1 TOTAL INFILTRATION VALUES & K=2 FRACTION FOR THIS PART
C
      DO 96 NP=1,NPART
C
C  READ THE NUMBER OF VALUES IN THE TABLE
      READ(20,'(A80)') STRING
C
      READ(20,*) INDEX(NP)
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(I5)') INDEX(NP)
C
C
      READ(20,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(20,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
C
      DO 98 M=1,INDEX(NP)
```

```
C
C
      READ(20,*) INFINT(NP,M,1),INFINT(NP,M,2)
      IF (IDBUG .NE. 0) WRITE(6,'(E10.3,5X,E10.3)') INFINT(NP,M,1),
     1  INFINT(NP,M,2)
C
C
   98 CONTINUE
C
C
C
C
C
C
C   NOW CALCULATE THE INFILTRATION FOR THIS PART
C
C
C
C
C   NOW INTERPOLATE BASED ON THE TOTAL INFILTRATION TINFIL
C
      IF (TINFIL .LE. INFINT(NP,1,1)) THEN
C
      INFIL(NP)=TINFIL*INFINT(NP,1,2)/INFINT(NP,1,1)
C
C
      ELSE IF (TINFIL .GE. INFINT(NP,INDEX(NP),1))  THEN
C
      INFIL(NP)=TINFIL*INFINT(NP,INDEX(NP),2)/INFINT(NP,INDEX(NP),1)
C
C
      ELSE
C
C
C   INTERPOLATE BASED ON NDEX
C
C
      NDEX=0
C
   97 NDEX=NDEX+1
C
C
      IF (TINFIL .GE. INFINT(NP,NDEX,1)) GOTO 97
C  NDEX EQUALS THE FIRST VALUE LARGER THAN TINFIL
C
C   FIRST IS THE TWO VALUES FOR THIS PART ARE EQUAL THEN
C   THERE IS NO NEED TO INTERPOLATE
C
      IF (INFINT(NP,NDEX,2) .EQ. INFINT(NP,NDEX-1,2)) THEN
C
      INFIL(NP)=INFINT(NP,NDEX,2)
C
C
C
C   NOW INTERPOLATE BETWEEN THE TWO DIFFERENT VALUES
C
      ELSE
C
C
      INFIL(NP)=(INFINT(NP,NDEX,2)-INFINT(NP,NDEX-1,2))*
```

```
     1    (TINFIL - INFINT(NP,NDEX-1,1))/
     2   (INFINT(NP,NDEX,1)-INFINT(NP,NDEX-1,1)) + INFINT(NP,NDEX-1,2)
C
      ENDIF
C
      ENDIF
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(A,E10.3,5X,A,E10.3)')
     1 'TOTAL INFILTRATION = ',TINFIL,'FOR THIS PART = ',INFIL(NP)
C
C
   96 CONTINUE
C
C
C
C
C
C   READ IN MATRIX FRACTION TABLES FOR INTERPOLATION OF THE FRACTION
C   OF MATRIX FLUX MATINT(I,J,K) I=NUMBER OF LAYERS, J=NUMBER OF
C   VALUES TO BE READ, AND K=1 (FRACTION OF INFILTRATION)
C   OR =2 (FRACTION OF MATRIX FLOW)
C
      DO 196 I=1,NLYERU
C
C  READ THE NUMBER OF VALUES IN THE TABLE
      READ(21,'(A80)') STRING
C
      READ(21,*) MTINDX(I)
C
C
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(I5)') MTINDX(I)
C
C
      READ(21,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(21,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
C
      DO 196 M=1,MTINDX(I)
C
C
      READ(21,*) MATINT(I,M,1),MATINT(I,M,2)
      IF (IDBUG .NE. 0) WRITE(6,'(E10.3,5X,E10.3)') MATINT(I,M,1),
     1  MATINT(I,M,2)
C
C
C
C
C
  196 CONTINUE
C
C
C
C
C
C
C   READ THE INFORMATION SPECIFIC TO AN AREA INSIDE THE AREA LOOP
C
```

```
C
C
C
      DO 500 NP=1,NPART
C
C
C  READ THE AREA DIMENSIONS OF THE INLET AND THE OUTLET
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      READ(5,'(2F10.0)') AREA(NP,1),AREA(NP,2)
      IF (IDBUG .NE. 0) WRITE(6,'(2(E10.3,5X))') AREA(NP,1),AREA(NP,2)
C
C
C   READ THE THICKNESS FOR EACH LAYER FOR THIS AREA
C
      READ(5,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
C
      READ(5,*) (LEGLEN(NP,J),J=1,NLYERS)
      IF (IDBUG .NE. 0) WRITE(6,'(8F10.3)') (LEGLEN(NP,J),J=1,NLYERS)
C
C
C  READ THE SATURATED THICKNESS FOR EACH LAYER
C
      READ(5,'(A80)') STRING
      READ(5,*) (SATLEN(NP,J),J=1,NLYERS)
C
      IF (IDBUG .NE. 0) WRITE(6,'(A80)') STRING
      IF (IDBUG .NE. 0) WRITE(6,'(8F10.3)') (SATLEN(NP,J),J=1,NLYERS)
C
C
C
C
C
C
C
C
C
C
C
C       NOW CALCULATE THE MASS FRACTION FOR EACH UNSATURATED LAYER
C      IN THIS AREAL SECTION  THE NP-TH PART
C
      DO 1000 I=1,NLYERU
C
C
C
C   IF THE INFILTRATION EXCEEDS THE LARGEST FLUX IN THE INTERPOLATION
C   TABLE THEN FIX THE MATRIX MASS FRACTION TO THE VALUE AT THAT FLUX
C
C
C
C
      IF (INFIL(NP) .GT. MATINT(I,MTINDX(I),1)*COND(I,1)) THEN
C
C  SET MATRIX VALUE TO ITS' UPPER LIMIT
C
```

