CNWRA 96-005

USER'S MANUAL FOR MULTIFLO: PART I
METRA 1.0 β
TWO-PHASE NONISOTHERMAL FLOW SIMULATOR

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

May 1996

CNWRA 96-005

USER'S MANUAL FOR MULTIFLO: PART I

METRA 1.0 β

TWO-PHASE NONISOTHERMAL FLOW SIMULATOR

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Mohan S. Seth
Technical Software and Engineering

Peter C. Lichtner
Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

May 1996

CONTENTS

CONTENTS (cont'd)

TABLES

FIGURES

ACKNOWLEDGMENTS

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. NRC-02-93-005. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards (NMSS), Division of Waste Management (DWM). The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC. Sources of data are referenced in each chapter. The respective sources of these data should be consulted for determining their levels of quality assurance.

The authors gratefully acknowledge the technical reviews of Ronald T. Green and Goodluck I. Ofoegbu and the programmatic review of Wesley C. Patrick. Appreciation is due to Bonnie L. Garcia for her assistance in the preparation of this report.

1 INTRODUCTION

The computer code MULTIFLO describes multicomponent-multiphase reactive transport in nonisothermal systems in 1, 2, and 3 spatial dimensions. MULTIFLO is a general code for simulating multiphase, multicomponent transport processes with chemical reactions and reversible and irreversible phase changes in solids, liquids, and gases. The code MULTIFLO consists of two sequentially coupled submodules: Mass and Energy TRAnsport (METRA) and General Electrochemical Migration (GEM). METRA solves mass balance equations for water and air and an energy balance equation. The submodule GEM solves mass balance equations for multicomponent reactive transport of solute species. GEM takes into account aqueous speciation of solutes, gaseous speciation, mineral precipitation and dissolution reactions, and ion-exchange and sorption. GEM uses the flow field, temperature, pressure and saturation state computed by the METRA submodule for computing reactive transport of aqueous and gaseous species. The flow field computed by METRA can be altered through changes in porosity and permeability computed from GEM. Both METRA and GEM can also be run in stand-alone mode.

This User's Manual applies to the METRA simulator and is intended primarily for providing information and instructions to users in the preparation of input data. In addition to a detailed description of input data formats and structure, the report also provides an overview on the general capability of the simulator, explanation of various output results, procedure for compilation and run execution, and the other user-oriented features.

1.1 OVERVIEW OF METRA

METRA is a two-phase, three-dimensional (3D), nonisothermal flow simulator. Simulation of simpler systems are also possible for zero, one, or two spatial dimensions with a single (all liquid or gas) or two-phase fluid. Rock with zero porosity allows simulation of pure heat conduction problems. The equation of state for pure water allows temperature ranges of 1–800 °C and pressures below 165 bars. METRA is based on a fully implicit formulation using a variable substitution approach. Space discretization is based on a block-centered grid. The primary variables chosen are listed in Table 1-1, in which p is the total system pressure corresponding to liquid pressure for a pure liquid system and total gas pressure for a two-phase or pure gas-phase system.

Table 1-1. Choice of primary variables for different fluid states

Fluid State	Primary Variables			Remarks
Single-Phase Liquid	p_l	X_a	T	X_a = mole fraction of air
Two-Phase	p_g	P_a	S_g	P_a = air partial (Pa)
Single-Phase Gas	p_g	P_a	T	T = temperature (°C)

The three primary equations solved by METRA are: (a) total mass balance equation, (b) air mass balance equation, and (c) energy balance equation. The three equations for each grid block are solved simultaneously using the Thomas algorithm [for one-dimensional (1D) problems], and, optionally,

the D4-ordered direct method, or WATSOLV package (van der Kwaak et al., 1995). The D4-solver is suitable for two-dimensional (2D) problems. The WATSOLV package is based on incomplete factorization (ILU) followed by acceleration employing generalized minimum residual (GMRES) or biconjugate gradient stabilized (CGSTAB) procedures. This package is suitable for both 2D and 3D problems.

The code METRA treats mass transport by (i) Darcy's law (as modified by the relative permeability) which includes capillarity, gravity, and viscous forces; and (ii) diffusion of air-vapor. Energy transfer is treated for convective and conductive processes. Arbitrary user-specified spatial variation in porosity, absolute rock permeability (in all three spatial directions), rock tortuosity, thermal conductivity, rock characteristic curves (relative permeability and capillary pressure), and other pertinent factors are included in the code. The code can be run in cartesian or cylindrical geometries. Other WP coordinates may be readily treated by directly specifying the transmissibility and rock bulk volume. A matrix-fracture system may be represented by employing an equivalent continuum concept. Vapor-liquid equilibria may be defined with lowered vapor pressure due to capillary forces. Efforts were made to design the code efficiently from a computational as well as memory consideration. It is expected that with 128 MB memory, 3D two-phase problems with over 50,000 grid blocks can be accommodated. For single-phase flow or heat conduction only, over 100,000 grid blocks should be easily accommodated. Time-dependent boundary conditions are accommodated in the code and include both Dirichlet (constant-field variables) and Neumann (constant flux) types. The boundary conditions can be readily modified with time. Heat and mass sources and sinks may be designated at desired grid nodes as a function of time. Tabular data are linearly interpolated corresponding to the midpoint of the time step.

Some of the other features included in the simulator are as follows. Flexible output can be obtained at specified times or time-steps. Output also includes data files for graphical post processing and restart files. Intermediate variables may be printed by debug options. Mass transfer in the gas phase includes enhanced binary diffusion. Extensive default parameters are provided for properties of air, convergence criteria, method of solution, diffusion parameters, and a host of others. PVT properties for H_2O are calculated optionally by a rapid table look-up procedure or are provided directly using the International Formulation Committee (1967) functions. Gas mixture properties are calculated assuming the ideal gas law. Gas viscosity is computed using the kinetic theory of gases (Hirschfelder et al., 1954). Arbitrary general heterogeneity for all rock parameters including capillary pressure and relative permeability curves is possible. Options are available for van Genuchten, Linear, Tabular, and Corey's equations for material properties. Rocks with matrix and fractures may be modeled using the equivalent continuum concept with the assumption that the capillary pressure for the matrix equals that of fractures. Vapor pressure lowering of H_2O-phase behavior optionally accounts for the lowered vapor pressure due to the capillary pressure as defined by Kelvin's equation. Options exist for automatic time-step-size calculations, and provision to cut a time-step repeatedly (specified number of times) to ensure that the solution vector changes within a specified limit. Partial dynamic memory management based on run time parameters has been implemented in the code. Mass and energy balances are calculated based on cumulative and incremental balances to monitor the accuracy of the solution.

2 INPUT-DATA STRUCTURE

All input data are grouped into three types:

(i) Fixed Data: Consists of just two title lines and a line indicating restart or initial run. A restart run is a run that is resumed from an earlier run obtained from a saved restart file.

(ii) Initialization Data: Time-independent data specifying material properties and initial conditions. Initialization data constitute the bulk of the input data in most cases.

(iii) Recurrent Data: Time-dependent data including source/sink tables, method of solution, output frequency, restart file writing, boundary conditions, tolerances, iterations, time-step sizes, target time, and such other parameters.

All data are in essentially free-format and associated with a keyword. Thus, the order of input in any particular group is generally immaterial except for the fixed data which must start with two title lines followed by the 'RSTART' keyword (discussed in Section 3.2).

For a normal run (as opposed to a Restart Run), the input data must be in the order of (i) Fixed Data, (ii) Initialization Data, and (iii) Recurrent Data. For a Restart Run, the Initialization Data must be skipped, and Recurrent Data should include data only from the time the run is to be resumed. If only Initialization is to be carried out, the run can be stopped without marching in time with a run terminator keyword 'END' after the Initialization Data.

All input data must not exceed 10 entries per line for floating point (real) numbers and 20 entries for integers per line. The record length must not exceed 120 characters.

2.1 KEYWORDS

All keywords are followed by either none, one, or more set(s) of data lines depending on the type of data. A keyword can be up to 20 characters long; however, only the first four characters are used for data identification purposes, and can start anywhere on a line (the keyword does not have to start from column 1) EXCEPT that there should be no other nonblank characters before the keyword on a keyword line. A keyword may follow one or more variables as dictated by the input requirements. Keywords may be in upper case, lower case, or a combination of upper and lower case letters, but must NOT have any embedded blanks. For example, the keyword THERMAL DATA is inadmissible, but THERMAL-DATA or Thermal-Data is acceptable.

2.2 COMMENT LINES

The code provides the ability to include comment lines in the input file which are ignored during processing. Comment lines are identified by a colon ':'. The colon and the following text/data contained on a single line are not processed by the code. The colon can also follow data appearing on a line, in which case the information after and including the colon will be ignored. To deactivate trailing data on a line or to introduce some comment on a data line, an exclamation point (!) may also be introduced such that after the exclamation point and including it, all the information on a data line is ignored.

2.3 SKIP/NOSKIP KEYWORDS

A facility is provided to skip over one or more blocks of data in the input file without physically deleting these lines of data. This procedure is invoked by means of the SKIP and NOSKIP keywords.

Read: ISKIP

ISKIP = SKIP marks the beginning of data to be skipped. Input data are skipped and are not to be processed following this line until a NOSKIP keyword is encountered.

= NOSKIP marks the end of the input data to be skipped. Data following this keyword is to be processed. This keyword has no effect if a SKIP keyword is not introduced prior to the NOSKIP keyword. In effect, the keyword NOSKIP negates the effect of the SKIP keyword.

If a segment or block(s) of data is not to be processed, insert a SKIP keyword at the beginning of such a block and enter a NOSKIP keyword at the end of the block. Repeated use of SKIP and NOSKIP keywords has the effect of removing the data between the SKIP and NOSKIP keywords in the data file during execution while preserving the entire input data file intact. If a NOSKIP keyword is not encountered following a SKIP keyword, all data following the SKIP keyword will be removed.

Example:

```
|
|
Data
|
|
SKIP
|
|
Data        Remove data beginning with the SKIP keyword until the NOSKIP keyword is
            encountered.
|
|
NOSKIP
|
|
Data        Process data following the NOSKIP keyword until another SKIP keyword is
            encountered.
|
|
SKIP
```

3 FIXED DATA

3.1 TITLE LINES

Two title lines are required, and must be the first two lines in a data file. Any alphanumeric characters may be assigned for the title with up to 80 characters per line. If no title or only a one line title is desired, simply read a blank line(s) to fulfill the two-line requirement. Comment lines beginning with a ':' (colon) can precede the title data but cannot be inserted between the two title lines.

The 'title lines' should precede the run-type keyword as below.

3.2 RUN TYPE LINE

The run-type keyword identifies whether a run is a 'normal run' or a 'restart run'.

Read: RSTArt NRST

RSTArt = Keyword defining run type.

NRST = Number of restart records from which the run is to be resumed. During a normal run, a set of restart records may have been written at different times. 'NRST' specifies the particular set of records from which to resume a run. For a normal run, read 'NRST' as zero, or simply leave the field blank.

Example: Fixed Data

```
This is a test run for a 3D problem
        April 10, 1995
:
RSTArt  0       :normal run

RSTArt  6       :restart run starting after the sixth restart data set.
```

Note: The text on the RSTArt keyword following and including the colon is not processed and is included for convenience and clarity.

For a 'Restart Run', skip the Initialization Data and proceed directly to the Recurrent Data section.

4 INITIALIZATION DATA KEYWORDS

The Initialization Data constitute the bulk of the data file which are specified by different keywords. The data can be entered in any order with the exception of the GRID keyword which must be first.

The keywords used are:

GRID	EQUIlibration
CMP2	INIT
CONDuction	LIQUid
CXYZ	MONItor
DEBUg	PCKR
DEPTh	PHIK
DXYZ	THERmal
ENDS	

Since the code uses only the first four characters, the remaining characters in several of the key words (shown in lower case) are used only for a more descriptive identification of the keyword.

The presentation below is in alphabetical order, although the data may be entered in any order, except for the GRID keyword which must be at the beginning of the Initialization Data as it specifies the memory size to be used for dimensioning purposes.

4.1 GRID GEOMETRY DATA [GRID]

Initialization Data — Keyword: **GRID** Required

The GRID keyword defines the grid geometry (radial or cartesian), the size, and other optional parameters.

Read: GRID GEOMETRY NX NY NZ IVPLWR IPVCAL IOUTPT PREF TREF HREF

GRID = Keyword defining grid geometry and optional parameters.

GEOMETRY = RADIAL for cylindrical geometry
= XYZ for cartesian geometry.

NX = Number of grid blocks in x or r direction.

NY = Number of grid blocks in y or theta direction.

NZ = Number of grid blocks in z direction.

IVPLWR = Index for vapor pressure lowering.
= 0, Do not invoke vapor-pressure lowering.
= 1, Invoke vapor-pressure lowering of H_2O due to capillary suction

IPVCAL = Index for calculating water PVT properties.

= 0, Construct lookup tables using correlations.
= 1, Use correlations for water properties.

IOUTPT = Index for output of Initialization data.
= 0 Suppress all initialization output.
= 1 Suppress initial porosity, permeability distribution in addition to items corresponding to the value = 2.
= 2 Suppress initial condition, transmissibility and related output.
= 3 Print full initialization output.

PREF = Reference pressure for pore compressibility [Pa]. [Default = initial pressure of block (1,1,1)].

TREF = Reference temperature for pore compressibility [°C]. [Default = initial temperature of block (1,1,1)].

HREF = Reference depth of grid block (1,1,1). That is, HREF specifies the depth of the first block (top). Other block depths are computed internally with reference to this depth. Note that the depth increases (positive) downwards.

By appropriate assignment of NX, NY, and NZ, 1, 2, and 3D problems may be defined. A zero dimensional (single block) problem can be defined by setting all NX = NY = NZ = 1. Computational limitations of the number of blocks in any direction and the total number of blocks can be specified by the user (see section on Dimensioning and Compilation).

