

QA: QA

**SOFTWARE USERS MANUAL (UM)
for the
FEHM Application Version 2.21**

STN: 10086-2.21-00

REV. NO. 00

DOCUMENT ID: 10086-UM-2.21-00

October 2003

Effective date

Prepared by:

Zora V. Dash
Los Alamos National Laboratory, EES-6

Date

Verified by:

Myles Fitzgerald
Los Alamos National Laboratory, EES-9

Date

Fred Pollock
Bechtel SAIC, LLC
Software Independent Verification & Validation

Date

Los Alamos National Laboratory

Copyright, 2003, The Regents of the University of California.

This program was prepared by the Regents of the University of California at Los Alamos National Laboratory (the University) under contract No. W-7405-ENG-36 with the U.S. Department of Energy (DOE). All rights in the program are reserved by the DOE and the University. Permission is granted to the public to copy and use this software without charge, provided that this Notice and any statement of authorship are reproduced on all copies. Neither the U.S. Government nor the University makes any warranty, express or implied, or assumes any liability or responsibility for the use of this software.

CHANGE HISTORY

<u>Revision Number</u>	<u>Effective Date</u>	<u>Description of and Reason for Revision</u>
2.21-00	9/16/03	Initial revision, based on AP-SI.1Q.

TABLE OF CONTENTS

LIST OF FIGURES	6
LIST OF TABLES	7
1.0 PURPOSE	8
2.0 DEFINITIONS AND ACRONYMS	8
2.1 Definitions	8
2.2 Acronyms	8
3.0 REFERENCES	8
4.0 PROGRAM CONSIDERATIONS	10
4.1 Program Options	10
4.2 Initialization	10
4.3 Restart Procedures	11
4.4 Error Processing	11
5.0 DATA FILES	17
5.1 Control file (iocntl)	17
5.2 Input file (inpt)	17
5.3 Geometry data file (incoor)	17
5.4 Zone data file (inzone)	18
5.5 Optional input files	18
5.6 Read file (iread)	18
5.7 Multiple simulations input file	19
5.8 Type curve data input file	19
5.9 Output file (iout)	19
5.10 Write file (isave)	20
5.11 History plot file (ishis)	20
5.12 Solute plot file (istrc)	20
5.13 Contour plot file (iscon)	21
5.14 Contour plot file for dual or dpdp (iscon1)	21
5.15 Stiffness matrix data file (isstor)	21
5.16 Input check file (ischk)	22
5.17 Submodel output file (isubm)	22
5.18 Output error file (ierr)	22
5.19 Multiple simulations script files	23
5.20 PEST output files (ispest, ispst1)	23
5.21 Streamline particle tracking output files (isptr1, isptr2, isptr3)	23
5.22 SURFER and TECPLOT output files	24
5.23 Advanced Visual Systems (AVS) output files	24

6.0	INPUT DATA	27
6.1	General Considerations	27
6.2	Individual Input Records or Parameters	33
7.0	OUTPUT	149
7.1	Output file (<i>filen.out</i>)	149
7.2	Write file (<i>filen.fin</i>)	149
7.3	History plot file (<i>filen.his</i>)	151
7.4	Solute plot file (<i>filen.trc</i>)	152
7.5	Contour plot file (<i>filen.con</i>)	152
7.6	Contour plot file for dual or dpdp (<i>filen.dp</i>)	153
7.7	Stiffness matrix data (<i>filen.stor</i>)	153
7.8	Input check file (<i>filen.chk</i>)	154
7.9	Submodel output file (<i>filen.subbc</i>)	154
7.10	Error output file (<i>fehmn.err</i>)	154
7.11	Multiple simulations script files (<i>fehmn.pre</i> , <i>fehmn.post</i>)	154
7.12	PEST output files (<i>filen.pest</i> , <i>filen.pest1</i>)	154
7.13	Streamline particle tracking output files (<i>filen.sptr1</i> , <i>filen.sptr2</i> , <i>filen.sptr3</i>)	155
7.14	SURFER and TECPLOT output files (<i>surzone_number.txt</i> , <i>veczone_number.txt</i> , <i>trczone_number.txt</i> or <i>teczone_number.plt</i> , <i>veczone_number.plt</i> , <i>trczone_number.plt</i>)	156
7.15	AVS log output file (<i>filen.10001_avs_log</i>)	157
7.16	AVS header output files (<i>filen.number_type_head</i>)	157
7.17	AVS geometry output file (<i>filen.10001_geo</i>)	157
7.18	AVS data output files (<i>filen.number_type_node</i>)	158
8.0	SYSTEM INTERFACE	160
8.1	System-Dependent Features	160
8.2	Compiler Requirements	160
8.3	Hardware Requirements	160
8.4	Control Sequences or Command Files	160
8.5	Software Environment	160
8.6	Installation Instructions	160
9.0	EXAMPLES AND SAMPLE PROBLEMS	162
9.1	Constructing an Input File	162
9.2	Code Execution	165
9.3	Heat Conduction in a Square	170
9.4	DOE Code Comparison Project, Problem 5, Case A	177
9.5	Reactive Transport Example	182
10.0	USER SUPPORT	189