```
      MASFRC(I,1)=(MATINT(I,MTINDX(I),2)/MATINT(I,MTINDX(I),1))*
     1  INFIL(NP)/COND(I,1)
C
C
C   IF THE MASS FRACTIONS ARE EQUAL THEN ALL FLUX IS IN THE
C   MATRIX SO INSURE THIS BY SETTING MASSFRC(I,1)=1.0
C   (THIS PRECLUDES ROUND OFF ERROR IN CALCUALTION)
C
      IF (MATINT(I,MTINDX(I),1) .EQ. MATINT(I,MTINDX(I),2))
     1  MASFRC(I,1)=1.0
C
C
C
C
C
C
C
C  IF THE INFILTRATION IS LESS THAN THE SMALLEST VALUE IN THE
C  INTERPOLATION TABLE THEN SET THE MATRIX MASS FRACTION TO VALUE
C  AT THAT FLUX
C
      ELSE IF (INFIL(NP) .LT. MATINT(I,1,1)*COND(I,1)) THEN
C
      MASFRC(I,1)=(MATINT(I,1,2)/MATINT(I,1,1))*INFIL(NP)/COND(I,1)
C
C
C
C   IF THE MASS FRACTIONS ARE EQUAL THEN ALL FLUX IS IN THE
C   MATRIX SO INSURE THIS BY SETTING MASSFRC(I,1)=1.0
C   (THIS PRECLUDES ROUND OFF ERROR IN CALCUALTION)
C
      IF (MATINT(I,1,1) .EQ. MATINT(I,1,2))
     1  MASFRC(I,1)=1.0
C
C
C
C
C
C
C
C  CALCULATE THE MASS FRACTION MATRIX BASED ON
C  A LINEAR INTERPOLATION BETWEEN THE TWO CLOSEST VALUES IN THE
C  INTERPOLATION TABLE, WHERE NDEX IS THE HIGHER LOCATION IN THE
C  TABLE
C
      ELSE
C
C
C
C   INTERPOLATE BASED ON NDEX
C
C
      NDEX=0
C
  197 NDEX=NDEX+1
C
C
      IF (INFIL(NP) .GE. MATINT(I,NDEX,1)*COND(I,1)) GOTO 197
C  NDEX EQUALS THE FIRST VALUE LARGER THAN INFIL(NP)
C
```

```
C
C
C  FIRST IF THE TWO MATRIX FRACTIONS ARE EQUAL WE DO NOT
C  INTERPOLATE
C
      IF (MATINT(I,NDEX,2) .EQ. MATINT(I,NDEX-1,2)) THEN
C
      MASFRC(I,1)=MATINT(I,NDEX,2)*COND(I,1)/INFIL(NP)
C
C
C
C  NOW INTERPOLATE BETWEEN TWO DIFFERENT VALUES OF MATRIX FLUX FRACTION
C
      ELSE
C
C
C
C
      MASFRC(I,1)=(MATINT(I,NDEX,2)-MATINT(I,NDEX-1,2))*
     1   (INFIL(NP)/COND(I,1) - MATINT(I,NDEX-1,1))/
     2  (MATINT(I,NDEX,1)-MATINT(I,NDEX-1,1)) + MATINT(I,NDEX-1,2)
C
      ENDIF
C
C
C   IF THE MASS FRACTIONS ARE EQUAL THEN ALL FLUX IS IN THE
C   MATRIX SO INSURE THIS BY SETTING MASSFRC(I,1)=1.0
C   (THIS PRECLUDES ROUND OFF ERROR IN CALCUALTION)
C
      IF (MATINT(I,NDEX,1) .EQ. MATINT(I,NDEX,2))
     1  MASFRC(I,1)=1.0
C
C
C
      ENDIF
C
C
C   NOW SET THE FRACTURE MASS FRACTION TO 1.0-(MATRIX MASS FRACTION)
C
      MASFRC(I,2)=1.0-MASFRC(I,1)
C
C
C
C
      IF (IDBUG .NE. 0) THEN
      WRITE(6,'(A,I5,5X,A,I5)') 'FOR PART #',NP,'AND LAYER #',I
C
      WRITE(6,'(A,E10.3,5X,A,E10.3)') 'MATRIX MASS FRACTION = ',
     1 MASFRC(I,1),'AND FRACTURE MASS FRACTION = ',MASFRC(I,2)
C
      ENDIF
C
C
C
C
C
 1000 CONTINUE
C
C
C
```