IPVCAL = 0 directs the simulator to construct a PVT-table for H_2O for a rapid table look-up procedure in subsequent calculations of fluid properties. The table is constructed at 2 °C intervals for the range of 5–369 °C for saturated fluid, compressed liquid and single-phase gas.

PREF and TREF are used to calculate porosity change with pressure and temperature. It is assumed that the values read-in for porosity are assigned at this reference pressure and temperature. If default values are used, the corresponding fields may be left blank or read as zeros.

Example:

```
:   geometry  nx  ny  nz  ivplwr  ipvcal  ioutput  pref  tref  href
GRID XYZ      21  1   32  0       0       1        0.    0.    0.
```

The above designates a cartesian grid of $21 \times 1 \times 32$ with no vapor pressure lowering, table look-up for H_2O properties, and default values for reference p, T, and depth. The comment line preceding the data line identifies the variables.

4.2 PVT PROPERTIES OF COMPONENT-2 [CMP2]

Initialization Data — Keyword: **CMP2** Optional

This keyword overrides the internally set values for properties of air (component-2). If the second component is CO_2, nitrogen, or any other gas, or if the default properties specified for air are unsatisfactory, they may be amended by this keyword. Component-1 is always designated as H_2O.

This keyword need not be used if air is the second component and the default properties are acceptable.

Read: CMP2 FMW2 SPHTC2 EQK2

CMP2 = Keyword for assigning this data set.

FMW2 = Molecular weight of component 2 (Default = 28.964).

SPHTC2 = Constant volume specific heat of component-2 [J/Kg-mole-K]. (Default = 21232 J/kg-mole-K).

EQK2 = Proportionality constant for Henry's Law for dissolution of component-2 in liquid water (Default = 10^{10} Pa)

Example:

```
: cmp2     fmw2   sphtc2  eqk2
 CMP2(co2) 16.0   0.      1.e9
```

In the above, default value for the specific heat will be used. (co2) is added to the keyword only for identification purposes and not processed internally.

4.3 KEYWORD FOR PURE HEAT CONDUCTION [CONDuction]

Initialization Data — Keyword: **CONDuction** Conditional

CONDuction keyword allows calculation of heat transfer by conduction only with zero porosity and permeability. If the rock thermal conductivities CKDRY and CKSAT read by the THERmal keyword are not equal, they are internally set such that CKSAT=CKDRY. All read-in porosity and permeability values are internally set to zero.

If all read-in porosity values are zero and this keyword is not invoked, the code will internally set the system for heat conduction only.

There are no associated data with this keyword.

Read: CONDuction

CONDuction = Keyword for designating the system for conduction-only heat flow with zero porosity and permeability.

Example:

CONDuction

4.4 GRID-BLOCK COORDINATES [CXYZ]

Initialization Data | Keyword: **CXYZ** Conditional

This keyword designates the coordinates of block boundaries with reference to an origin at the southwest corner of the top surface of block (1,1,1) such that the z-axis is positive downwards as shown in Figures 4-1 and 4-2 (left-handed coordinate system for Cartesian and radial coordinates). These coordinates are used to calculate the block sizes internally. Data corresponding to this keyword should be omitted if the block sizes are directly specified using the keyword DXYZ. If data corresponding to both keywords DXYZ and CXYZ are specified, the last values read-in will be retained.

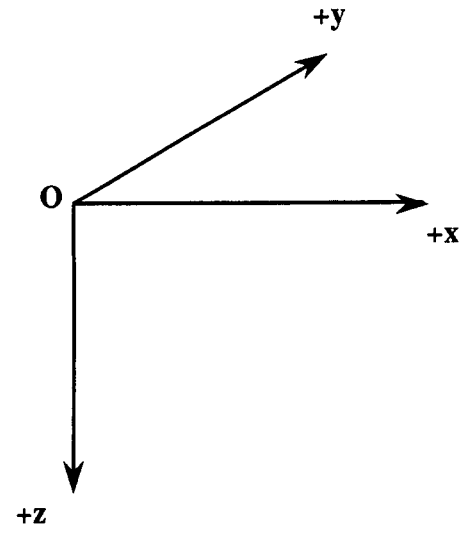

Figure 4-1. Cartesian xyz-coordinate system

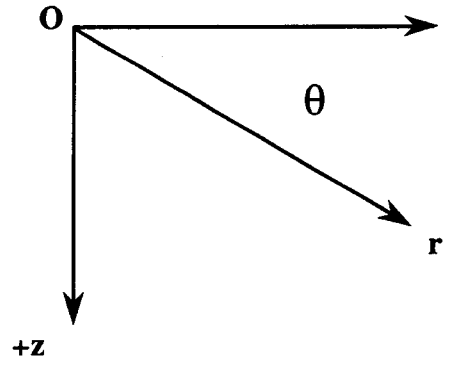

Figure 4-2. Radial coordinate system

Read: CXYZ IGRID RW RE

Read: (X(I), I = 1,NX)

Read: (Y(J), J = 1,NY)

Read: (Z(K), K = 1,NZ)

CXYZ = Keyword for reading grid-block interface coordinates.

IGRID = Index for grid-type.
= 0 Block centered.
= 1 Point-distributed (not implemented).
= 2 Boundary node at the surface corresponding to CTOUGH.

RW = Inner boundary radius for cylindrical system [meter].
RE = Outer boundary radius for cylindrical system [meter].

For cartesian geometry, RW and RE are not required.

X = X-direction coordinates for block-interfaces starting with the block interface between block 1 and 2 and ending with the outer boundary of NXth block with a total of NX values. For cylindrical system, X designates the radius to block center from the axis of the cylinder [meter].

Y = Y-direction coordinates for block interfaces in y-direction (similar to x-direction) [meters]. For cylindrical systems, Y, designates the angle with reference to an arbitrarily chosen zero degree plane [degrees].

Z = Z-direction coordinates for block interfaces in the z-direction where the z-axis is positive (increases) downwards [meter].

Ten values are read per line for X, Y, and Z. Read NX nonzero values for X, NY values for Y, and NZ values for Z. For cylindrical geometry, if fewer than NX values are read for X coordinates, the unspecified values are computed internally based on logarithmic spacing. RW must be $<X(1)$ and $RE>X(NX)$.

Examples:

Cartesian Geometry:

```
:     igrid  rw   re
CXYZ    0    0    0
:   x(1)  x(2) ....
    1. 2.  3.5  5.  7. 9.  12.  15.  20.  30.   ! x-direction
   45. 60. 100.
:   y(1)
    1.                                   ! y-direction
:   z(1) z(2) ...
    10. 20. 30. 45.  60.  75.  100.  150.       ! z-direction
```

The above designates coordinates for a cartesian grid system of 13 blocks in x-direction, 1 block in y-direction, and 8 blocks in z-direction.

Cylindrical geometry:

```
:     igrid   rw     re
CXYZ   0      .001   100.   ! radial geometry with inner and outer radius of cylinder .001 and 100 m.
:
:     r(1)  r(2)
      .5    1.              ! only 2 values are specified. The rest will be computed internally.
:
:     y(1)
```

```
      90.                          ! one angular sector of 90 degrees.
:     z(1)  z(2) ...
      10.   20.  30.  45.  60.  75.  100.  150.  ! z-direction
```

4.5 DEBUG OUTPUT FOR INITIALIZATION [DEBUg]

Initialization Data — Keyword: **DEBUg**, Optional

This keyword prints out intermediate variables for a designated region of grid blocks for debugging or examination of intermediate results. This option should not be used for routine production runs because of the extensive output it produces.

```
Read: DEBUg    IDBUG
Read: I1  I2  J1  J2  K1  K2
        .  .  .  .  .  .
        .  .  .  .  .  .
        0
```

DEBUg = Keyword for printing intermediate variables.

IDBUG = Index designating the level or extent of output.

= 0, turn-off the previously set debug output.
= 1, print minimum amount of debug output.
= 2, print intermediate level of debug output.
= 3, print full debug output.

I1, I2
J1, J2
K1, K2 = Indices bounding the region in x (r), y (theta), and z-directions for which more detailed output will be printed.

Read one region per line and terminate the read-sequence by reading a blank line or a line with one or more zeros.

The DEBUg option is turned off internally upon completion of initialization. A new DEBUg keyword must be introduced for debug output in the recurrent-data section.

Example:

```
Debug     1    ! Level = 1
1 3 1 1 1 5    ! Region 1 for debug output
5 9 1 1 7 11   ! Region 2 for debug output
0              ! Debug region terminator (must be the last line)
```

4.6 DEPTHS OF GRID BLOCKS [DEPTh]

Initialization Data — Keyword: **DEPTh**
Optional

This keyword defines the depth of grid blocks. It can be used to set the depth of each block to define uneven topographies or a constantly dipping bed. If omitted, a depth of zero corresponding to a dip angle of zero is set for the top layer and other layer depths are computed by adding the thicknesses of the overlying layer or layers.

Read: DEPTh IDEPTH DIP

DEPTh = Keyword for calculating the depth of a uniformly dipping formation.

IDEPTH = Index for the method for specifying the depth.

= 1 Use the dip of the formation to calculate depths.
= 2 Top layer depths are specified from which the other layer depths are computed by adding the thicknesses of the overlying layers.

= 3 Depths of all blocks are specified.

DIP = Dip in degrees. (Dip is formation inclination with respect to horizontal plane.) If IDEPTH > 1, the DIP is not used.

Read the following additional data for IDEPTH > 1 in the following format.

If IDEPTH = 2

```
do J = 1,NY
     read() (H(I,J,1),I=1,NX)
end do
```

If IDEPTH = 3

```
do K = 1,NZ
  do J = 1,NY
     read() (H(I,J,K),I=1,NX)
  end do
end do
```

1. The above read structure is as follows:

   - Read NX values of depths for the top layer for row 1 (J=1, K=1).

- If NX > 10, read 10 values per line on successive lines until all NX values are read for the row.

- Repeat Steps 1 and 2 for successive rows (J = 2,3,..NY) for the top layer (K = 1) starting each row on a new line.

- If IDEPTH = 3, repeat steps 1 to 3 for successive planes (K=2,3,..NZ) until all NZ planes are read-in.

2. The depths are measured POSITIVE (INCREASE) DOWNWARDS. The top layer corresponds to K = 1, and the bottom layer at K = NZ. That is, the layers are numbered from top to bottom. The depths are specified in units of meters.

3. The read-in depth should be at the center of the upper surface of a block. The block center depths (halfway through the thickness) are computed internally by adding half the thickness to the read-in values.

4. The calculated depths of lower layers (K=2,3..NZ) using option IDEPTH = 2 can introduce excessive errors if the formation is steeply dipping or highly undulating in which case the option IDEPTH = 3 is recommended.

5. All depths are to be specified with reference to the depth of block (1,1,1) to which HREF (read-in with the Keyword 'GRID') is added internally.

Examples:

Constant Dip Specified (IDEPTH = 1).

DEPTH 1 10. ! designates 10° dip of the formation for option IDEPTH = 1.

Top Layer Depths Specified (IDEPTH = 2)

```
DEPTH 2
15 20 30 40 50 70 90 100 110 120
130 140 150 160                    !(row = 1, Layer = 1)
20 25 35 45 55 75 95 110 120 130
140 160 180 200                    !(row = 2, layer = 1)
. . . . . .
. . . . . .
100 120 130 140 150 160 170 180 190 200
220 230 240 250                    !(row = NY, layer = 1)
```

If all Layer-depths are specified (IDEPTHS = 3), the same data structure as the above is used for each successive layer.

4.7 GRID-BLOCK SIZES [DXYZ]

Initialization Data — Keyword: **DXYZ**
Conditionally Required

This keyword designates the size of grid blocks in x (r), y (theta), and z directions. It also specifies the choice of grid as to block-centered (default) or point-distributed (not implemented). This data set should be omitted if the grid-block coordinates are specified with the keyword CXYZ.

Read: DXYZ IGRID RW RE

Read: (DX(I), I = 1, NX)
Read: (DY(J), J = 1, NY)
Read: (DZ(K), K = 1, NZ)

DXYZ = Keyword for reading grid-block sizes.

IGRID = Index for grid-type.
= 0 block centered.
= 1 Point-Distributed (not implemented).
= 2 Boundary node at the surface corresponding to CTOUGH.

RW = Inner Boundary radius for cylindrical system [meter].
RE = Outer Boundary radius for cylindrical system [meter].

For cartesian geometry, RW and RE are not required.

DX = Block sizes in x-direction [meter]. For a cylindrical system, DX designates the radius to block center from the axis of the cylinder.

DY = Block sizes in y-direction [meter]. For a cylindrical system, DY designates the sector angle of a block in degrees.

DZ = Block thicknesses [meter]. Note that the z-direction is positive downward with the top block numbered as 1 and the bottom block as NZ.

For cylindrical geometry, only the value of DX(1) may be read, and the rest will be computed internally based on logarithmic spacing. The inner radius RW must satisfy the inequality: RW < DX(1).

Ten values are read per line for DX, DY, and DZ starting DX, DY, and DZ each on a new line. Read NX nonzero values for DX, NY values for DY, and NZ values for DZ. If the values read are less, the code will assign the last nonzero values to the remaining sizes. For example, if all the blocks are of constant size, reading only the first block size will be sufficient. In case of a cylindrical system, if DX(3) through DX(NX) are to be computed internally, simply read DX(1) and DX(2) on one line. When a zero value for DX(I), DY(J), or DZ(K) is encountered, further reading for that variable is terminated and the data are internally set equal to the last nonzero DX(), DY() or DZ() value read.

Examples:

Cartesian Geometry:

```
:      igrid   rw     re
DXYZ     0      0     0
:    x(1)   x(2) ....
     1.  1.   1.5   1.5   2.  2.  3.  3.  5.  10.   ! x-direction
    15.  15.  40.
:    y(1)
     1.                                             ! y-direction
:    z(1) z(2) ...
     10.  20. 30.  45.   60.   50.   50.  150.        ! z-direction
```

The above designates block sizes for a cartesian grid system of 13 blocks in x-direction, 1 block in y-direction, and 8 blocks in z-direction.