LIST OF FIGURES

Figure 1.	AVS UCD formatted FEHM output files.	26
Figure 2.	Elements available with FEHM in 2-D and 3-D problems showing nodal numbering convention.	54
Figure 3.	Input control file for heat conduction example.	165
Figure 4.	Terminal query for FEHM example run.	167
Figure 5.	Schematic diagram of 2-D heat conduction problem.	170
Figure 6.	Finite element mesh used for 2-D heat conduction problem.	171
Figure 7.	FEHM input file for heat conduction example (heat2d.in).	172
Figure 8.	FEHM output from the 2-D heat conduction example.	173
Figure 9.	Comparison of analytical and model solution for 2-D heat conduction.	176
Figure 10.	Schematic diagram of the geometry and boundary conditions for the DOE code comparison project problem.	178
Figure 11.	FEHM input file for DOE problem.	179
Figure 12.	Comparison of FEHM production well temperatures with results from other codes.	180
Figure 13.	Comparison of FEHM production and observation well pressure drops with results from other codes.	180
Figure 14.	Contour plot of pressure at ten years for the DOE problem.	181
Figure 15.	Contour plot of temperature at ten years for the DOE problem.	181
Figure 16.	Schematic drawing of the geometry and boundary conditions for the cobalt transport problem.	183
Figure 17.	FEHM input file for reactive transport problem	184
Figure 18.	Comparison of FEHM and PDREACT for the breakthrough curves of aqueous species.	188
Figure 19.	Comparison of FEHM and PDREACT for the exit concentration versus time for solid species.	188

LIST OF TABLES

Table I.	Capabilities of FEHM with Macro Command References.	10
Table II.	Initial (Default) Values	11
Table III.	Error Conditions Which Result in Program Termination.	12
Table IV.	AVS File Content Tag.	25
Table V.	Macro Control Statements for FEHM	28
Table VI.	Required and Optional Macros by Problem Type.	163
Table VII.	Input Parameters for the 2-D Heat Conduction Problem	170
Table VIII.	Input Parameters for the DOE Code Comparison Project Problem	177
Table IX.	Input Parameters for the Reactive Transport Test Problem	183

1.0 PURPOSE

This User's Manual documents the use of the FEHM application.

2.0 DEFINITIONS AND ACRONYMS

2.1 Definitions

FEHM- Finite element heat and mass transfer code (Zyvoloski, et al. 1988)

FEHMN- YMP version of FEHM (Zyvoloski, et al. 1992).

The versions are now equivalent and the use of FEHMN has been dropped.

2.2 Acronyms

AVS - Advanced Visual Systems.

I/O - Input / Output.

LANL - Los Alamos National Laboratory.

N/A - Not Applicable.

PEST - Parameter Estimation Program.

SOR - Successive Over-Relaxation Method.

UCD - Unstructured Cell Data.