```
C
C
C
C
C
C  FILL UP A SATURATION VERSUS CONDUCTIVITY CURVE USING
C  THE VAN GENUCHTEN CURVE FROM DCM3D
C
C  SET THE SATURATIONS TO BE USED
C
      DO 5001 I=1,105
C
      SAT(I)=(I-5.)/100.
C
      IF (I .EQ. 1) SAT(I)=.0000001
      IF (I .EQ. 2) SAT(I)=.000001
      IF (I .EQ. 3) SAT(I)=.00001
      IF (I .EQ. 4) SAT(I)=.0001
      IF (I .EQ. 5) SAT(I)=.001
C
 5001 CONTINUE
C
      DO 5000 N=1,NLYERU
C
      LAMBDA=1. - 1./BETA(N,1)
C
      DO 5005 I=1,105
C
C
      KREL=(SAT(I)**.5)*(1.-(1.-SAT(I)**(1./LAMBDA))**LAMBDA)**2
C
      KMAT(I)=KREL*COND(N,1)
C
C
C
C
C
C
      LAMBDA=1. - 1./BETA(N,2)
C
      KREL=(SAT(I)**.5)*(1.-(1.-SAT(I)**(1./LAMBDA))**LAMBDA)**2
C
      KFRC(I)=KREL*COND(N,2)
C
C
 5005 CONTINUE
C
C
C   LOOP OVER THE K ARRAYS TO FIND THE SATURATION
C
C
C
C
      NSTEP=0
C
 5200 NSTEP=NSTEP+1
C
C  COMPARE CONDUCTIVITY TO FLUX
C
      IF (KMAT(NSTEP) .LT. MASFRC(N,1)*INFIL(NP)) GOTO 5200
```

```
C
      IF (NSTEP .EQ. 1) SATM(N)=0.0
      IF (NSTEP .GT. 1) SATM(N)=SAT(NSTEP)
C
C
C
C  CALCULATE THE PORE VELOCITY
C
      IF (MASFRC(N,1) .EQ. 0.0 .OR. SATM(N) .EQ. 0.0) THEN
      VEL(N,1)=0.0
      ELSE
      VEL(N,1)=MASFRC(N,1)*INFIL(NP)/(POR(N,1)*SATM(N))
      MC(N,1)=POR(N,1)*SATM(N)
      ENDIF
C
C
C
C  NOW DO THE FRACTURE
C
      NSTEP=0
 5201 NSTEP=NSTEP+1
C
C COMPARE  FRACTURE CONDUCTIVITY TO FLUX
C
      IF (KFRC(NSTEP) .LT. MASFRC(N,2)*INFIL(NP)) GOTO 5201
C
      IF (NSTEP .EQ. 1) SATF(N)=0.0
      IF (NSTEP .GT. 1) SATF(N)=SAT(NSTEP)
C
C
C
C
C  CALCULATE THE PORE VELOCITY
C
      IF (MASFRC(N,2) .EQ. 0.0 .OR. SATF(N) .EQ. 0.0) THEN
      VEL(N,2)=0.0
      ELSE
      VEL(N,2)=MASFRC(N,2)*INFIL(NP)/(POR(N,2)*SATF(N))
      MC(N,2)=POR(N,2)*SATM(N)
      ENDIF
C
C
C
 5000 CONTINUE
C
C
C
C
C    CALCULATE THE MASS FRACTION FOR EACH AREAL PART OF THE
C    REPOSITORY AND FOR EACH OF THE POSSIBLE PATHS
C
C
C
C
C   LOOP OVER ALL THE PATHS WHICH ACCOUNT FOR MATRIX (=1) OR
C   FRACTURE FLOW IN EACH LAYER  (2**NLYERS POSSIBLE PATHS)
C
C
        NPATH=2**NLYERU
C
```

```
      DO 500 N=1,NPATH
C
C
C   SET AN INCREMENT TO KEEP TRACK OF THE NON-ZERO LEGS OF A PATH
      INC=0
C
C
C
C
C
C   SET THE PTHMAS ARRAY TO ONE FOR THIS PATH
C
      PTHMAS(NP,N) = 1.0
C
C
C
      IPATH=N
C
      DO 505 LAYUNS=1,NLYERU
C
C
      IMEDIA=IPATH/2
C
      MEDIA=1
C
      IF (IMEDIA*2 .EQ. IPATH) MEDIA=2
C
      IPATH=IMEDIA
C
C
C
C
      IF (IDBUG .NE. 0) THEN
C
       IF (LAYUNS .EQ. 1) WRITE(6,'(A,5X,I5)') 'PATH #',N
C
      IF (MEDIA .EQ. 1) WRITE(6,'(A,I5,5X,A)') 'LAYER #',LAYUNS,
     1 'IS MEDIA TYPE    MATRIX'
C
      IF (MEDIA .EQ. 2) WRITE(6,'(A,I5,5X,A)') 'LAYER #',LAYUNS,
     1 'IS MEDIA TYPE    FRACTURE'
C
      ENDIF
C
C
C    DETERMINE THE FLUX FRACTION FOR THIS LAYER
C
      LYFAC=MASFRC(LAYUNS,MEDIA)
C
C    CHANGE THE FACTOR IF THE LEG LENGTH IS ZERO
C
      IF (LEGLEN(NP,LAYUNS) .EQ. 0.0 .AND. MEDIA .EQ. 1) LYFAC=1.0
C
C   SO DOUBLE COUNTING IS NOT DONE SKIP THIS PATH IF THE
C   LEG LENGTH IS ZERO AND THE MEDIA IS ZERO
C   THIS IS DONE BY SETTING LYFAC=0
C
      IF (LEGLEN(NP,LAYUNS) .EQ. 0.0 .AND. MEDIA .EQ. 2) LYFAC=0.0
C
C
```