Cylindrical Geometry:

```
:      igrid    rw      re
DXYZ   0       .001    100.    ! radial geometry with inner and outer radius of cylinder .001 and 100 m.
:
:      r(1)  r(2)
       .5    1.                ! only 2 values are specified. The rest will be computed internally.
:
:      y(1)
       90.                     ! one angular sector of 90 degrees.
:      z(1)   z(2) ...
       10.     20.   30.   10.   20.  40.  100.  150.  ! z-direction
```

4.8 RUN TERMINATOR DATA [ENDS]

Initialization Data — Keyword: **ENDS** Conditional

This keyword should be read ONLY if the run is to be terminated after initialization. To complete initialization properly, ENDS should be the last keyword in the initialization data set. For a normal run with time-stepping, the ENDS keyword should be used only at the end of the Recurrent data.

Read: ENDS

ENDS = Keyword terminating the run.

Example:

```
Ends        ! No data will be processed after this data.
```

4.9 EQUILIBRATION DATA [EQUIlibration]

Initialization Data — Keyword: **EQUIlibration** Optional

This keyword sets the system initially in capillary-gravity equilibrium for zero flux boundary conditions. The fluid velocity in all directions is assumed to be zero. That is, the gas and the liquid phase potentials are constant throughout the system. At present, only van Genuchten functions are used for equilibration. Use of this keyword overrides the pressure, temperature and saturation specified by the keyword PHIK.

Read: EQUIL DEPTH PDEPTH TDEPTH TGRAD PARAM IEQUIL

EQUIL = Keyword for initial equilibration of the system.

DEPTH = Reference depth at which gas phase pressure (PDEPTH) and temperature (TDEPTH) are specified [m].

In a two-phase system, if a water table (gas-water contact) exists within the system, it is recommended that DEPTH correspond to the depth of the water-table. A water-table (gas-water contact) depth is defined as the highest depth at which 100 percent water saturation exists.

PDEPTH = Specified gas phase pressure at the specified depth DEPTH [Pa].

TDEPTH = Specified temperature at the specified depth DEPTH [°C].

TGRAD = Temperature gradient (a constant) used to calculate temperature of each grid block [°C/m].

PARAM = Parameter whose value is dependent on the variable IEQUIL. See IEQUIL below.

= Gas saturation at depth = DEPTH, if IEQUIL > 0 [fraction].
= Capillary pressure (Pg-Pw) at depth = DEPTH, if IEQUIL < 0 [Pa].
= Pressure gradient for gas pressure in a two-phase or pure gas system, and liquid pressure for a pure liquid system (a constant) if IEQUIL = 0 [Pa/m].

IEQUIL = Index for defining PARAM

= < 0, PARAM = capillary pressure specified at DEPTH.

= > 0, equal to the sequential number defining the van Genuchten relative-permeability function defined by the PCKR keyword. The integer must be in the range of 1 to NST, where NST is the number of PCKR functions read. The integer is used to calculate the capillary pressure corresponding to the given gas saturation, PARAM.

= 0, if PARAM = pressure gradient.

Examples:

```
:     depth  pdep  tdep  tgrad param  iequil
EQUIL  0.   1.e5  .25.  .025  0.4     2     ! equil with Sg specified

EQUIL  0.   1.e5  .25.  .025  1.e4    -1     ! equil with cap. press.

EQUIL  0.   1.e5  .25.  .025  9800.   0     ! equil with p-gradient.
```

4.10 DATA FOR SPECIFICATION OF INITIAL CONDITION [INIT]

Initialization Data — Keyword: **INIT** Required

The INIT keyword reads initial pressure, saturation, temperature and mole fraction of component 2 (air). For two-phase regions, temperature is used to calculate the partial pressure of air in the gas phase which is used as a primary variable.

Read: INIT FN
Read: I1 I2 J1 J2 K1 K2 P T SG XA SGM

INIT = Keyword for reading initial conditions.
FN = File name from which the initial condition data are read (optional). The file name must have an extension of '.int' in lower case letters, but the extension should not be included in FN specification. The file name 'FN' (excluding the extension) should not exceed 8 characters. The run will stop if the file name is misspelled or the file fn.int does not exist. Default = blank space, in which case the data is read from the input file.

I1, I2 = First and last index in x (r) direction.
J1, J2 = First and last index in y (theta) direction.
K1, K2 = First and last index in z-direction.

These indices define a region for which the properties are constant.

P = Pressure for the defined region [Pa].

T = Temperature for the defined region [°C].

SG = Gas phase saturation for the defined region [fraction].

XA = Mole fraction of air in the defined region [fraction].

SGM = Matrix gas phase saturation for the defined region, fraction. Ignored if the ECM option is not used [fraction]. Default for SGM = SG for ECM option.

XA is an independent variable only for a single-phase state, and, therefore, it is not used for a two-phase system.

Read as many lines as required to define the initial condition for the entire system, and terminate the sequence by reading a line with one or more zeros or blanks.

If two or more regions overlap, the last region read overrides the earlier ones.

Initial conditions must be specified for the entire system (for all grid blocks), otherwise results may be unpredictable because default values are not used. For the ECM option, P, T, and SG correspond to the values for the fracture continuum.

Example:

:	il	i2	j1	j2	k1	k2	p	t	sg	x2	sgm
	1	64	1	32	1	1	1.0e5	25.0	0.5	0.	.50

4.11 DATA FOR SINGLE PHASE LIQUID SYSTEM [LIQUid]

Initialization Data — Keyword: **LIQUid**
Conditional

The LIQUid keyword designates the system to be all liquid and isothermal. In this case, the computational work and the memory requirement are reduced by more than an order of magnitude. In addition, this option enables simulation of a nonaqueous fluid.

Read: LIQUid DWS VISW CWS

LIQUid = Keyword for designating the system to be isothermal and single phase liquid.

DWS = Density of liquid [kg/m^3]. (Default = water density computed internally).

VISW = Liquid viscosity [kg/m/s]. (Default = water viscosity computed internally).

CWS = Liquid compressibility [1/Pa]. (Default = water compressibility computed internally)

The default values correspond to the temperature and pressure of block (1,1,1).

Example:

```
Liquid  970 0 0
```

In the above, the density of liquid is assigned a value of 970 kg/m^3, while the viscosity and the compressibility are taken as that of water from the internally calculated pvt-properties.

4.12 DATA FOR MONITORING NODES [MONItor]

Initialization Data — Keyword: **MONItor** Optional

The keyword MONItor allows monitoring key variables of designated nodes as a function of time. The information is written in the file fn_prn, which may be monitored as the simulation proceeds. This file may also be used to generate time-history plots at the monitored nodes.

Read: MONItor m1 m2 m3 m4 m5

MONItor = Keyword for assigning elements to be monitored.

m1..m5 = Element numbers which are to be monitored. A maximum of five elements may be assigned.

An element number is calculated using the following relation:

$$m = i + (j-1)\, nx + (k-1)\, nx\, ny$$

where i, j, k are indices for a 3D grid, and m is the element number.

Example:

```
Monitor
1 3 7      ! primary variables for these three nodes will be written on lfn_prn file for each time
           step.
```

4.13 DATA FOR ROCK CHARACTERISTIC PROPERTIES [PCKR]

Initialization Data — Keyword: **PCKR** Required

This keyword specified the relative permeability and capillary pressure characteristics for different rock types. The characteristic curves included are equivalent continuum, van Genuchten, Linear, Tabular, and Corey's.

Read: PCKR FN
Read: NO TYPE SWIRM RPMM ALPHAM SWEXT SGC IECM
If IECM > 0, then
Read: SWIRF RPMF ALPHAF PHIM PHIF PERMM PERMF

PCKR = Keyword defining relative permeability and capillary pressure characteristic curves.

FN = File name from which data for potential properties are read (optional). The file name must have an extension of '.pck' in lower case letters, but the extension should not be included in FN specification. The file name 'FN' (excluding the extension) should not exceed 8 characters. The run will stop if the file name is misspelled or the file fn.pck does not exist. Default = blank space, in which case the data is read from the input file.

NO = Sequential number.

TYPE = 3 character string identifying the type-curve.

= VAN-gen, van Genuchten Curves.
= LINear, Linear curve.
= TABular, Data are read-in in Tabular form.
= CORey, Corey's curves.

If TYPE = Van-Gen read the following:

SWIRM = Irreducible liquid saturation (Krw = 0 below it) [fraction]. For the Equivalent Continuum Model (ECM), it represents the irreducible liquid saturation for the matrix.

RPMM = Exponent 'm' in van Genuchten equation [see Eq. (A-1)]. For the ECM, RPMM represents the exponent for the matrix.

ALPHAM = Parameter α in van Genuchten equation [1/Pa]. For the ECM, ALPHAM represents the matrix parameter.

The preceding three variables refer to the matrix continuum for ECM formulation, IECM > 0.

SWEXT = >0 Liquid saturation below which the capillary pressure is calculated based on the slope dPcw/dSw evaluated at SWEXT. SWEXT must be greater than SWIRM. Default value for SWEXT = SWIRM+.05. A value of SWEXT very close to SWIRM may result in performance degradation. The effect is likely to be more severe with vapor pressure lowering option. SWEXT is set internally for IECM = 1 (see below).
<0 Cutoff value for capillary pressure [Pa].

SGC = Critical (immobile) Gas Saturation [fraction].

IECM = Index for invoking the ECM. Available only for van Genuchten functions.

= 0 Do not invoke the ECM.
= 1 Invoke the ECM using unequally spaced saturation versus Pc/Kr tables generated internally as in CTOUGH.
= 2 ECM with equally spaced saturation tables generated internally.
= 3 User-specified PCKR tables for ECM.

SWIRF = Irreducible liquid saturation for the fracture [fraction].

ALPHAF = Van Genuchten parameter α for the fracture [1/Pa].

PHIM = Matrix porosity [fraction].

PHIF = Fracture porosity [fraction].

PERMM = Matrix permeability [m^2].

PERMF = Fracture permeability [m^2].

If IECM = 3,

Read: (swt(i), fkrwt(i), Fkrgt(i), Pcwt(i), i = 1,nn)

SWT = liquid saturation, fraction (0 <= SWT <= 1).

FKRWT = Liquid relative permeability corresponding to SWT [m^2].

FKRGT = Gas relative permeability corresponding to SWT [m^2].

PCWT = Liquid-gas capillary pressure [Pa].

This is a general case in that any functional forms of the relative permeability and the capillary pressure may be used to generate a table and entered with this option.

Read one entry per line with a total of nn entries followed by a line with zeros or blanks.

SWT must cover the range from SWIRM to 1, and must be read in ascending order.

Sufficient number of entries should be read to define a smooth variation in the relative permeability and the capillary pressure values.

If TYPE = LINear, read the following:

SWIRM = Irreducible liquid saturation (Krw = 0 below it).

RPMM = Not used (simply read zero, not blank)

ALPHAM = Not used (simply read zero, not blank)

SWEXT = Capillary Pressure at zero liquid saturation [Pa].

SGC = Critical (immobile) Gas Saturation [fraction].

IECM = 0

The relative permeabilities are linear functions ranging from 0 to 1 such that KRW = 0 at or below SWIRM liquid saturation, and KRG = 0 at or above (1-SGC) liquid saturation. The capillary pressure is linear between 0 to SWEXT corresponding to liquid saturations of 1 and 0, respectively.

If TYPE = TABular, read the following:

SWIRM = Irreducible liquid saturation (Krw = 0 below it) [fraction].

RPMM = Not used (simply read zero, not blank).

ALPHAM = Not used (simply read zero, not blank).

SWEXT = Same as for TYPE = 1.

SGC = Critical (immobile) Gas Saturation [fraction].

IECM = 0.

Read: (swt(i), fkrwt(i), Fkrgt(i), Pcwt(i), i = 1,nn)

SWT = liquid saturation, fraction (0 <= SWT <= 1).

FKRWT = Liquid relative permeability corresponding to SWT [m^2].

FKRGT = Gas relative permeability corresponding to SWT [m^2].

PCWT = Liquid-gas capillary pressure [Pa].

This is a general case in that any functional forms of the relative permeability and the capillary pressure may be used to generate a table and entered with this option.

Read one entry per line with a total of nn entries followed by a line with zeros or blanks.

SWT must cover the range from SWIRM to 1, and must be read in ascending order.

Sufficient number of entries should be read to define a smooth variation in the relative permeability and the capillary pressure values.

If TYPE = COREy's, read the following:

SWIRM = Irreducible liquid saturation (Krw = 0 below it).

RPMM = Not used (simply read zero, not blank).

ALPHAM = Not used (simply read zero, not blank).

SWEXT = Not used (simply read zero, not blank).

SGC = Critical (immobile) gas saturation [fraction].

IECM = 0.

The relative permeability is computed using the following function.

```
sstar=(usw-swirm)/(1-swirm-sgc)
fkrw=sstar**4
fkrg=(1-sstar*sstar)*(1-sstar)**2
```

The capillary pressure is internally set to zero for this case.

Same type-curves with different parameters may be read-in as often as desired. The sequential number does not have to start in ascending order but should cover the range of the number of lines read. That is, if there are 10 curves designated, the number should range between 1 to 10 in any desired order with each set of parameters on a new line.

Read as many data as desired and terminate the sequence by reading a blank line, or a line with one or more zeros.

Note that the data on the second line should be read only if IECM > 0 and is applicable only to van Genuchten functions (TYPE = 1).