YMP - Yucca Mountain Site Characterization Project.

3.0 REFERENCES

- Burnett, R. D., and E. O. Frind "Simulation of Contaminant Transport in Three Dimensions. 2. Dimensionality Effects," *Water Resources Res.* **23**:695-705 (1987). TIC:246359
- Carslaw, H. S., and J. C. Jaeger, *Conduction of Heat in Solids*, 2nd Edition, Clarendon Press (1959). TIC:206085
- Conca, J. L., and J. V. Wright, "Diffusion and Flow in Gravel, Soil, and Whole Rock," *Applied Hydrogeology* **1**:5-24 (1992). TIC:224081
- Golder Associates, "User's Guide GoldSim Graphical Simulation Environment", Version 7.40, Golder Associates Inc., Redmond, Washington (2002). MOL.20030130.0347, TIC:235624
- Ho, C. K., "T2FEHM2 Post-Processor to Convert TOUGH2 Files to FEHM-Readable Files for Particle Tracking User's Manual," Sandia National Laboratories, November 1997. MOL.19980218.0246
- Installation Test Plan for the FEHM Application Version 2.21, 10086-ITP-2.21-00.
- Lichtner, P.C., S. Kelkar, and B. Robinson, "New Form of Dispersion Tensor for Axisymmetric Porous Media with Implementation in Particle Tracking," *Water Resources Research*, **38**, (8), 21-1 to 21-16. Washington, D.C.: American Geophysical Union (2002). 163821, TIC:254597
- Molloy, M. W., "Geothermal Reservoir Engineering Code Comparison Project," *Proceedings of the Sixth Workshop on Geothermal Reservoir Engineering*, Stanford University (1980). TIC:249211
- Software Installation Test Plan for the FEHM Application Version 2.21, 10086-ITP-2.21-00.

- Tompson, A. F. B., E. G. Vomvoris, and L. W. Gelhar, "Numerical Simulation of Solute Transport in Randomly Heterogeneous Porous Media: Motivation, Model Development and Application," Lawrence Livermore National Laboratory report, UCID 21281 (1987). MOL.19950131.0007
- Tseng, P. -H., and G. A. Zyvoloski, "A Reduced Degree of Freedom Method for Simulating Non-isothermal Multi-phase Flow in a Porous Medium," *Advances in Water Resources* **23**:731-745 (2000). TIC:254768
- Validation Test Plan for the FEHM Application Version 2.21, 10086-VTP-2.21-00.
- Watermark Computing, "PEST Model-independent parameter estimation: User's Manual," Oxley, Australia: Watermark Computing (1994). MOL.19991028.0052
- Zyvoloski, G. A., and Z. V. Dash, "Software Verification Report FEHMN Version 1.0," LA-UR-91-609 (1991). NNA.19910806.0018
- Zyvoloski, G. A., Z. V. Dash, and S. Kelkar, "FEHM: Finite Element Heat and Mass Transfer Code," LA-11224-MS (1988). NNA.19900918.0013
- Zyvoloski, G. A., Z. V. Dash, and S. Kelkar, "FEHMN 1.0: Finite Element Heat and Mass Transfer Code," LA-12062-MS, Rev.1 (1992). NNA.19910625.0038
- Zyvoloski, G. A., and B. A. Robinson, *GZSOLVE Application*, ECD-97 (1995). MOL.19950915.0248
- Zyvoloski, G. A., B. A. Robinson, and Z. V. Dash, *FEHM Application*, SC-194 (1999). MOL.19990810.0029

4.0 PROGRAM CONSIDERATIONS

4.1 Program Options

The uses and capabilities of FEHM are summarized in Table I with reference to the macro input structure discussed in Section 6.0.