```
      IF (LYFAC .EQ. 0.0) GOTO 499
C
C
C
C
C   WE NOW HAVE A BONAFIDE PATH SO SET UP THE INCREMENT
C
      INC=INC+1
C
C
C
C
C   STORE THE LEG LENGTHS
C
      NEFLEN(INC)=LEGLEN(NP,LAYUNS)
C
C
C
C
C   STORE THE MOISTURE CONTENT (USED AS POROSITY FOR NEFTRAN)
C
      MOISTC(INC)=MC(LAYUNS,MEDIA)
C
C
C
C
C
C   STORE THE VELOCITIES
C
      NEFVEL(INC)=VEL(LAYUNS,MEDIA)
C
C
C
C
C
C
C  CALCULATE AND SAVE THE NEW MASS FLUX FOR THIS PATH
C
      PTHMAS(NP,N)=PTHMAS(NP,N)*LYFAC
C
C
C
C
C
C
C   STORE THE RD'S FOR ALL THE ISOTOPES FOR EACH LAYER
C
C
C
C   LOOP OVER THE ISOTOPES TO OBTAIN THE RD FOR THIS LAYER
C   AND FOR THE MEDIA TYPE BE IT MATRIX(1) OR FRACTURE(2)
C
      DO 505 II=1,NISO
C
      NEFRD(II,INC)=RD(II,LAYUNS,MEDIA)
C
C
C
  505 CONTINUE
C
```

```
C
C
C
C  LET'S ADD THE SATURATED PORTION
C
C LEG COUNTER STARTS WITH THE INC THAT THE UNSATURATED LEGS ENDED WITH
C
      DO 520 I=1,NLYERS
C
      IF (SATLEN(NP,I) .EQ. 0.0) GOTO 520
C
C
      INC=INC+1
C
C
C    SET THE LENGTH FOR THIS LEG
      NEFLEN(INC)=SATLEN(NP,I)
C
C   SET THE POROSITY TO THE FRACTURE VALUE
C
      MOISTC(INC)=POR(I,2)
C
C
C   CALCULATE THE VELOCITY BASED ON THE GRADIENT AND THE
C   SATURATED CONDUCTIVITY
C
      NEFVEL(INC)=GRAD*COND(I,2)/MOISTC(INC)
C
C
C
C
C  CALCULATE THE RETARDATION BASED ON THE FRACTURE VALUES
C
      DO 520 II=1,NISO
C
      NEFRD(II,INC)=RD(II,I,2)
C
C
  520 CONTINUE
C
C
C
C
C
C  SET THE NEFLEG VALUE TO ACCOUNT FOR THE SATURATED LEGS
C
      NEFLEG=INC
C
C
C
C
C
C
C   NOW LET'S WRITE THE NEFTRAN INPUT FILE FOR THIS PATH
C    (UNITS WILL BE CONVERTED FROM MKS TO ENGLISH SYSTEM)
C
C
C
C
      IF (IDBUG .NE. 0) GOTO 499
```

```
C
      WRITE(35,'(1X,A,I4,A,I4,2X,A,E10.3,2X,A,E10.3)')
     1   'AREA & PATH #S',NP,',  ',N,'AREA INFIL. = ',INFIL(NP),
     2  'PATH MASS FRC.= ',PTHMAS(NP,N)
C
      WRITE(35,'(5X,A)') 'GROUP 1 - OPTIONS, STANDARD PRINT, 0=NO PRINT'
C
      WRITE(35,'(I5,5X,A)') 1,'LIST IF ALL THE  PARAMETERS'
C
      WRITE(35,'(I5,5X,A)') 1,'THE NETWORK LEG PROPERTIES'
C
      WRITE(35,'(I5,5X,A)') 1,'THE NETWORK JUNCTION PROPERTIES'
C
      WRITE(35,'(I5,5X,A)') 1,'THE MIGRATION PATH PROPERTIES ARRAY'
C
      WRITE(35,'(I5,5X,A)') 1,'THE DECAY CHAIN PROPERTIES ARRAY'
C
      WRITE(35,'(I5,5X,A)') 1,'THE ELEMENT PROPERTIES ARRRAY'
C
      WRITE(35,'(I5,5X,A)') 1,'FLUID VELOCITIES & SOURCE/DISCH FLOWS'
C
      WRITE(35,'(I5,5X,A)') 1,'SUBCHAINS, PRODUCTION & VELOCITIES'
C
      WRITE(35,'(I5,5X,A)') 1,'SPACE & TIME STEPS AND SOURCE TYPE'
C
      WRITE(35,'(I5,5X,A)') 1,'TIME INDEPENDENT OUTPUT, IF +-N, RATES/CON
     1CS EVERY NTH TIME STEP'
C
      WRITE(35,'(I5,5X,A)') 1,'DISCHARGE SUMMARY'
C
      WRITE(35,'(5X,A)') 'EXTERNAL FILES, NONZERO => FILE WRITTEN/READ'
C
      WRITE(35,'(I5,5X,A)') 1,'SRATE.DAT, SOURCE RATES (CI/Y) - WRITTEN
     1 TO UNIT 25'
C
      WRITE(35,'(I5,5X,A)') -1,'DRATE.DAT, DISCHARGE RATES (CI/Y) -
     1 WRITTTEN TO UNIT 30'
C
      WRITE(35,'(I5,5X,A)') 0,'EXTERNAL SOURCE FLAG (=1 =>READ UNIT 14)'
C
      WRITE(35,'(I5,5X,A)') 0,'SAMP.DAT, SAMPLED DATA FOR REPEATED TRIALS
     1  - READ FROM UNIT 10'
C
      WRITE(35,'(I5,5X,A)') 0,'VFIELD.DAT, UNIT 11, TIME-DEPENDENT
     1 VELOCITIES READ FROM'
C
      WRITE(35,'(I5,5X,A)') 0,'SFLOW.DAT, UNIT 12, TIME-DEPENDENT SOURCE
     1 FLOW RATES READ FROM'
C
      WRITE(35,'(20X,A)') 'RUN CONTROLS'
C
      WRITE(35,'(I5,5X,A)') 0,'USE DVM (0) OR ANALYTIC SOLN (NONZERO)'
C
      WRITE(35,'(I5,5X,A)') 1,'SOLVE NETWORK (0) OR INPUT VELOCITIES
     1(NONZERO)'
C
      WRITE(35,'(I5,5X,A)') 0,'LEACH (0), SOLUBILITIES (1), OR BOTH (2)
     1 FOR SOURCE RATES'
C
      WRITE(35,'(I5,5X,A)') 0,'FLOWTHRU (0), MIXCELL (1), OR CHOOSE (2)
```