Example:

```
PCKR
: No.  type   swir   lambda    alpha     swext   sgc   iecm
  1    Van    .01    .75       1.31e-5   0.      0.    1
: swirf  lambdaf  alphaf   phim   phif   fkm    fkf
  .2     .36      8.4e-7   .3     1.e-3  1.e-18 1.e-12
  2    Van    .01    .75       1.31e-5   0.      0.    1
  .3     .44      6.1e-7   .35    1.e-3  1.e-16 1.e-12
  0                                                    ! data terminator
```

4.14 SPECIFICATION OF ROCK PROPERTIES [PHIK]

Initialization Data — Keyword: **PHIK** Required

This keyword assigns rock porosity, permeability, depth, and the associated type relative permeability capillary pressure characteristics.

Read: PHIK FN
Read: I1 I2 J1 J2 K1 K2 IST ITHRM VB POR PERMX PERMY PERMZ PORM PERMM

PHIK = Keyword for reading porosity/permeability data.

FN = File name from which porosity/permeability data are read (optional). The file name must have an extension of '.phk' in lower case letters, but the extension should not be included in FN specification. The file name 'FN' (excluding the extension) should not exceed 8 characters. The run will stop if the file name is misspelled or the file fn.phk does not exist. Default = blank space, in which case the data is read from the input file.

I1, I2 = First and the last index of a region in x (r) direction.

J1, J2 = First and the last index of a region in y (theta) direction.

K1, K2 = First and the last index of a region in z direction.

The properties of a region bounded by the above indices are defined to be constant.

IST = Characteristic curve number for relative permeability and capillary pressure. The curve numbers are identified by the sequential numbers read in PCKR data.

ITHRM = Thermal properties data set number read-in by the keyword THERmal. The sequential number of rock thermal properties must be assigned.

VB = Block volume [m^3]. If read as zero, it is computed internally based on the grid-block sizes (DX, DY, DZ).

POR = Porosity [fraction].

PERMX = x (r) directional absolute rock permeability [m^2].

PERMY = y (theta) directional absolute rock permeability [m^2].

PERMZ = z directional absolute rock permeability [m^2].

PORM = Matrix porosity. [Defaults to PHIM(IST).]

PERMM = Matrix permeability [m^2]. [Defaults to FKM(IST).]

In case the ECM-formulation is invoked, if any of the porosity or permeability values are read as zero, they are set to the values assigned by the PCKR keyword. For a non-ECM system, the values are used as specified here. Also, for the ECM formulation, POR, PERMX, PERMY, and PERMZ refer to the values for fracture continuum where PERMY and PERMZ are internally set equal to PERMX, as the ECM option applies only to isotropic systems.

Read as many lines as required to define the data for the entire system, and terminate the sequence by reading a line with a zero. Termination with a negative integer or with value -1 or -2 activates other options which are discussed below. However, these options apply to only non-ECM systems.

If regions overlap, the last region read overrides the former ones.

If the data are not specified for the entire system (for all grid blocks), it may cause unpredictable results.

Termination with I1 = -1

This option permits a more convenient way to assign the values for porosity, permeability, gas saturation, and the relative permeability functions for generally heterogenous rock properties. One or more of these properties must be created and saved on a separate file which must be named as filename FN. The data in file FN should be in the following format:

Keyword fac fky fkz

data in free format corresponding to this keyword.
. . . . . .
. . . . . .
. . . . . .
Return

Keyword	Description
PERMeability	For reading x-direction permeability
POROsity	For reading porosity, fraction
SGINIt	For reading the initial gas saturation
SATF	For reading relative permeability/capillary function table number. They must be read as integers.

fac = Factor by which the read-in values to be multiplied. This is especially helpful to convert data in the proper units. This factor is ignored for keyword 'SATF'.

fky,fkz = Factors by which x-direction permeability to be multiplied to obtain y and z direction permeabilities. These are ignored for other keywords.

The values corresponding to the keywords may be read in any order. The last line on the file must have a 'Return' statement.

The data are read in the order of increasing i, j, and k indices, respectively. For example, the porosity array is read as:

```
do k = 1,nz
  do j = 1,ny
    read (lfn,*) (por(i,j,k), i = 1,nx)
  end do
end do
```

It should be ensured that values for each elements are assigned. Permeability in y and z directions are automatically computed as a factor of x-direction values. If fky or fkz are read as zero, the corresponding directional permeability is computed as zero. Comments are not permitted within the grid.inp file.

Example:

```
Permeability  1. .5 .1
  1.e-12 1.e-11 2.e-13 4.e-13 .....    (i= 1, nx, j = 1, k = 1)  !
  2.e-13 3.e-12 1.e-11 1.e-15 .....    (i= 1,nx, j = 2, k = 1)
  . . .  .  .  .
  3.1e-11 4.6e-12  2.9e-12 3.e12 ...   (i= 1,nx, j = ny,k = 1)
  1.1e-12  2.3e-14 3.1e-15 3.e-12...   (i= 1,nx, j = 1, k = 2)
  2.e-11  3.e-11  5.e-11  6.e-13 ...   (i= 1,nx, j = 2, k = 2)
  . . .  .  .  .
  7.e-11  8.e-12 9.e-13  5.e-11 ...    (i= 1,nx, j = ny,k = 2)
  . . . . . . . . . . . . . . .
  . . . . . . . . . . . . . . .
  . . . . . . . . . . . . . . .
  5.e-13 5.e-12 5.e-12 3.e-14 .....    (i= 1,nx, j = ny,k =nz)
```

The lines in parentheses should not be included in the 'fn.phk' file. They are shown here for explanatory purposes only.

Porosity 1.
porosity values in the same pattern as the above

Sginit 1.
gas saturation values in the same pattern as the above

RETURN

Termination with I1 = -2

In this case, the permeability values are internally generated using a random number generating algorithm. The indices I2 and J1 have additional meaning as outlined below. Currently, this option is available for 2D problems only.

I2:	n	size of equal permeability cluster
J1:	0	compute permeability field only
	1	compute correlated porosity and permeability

The average permeability of the medium is taken from PERMX.

4.15 ROCK THERMAL PROPERTIES AND BINARY DIFFUSION DATA [THERmal]

Initialization Data — Keyword: **THERmal** Required

The keyword THERmal assigns the thermal properties of the rock and binary diffusion parameters.

Read: THERmal
Read: NO RHO CPR CKDRY CKSAT CRP CRT TAU CDIFF CEXP ENBD

THER = Keyword for reading the thermal properties.

NO = Sequential number of the data set.

RHO = Rock density [kg/m^3].

CPR = Rock specific heat [J/kg-K].

CKDRY = Thermal conductivity of dry rock [J/s/m-K]. (Default = CKSAT)

CKSAT = Thermal conductivity of liquid saturated rock [J/s/m-K]. (Default = CKDRY).

CRP = Pore compressibility, $\frac{1}{\phi}\partial\phi/\partial p$, with pressure at constant T [1/Pa].

CRT = Absolute value of pore expansivity, $\frac{1}{\phi}|\partial\phi/\partial T|$, with temperature at constant pressure [1/T].

TAU = Tortuosity for binary diffusion (Default = 0).

CDIFF = Vapor-air diffusion coefficient [m^2/s]. (Default = 2.13e-5 m^2/s).

CEXP = Exponent for binary diffusion (Default = 1.8).

ENBD = Enhanced binary diffusion coefficient (Default = 0).
= 0 enhanced binary diffusion not invoked
> 0 invoke enhanced binary diffusion

The sequential number does not have to start in ascending order but should cover the range of the number of lines read. That is, if there are 10 rock-types designated, the number should range from 1 to 10 in any desired order with each set of parameters on a new line.

Read as many data as desired and terminate the reading sequence by reading a blank line, or a line with one or more zeros.

Example:

```
Thermal
: rho    cpr  ckdry cksat crp  crt  tau  cdiff  cexp  enbd
1 2580.  840.  1.7  2.1   0.   0.  .5   2.13e-5  1.8   0
```

5 RECURRENT DATA KEYWORDS

Recurrent data can be changed with time and include such items as the method of solution, source/sink tables, time step sizes, output frequency, boundary condition specifications, etc. A given type of data will remain effective until it is modified or replaced by a new set at subsequent time intervals. Thus, the method of solution may be kept fixed once defined initially, but output may be changed as desired during different time intervals. All data between the RECUrrent-data and the first TIME keyword or between two consecutive TIME keywords constitutes one set of RECUrrent data. The keywords associated with Recurrent data are:

AUTO-step	RSTArt
BCONdition	SOLVe
DEBUg	SOURce
ENDS	STEAdy
LIMIt	TIME
OUTPut	TOLR
RECUrrent	

They may be read in any order, EXCEPT the RECUrrent keyword which must be the first one. Once a TIME keyword is encountered, it is assumed that all the recurrent data have been assigned or changed as desired, and will proceed with the calculations until the target time assigned on the TIME keyword is reached. The last keyword in the data set must be 'ENDS' which signals the completion of a run. They are presented below in alphabetical order for convenience.

5.1 AUTOMATIC TIME STEP CONTROL DATA [AUTOstep]

Recurrent Data — Keyword: **AUTO-step**
Optional

This keyword will override the default parameters used for automatic time step size calculations. It is best to leave the default parameters unchanged unless they are unsatisfactory based on the performance of a run or user experience for a specific problem.

Read: AUTO-step DPMXE DSMXE DTMPMXE DP2MXE TACCEL IAUTODT FAC1

AUTO = Keyword for amending automatic time step size calculation parameters.

DPMXE = Estimated maximum pressure change in any grid block during a time step [Pa]. (Default = .10e5 Pa).

DSMXE = Estimated maximum saturation change in any grid block during a time step. (Default = .04).

DTMPMXE = Estimated maximum temperature change in any grid block during a time step [°C]. (Default = 5 °C).

DP2MXE = Estimated maximum air-phase pressure change in any grid block during a time step [Pa]. (Default = .10e5 Pa).

TACCEL = Acceleration parameter for automatic time-step size calculation for IAUTODT = 1. For IAUTODT = 0, TACCEL is not used. (Default value for TACCEL = 1.e-4).

IAUTODT = 0 Algorithm 1 is used for automatic time-step calculations.
= 1 Algorithm 2 is used.

FAC1 = Factor which the ratios DSMXE/DSMX, DPMXE/DPMX, etc. should not exceed. (Default = .5) See LIMIT keyword. (Recommended value .5 < FAC1 < .7). If any of these ratios exceed FAC1, then DSMXE, DPMXE, etc. are reset internally to satisfy this condition.

Note that the above estimates of changes in primary variables are used to calculate the next time step size and does not guarantee that the changes will not exceed these limits. An absolute upper limit on changes are specified by 'LIMIT' keyword. These values will remain in effect until they are modified by any of the subsequent Recurrent Data set entries.

Example:

```
:          dpmxe  dsmxe  dtmpmxe  dpamxe  taccel  iauto  fac1
Auto-Step  1.e4   .03     3        1.e4    0.
```

5.2 BOUNDARY CONDITION DATA [BCON]

Recurrent Data — Keyword: **BCON**
Conditional

This keyword assigns boundary conditions which can be changed at subsequent time intervals. The boundary conditions include both Dirichlet and Neumann types.

Read: BCON NBC FN
Read: ITYPE IFACE I1 I2 J1 J2
Read: TIMBC QBC PBC TBC SGBC XABC

BCON = Keyword for assigning boundary conditions.

NBC = Number of boundary conditions.

FN = File name from which the time varying boundary condition data are read (optional). The file name must have an extension of '.bc' in lower case letters, but the extension should not be included in FN specification. The file name 'FN' (excluding the extension) should not exceed 8 characters. The run will stop if the file name is misspelled or the file fn.bc does not exist. Default = blank space, in which case the data is read from the input file.

ITYPE = Type of boundary condition.
= 1 Dirichlet (constant field variables).
= 2 Neumann (constant flux).
= 3 velocity flux.

IFACE = Surface at which the boundary condition is imposed.

=	RIGHT,	right face, that is at I = NX.
=	LEFT,	left face, that is at I = 1 (x = r = 0).
=	TOP,	top surface, that is at K = 1.
=	BOTTOM,	bottom surface, that is at K = NZ.
=	FRONT,	front surface of the block at J = 1.
=	BACK,	Back surface of the block at J = NY.

I1,I2
J1,J2 = These indices define the region on the designated face where the conditions are imposed.

TIMBC = Time [seconds].

QBC = not used if ITYPE = 1.
= Flux rate [kg/m^2/s], if ITYPE = 2.
= velocity [meters/year], if ITYPE = 3.

Flux and velocity are positive if mass is entering the system and negative if mass is leaving the system.

PBC = Pressure at the designated surface [Pa]. If PBC ≤ 0, then the boundary represents a zero mass flux boundary and heat flux dry conduction only. In this case, SGBC and XABC are not used.

TBC = Temperature at the designated surface [°C].

SGBC = Gas phase saturation at the designated surface.

XABC = Mole fraction of air at the surface. This is required only for single phase systems when SGBC = 0. or 1.0

The variables PBC and SGBC are not used for Neumann boundary condition, that is for ITYPE = 2. Also, SGBC and XABC are not used for pure liquid isothermal systems defined by LIQUid keyword. For the Equivalent Continuum Model, they correspond to the average values of matrix and fracture.

Read one line defining each region followed by a set of time varying input data terminated with the digit 0. The number of entries and the time interval in each table (if more than one table) need not be the same. A combination of both Neumann and Dirichlet conditions may be imposed on a region by specifying the same region through indices I1,I2,J1, and J2.

Example:

```
Bcondition    2
: itype  iface  i1  i2  j1  j2
   1          Right     1    10   1    5
: timebc    qbc  pbc  tbc  sgbc      xabc
   0.       0.   1.1e5    30   .4   0
   1.e10    0.   1.2e5    30   .4   0
   0.
: itype  iface  i1  i2  j1  j2
   3    Left      1    10   1    5
: timebc    qbc  pbc  tbc  sgbc      xabc
   0.       1.e-4     1.0e5     30   0.   0.
   1.e10         1.e-4     1.0e5     30   0.   0.
   0.
```

5.3 DATA FOR DEBUG OUTPUT [DEBUg]

Recurrent Data — Keyword: **DEBUg**
Optional

This keyword prints out intermediate variables for designated region or grid blocks for debugging or examination purposes. This option should not be used for routine production runs.

```
Read: DEBUg    IBUG
Read: I1 I2 J1 J2 K1 K2
      . . . . . .
      . . . . . .
      0
```

DEBUg = Keyword for printing intermediate variables.