Table I. Capabilities of FEHM with Macro Command References	
I.	Mass, energy balances in porous media
A.	Variable rock properties (rock)
B.	Variable permeability (perm, fper)
C.	Variable thermal conductivity (cond, vcon)
D.	Variable fracture properties, dual porosity, dual porosity/dual permeability (dual, dpdp, gdpm)
II.	Multiple components available
A.	Air-water isothermal mixture available (airwater, bous, head), fully coupled to heat and mass transfer (ngas, vapl, adif)
B.	Up to 10 solutes with chemical reactions between each (trac, rxn)
C.	Multiple species particle tracking (ptrk, mptr, sptr)
D.	Different relative permeability and capillary pressure models (rlp, exrl)
III.	Equation of state flexibility inherent in code (eos)
IV.	Pseudo-stress and storativity models available
A.	Linear porosity deformation (ppor)
B.	Gangi stress model (ppor)
V.	Numerics
A.	Finite element with multiple element capabilities (elem)
B.	Short form input methods available (coor, elem, fdm)
C.	Flexible properties assignment (zone, zonn)
D.	Flexible solution methods
1.	Upwinding, implicit solution available (ctrl)
2.	Iteration control adaptive strategy (iter)
E.	Finite volume geometry (finv, isot)
VI.	Flexible time step and stability control (time)
VII.	Time-dependent fixed value and flux boundary conditions (flow, boun, hflx)

4.2 Initialization

The coefficient arrays for the polynomial representations of the density (crl, crv), enthalpy (cel, cev), and viscosity (cvl, cvv) functions are initialized to the values enumerated in Table III of the "Models and Methods Summary" of the FEHM Application (Zyvoloski et al. 1999), while values for the saturation pressure and temperature function coefficients are found in Table IV of that document. All other global array and scalar variables, with the exception of the variables listed in Table II, are initialized to zero if integer or real, character variables are initialized to a single blank character, and logical variables are initialized as false.

Variable	Value	Variable	Value	Variable	Value
aiaa	1.0	contim	1.0e+30	daymax	30.0
daymin	1.0e-05	g1	1.0e-06	g2	1.0e-06
g3	1.0e-03	iad_up	1000	iamx	500
icons	1000	irlp	1	nbits	256
ncntr	10000000	nicg	1	rnmax	1.0e+11
str	1.0	strd	1.0	tmch	1.0e-09
upwgt	1.0	upwgt_a	1.0	weight_factor	1.0e-3

4.3 Restart Procedures

FEHM writes a restart file for each run. The restart output file name may be given in the input control file or as terminal input, or if unspecified will default to *fehmn.fin* (see Section 6.2.1 on page 33). The file is used on a subsequent run by providing the name of the generated file (via control file or terminal) for the restart input file name. It is recommended that the restart input file name be modified to avoid confusion with the restart output file. For example, by changing the suffix to *.ini*, the default restart output file, *fehmn.fin* would be renamed *fehmn.ini*, and that file name placed in the control file or given as terminal input. Values from the restart file will overwrite any variable initialization prescribed in the input file. The initial time of simulation will also be taken from the restart file unless specified in the macro **time** input (see Section 6.2.68 on page 133).

4.4 Error Processing

Due to the nonlinearity of the underlying partial differential equations, it is possible to produce an underflow or overflow condition through an unphysical choice of input parameters. More likely the code will fail to converge or will produce results which are out of bounds for the thermodynamic functions. The code will attempt to decrease the time step until convergence occurs. If the time step drops below a prescribed minimum the code will stop, writing a restart file. The user is encouraged to look at the input check file which contains information regarding maximum and minimum values of key variables in the code. All error and warning messages will be output to an output error file or the main output file.

Table III provides additional information on errors that will cause FEHM to terminate.