```
     1 FOR SOURCE RATES'
C
      WRITE(35,'(I5,5X,A)') 0,'CONSTANT (0) OR EXPONENTIAL (1) LEACH
     1 RATE MODEL'
C
      WRITE(35,'(I5,5X,A)') 0,'FORCE (NONZERO) USE OF LEG-TO-LEG TRANSFER
     1 ALGORITHM'
C
C
C
      WRITE(35,'(20X,A)') 'DEBUG PRINTS, NONZERO => PRINT'
C
      WRITE(35,'(I5,5X,A)') 0,'DVM MULTIPLIERS FOR TRANSPORT, DISCHARGE,
     1 AND SOURCE'
C
      WRITE(35,'(I5,5X,A)') 0,'DIAGNOSTIC INFO FOR SUBROUTINE BAND'
C
      WRITE(35,'(I5,5X,A)') 0,'DIAGNOSTIC INFO FOR FUNCTION GIT'
C
      WRITE(35,'(I5,5X,A)') 0,'MATRIX/VECTOR SYSTEM SOLVED FOR FLOW'
C
      WRITE(35,'(I5,5X,A)') 0,'THE DATA READ FROM UNIT 10, SAMP.DAT'
C
      WRITE(35,'(I5,5X,A)') 0,'CONC. FLAG (=1 => WRITE TO UNIT 29 IN
     1 CI/FT**3)'
C
      WRITE(35,'(I5,5X,A)') 1,'INVENTORY AT RELEASE TIME'
C
      WRITE(35,'(I5,5X,A)') 0,'RESTRICTIONS PLACED ON THE TIME STEPS'
C
      WRITE(35,'(I5,5X,A)') 0,'LEG/JUNCTION & JUNCTION/LEG CONNECTIONS'
C
      WRITE(35,'(I5,5X,A)') 0,'VELOCITY FIELDS FROM UNIT 11'
C
      WRITE(35,'(I5,5X,A)') 0,'THE ATOM COUNT SUMMAARY'
C
      WRITE(35,'(I5,5X,A)') 0,'JUNCTION PRESSURES & LEG FLOWRATES'
C
      WRITE(35,'(I5,5X,A)') 0,'LEG-TO-LEG TRANSFER FRACTIONS'
C
      WRITE(35,'(I5,5X,A)') 0,'TIME SPENT AS EACH ISOTOPE IN A SUBCHAIN'
C
      WRITE(35,'(I5,5X,A)') 0,'TIME DEPENDENT FLOWRATES THROUGH SOURCE
     1 REGIME'
C
      WRITE(35,'(20X,A)') 'GROUP 2  -  PROBLEM SIZE'
C
      WRITE(35,'(I5,5X,A)') NEFLEG+1,'NUMBER OF NETWORK LEGS'
C
      WRITE(35,'(I5,5X,A)') 0,'NUMBER OF NETWORK JUNCTIONS'
C
      WRITE(35,'(I5,5X,A)') NEFLEG+1,'NUMBER OF MIGRATION PATH LEGS'
C
      WRITE(35,'(I5,5X,A)') NCHAIN,'NUMBER OF DECAY CHAINS INPUT'
C
      WRITE(35,'(I5,5X,A)') NISO,'NUMBER OF ELEMENTS INPUT'
C
      WRITE(35,'(A)') '   CHN1  CHN2  CHN3  CHN4  CHN5  - # PER CHAIN'
C
      WRITE(35,'(10I5)') (MEMPCH(II), II=1,NCHAIN)
```