IDBUG = Index designating the level or extent of output.

= 0, turn-off the debug output, if previously invoked.
= 1, print minimum amount of debug output.
= 2, print intermediate level of debug output.
= 3, print full debug output.
= 4, special customized output.

I1, I2
J1, J2
K1, K2 = Indices bounding the region in x (r), y (theta) and z-directions for which the requested output will be printed.

Read one region per line and terminate the read-sequence by reading a blank line or a line with one or more zeros.

Since DEBUg can generate a large amount of output, it is desirable to limit this option for a small number of grid blocks and for a limited number of time steps.

Once activated, debug can be deactivated by reading another DEBUg keyword in a subsequent Recurrent data set with IBUG = 0, and no other data.

Example:

```
Debug      2
1  3  1  2  1  2
9  11 1  2  3  5
0
```

5.4 RUN COMPLETION KEYWORD [ENDS]

Recurrent Data — Keyword: **ENDS** Required

This keyword should be read ONLY if the run is to be terminated after the last target time. As such, this should be the last keyword in a data file.

READ ENDS

ENDS = Keyword terminating the run

Any data after this keyword will remain intact and not processed.

Example:

```
: run completion keyword
Ends
```

5.5 MAXIMUM TIME STEP LIMIT [LIMIt]

Recurrent Data | Keyword: **LIMIt** Optional

This keyword modifies the default values which are used in limiting the maximum change in the primary variables during a time step. If during any time step the maximum change in a primary variable exceeds the value defined by LIMIt data (or the default values), the time step is successively cut (up to a specified number) to ensure that they are within these specified limits. If a cut in the time-step results in a new time step which is > 0.90 of the original time step, the time step will not be cut in which case the solution vector will slightly exceed LIMIt data. This is to avoid resolving the step with only a small difference in the time step size.

Read: LIMIt DPMX DSMX DTMPMX DPAMX DTMN DTMX ICUTMX

LIMIt = Keyword for modifying the default values (or earlier specified) for maximum change of solution vector.

DPMX = Maximum change in pressure over a time step in any block [Pa]. (Default = .5e5 Pa.)

DSMX = Maximum change in saturation over a time step. (Default = .10.)

DTMPMX = Maximum change in temperature over a time step [T]. (Default = 10 °C.)

DPAMX = Maximum change in partial air pressure over a step [Pa]. (Default = .5e5 Pa.)

DTMN = Minimum time step size allowed with units as specified on TIME or STEAdy keyword (Default = 1 sec.)

DTMX = Maximum time step size allowed with units as specified on TIME or STEAdy keyword (Default = 3.15e9 sec.)

DTMN and DTMX are in the same units as the units for the time-step size (see TIME keyword).

ICUTMX = Maximum number of time-step cuts allowed during a time step (Default = 5).

DTMN and DTMX will control if the automatic time step size falls beyond these limits.

Note that DPMXE, DSMXE, DPAMXE, and DTMPMXE must at least equal to FAC1 times DPMX, DSMX, DPAMX, and DTMPMX, respectively. The former are specified by AUTO-step keyword (see AUTO-step).

5.6 OUTPUT DATA [OUTPut]

Recurrent Data — Keyword: **OUTPut**
Optional

This keyword enables output of desired variables at specified time intervals or at specified target times. The frequency of output may be changed with repeated use of this keyword in subsequent Recurrent Data sets.

Read: OUTPut A=l B=m C=k G=m L=n P=n Q=j S=k T=i V=k Y=n

OUTPut	=	Keyword for specifying output frequency.
A	=	All the important arrays in tabular form.
B	=	Boundary condition fluxes summary.
C	=	Output of convergence summary.
G	=	Output of gas saturation array.
L	=	Output of liquid saturation.
P	=	Output of pressure array.
Q	=	Source/Sink summary.
S	=	Frequency of one-line summary output on the screen.
T	=	Output of temperature array.
V	=	Output of all directional velocity arrays.
Y	=	Output of mole fraction of air in the gas phase.

Integers n, m, etc., following the equal sign designate the frequency of output. If the integer (say m) is negative, the output will be produced at every abs(m)th time step. If m is positive, it will be produced at every mth target time (see TIME keyword for the definition of target time). If output option A=l (where l is an integer as defined above) is specified such that 'l' is a multiple of n, m, etc., for any of the other variables the output corresponding to these variables will not be produced, as 'A' prints out all the variables in tabular form. The user should become familiar with these options to properly manipulate the output as desired.

Examples:

OUTPUT	A=1	Print all variables at every target time.
OUTPUT	C=-1 A=2	Print the convergence summary at every time step and rest of the variables at every second target time.
OUTPUT	V=1 C=1 T=2 G=2	Output velocities and the convergence summary at every target time, the temperature and gas saturations at every second target time.

5.7 PLOTS DATA [PLOTs]

Recurrent Data — Keyword: **PLOTs**
Optional

This keyword designates the frequency of writing plot/graphic files for spatial and time-history post-processing.

Read: PLOTs IPLOTS MXNDPLT (NDPLTS (I), I=1, MXNDPLT)

PLOTs = Keyword for reading plots/graphic files output.

IPLOTS = 0 No graphic/plot files requested.
= n<0 Frequency in number of time steps for writing graphic files.
= m>0 Frequency in number of target times for writing graphic files.

'm' and 'n' are integers with absolute value <99999.

MXNDPLT = Number of nodes at which time-history data is stored.

NDPLTS(I) = Node number for which the plot data are saved for post-processing.

Examples:

```
PLOTs  1     3  1 7 11   ! writes plot file at every target time, and time history at nodes 1,
                          7 and 11.

PLOTs  -10   3  1  7  11  ! writes plot file at every tenth time step, and time history at nodes
                          1, 7 and 11.

PLOTs 1  3  10  20  30
```

5.8 KEYWORD FOR BEGINNING OF RECURRENT DATA [RECUrrent]

Recurrent Data — Keyword: **RECUrrent** Required

This keyword has no associated data and signals the completion of all initialization data and the beginning of the recurrent data section. The simulator proceeds with initialization calculations upon encountering this keyword or the 'END' keyword. This keyword must be the last keyword in the initialization data set and the first keyword of the recurrent data set.

Read: RECUrrent

RECU = Keyword signaling the end of initialization data. The control will pass to Recurrent data after this keyword unless an 'END' is encountered after this keyword.

5.9 RESTART DATA [RSTArt]

Recurrent Data — Keyword: **RSTArt**
Optional

This keyword designates the frequency of writing restart files.

Read: RSTArt IRST1 IRST2

RSTArt = Keyword for reading restart/graphic files output.

IRST1 = 0 Frequency of writing cut-and-paste data.
= n<0 Frequency in number of time periods for writing cut-and-paste data.
= m>0 Frequency in number of time steps for writing cut-and-paste data.

IRST2 = 0 Frequency of writing restart file.
n<0 Frequency in number of time periods for writing restart data.
m>0 Frequency in number of time steps for writing restart data.

Cut-and-paste file can be extracted from lfn_out and cut and pasted into an initial run. The file is written in the required format for INIT data.

Restart file is a binary file which save all the variables required for resuming a run from the time at which such file is saved.

Examples:

```
RSTArt    0 1     ! writes restart data at every target time

RSTArt    -10     ! writes a cut-and-paste data at every tenth time step

RSTArt    0 3     ! writes restart data at every third target time
```

5.10 DATA FOR METHOD OF SOLUTION [SOLVe]

Recurrent Data — Keyword: **SOLVe** Required

This keyword specifies the linear equation solver method, and overrides the one designated by default or previously specified.

Read: SOLVe ISOLVE NEWTNMN NEWTNMX NORTH NITMAX LEVEL

SOLVe = Keyword for designating matrix solver and iterations.

ISOLVE = 1 Thomas Algorithm for 1D problem. For such problems, this method is internally set and will override any other specified method.

= 2 D4 method for 2D problems. This is a default solver for 2D problems (not available for single phase liquid or heat conduction only).

= 3 Watsolv (ILU factorization) with GMRES accelerator.

= 4 Watsolv (ILU factorization) with CGSTAB accelerator. (See Watsolv User's Guide for details.)

NEWTNMN = Minimum number of Newtonian iterations before a check on convergence is made. (Default = 2. For pure conduction or pure liquid problem, Default = 1.)

NEWTNMX = Maximum number of Newtonian iteration for convergence. Time step will be cut and equations resolved if the iterations exceed NEWTNMX. (Default = 7. For pure conduction and pure liquid system, Default = 1.)

NORTH = Maximum number of vectors for orthogonalization for Watsolv. Used only if ISOLVE = 3 (GMRES). (Default = 4).

NITMAX = Number of inner iterations for Watsolv. (Default = 100).

LEVEL = Degree of fill in incomplete LU factorization for Watsolv. (Default = 1).

Example:

```
:      isolve newtnmn newton  north nitmax level
SOLVE  3      2      0       3     75     2
```

In the above, Watsolv solver with GMRES with 3 orthogonalization vectors will be used. The level of fill is declared as 2 with 2 minimum Newtonian iterations and default value for maximum Newtonian iterations.

5.11 SOURCE SINK DATA [SOURce]

Recurrent Data — Keyword: **SOURce**
Conditional

This keyword assigns a set of tables for sources and sinks as a function of time.

Read: SOURce NS SCALH SCALM FN

Read: IS1 IS2 JS1 JS2 KS1 KS2 ISTYPE

Read: TIMEQ QHT QMT
. . .
. . .
0

SOURce = Keyword for assigning source/sink table.

NS = Total number of sources and sinks.

SCALH = Scale factor or multiplier to read-in values of QHT.

SCALM = Scale factor or multiplier to read-in values of QMT.

FN = File name from which the time varying source and sink data are read (optional). The file name must have an extension of '.src' in lower case letters, but the extension should not be included in FN specification. The file name 'FN' (excluding the extension) should not exceed 8 characters. The run will stop if the file name is misspelled or the file fn.src does not exist. Default = blank space, in which case the data is read from the input file.

IS1, IS2 = Indices bounding the source/sink region in I-direction.
JS1, JS2 = Same as above in the J, and K directions respectively.
KS1, KS2

ISTYPE = Index for type of source/sink. It is a two digit number, say 'ij' where

i = 1 mass of water.
i = 2 mass of air (not available yet).
i = 3 heat source with no mass.

j = 1 rate for each grid block in the region.
j = 2 rate per unit volume from which the total strength is computed based on block pore volume for the mass source, and the bulk block volume for the heat source.

j = 3 rate for the total specified region which is internally allocated to individual grid blocks based on pore volume for the mass and the bulk volume of blocks for the heat.

TIMEQ = Time in seconds.

QHT = Temperature [°C], if ISTYP = 1j or 2j (j=1,2 or3).
= Heat rate [J/sec], if ISTYP = 3j (j = 1,2,3).

QMT = mass rate [kgm/sec], if ISTYP = 1j or 2j (j = 1,2,3).
= 0, if ISTYP = 3j (j = 1,2,3).

The read-in values for fluid and energy withdrawal must be negative (sink), and injection is positive (source). If QMT is specified as a negative value corresponding to a sink, QHT is not used. If QMT is a positive value corresponding to a mass source, and QHT is read as a negative value, it will be interpreted as specific enthalpy rather than temperature.

Read TIMEQ, QHT and QMT on one line successively in ascending order of TIMEQ. Once the complete desired time range is entered, terminate the read sequence with a blank line or a line with one or more zeros.

As simulation proceeds, source and sink rates are linearly interpolated from this table corresponding to the mid-point of the time-step size. If a single entry is specified, a constant heat or mass source (sink) will be used. If simulation time exceeds the time in the above table, the last table entry will be used.

This completes data for one source.

Repeat the above sequence starting with reading IS1, IS2, etc. for the next source until all NS sources are specified. The number of entries or the time interval in each table (if more than one table) need not be the same.

Example:

```
Source  3  .5  0.      ! read-in heat source will be multiplied by .5
:i1 i2 j1 j2 k1 k2 istyp
  1  2  1  1 1  2  32  ! heat source per unit volume is specified in
:time qh/vol
0.  1.3                    the region.
1.e7  1.2
5.e8  .9
0
:i1 i2 j1 j2 k1 k2 istyp
  5 8   1 1  1 2   31     ! a uniform heat source is specified for each
:time qh/blk
0  10.                 ! grid block
3.e7  7.
3.e9  5
```

```
0
:i1 i2 j1 j2 k1 k2 istyp
  1  7  2  2  1  1  33     ! total heat source specified for the region
:time qh for the region
0   50.                   ! which is allocated to individual grid blocks
1.e7 40                   ! proportional to the bulk volume for the blocks.
1.e8 30
1.e9 20
1.e10 10
0
```

5.12 STEADY STATE DATA [STEAdy]

Recurrent Data — Keyword: **STEAdy**
Conditional

This keyword sets the run in a mode that it will proceed marching in time until a steady state is reached. The steady state is determined based on a maximum saturation change, maximum temperature change, and maximum pressure change per year as set by the user with this keyword. Upon attaining the steady state, the time and the time-step numbers are internally re-initialized to zero and new recurrent data-sets may be read. The run may be terminated with an 'ENDS' keyword following this keyword if only steady-state distribution is to be achieved.

Read: STEAdy DPSTDY DSSTDY DTMPSTDY

STEAdy[u] = Keyword for designating the run to march in time until a steady state is reached. The variable [u] in the bracket sets the units for time for monitoring the steps. See the keyword 'TIME' for details on [u].

DPSTDY = Maximum pressure change per year to define a steady state [Pa] (Default = 1.e−6).