Table III. Error Conditions Which Result in Program Termination	
Error Condition	Error Message
I/O file error	
Unable to create / open I/O file	**** Error opening file <i>fileid</i> **** . . . ****-----**** **** JOB STOPPED **** ****-----****
Coefficient storage file not found	program terminated because coefficient storage file not found
Coefficient storage file can not be read	error in parsing beginning of stor file -or- stor file has unrecognized format:quit -or- stor file has neq less than data file:quit
Coefficient storage file already exists	>>> changing name of new *.stor (old file exists) new file name is fehmn_temp.stor -and- >>> name fehmn_temp.stor is used : stopping
Optional input file not found	ERROR nonexistent file <i>filename</i> STOPPED trying to use optional input file
Unable to open optional input file	ERROR opening <i>filename</i> STOPPED trying to use optional input file
Unable to determine file prefix for AVS output files	FILE ERROR: nmfil2 file: <i>filename</i> unable to determine contour file prefix
Unable to determine file prefix for pest output files	FILE ERROR: nmfil15 file: <i>filename</i> unable to determine pest file name -or- FILE ERROR: nmfil16 file: <i>filename</i> unable to determine pest1 file name
Unable to determine file prefix for streamline particle tracking output files	FILE ERROR: nmfil17 file: <i>filename</i> unable to determine sptr1 file name -or- FILE ERROR: nmfil18 file: <i>filename</i> unable to determine sptr2 file name -or- FILE ERROR: nmfil19 file: <i>filename</i> unable to determine sptr3 file name
Unable to determine file prefix for submodel output file	FILE ERROR: nmfil24 file: <i>filename</i> unable to determine submodel file name
Input deck errors	
Coordinate or element data not found	**** COOR Required Input **** -or- **** ELEM Required Input **** ****-----**** **** JOB STOPPED **** ****-----****

Table III. Error Conditions Which Result in Program Termination	
Error Condition	Error Message
Inconsistent zone coordinates	inconsistent zone coordinates <i>izone = izone</i> please check <i>icnl</i> in macro CTRL
Invalid AVS keyword read for macro cont	ERROR:READ_AVS_IO unexpected character string (terminate program execution) Valid options are shown: . . .
Invalid keyword or input order read for macro boun	The invalid string was: <i>string</i> time change was not first keyword, stop -or- illegal keyword in macro <i>boun</i> , stopping
Invalid keyword read for macro subm	>>>> error in keyword for macro <i>subm</i> <<<<
Invalid macro read	**** error in input deck : <i>char</i> ****
Invalid parameter values (macros using loop construct)	Fatal error - for array number <i>arraynum</i> macro - <i>macro</i> Group number - <i>groupnum</i> Something other than a real or integer has been specified -or- Line number - <i>line</i> Bad input, check this line -or- Fatal error, too many real inputs to <i>initdata2</i> -or- Fatal error, too many integer inputs to <i>initdata2</i>
Invalid streamline particle tracking parameter	<i>ist</i> must be less than or equal to 2
Invalid tracer input	** Using Old Input Enter Temperature Dependency Model Number: 1 - Van Hoff 2 - awwa model, see manual for details **
Invalid transport conditions	Fatal error You specified a Henrys Law species with initial concentrations input for the vapor phase (<i>icns = -2</i>), yet the Henrys Constant is computed as 0 for species number <i>speciesnum</i> and node number <i>nodenum</i> . If you want to simulate a vapor-borne species with no interphase transport, then you must specify a gaseous species (<i>icns = -1</i>).
Invalid flag specified for diffusion coefficient calculation	ERROR -- Illegal Flag to <i>concadiff</i> Code Aborted in <i>concadiff</i>
Optional input file name can not be read	ERROR reading optional input file name STOPPED trying to use optional input file