```
C
      WRITE(35,'(A)') '   CHN1  CHN2  CHN3  CHN4  CHN5 - TRANSPORT-0,
     1SOURCE-1, NEITHER-2'
C
      WRITE(35,'(10I5)') (IZERO(II),II=1,NCHAIN)
C
C
C
      WRITE(35,'(20X,A)') 'GROUP 3 - SOURCE/FLOW PARAMETERS'
C
      WRITE(35,'(E10.3,5X,A)') PTHMAS(NP,N),'FRACTION OF INVENTORY
     1ACCESSED'
C
      WRITE(35,'(E10.3,5X,A)') 0.0,'PORE VOLUME CONTAINING SOURCE (FT**3)
     1 USED FOR MIXING CELL'
C
      WRITE(35,'(E10.3,5X,A)') AREA(NP,1)*MOISTC(1)*3.28*3.28,
     1 'PORE AREA OF SOURCE  (FT**2)'
C
      WRITE(35,'(E10.3,5X,A)') 1.0E-5,'LEACH RATE (1/Y) IF LEACH LTD'
C
      WRITE(35,'(5X,F5.2,5X,A)') 62.4,
     1  'DENSITY OF SOURCE FLUID (LB/FT**3)'
C
      WRITE(35,'(E10.3,5X,A)') AREA(NP,2)*MOISTC(NEFLEG)*3.28*3.28,
     1 'PORE AREA AT DISCHARGE (FT**2)'
C
C
C
      WRITE(35,'(20X,A)') 'GROUP 4 - TIME PARAMETERS'
C
      WRITE(35,'(E10.3,5X,A)') ETIME,'TIME TO END OF SIMULATION'
C
      WRITE(35,'(E10.3,5X,A)') 0.0,'TIME OF ONSET OF MIGRATION (Y)'
C
      WRITE(35,'(E10.3,5X,A)') 0.0,'TIME OF ONSET OF LEACHING (Y)'
C
      WRITE(35,'(E10.3,5X,A)') 0.0,'TIME STEP FOR SOURCE (Y)'
C
      WRITE(35,'(A)') 'CHAIN1 CHAIN2 CHAIN3 CHAIN4 CHAIN5 TIME STEPS BY
     1 CHAIN'
C
      WRITE(35,'(10(F5.1,2X))') (ZERO(II), II=1,NCHAIN)
C
C
C
      WRITE(35,'(10X,A)') 'GROUP 5 - INCREMENT DETERMNATION'
C
      WRITE(35,'(I5,5X,A)') 0,'USE DEFAULTS (0) OR SUPPLY ALL 16
     1 VALUES'
C
      WRITE(35,'(10X,A)') 'NETWORK LEG PROPERTIES ARRAY'
C
      WRITE(35,'(A)') 'LEG  INLET  OUTLET    LENGTH    AREA    HYDRAULIC
     1   BRINE'
      WRITE(35,'(A)') ' #    JCT    JCT       (FT)    (SQ-FT)   K  (FT/D)
     1   CONC.'
C
C
      WRITE(35,'(I2,5X,I2,5X,I2,4(5X,F5.1))') 1,1,2,10.*3.28,0.0,
```

```
    1  0.0,0.0
C
      DO 900 II=1,NEFLEG
C
      WRITE(35,'(I2,5X,I2,5X,I2,4(5X,F5.1))') II+1,II+1,
     1  II+2,NEFLEN(II)*3.28,0.0,0.0,0.0
C
  900 CONTINUE
C
C
C
      WRITE(35,'(20X,A)') 'MIGRATION PATH PROPERTIES ARRAY'
C
      WRITE(35,'(A)') 'LEG   DISPERS.   SPA. STEP  DIFFUS  MOBILE
     1IMMOB   MASS  XFER   VELOCITY'
      WRITE(35,'(A)') ' #      (FT)        (FT)    N/Y=0/1  POROS.
     1POROS.  COEF(1/Y)     (FT/YR)'
C
      WRITE(35,'(I3,5X,F5.1,5X,F5.1,5X,I2,3X,E10.3,2X,E10.3,2X,E10.3,2X,
     1  E10.3)') 1,DISPER*3.28,1.0*3.28,0,MOISTC(1),0.0,0.0,
     2  NEFVEL(1)*3.28E-3
C
C
      DO 901 II=1,NEFLEG
C
      WRITE(35,'(I3,5X,F5.1,5X,F5.1,5X,I2,3X,E10.3,2X,E10.3,2X,E10.3,2X,
     1  E10.3)') II+1,DISPER*3.28,1.0*3.28,0,MOISTC(II),0.0,0.0,
     2  NEFVEL(II)*3.28E-3
C
  901 CONTINUE
C
C
C
C
C
      WRITE(35,'(30X,A)') 'DECAY CHAIN ARRAY'
C
      WRITE(35,'(A)')   'NAME      ATOM  ELE-    LOC  PAR  PAR  FRAC    FRA
     1C      INVEN     HALF LIFE  WEIGH'
      WRITE(35,'(A)')   '  A6      MASS  MENT    NDX   #1   #2  FROM1   FRO
     1M2     (CI)       (Y)       FAC'
C
C
      NELM=0
C
      DO 902 II=1,NCHAIN
C
      DO 902 J=1,MEMPCH(II)
C
      NELM=NELM+1
C
C
      WRITE(35,'(A1,A6,A1,F6.0,2I6,2I5,2(2X,F5.3),2X,2E10.3,2X,F5.3)')
     1  '''',NAME(NELM),
     1 '''',AMASS(NELM),NELM,J,PAR1(NELM),PAR2(NELM),FRCFM1(NELM),
     2  FRCFM2(NELM),INV(NELM),HLIFE(NELM),EPAWGT(NELM)
C
C
  902 CONTINUE
C
```

```
C
C
C
C
C
      WRITE(35,'(20X,A)') 'ELEMENT PROPERTIES ARRAY'
C
C
      WRITE(35,'(A)') 'ELEM.    SOLUBILITY    LEG   MOBIL RD       IMMOBI
     1LE RD'
C
      WRITE(35,'(A)') 'INDEX       (G/G)       #    '
C
C
      NELM=0
C
      DO 903 II=1,NCHAIN
C
      DO 903 J=1,MEMPCH(II)
C
      NELM=NELM+1
      DO 903 KK=1,NEFLEG
C
C
C
      IF (KK .EQ. 1) THEN
C
      WRITE(35,'(I5,3X,E10.3,2X,I5,3X,E10.3,5X,E10.3)') NELM,0.0,KK,
     1  NEFRD(NELM,KK),NEFRDI(NELM,KK)
C
      WRITE(35,'(8X,10X,2X,I5,3X,E10.3,5X,E10.3)') KK+1,
     1  NEFRD(NELM,KK),NEFRDI(NELM,KK)
C
C
      ELSE
C
      WRITE(35,'(8X,10X,2X,I5,3X,E10.3,5X,E10.3)') KK+1,
     1  NEFRD(NELM,KK),NEFRDI(NELM,KK)
C
      ENDIF
C
C
C
C
  903 CONTINUE
C
C
C
C
C
C
C
C
  499 CONTINUE
      IF (IDBUG .EQ. 0) GOTO 500
C
      IF (LYFAC .EQ. 0.0) THEN
C
C
      WRITE(6,'(A,I5,A,I5,A)') 'AREA #',NP,' AND PATH #',N,
```