DSSTDY = Maximum saturation change in any grid block per year. (Default = 1.e−5 1/y.)

DTMPSTDY = Maximum temperature change in any grid block per year [T/time unit]. (Default = 1.e−8 °C/y.)

Once the simulation proceeds up to a time when the absolute value of the maximum pressure change in any block divided by the time step in years is less than DPSTDY, the run will process the remaining data. (Default value for DPSTDY = 1.e−6 Pa/y.)

Examples:

STEAdy[y] 1.e-5

The run will march in time until the maximum change per year in any block is less than 1.e-6 Pa for pressure, less than $1.e^{-5}$ for saturation, and less than $1.e^{-8}$ °C for temperature.

5.13 TARGET TIME DATA [TIME]

Recurrent Data — Keyword: **TIME** Required

This keyword sets the target time up to which simulation will proceed for a given set of Recurrent data. Upon completion to the target time, one or more recurrent data sets may be modified, and the next target time is specified. The recurrent data initially specified will remain effective throughout the simulation unless modified. New Recurrent Data sets not defined earlier may also be introduced at any time during a run.

A set of data between two target times is referred to as one set of Recurrent Data.

Read: TIME TTARGET DTT DPMXE DSMXE DTMPMXE DP2MXE

TIME[u] = Keyword for time step specifications. The variable [u] in the bracket sets the units for the target time and the time step size, which may be set as under:

units of time	variable [u]
years	[y], [yr], [yrs], or [years]
months	[m], [mo], [mos], or [months]
days	[d], [day], or [days]
seconds	[s], [sec], or [secs]

The units of the target-time and the time-step size may be changed as often as desired by specifying different units for the variable [u] on subsequent TIME keywords. If an inadmissible value for [u] is specified, or no value is specified for it, the time defaults to 'seconds'. Note that TIME[u] is treated as a single keyword, and as such there should not be any embedded blanks.

Unlike other Recurrent Data, the units for the target time must be set with <u>each</u> target time. Not assigning the units will default to the units of seconds.

The time for plot files will use the units of time as specified here.

TTARGET = Target time up to which simulation will proceed with the data set specified before. If read as a negative value, simulation will proceed abs(TTARGET) steps beyond this and stop, ignoring all subsequent input data. The units of the target time is as declared by the variable [u] in the keyword.

IF METRA IS RUN IN COUPLED MODE WITH GEM, TTARGET IS OVERRIDDEN BY THE VARIABLE TPLOT() in GEM. (see Gem's User's manual for details.)

DTT > 0, fixed time step size of magnitude DTT.
= 0, compute internally.

The first time step by default is 10*DTMIN as specified in the LIMIt keyword, if read as zero. It may be better to read a larger value. The units of the time step-size are the same as declared by variable [u] with the keyword.

DPMXE = Estimated maximum pressure change in any grid block during a time step [Pa]. (Default = .10e5 Pa.)

DSMXE = Estimated maximum saturation change in any grid block during a time step. (Default = .04.)

DTMPMXE = Estimated maximum temperature change in any grid block during a time step [°C]. (Default = 5 °C.)

DP2MXE = Estimated maximum air-phase pressure change in any grid block during a time step [Pa]. (Default = 0.1e5 Pa.)

Note that the above estimates of changes in primary variables are used to calculate the next time step size, and do not guarantee that the changes will not exceed these limits. An absolute upper limit on changes is specified by 'LIMIT' keyword. These values will remain intact until they are modified during any of the subsequent data sets.

Also, if these values are read as zero (or field left blank), they are assigned from AUTO-step keyword data or from default values.

If the calculated time step is within 20 percent of the target time, the time-step size will be increased to hit the target time which may result in a greater change in the field variables than estimated, but will avoid taking an extra small step to reach the target time.

Examples:

Time	1000.	100.	(time in the units of second by default)
Time[sec]	1000.	100.	(time in the units of seconds)
Time[yrs]	1.	.02	(time in the units of years)

Time[s], Time[d], Time[m], Time[y] etc. are acceptable keywords for specifying times in seconds, days, months, and years, respectively.

5.14 DATA FOR CONVERGENCE TOLERANCES [TOLR]

Recurrent Data — Keyword: **TOLR**
Optional

This keyword overrides or modifies the tolerances for convergence. If one or more of the tolerances are read as zero, they are assigned internally set default values.

Read: TOLR TOLP TOLS TOLT TOLP2 TOLM TOLA TOLE RTWOTOL RMXTOL SMXTOL

TOLR	=	Keyword for assigning tolerances.
TOLP	=	Pressure tolerance [Pa]. (Default = 10 Pa.)
TOLS	=	Saturation tolerance. (Default = 0.0001.)
TOLT	=	Temperature tolerance [°C]. (Default = 0.001.)
TOLP2	=	Air partial pressure tolerance [Pa]. (Default = 10 Pa.)
TOLM	=	Tolerance on mass residual (L2 error norm). (Default = 1.e-5.)
TOLA	=	Tolerance on air residual (L2 error norm). (Default = .001.)
TOLE	=	Tolerance on energy residual (L2 error norm). (Default = .001.)
Residual (L2)	=	Max abs(Ax-b) for total mass, error and energy equations = [abs(Ax-b)/Mass or energy] for each block.
RTWOMAX	=	RTWOTOL*(initial residual norm). (Default = 1.e-7.)
RMXTOL	=	Absolute value of residual. (Default = 1.e-7.)
SMXTOL	=	Max of (dx, dx/x), where x = solution vector. (Default = 1.e-7.)

For pure conduction and liquid options, the last three tolerances for Watsolv default to 1.e-12 (refer to the Watsolv User's Guide for details).

6 SAMPLE INPUT DATA

6.1 STRUCTURE OF INPUT DECK

The structure of the input deck is as follows:

Title (2 lines)
RSTArt
GRID
DXYZ (or CXYZ)
THERmal
PCKR
PHIK
INIT
RECUrrent-data
OUTPut*
SOLVe
TOLR*
LIMIt*
AUTO*
BCON*
SOURce*
PLOTs*
TIME [ ]
STEAdy [ ]*
ENDS

All keywords are required with the exception of those marked with an asterisk (*).

6.2 ONE-DIMENSIONAL CARTESIAN GEOMETRY

This problem describes the thermal hydrologic effects of emplacement of HLW in a partially saturated host rock such as the proposed repository at Yucca Mountain, Nevada. The following data set describes a 1D vertical system of 560 meter height and 3.14 sq meter area with a time-varying decaying heat source with an initial heat load of 114 kw/acre placed at a depth of 225 meters. The matrix/fracture system is represented by the equivalent continuum model (ECM) (Lichtner and Walton, 1994) where the permeabilities for the fracture and the matrix are assigned constant values of 1.e-11 m^2 and 1.9e-18 m^2, respectively, and the corresponding porosity values are 1.8e-3 and 0.10. The system is initially in nonequilibrium with uniform gas pressure of 1 bar throughout, liquid saturation of 0.50, and the temperature of 25 °C. Dirichlet boundary conditions at the top and the bottom are imposed using large volume blocks at these boundaries. No flow at the lateral boundaries is assumed. The system is discretized into 238 elements of varying thickness ranging from 0.10 to 25 meters. The simulation run is set to proceed to 100,000 years.

Example:

```
Test Data for Multiflo Simulator (initial data : 1D, Yucca Mt.)
                    April 28, 1995

RSTART  0
:
:       XYZ                 =  1 table look-up,; pref = ref. press.
:       RADIAL              =  0 correlations;   tref = ref temp.
:       OTHER
:
:grid  geometry  nx ny   nz ivplwr ipvcal  iout   pref   tref  href
Grid   XYZ       1  1  238   0       1      0     0      0
:
Monitor
 135
debug 1
0
Pckr                        :relative perm and pc keyword
:       i  type-curv swirm   rpmm(lamda)  alpham    swext  sgc  iecm
        1  Van-Gen   0.001  .7635       1.315e-4  .05  0.  1
:                    swirf  rpmf(lamda)  alphaf  phim    phif   permm   permf
                     0.001  0.4438      5.8e-7  0.10    1.8e-3 1.9e-18 1.e-11
0                 :blank line
:
Debug  1
0
Thermal-prop
: no rho       cpr     ckdry  cksat   crp  crt   tau  cdiff    cexp  enbd
  1  2.580e+03 840.0   1.74   2.3     0    0    .0  2.13e-5  1.8
  2  2.580e+03 840.    1.74   2.3     0    0    .0  2.13e-5  1.8
0
:       igrid   rw      re
DXYZ 2
: (dx(i),i=1,nx)
3.14
: (dy(j),j=1,ny)
1.
: (dz(k),k=1,nz)    ten entries/line
:
  2.5e1 2.5e1 2.5e1 2.5e1 1.7500e1 1.0e1 1.0e1 1.0e1 1.0e1 1.0e1
  1.0e1 1.0e1 1.0e1 1.0e1 7.5e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0
  5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0
  5.0e0 5.0e0 5.0e0 5.0e0 3.5e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0
  2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0
  2.0e0 2.0e0 2.0e0 2.0e0 1.5e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0
  1.0e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0 1.0e0
  1.0e0 1.0e0 7.5e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1
```

```
  5.e-1 5.e-1 5.e-1 5.0e-1 5.0e-1 5.0e-1 3.7500e-1 2.5e-1 2.5e-1 2.5e-1
  2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1
  2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1
  2.5e-1 2.5e-1 2.5e-1 2.7499e-1 2.5e-1 2.e-1 2.0e-1 2.e-1 2.e-1 2.e-1
  2.e-1 2.0e-1 2.0e-1 1.5e-1 1.E-01 1.E-01 1.e-1 1.e-1 1.e-1 1.e-1
  1.e-1 1.e-1 1.e-1 1.e-1 1.e-1 1.e-1  1.e-1 1.e-1 1.e-1 1.e-1
  1.e-1  1.e-1 1.e-1 1.e-1 1.e-1 1.e-1  1.e-1 1.e-1 1.e-1 1.e-1
  1.e-1  1.e-1 1.e-1 1.e-1 1.e-1 1.4999e-1 2.e-1 2.0e-1 2.0e-1 2.0e-1
  2.e-1 2.001e-1 2.0e-1 2.e-1 2.e-1 2.e-1 2.251e-1 2.5e-1 2.5e-1 2.5e-1
  2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1
  2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 2.5e-1 3.75e-1
  5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1
  5.0e-1 5.0e-1 5.0e-1 5.0e-1 5.0e-1 7.5e-1 1.0e0 1.0e0 1.0e0 1.0e0
  1.0e0 1.0e0 1.5e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0 2.0e0
  2.0e0 2.0e0 3.5e0 5.0e0 5.0e0 5.0e0 5.0e0 5.0e0 7.5e0 1.0e1
  1.0e1 1.0e1 1.0e1 1.5e1 2.0e1 2.0e1 2.0e1 2.0e1
:
PhiK
: i1 i2 j1 j2 k1 k2 iist ithrm vb  por permx permy  permz  pormm permm
   1  1  1  1  1  1  1   2   7.854e7
   1  1  1  1  2  237 1   1   0.
   1  1  1  1 238 238 1   2   6.2832e7
   0
Init
: i1  i2  j1  j2  k1  k2   p       t     sg    x2  sgm
   1   1   1   1   1  238  1.0e5    25.0  0.5  0.  .50
   0
Source  1
: is1  is2  js1  js2  ks1  ks2  istyp
    1    1    1    1   135  135  31
: timeq(sec)   T/qht (C/(J/s))    qmt (kg/s)
   .0000E+00   8.8496E+01
  6.3115E+07   8.2633E+01
  2.3668E+09  2.6482E+01
   ~  ~  ~  ~  ~  ~   ~  ~  ~  ~  ~
   ~  ~  ~  ~  ~  ~   ~  ~  ~  ~  ~
  3.1558E+12  8.5011E−02
  4.7336E+12  6.1731E−02
0
Recurrent-data
Output   A=1   C=1
:
:Debug 2
:1 1 1 1 134  136
:0
: isolv  newtnmn  newtnmx
Solve 1     2        10
:
```

```
:AUTO-step  DPMXE    DSMXE  DTMPMXE   DP2MXe
AUTO-step   1.0E+4   0.03    5.0       1.e4
:
:TOLR TOLP  TOLS  TOLT  TOLP2  TOLM  TOLA  TOLE
Tolr   10. 1.e-4  1.e-3  1.e+1 1.e-5 1.e-3 1.e-30 0. 0. 0.
:
:Limit dpmx    dsmx dtmpmx dp2mx dtmn dtmx icutmx
LIMIT  1.e5    .08    10.   1.e5  1.e-5 1000.

:Output Print all arrays and convergence summary at every target time
OUTPUT   A=1 C=1
PLOTS 1
:     target  dt    dpmx   dsmx   dp2mx   dtmpmx

Time[y]  0.01
Time[y]  10.
Time[y]  100.
Time[y]  1000.
Time[y]  10000.
Time[y]  20000.
Skip
Debug 3
1 1 1 1 211  213
0
Noskip
Time[y]  50000.
Time[y]  100000.
Ends
```

6.3 TWO-DIMENSIONAL RADIAL GEOMETRY

This problem describes the thermal-hydrologic effects of emplacement of HLW in a partially saturated host rock in two spatial dimensions. The waste is represented by a circular disk, and cylindrical symmetry is assumed. An initial heat load of 114 kw/acre is used in the calculation. The equivalent continuum model is not used in this example.

A permeability of 1.e-15 m^2 is assigned to each block. Temperature is fixed at the top and bottom of the computational domain by assigning large block volumes. No flow at the outer boundary is imposed on the system.