Table III. Error Conditions Which Result in Program Termination	
Error Condition	Error Message
Optional input file contains data for wrong macro	ERROR --> Macro name in file for macro <i>macroname</i> is <i>wrong_macroname</i> STOPPED trying to use optional input file
Option not supported	This option (<i>welbor</i>) not supported. Stop in input -or- specific storage not available for non isothermal conditions : stopping -or- gangi model not yet available for air-water-heat conditions : stopping -or- Gencon not yet set for <i>rdldof</i> . Stop in <i>gencon</i>
Parameter not set	>>> gravity not set for head problem: stopping <<<
Relative permeabilities specified for non-dual or -double porosity model.	***** f-m terms but no <i>dpdp</i> : stopping *****
Invalid parameters set	
Dual porosity	**** check fracture volumes, stopping**** **** check equivalent continuum VGs ****
Finite difference model (FDM)	>>> dimension (<i>icnl</i>) not set to 3 for FDM: stopping <<<
Maximum number of nodes allowed is less than number of equations	**** <i>n0</i> (<i>n0</i>) .lt. <i>neg</i> (<i>neg</i>) **** check parameter statements ***
Node number not in problem domain (macros <i>dvel</i> , <i>flxo</i> , <i>node</i> , <i>nod2</i> , <i>nod3</i> , <i>zone</i> , <i>zonn</i>)	**** Invalid input: macro <i>macro</i> ****' **** Invalid node specified, value is greater than <i>n0</i> (<i>n0</i>): stopping ****
Noncondensable gas	cannot input <i>ngas</i> temp in single phase -or- <i>ngas</i> pressure lt 0 at temp and total press given max allowable temperature <i>temp</i> -or- <i>ngas</i> pressure gt total pressure <i>i= i</i> -or- <i>ngas</i> pressure lt 0.
Particle tracking	ERROR: <i>Pcnsk</i> in <i>ptrk</i> must be either always positive or always negative. Code aborted in <i>set_ptrk.f</i>
Relative permeabilities	cannot have anisotropic perms for <i>rlp</i> model 4 or <i>rlp</i> model 7 with equivalent continuum stopping

Table III. Error Conditions Which Result in Program Termination	
Error Condition	Error Message
Tracer	ERROR: Can not have both particle tracking (ptrk) and tracer input (trac). Code Aborted in concen.f -or- Gencon not yet set for rd1dof. Stop in gencon -or- ERROR - solute accumulation option cannot be used with cnsk<0 -or- ** On entry to <i>SRNAME</i> parameter number <i>I2</i> had an illegal value
Insufficient storage	
Boundary conditions	exceeded storage for number of models
Dual porosity	***** n > n0, stopping *****
Generalized dual porosity	In <i>gdpm</i> macro, <i>ngdpmnodes</i> must be reduced to reduce storage requirements A value of <i>ngdpm_actual</i> is required The current value set is <i>ngdpmnodes</i> -or- Fatal error in <i>gdpm</i> macro A value of <i>ngdpm_actual</i> is required' The current value set is <i>ngdpmnodes</i> Increase <i>ngdpmnodes</i> to <i>ngdpm_actual</i> and restart
Geometric coefficients	program terminated because of insufficient storage
Tracer	**** memory too small for multiple tracers ****
Invalid colloid particle size distribution	Fatal error, the colloid particle size distribution must end at 1
Invalid particle diffusion	Fatal error For a <i>dmdp</i> simulation, Do not apply the matrix diffusion particle tracking to the matrix nodes, only the fracture nodes
Invalid particle state	Initial state of particles is invalid Particle number <i>i1</i>
Error computing geometric coefficients	iteration in zone did not converge, <i>izone</i> = <i>zone_number</i> please check <i>icnl</i> in macro CTRL
Too many negative volumes or finite element coefficients	too many negative volumes: stopping -or- too many negative coefficients: stopping
Unable to compute local coordinates	iteration in zone did not converge, <i>izone</i> = <i>zone</i> please check <i>icnl</i> in macro CTRL
Unable to normalize matrix	cannot normalize
Singular matrix in LU decomposition	singular matrix in <i>ludcmp</i>

Table III. Error Conditions Which Result in Program Termination	
Error Condition	Error Message
Singular matrix in speciation calculations	Speciation Jacobian matrix is singular! -or- Scaled Speciation Jacobian matrix is singular! -or- Speciation scaling matrix is singular!
Solution failed to converge	timestep less than daymin <i>timestep_number</i> <i>current_timestep_size</i> <i>current_simulation_time</i> -or- Tracer Time Step Smaller Than Minimum Step Stop in resettrc -or- Newton-Raphson iteration limit exceeded in speciation subroutine! -or- Newton-Raphson iteration limit exceeded in scaled speciation subroutine! Failure at node <i>i</i>

5.0 DATA FILES

5.1 Control file (iocntl)

5.1.1 Content

The control file contains the names of the input and output files needed by the FEHM code. In addition to listing the I/O file names, the terminal (tty) output option and the user subroutine number are given. The control file provides the user an alternate means for inputting file names, terminal output option, and user subroutine number than through the terminal I/O. It is useful when long file names are used or when files are buried in several subdirectories, or for automated program execution. The elements of the file and input requirements are described in Section 6.2.1.