```
     1  '  HAS A MASS FLOW OF 0.0'
C
      ELSE
C
      WRITE(6,'(A,I5,A,I5,A,E10.3)') 'AREA #',NP,' AND PATH #',N,
     1  '  HAS A MASS FLOW OF ',PTHMAS(NP,N)
C
      ENDIF
C
C
C
  500 CONTINUE
C
C
C
      STOP
      END
/*
//READIT   EXEC  EDSIN,NAME='WDCCTUI.BUDHI6.DATA',DISK=FILE13
//READIT2   EXEC  EDSIN,NAME='WDCCTUI.INFIL.DATA',DISK=FILE13,
//          INPUT='&INFIL'
//READIT3   EXEC  EDSIN,NAME='WDCCTUI.LAYER.DATA',DISK=FILE13,
//          INPUT='&LAYER'
//DOIT     EXEC   FORVLKGO,REGION=5000K
//GO.SYSIN    DD   DSN=&INPUT,DISP=(OLD,DELETE)
//GO.FT20F001    DD   DSN=&INFIL,DISP=(OLD,DELETE)
//GO.FT21F001    DD   DSN=&LAYER,DISP=(OLD,DELETE)
//GO.FT35F001     DD     DSN=WDCCTUI.NEFTRAN2.SYSINP,UNIT=FILE,
//              VOL=SER=FILE13,SPACE=(TRK,(5,3)),DISP=(NEW,KEEP),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
```

25/30

```
@@@@@@@@@@@@          @@@@@@@@      @@@@         @@@@       @@@@
  @@@@    @@@@          @@@@      @@@@@@@@       @@@@@@     @@@@
  @@@@    @@@@          @@@@    @@@@    @@@@     @@@@@@@@   @@@@
  @@@@@@@@@@            @@@@    @@@@    @@@@     @@@@  @@@@@@@@
  @@@@  @@@@    @@@@    @@@@    @@@@@@@@@@@@     @@@@    @@@@@@
  @@@@    @@@@  @@@@    @@@@    @@@@    @@@@     @@@@      @@@@
@@@@@@    @@@@    @@@@@@@@      @@@@    @@@@     @@@@      @@@@
```

```
   @@@@  @@    @@ @@    @@ @@@@@@     @@
  @@  @@ @@@   @@ @@    @@   @@   @@   @@@@
 @@      @@@@ @@ @@    @@   @@   @@ @@   @@
 @@      @@ @@@@ @@ @ @@   @@@@@   @@   @@
 @@      @@  @@@ @@@@@@@   @@ @@   @@@@@@
  @@  @@ @@   @@ @@@ @@@   @@  @@ @@   @@
   @@@@  @@    @@ @@    @@ @@@   @@ @@   @@
```

Created at.........: 09:44:32 on: 03-10-92

Job Number.........: 79

```
THE TIME TO END THE SIMUALTION IN YEARS
   1.0E4
NUMBER OF AREAS,  MAX # OF LAYERS, AND MAX # OF UNSATURATED LAYERS
    2    4    4
INFILTRATION (MM/YR)  AND SATURATED GRADIENT FOR THIS REALIZATION
       0.3         0.1
DISPERSION LENGTH (M) USED FOR ALL NEFTRAN LEGS
     10.0
THE NUMBER OF CHAINS AND THE NUMBER OF ISOTOPES
      2    5
THE NUMBER OF ISOTOPES PER CHAIN
   3    2
ISOTOPE INFORMATION - 3 LINES PER ISOTOPE
LN 1 NAME, AMAS, P#1, P#1, FRC F1, FRC F2, INV(CI), HLIFE(Y), EPA WGT
LINE 2 MATRIX RD'S PER LAYER
LINE 3 FRACTURE RD'S PER LAYER
'NP237'     237   0   0     0.0       0.0       1000.    2.14E6    1.
   20.   30.    40.   50.
   5.    6.     7.    8.
'U233'      233   1   0   1.0    0.0          1000.     1.62E5     1.
   120.  130.  140.   150.
   15.   16.  17.  18.
'TH229'     229   2   0    1.0    0.0           1000.     7.3E3      1.
  220.   230.   240.  150.
  25.    26.    27.  28.
'MC230'  230    0    0    0.0    0.0    500.   5.            1.
  90.   91.  92.   93
  9.     8.   7.   6.
'MC232'  232    1    0    1.0    0.0    500.   20.           1.
 80.   81.   82.  83.
 18.   17.   16.  15.
TOPOPAH SPRING  CONDUCTIVITY(MM/YR), POROSITY, AND BETA
        .6       .11       1.8
       30.    4.1E-5       4.2
CALICO HILLS VITRIC CONDUCTIVITY(MM/YR), POROSITY, AND BETA
     8500.       .46       2.5
      300.    4.6E-5       4.2
CALICO HILLS ZEOLITIC CONDUCTIVITY(MM/YR), POROSITY, AND BETA
        .6       .28       1.6
      300.    4.6E-5       4.2
PROW PASS CONDUCTIVITY(MM/YR), POROSITY, AND BETA
       60.       .25       2.2
      300.    4.6E-5       4.2
********THIS INFORMATION IS REPEATED FOR EACH PART *****PART #1
INLET AND OUTLET AREAS (SQUARE METERS) FOR AREA #1
      100.      200.
AREA #1 LAYERS (M)
     50.   100.    10.   200.
SATURATED LENGTHS (M)
    55.  105.  15. 0.0
********THIS INFORMATION IS REPEATED FOR EACH PART *****PART #2
INLET AND OUTLET AREAS (SQUARE METERS) FOR AREA #2
      110.      220.
AREA #1 LAYERS (M)
     55.    110.    11.    220.
SATURATED LENGTHS (M)
    60.  0.0  0.0   0.0
```

27/30