Example:

```
Test Data for Multiflo simulator (initial data : 2D, Yucca Mt.)
                    Sept. 7, 1995

RSTART  0
:
:     XYZ              = 1 table look-up,; pref = ref. press.
```

```
:     RADIAL            = 0 correlations;   tref = ref temp.
:     OTHER
:grid  geometry  nx  ny  nz ivplwr ipvcal  iout pref tref  href
Grid   RADIAL    31   1  28  0      1      1    0    0     0
:
Pckr                          :relative perm and pc keyword
:   i  type-curv swir   rpm(lamda)  alpha    swext  iecm
    1  Van-Gen   0.1    .4400      5.8E-7   0.
0            :blank line
:
Thermal-prop
: no rhoi   cpr  ckdry  cksat   crp  crt   tau  cdiff   cexp  enbd
  1  2300.  840. 1.74    2.3     0    0    .0  2.13e-5  1.8    1.
  2  2300.  840. 1.74    2.3     0    0    .0  2.13e-5  1.8    1.
0
:    igrid   rw      re
DXYZ  2      0.      2.e3
: (dx(i),i=1,nx)
2.5E1  7.5E1  1.25E2  1.75E2  2.25E2  2.75E2  3.25E2  3.75E2  4.25E2  4.750E2
5.25E2  5.75E2  6.25E2  6.75E2  7.25E2  7.75E2  8.25E2  8.75E2  9.25E2  9.75E2
1.025E3 1.075E3 1.125E3 1.175E3 1.225E3 1.275E3 1.325E3 1.375E3 1.425E3 1.475E3
1.525E3 1.575E3  1.625E3  1.75E3  1.95E3
:
: (dy(j),j=1,ny)
360.
:
: (dz(k),k=1,nz)
2.5E1  2.5E1  2.5E1  2.5E1  2.5E1  1.25E1  1.25E1  1.25E1  1.25E1  2.5E0
2.5E0  2.5E0  2.5E0  1.8750E  1.250E  1.8750E  2.5E0  2.5E0  2.5E0  2.5E0
1.25E1  1.25E1  1.25E1  1.25E1  1.25E1  1.25E1  1.25E1  1.25E1
:
PhiK
: i1 i2  j1  j2  k1  k2  iist ithrm vb    porm  permx   permy    permz
  1  31  1   1   1   1    1   2    1.e10 .10  1.e-15   1.e-15  1.e-15
  1  31  1   1   2   27   1   1    0.0   .10  1.e-15   1.e-15  1.e-15
  1  31  1   1   28  28   1   2    1.e10 .10  1.e-15   1.e-15  1.e-15
  0
Init
: i1  i2  j1  j2  k1  k2   p       t     sg    x2   sgmtx
   1  31   1   1   1  28  1.0e5    25.0  0.5   0.
   0
Recurrent-data
:         ns   scalh  scalm
Source    2
: is1   is2   js1   js2   ks1   ks2   istyp
   1      1     1     1    15    15     31
: timeq(sec)   T/qht (C/(J/s))    qmt (kg/s)
      0.       2.2124E+05
```

```
  3.1558E+08  1.7047E+05
  6.3115E+08  1.4210E+05
  9.4673E+08  1.2062E+05
  ~~~~~~~~~~~~~~~~~~~~~~~~~~
  2.2090E+12   1066.
  2.5246E+12   879.6
  2.8402E+12   742.4
  3.1558E+12   638.0
  4.7336E+12   463.1
  0
: is1   is2   js1   js2   ks1   ks2   istyp
   2     4     1     1    15    15     33
: timeq(sec)   T/qht (C/(J/s))    qmt (kg/s)
      0.      1.1062E+06
  3.1558E+08  8.5233E+05
  6.3115E+08  7.1052E+05
  9.4673E+08  6.0311E+05
  1.2623E+09  5.1807E+05
  ~~~~~~~~~~~~~~~~~~~~~~~~~~
  1.8935E+12   2218.
  2.2090E+12   1777.
  2.5246E+12   1466.
  2.8402E+12   1237.
  3.1558E+12   1063.
  4.7336E+12   771.8
  0
: isolv  newtnmn  newtnmx  north   inritr  level
Solve 2    2        7        4       0      2
:Debug 2
:0
:
Output   A=1 C=1
Plots  1
:        target   dt    dpmx    dsmx    dp2mx    dtmpmx
:Time[yr]   1.
Time[yr]   10.
Time[yr]   50.
Time[yr]   100.
Time[yr]   1000.
Time[yr]   5000.
Time[yr]   10000.
Time[yr]  1.e5
Ends
```

7 EXPLANATION OF OUTPUT RESULTS

Although most of the output is self-explanatory, some of the important output variables require further explanation or clarification. The items are discussed in the order in which they appear in a typical output and grouped as Initialization and Recurrent Output sections.

7.1 INITIALIZATION OUTPUT

7.1.1 Mass and Energy in Place

These values are computed by simply summing up the mass and energy content of each grid block:

$Mass_i$ = PV (Dw Sw+Dg Sg) X_i MW_i for *i*th component
Energy = PV (Dw Sw Uw+Dg Sg Ug)+Vb (1-Por) Dr Cpr T

where

PV	=	Pore volume of a grid block [m^3]
Dw	=	Liquid molar density [moles/m^3]
Dg	=	Gas molar density [moles/m^3]
Sw	=	Liquid saturation [fraction]
Sg	=	Gas Saturation [fraction]
X_i	=	Mole fraction of the *i*th component
MW_i	=	Molecular weight of the *i*th component
VB	=	Bulk block volume [m^3]
Por	=	Porosity [fraction]
Dr	=	Rock density [Kg/m^3]
Cpr	=	Rock specific heat [J/m^3-C]
T	=	Temperature [°C]
Uw	=	Molar internal energy of liquid [J/mole]
Ug	=	Molar internal energy of gas [J/mole]

Bulk volume of a block for cartesian and the cylindrical geometries are computed as:

VB = DX * DY * DZ for cartesian geometry

$= pi\ (r_{i+1/2} + r_{i-1/2})\ (r_{i+1/2} - r_{i-1/2})\ DZ$

where

DX,DY,DZ	=	Block dimensions in x, y and z direction, respectively.
pi	=	3.14159
$r_{i+1/2}$	=	Outer radius of *i*th block.
$r_{i-1/2}$	=	Inner radius of *i*th block.

The pore volume of a block (PV) is computed as VB × Por.

7.1.2 Initial Conditions

Initial condition variables include pressure (P), temperature (T), gas saturation (Sg), and mole fraction of air component (Xg2). In a two-phase region, Xg2 is a function of P and T which is computed using the ideal gas law as:

$$Xg2 = (P - Pv)/P$$

where

Pv = Saturation pressure of water

Pv is a function of T only if the capillary pressure effect is neglected. In the presence of capillary forces, Pv is a function of capillary pressure, liquid water density, temperature, and saturation pressure.

7.1.3 Transmissibility

The constant component of fluid transmissibility (Tx, Ty, and Tz) is an interblock property and calculated in each direction as follows. For Cartesian Geometry:

$$Tx_i = \frac{2\ K_i\ K_{i-1}\ DY\ DZ}{DX_i\ K_{i-1} + DX_{i-1}\ K_i}\ , \tag{7-1}$$

$$Ty_j = \frac{2\ K_j\ K_{j-1}\ DX\ DZ}{DY_j\ K_{j-1} + DY_{j-1}\ K_j}\ , \tag{7-2}$$

$$Tz_k = \frac{2\ K_k\ K_{k-1}\ DX\ DY}{DZ_k\ K_{k-1} + DZ_{k-1}\ K_k}\ . \tag{7-3}$$

For Cylindrical Geometry:

$$Tx_i = \frac{2\ \pi\ K_i\ K_{i-1}\ DZ}{K_{i-1}\ \ln\frac{r_i}{r_{i-1/2}} + K_i\ \ln\frac{r_{i-1/2}}{r_{i-1}}}\ , \tag{7-4}$$

$$Tz_k = \frac{\pi\ K_k\ K_{k-1}\left[(r_{i+1/2})^2 - (r_{i-1/2})^2\right]}{DZ_k K_{k-1} + DZ_{k-1}\ K_k}\ . \tag{7-5}$$

where K_i and K_{i-1} are the permeability values in I-direction (x or r) for *i*th and (i−1)*st* blocks. Similarly, K_j and K_{j-1}, and K_k and K_{k-1} are values in y or angular and vertical directions, respectively. The block center radius for a cylindrical system is r_i where $r_{i-1/2}$ and $r_{i+1/2}$ are the inner and outer radii of the block boundaries.

For a cylindrical system, the above transmissibilities are multiplied by a factor (angle/2 pi) if the system is not a full 360°.

Since the transmissibilities are at block interfaces, they correspond to the space location at $(i_{1-1/2},j,k)$ between blocks i and i−1, and similarly in j and k directions. Since fractional indices are not permitted in the code, they are designated at (i,j,k). In other words, the transmissibility between *i*th and (i−1)*st* block is designated as the *i*th transmissibility. Thus, all the boundary transmissibilities at Tx(1,j,k), Tx(nx+1,j,k), Ty(i,1,k), Ty(i,ny+1,k), Tz(i,j,1), and Tz(i,j,nz+1) are zero. In the actual code, the far boundary transmissibility values are not even stored as they will remain zero and the indices are compatible with the block-centered variables.

7.1.4 Binary-Diffusion Transmissibility

In the presence of vapor-air binary diffusion, the constant component of these transmissibilities (Txdfsn, Tydfsn, and Tzdfsn) are calculated from the following equation.

$$T_{x_d} = \frac{2\omega\sqrt{\phi_i\tau_i\phi_{i-1}\tau_{i-1}}\; D^0_{va}\; \Delta Y\; \Delta Z}{R\; T_o^{\theta}\; (\Delta X_i + \Delta X_{i-1})} \qquad (7\text{-}6)$$

and similarly in y- and z-directions, where

ω	=	Enhancement factor
τ	=	Tortuosity factor
ϕ	=	Porosity
D^0_{va}	=	Binary diffusion coefficient [m^2/s] (read-in as cdiff with THERM keyword)
θ	=	Exponent in the relation (read-in as cexp)
T_o	=	273.15 (absolute temperature at 0 °C)

7.2 RECURRENT OUTPUT

7.2.1 Mass and Energy in Place

These values represent mass and energy in place at any time (t) and are computed in a manner identical to the initial values discussed above.

7.2.2 Inner Iteration Counter

The numbers INRITRS represent the inner iterations during each Newtonian iteration and their sum over a time step. The inner iterations correspond to the number of iterations required for

convergence by an iterative solver. For direct solvers such as band-solve or D4, Newtonian iterations equal the inner iterations and, therefore, they are they not printed when the direct method is specified as a solver. Note that 1D systems are solved by the direct method as default. The number of Newtonian iterations required for convergence corresponds to the number of times a linearized equation solver is called excluding the call to solver when a time-step cut was performed.

7.2.3 Err-H_2O, Err-Air, Err-Enrg

These numbers represent the actual error in the solution over a time step. Thus, the numbers identify the actual mass (kg) or energy (joules) of H_2O, air, and energy which is unaccounted for. Err-H_2O is computed as follows:

$$\text{Err}-H_2O = H_2O(t+\text{delt}) - H_2O(t) - H_2O \text{ Mass injected.}$$

Err−Air and Err−Enrg for air and the energy are calculated analogously.

The solution errors can be reduced by tighter tolerances (TOLR-Keyword) at the expense of increased computing time. However, due to machine round-off errors (number of significant digits for computations), these errors cannot be reduced beyond certain finite values. In practical terms, an excessively small solution error is of little significance as the results will remain essentially unchanged.

7.2.4 ICONV

This variable assumes a value of 1 or 2 and indicates whether the convergence is achieved based on maximum changes in the solution variables (ICONV=1) or attaining the residual errors (ICONV=2).

Thus, for ICONV=1,

$$\begin{aligned} |P(k+1)-P(k)| &< \text{tolp} \\ |Pa(k+1)-Pa(k)| &< \text{tola} \\ |Sg(k+1)-Sg(k)| &< \text{tols} \\ |T(k+1)-T(k)| &< \text{tolt} \end{aligned} \qquad (7\text{-}7)$$

where 'k' is the Newtonian iteration number. The parameters tolp, tolpa, tols, and tolt are specified as the first four variables with the TOLR keyword. The next three variables on this keyword control the convergence for ICONV=2.

For ICONV=2,

$$\frac{|M(k+1)-M(k)|}{M(k)} < \text{tolm}$$

$$\frac{|Ma(k+1)-M_a(k)|}{M_a(k)} < \text{tola} \qquad (7\text{-}8)$$

$$\frac{|E(k+1)-E(k)|}{E(k)} < \text{tole}$$

If convergence is desired based on only ICONV=1 or 2, the parameters corresponding to the undesired criteria may be set to very small numbers (e.g., 1.e-30).

7.2.5 DPMAX, DSMAX, and DTMPMX

These variables denote the maximum change in pressure, gas saturation, and temperature over a time step. The block which undergoes such a maximum change is identified in parenthesis with reference to its (i,j,k) location.

7.2.6 Balances

Similar to Err−H_2O, etc., the balance is an another indicator of the accuracy of the solution. Two different types of balances are computed, namely the cumulative balance and the incremental balance. The cumulative balance becomes relatively less sensitive after a sufficient amount of fluid/energy is injected/produced (including influx and afflux) but reflects the overall error in the system from the initial time. Thus, a significant error during a few time steps may not reflect very adversely on the cumulative balance. The incremental balance takes cognizance of the error only over a time step. They are computed as follows:

$$\text{Cumulative Balance} = \frac{\text{Mass}(t) - \text{Mass}(t=0)}{\text{Net cumulative injection}} \qquad (7\text{-}9)$$

$$\text{Incremental Balance} = \frac{\text{Mass}(t+\Delta t) - \text{Mass}(t)}{\text{Net injection during } \Delta t} \qquad (7\text{-}10)$$

Balances for H_2O, Air, and Energy are computed as above and printed out as H_2OBAL, AIRBAL, ENRGBAL for H_2O, air and energy, respectively. H_2OINC, AIRINC, and ENRGINC represent the corresponding incremental balances.