5.1.2 Use by Program

The control file provides the FEHM application with the names of the input and output files, terminal output units, and user subroutine number to be utilized for a particular run. The default control file name is *fehmn.files*. If the control file is found, it is read prior to problem initialization. If not present, terminal I/O is initiated and the user is prompted for required information. A control file may use a name other than the default. This alternate control file name would be input during terminal I/O. See Section 6.1.1.1.

5.1.3 Auxiliary Processing

N/A

5.2 Input file (inpt)

5.2.1 Content

The input file contains user parameter initialization values and problem control information. The form of the file name is *filen* or *filen.** where "*filen*" is a prefix used by the code to name auxiliary files and ".*" represents an arbitrary file extension. If a file name is not specified when requested during terminal I/O, the file *fehmn.dat* is the default. The organization of the file is described in detail in Section 6.2.

5.2.2 Use by Program

The input file provides the FEHM application with user parameter initialization values and problem control information. The input file is read during problem initialization.

5.2.3 Auxiliary Processing

N/A

5.3 Geometry data file (incoor)

5.3.1 Content

The geometry data file contains the mesh element and coordinate data. This can either be the same as the input file or a separate file.

5.3.2 Use by Program

The geometry data file provides the FEHM application with element and coordinate data. The geometry data file is read during problem initialization.

5.3.3 Auxiliary Processing

N/A

5.4 Zone data file (inzone)

5.4.1 Content

The zone data file contains the zone information (see macro **zone**). This can either be the same as the input file or a separate file.

5.4.2 Use by Program

The zone data file provides the FEHM application with initial geometric zone descriptions. The zone data file is read during problem initialization.

5.4.3 Auxiliary Processing

N/A

5.5 Optional input files

5.5.1 Content

The optional input files contain user parameter initialization values and problem control information. The names of optional input files are provided in the main input file to direct the code to auxiliary files to be used for data input. Their use is described in detail in Section 6.2.4

5.5.2 Use by Program

The optional input files provide the FEHM application with user parameter initialization values and problem control information. The optional input files are read during problem initialization.

5.5.3 Auxiliary Processing

N/A

5.6 Read file (iread)

5.6.1 Content

The read file contains the initial values of pressure, temperature, saturation, and simulation time (the restart or initial state values). It may also contain initial species concentrations for transport simulation or particle tracking data for particle tracking simulation restarts. The naming convention is similar to that for the output file. The generated name is of the form *filen.ini*.

5.6.2 Use by Program

The FEHM application uses the read file for program restarts. The read file is read during problem initialization.

5.6.3 Auxiliary Processing

N/A

5.7 Multiple simulations input file

5.7.1 Content

The multiple simulations input file contains the number of simulations to be performed and, on UNIX systems, instructions for pre- and post-processing input and output data during a multiple realization simulation. The file name is *fehmn.msim*.

5.7.2 Use by Program

The FEHM application uses the multiple simulations input file to setup control for a multiple realization simulation. It is accessed at the beginning the program.

5.7.3 Auxiliary Processing

N/A

5.8 Type curve data input file

5.8.1 Content

The type curve data input file contains parameter and data values necessary to compute dispersion delay times for the particle tracking models using type curves.

5.8.2 Use by Program

The FEHM application uses the type curve data input file to read the parameter and data values necessary to simulate dispersion delay times for the particle tracking models. It is accessed at the beginning the program if a particle tracking simulation using type curves is run.

5.8.3 Auxiliary Processing

N/A

5.9 Output file (iout)

5.9.1 Content

The output file contains the FEHM output. The file name is provided in the input control file or as terminal input, or may be generated by the code from the name of the input file if terminal I/O is invoked. The generated name is of the form *filen.out* where the “*filen*” prefix is common to the input file.