```
@@@@@@@@@@@@          @@@@@@@@      @@@@        @@@@      @@@@
  @@@@    @@@@          @@@@      @@@@@@@@      @@@@@@    @@@@
  @@@@    @@@@          @@@@    @@@@    @@@@    @@@@@@@@  @@@@
  @@@@@@@@@@            @@@@    @@@@    @@@@    @@@@  @@@@@@@@
  @@@@  @@@@    @@@@    @@@@    @@@@@@@@@@@@    @@@@    @@@@@@
  @@@@    @@@@  @@@@    @@@@    @@@@    @@@@    @@@@      @@@@
@@@@@@    @@@@    @@@@@@@@      @@@@    @@@@    @@@@      @@@@
```

```
  @@@@  @@    @@ @@    @@ @@@@@@    @@
 @@  @@ @@@   @@ @@    @@   @@    @@  @@@@
@@      @@@@ @@ @@    @@   @@    @@ @@   @@
@@      @@ @@@@ @@ @ @@   @@@@@   @@   @@
@@      @@  @@@ @@@@@@@   @@ @@   @@@@@@
 @@  @@ @@    @@ @@@ @@@   @@   @@ @@   @@
  @@@@  @@    @@ @@    @@ @@@   @@ @@   @@
```

Created at.........: 09:45:02 on: 03-10-92

Job Number.........: 80

THE NUMBER OF VALUES IN THE TABLE FOR AREA PART #1
8
NOW INPUT THE VALUES (ONE SET/LINE) TOTAL INFIL., INFIL. FOR THIS PART
GOING FROM LOW TO HIGH VALUES (MM/YR)

.01	.01
.03	.03
.1	.1
.2	.18
.3	.28
.4	.3
.5	.4
10.	.6

THE NUMBER OF VALUES FOR AREA PART #2
6

TOTAL INFILTRATION, LOW TO HIGH VALUES	INFILTRATION FOR THIS PART (MM/YR)
.01	.01
.1	.1
.6	.5
1.	.52
4.	.53
10.	.54

29/30

```
@@@@@@@@@@@@          @@@@@@@@      @@@@        @@@@      @@@@
  @@@@    @@@@          @@@@      @@@@@@@@      @@@@@@    @@@@
  @@@@    @@@@          @@@@    @@@@    @@@@    @@@@@@@@  @@@@
  @@@@@@@@@@            @@@@    @@@@    @@@@    @@@@  @@@@@@@@
  @@@@  @@@@    @@@@    @@@@    @@@@@@@@@@@@    @@@@    @@@@@@
  @@@@    @@@@  @@@@    @@@@    @@@@    @@@@    @@@@      @@@@
@@@@@@    @@@@    @@@@@@@@      @@@@    @@@@    @@@@      @@@@
```

```
  @@@@  @@    @@ @@    @@ @@@@@@    @@
 @@  @@ @@@   @@ @@    @@   @@   @@   @@@@
@@      @@@@ @@ @@    @@   @@   @@ @@   @@
@@      @@ @@@@ @@ @ @@    @@@@@  @@   @@
@@      @@  @@@ @@@@@@@    @@ @@   @@@@@@
 @@  @@ @@    @@ @@@ @@@    @@   @@ @@   @@
  @@@@  @@    @@ @@    @@ @@@    @@ @@   @@
```

Created at.........: 09:45:04 on: 03-10-92

Job Number.........: 81

TOPOPAH SPRING MATRIX INTERPOLATION TABLE # OF ENTRIES

6

VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION

0.1	0.1
0.4	0.4
0.7	0.7
0.8	0.75
1.0	.9
3.0	.95

CAILICO HILLS MATRIX INTERPOLATION TABLE # OF ENTRIES

5

VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION

0.2	.2
0.5	0.5
.8	.7
1.2	.8
5.	.81

ME AND YOU MATRIX INTERPOLATION TABLE # OF ENTRIES

4

VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION

0.3	0.3
1.0	.8
4.0	.85
9.0	.87

JUST YOU MATRIX INTERPOLATION TABLE # OF ENTRIES

6

VALUES ARE ENTERED AS FRACTION OF TOTAL INFILTRATION AND
FRACTION OF MATRIX FLOW IN ASCENDING VALUES OF INFILTRATION

0.3	0.3
0.5	0.5
0.9	0.9
1.0	.93
2.0	.94
8.0	.96