For a perfect solution, both the incremental and cumulative balances should be unity. Note that if there is zero net injection/influx, the balance calculations result in an indeterminate quantity. In this case, monitoring the errors, Err−H_2O, etc., provides the needed control to assess the accuracy of the solution.

Balances are saved in a separate file, lfn_bal, which should be reviewed at the end of each run to ensure satisfactory results.

7.2.7 Boundary Condition Influx Summary

The variables designated as cummbc, cumcbc, and cumhbc represent the cumulative mass, air mass, and total energy influx from time = 0 into the system. Negative values represent flux out of the system. The rate of influx for the corresponding quantities is given by qm, qa, and qh.

7.2.8 Source/Sink Summary

The printout is analogous to the boundary condition influx with similar meaning of the variables.

7.2.9 Velocities

The velocities represent superficial Darcy velocities. They are computed at block interfaces in lfn_out file and at the block center in the plot files.

7.2.10 Plot Files

A number of plot files are written out in a spreadsheet-like format for ease in plotting results. Both spatial profiles printed at target times and time history plots at specified node points are available. Time history plots consist of five files with the variables shown in Table 7-1.

Table 7-1. List of time history plot files and their associated variables

File Name	Variables
jobname_his.xyp	time step
jobname_tmp.xyp	temperature
jobname_sat.xyp	saturation
jobname_press.xyp	pressure
jobname_rh.xyp	relative humdity

There are two spatial plot files—one for field variables and one for flow velocities and fluxes. The units for velocity are meters per year. Fluxes are in units of $kg/m^2/s$. The structure of the spatial plot files is shown in Table 7-2.

Table 7-2. List of spatial plot files printed at target times and their associated variables

File Name	Variables
jobname_fld.xyp	[xyz], press, temp, sl, sg, xairl, xairg, pcap
jobname_flx.xyp	[xyz], vgx, vlx, vgy, vly, vgz, vlz, fgx, flx, fgy, fly, fgz, flz

8 DISK FILE REQUIREMENTS

The following disk files are automatically opened upon initiation of a run. The name of the files are assigned based on the name of the input data file as indicated below.

UNITS	FILE NAME	DISPOSITION/REMARKS
51	fn.dat	Input data file
52	fn_out	Output file
53	fn_errs	Run Errors and diagnostics file
53	fn_bal	Energy and Mass balance file
55	fn_vel	Velocity file
56	fn_rsi	Restart file to resume a run
57	fn_rso	Restart file
58	fn_bug	Debug output file
59	fn_prn	Monitoring file
60	fn.bc	Boundary condition data
60	fn.int	Initial condition data
60	fn.pck	Material properties data
60	fn.phk	Porosity/permeability data
60	fn.src	Source data

The files are closed upon successful completion of a run. Note that the data file must have extension xx.dat. The output files have a suffix separated by '_' (underscore).

For restarting a run from a saved file 'fn_rso', change its name to 'fn_rsi'.

9 RUN EXECUTION

Once the simulator is compiled and installed on a computing system, the execution is carried out by simply typing:

```
multi  test
```

where multi is the name of the executable module, and test.dat is the name of the data file. The data file name should not be longer than 7 characters and MUST have 'dat' as the extension. The above command will execute the job interactively.

If the job is to be processed in background mode, type:

```
multi test &
```

or

```
nohup multi test &
```

Note that the output files will have names with the prefix the same as the data file prefix (test in the example). This enables simultaneous execution of different data sets having different names in the same subdirectory.

Upon completion of a run or upon abnormal termination, carefully examine the fn_bal (balance file) and the fn_errs (error file) files to ensure that there are no errors and the mass and the energy balances are reasonably conserved. For a near perfect solution, the cumulative balances should not deviate from unity more than 0.1 percent. The incremental balances may vary more sporadically.

10 DIMENSIONING, MACHINE SPECIFIC ROUTINES, AND COMPILATION

10.1 REDIMENSIONING

For different-sized problems, it may be desirable or even necessary to alter the dimensions of certain arrays. There are two places in the code where the array sizes can be altered.

(i) Reset the dimensioning parameters in file 'paramtrs.h' as specified.

(ii) If during execution the run terminates due to insufficient size of array 'aa' in the main program, reset the array size to a larger value. Also, the data statement 'maxaa' must be reset to the same value as the dimensions of the 'aa' array. Prior to the termination of a run, a message will be printed indicating the minimum required size of array 'aa'.

10.2 MACHINE SPECIFIC ROUTINES

Since different computing systems use different system timing routines, call to available timing routine for a particular system should be invoked. This should require a change only in the seconds.f subroutine within the cputim.f group of routines. At present, it is set for Sun work stations. Calls to timing routines for the Lahey compiler for PC and LPI compiler for PC/UNIX are deactivated. The timing routine should compute the time in cpu-seconds.

10.3 COMPILATION

A makefile is included with the source code. The code is written in double precision, so no special compiler options for double precision should be specified. For Cray and other 64 bit machines, the statement 'implicit real*8 (a-h,o-z)' in impl.h file should be commented out. The executable file 'metra' is produced.

11 REFERENCES

Brooks, R.H., and A.T. Corey. 1966. Properties of porous media affecting fluid flow. *Journal of Irrigation and Drainage Division* 93(3): 61–68.

Coats, K.H., and H.S. Price. 1974. Direct methods in reservoir simulation. *SPEJ* 257: 295–306.

Edlesfen, N.E., and B.C. Anderson. 1943. Thermodynamics of soil moisture. *Hilgardia* 15(2): 31–298.

Hirschfelder, J.O., C.F. Curtiss, and R.B. Bird. 1954. *Molecular Theory of Gases and Liquids*. New York, NY: John Wiley & Sons.

International Formulation Committee. 1967. *A Formulation of the Thermodynamic Properties of Ordinary Water Substance*. Dusseldorf, Germany: IFC Secretariat.

Lichtner, P.C., and J.C. Walton. 1994. *Near-field Liquid-Vapor Transport in a Partially Saturated High-Level Nuclear Waste Repository*. CNWRA 94-022. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Somerton, W.H., A.H. El-Shaarani, and S.M. Mobarak. 1974. High temperature behavior of rocks associated with geothermal-type reservoirs. Paper SPE-4897. *Proceedings of the 44th Annual California Regional Meeting of the Society of Petroleum Engineers*. Richardson, TX: Society of Petroleum Engineers.

van der Kwaak, J.E., P.A. Forsyth, and E.A. Sudicky. 1995. *WATSOLV, Sparse Matrix Iterative Solver Package, User's Guide*. Waterloo, Ontario, Canada: Waterloo Centre for Groundwater Research, University of Waterloo.

van Genuchten, M.T. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of American Journal* 44: 892–898.

Vargaftik, N.B. 1975. *Tables on the Thermophysical Properties of Liquid and Gases*. 2nd Ed. New York, NY: John Wiley & Sons.

Walton, J.C., and P.C. Lichtner. 1995. Quasi-steady state model for coupled liquid, vapor, and heat transport: Application to the proposed Yucca Mountain high-level Waste Repository. Submitted to *Water Resources Research*.

APPENDIX

APPENDIX

This appendix provides a brief mathematical description of the underlying equations solved by METRA.

A.1 SATURATION FUNCTIONS

Capillary saturation functions provided in METRA are van Genuchten, linear, and Brook-Corey functions. In addition, tabular data may be provided by the user.

A.1.1 VAN GENUCHTEN SATURATION FUNCTIONS

Capillary pressure is related to saturation by various phenomenological relations, one of which is the van Genuchten (1980) relation given by

$$s_l^{eff}(p_c) = \left[1 + \left(\alpha|p_c|\right)^m\right]^{-\lambda}, \tag{A-1}$$

where p_c represents the capillary pressure [Pa], and s_l^{eff} is defined by

$$s_l^{eff} = \frac{s_l - s_l^r}{s_l^0 - s_l^r}, \tag{A-2}$$

where s_l^r denotes the residual saturation, and s_l^0 denotes the maximum saturation. The quantity m is related to λ by the expression

$$\lambda = 1 - \frac{1}{m}, \; m = \frac{1}{1-\lambda}. \tag{A-3}$$

Relative permeability for the liquid phase is given by the expression

$$k_{rl} = \sqrt{s_l^{eff}}\left[1 - \left(1 - \left(s_l^{eff}\right)^{1/\lambda}\right)^{\lambda}\right]^2, \tag{A-4}$$

and for the gas phase by

$$k_{rg} = 1 - k_{rl}. \tag{A-5}$$

A.1.2 Brook-Corey Functions

The Brook-Corey functions for relative permeability are defined by

$$k_{rl} = s_*^4, \; k_{rg} = \left(1 - s_*^2\right)\left(1 - s_*\right)^2, \; s_* = \frac{s_l - s_l^r}{1 - s_l^r - s_g^r}. \tag{A-6}$$

Capillary pressure is not incorporated in the Brook-Corey functions.

A.1.3 Linear

The linear relation is defined by

$$k_{rl} = s_*, \; k_{rg} = 1 - k_{rl}$$
$$s_* = \frac{s_l - s_l^r}{1 - s_l^r} . \tag{A-7}$$

A.2 KELVIN'S EQUATIONS FOR VAPOR PRESSURE LOWERING

Vapor pressure lowering resulting from capillary suction is described by Kelvin's equation given by

$$p_v = p_{sat}(T) e^{-p_c/n_l RT}, \tag{A-8}$$

where p_v represents the vapor pressure, p_{sat} the saturation pressure of pure water, T denotes the absolute temperature and R denotes the gas constant. Note that the density of the liquid phase, η_l, is represented on a molar basis.

A.3 THERMAL CONDUCTIVITY

Thermal conductivity is determined from the equation (Somerton et al., 1974)

$$\mathrm{CK} = \mathrm{CKDRY} + \sqrt{s_l}\,(\mathrm{CKSAT}-\mathrm{CKDRY}), \tag{A-9}$$

where CKDRY and CKSAT are dry and fully saturated rock thermal conductivities.

A.4 VAPOR-AIR DIFFUSION EQUATION

The conservation equation for the H_20 component (w) is given by

$$\frac{\partial}{\partial t}\left[\phi\left(s_l n_l X_w^l + s_g n_g X_w^g\right)\right]$$
$$+ \nabla \cdot \left(q_l n_l X_w^l + q_g n_g X_w^g - D_g^{eff} n_g \nabla X_w^g\right) = Q_w, \tag{A-10}$$

and for the air component (a) by

$$\frac{\partial}{\partial t}\left[\phi\left(s_l n_l X_a^l + s_g n_g X_a^g\right)\right] + \nabla\cdot\left(q_l n_l X_a^l + q_g n_g X_a^g - D_g^{eff} n_g \nabla X_a^g\right) = Q_a, \tag{A-11}$$

with source terms Q_w and Q_a. In these equations, the Darcy fluxes for liquid, q_l, and gas, q_g, are defined by

$$q_l = -\frac{kk_{rl}}{\mu_l}\nabla\left(p_l - \rho_l g z\right), \tag{A-12}$$

and

$$q_g = -\frac{kk_{rg}}{\mu_g}\nabla\left(p_g - \rho_g g z\right), \tag{A-13}$$

where k denotes the saturated permeability, $p_{l,g}$ the fluid pressure, $\mu_{l,g}$ the viscosity of the liquid and gas phases with mass densities $\rho_{l,g}$, and g the acceleration due to gravity. The quantities $n_{l,g}$ refer to the density of the liquid and gas phases based on a molar representation. The mole fractions $X_w^{l,g}$ and $X_a^{l,g}$ satisfy the relations

$$X_w^l + X_a^l = 1, \quad X_w^g + X_a^g = 1. \tag{A-14}$$

Diffusion of water in the aqueous phase is neglected. The liquid and gas pressures are related through the capillary pressure:

$$p_l = p_g - p_c. \tag{A-15}$$

The effective binary gas diffusion coefficient is defined in terms of temperature, pressure, and material properties by

$$D_g^{eff} = \omega\tau\phi s_g D_g^0 \frac{p_0}{p}\left[\frac{T + T_0}{T_0}\right]^\theta, \tag{A-16}$$

where T_0 and p_0 denote reference temperature and pressure, θ is an empirical constant, and ω is an enhancement factor (Walton and Lichtner, 1995). The enhancement factor is usually considered to be inversely proportional to the gas saturation, s_g, which thus cancels from the expression for the effective gas diffusion coefficient.

Adding Eqs. (A-10) and (A-11) eliminates the diffusive terms providing the total mass balance equation for air and water as:

$$\frac{\partial}{\partial t}\left[\phi\left(s_l n_l + s_g n_g\right)\right] + \nabla \cdot \left(q_l n_l + q_g n_g\right) = Q_w + Q_a . \qquad \text{(A-17)}$$

A.5 ENERGY BALANCE EQUATION

The energy balance equation, assuming thermodynamic equilibrium between rock and fluid, is given by

$$\begin{aligned} &\frac{\partial}{\partial t}\left[\phi\left(s_l n_l U_l + s_g n_g U_g\right)\right] + \nabla \cdot \left(q_l n_l H_l + q_g n_g H_g\right) \\ &\quad + \frac{\partial}{\partial t}\left[(1-\phi) C_p^{rock} \rho_{rock} T\right] - \nabla \cdot \kappa \nabla T = Q_e , \end{aligned} \qquad \text{(A-18)}$$

where T denotes the temperature, U_π the total internal energy, and H_π the total enthalpy of the πth fluid phase, C_p the heat capacity, κ the thermal conductivity, and Q_e a source term. Heat produced by chemical reactions is ignored.

Together with appropriate initial and boundary conditions, the mass conservation equations for air, Eq. (A-11), the total mass balance equation, Eq. (A-17), and the heat balance equation [Eq. (A-18)] are solved simultaneously using an implicit finite difference scheme. Upstream weighting is employed at interfaces at which a change in phase occurs.

A detailed description of the equivalent continuum model (ECM) can be found in Lichtner and Walton (1994).