5.9.2 Use by Program

The FEHM application uses the output file for general program time step summary information. It is accessed throughout the program as the simulation steps through time.

5.9.3 Auxiliary Processing

This file may be accessed by scripts or user developed programs to extract summary information not recorded in other output files.

5.10 Write file (isave)

5.10.1 Content

The write file contains the final values of pressure, temperature, saturation, and simulation time for the run. It may also contain final species concentrations for transport simulations or particle tracking data for particle tracking simulations. This file can in turn be used as the read file in a restart run. The naming convention is similar to that for the output file. The generated name is of the form *filen.fin*.

5.10.2 Use by Program

The FEHM application uses the write file for storing state data of the simulation. It is accessed at specified times throughout the program when state data should be stored.

5.10.3 Auxiliary Processing

This file may be accessed by scripts or user developed programs to extract final state information not recorded in other output files.

5.11 History plot file (ishis)

5.11.1 Content

The history plot file contains data for history plots of variables. The naming convention is similar to that for the output file. The generated name is of the form *filen.his*.

5.11.2 Use by Program

The FEHM application uses the history plot file for storing history data for pressure, temperature, flow, and energy output. It is accessed throughout the program as the simulation steps through time.

5.11.3 Auxiliary Processing

This file may be used to produce history plots by external graphics programs.

5.12 Solute plot file (istrc)

5.12.1 Content

The solute plot file contains history data for solute concentrations at specified nodes. The naming convention is similar to that for the output file. The generated name is of the form *filen.trc*.

5.12.2 Use by Program

The FEHM application uses the solute plot file for storing history data for tracer output. It is accessed throughout the program as the simulation steps through time.

5.12.3 Auxiliary Processing

This file may be used to produce history plots of tracers by external graphics programs.

5.13 Contour plot file (iscon)

5.13.1 Content

The contour plot file contains the contour plot data. The naming convention is similar to that for the output file. The generated name is of the form *filen.con*.

5.13.2 Use by Program

The FEHM application uses the contour plot file for storing contour data for pressure, temperature, flow, energy output, and tracer output. It is accessed at specified times throughout the program when contour data should be stored.

5.13.3 Auxiliary Processing

This file may be used to produce contour plots by external graphics programs.

5.14 Contour plot file for dual or dpdp (iscon1)

5.14.1 Content

The dual or dpdp contour plot file contains the contour plot data for dual porosity or dual porosity / dual permeability problems. The naming convention is similar to that for the output file. The generated name is of the form *filen.dp*.

5.14.2 Use by Program

The FEHM application uses the dual or dpdp contour plot file for storing contour data for pressure, temperature, flow, energy output, and tracer output for dual porosity or dual porosity / dual permeability problems. It is accessed at specified times throughout the program when contour data should be stored.

5.14.3 Auxiliary Processing

This file may be used to produce contour plots by external graphics programs.

5.15 Stiffness matrix data file (isstor)

5.15.1 Content

The stiffness matrix data file contains finite element coefficients calculated by the code. It is useful for repeated calculations that use the same mesh, especially for large problems. The naming convention is similar to that for the output file. The generated name is of the form *filen.stor*.

5.15.2 Use by Program

The stiffness matrix data file is both an input and an output file the FEHM application uses for storing or reading finite element coefficients calculated by the code. The stiffness matrix data file is read during problem initialization if being used for input. It is accessed after finite element coefficients are calculated if being used for output.

5.15.3 Auxiliary Processing

N/A

5.16 Input check file (ischk)

5.16.1 Content

The input check file contains a summary of coordinate and variable information, suggestions for reducing storage, coordinates where maximum and minimum values occur, and information about input for variables set at each node. The naming convention is similar to that for the output file. The generated name is of the form *filen.chk*.

5.16.2 Use by Program

The FEHM application uses the input check file for writing a summary of the data initialization. The input check file is accessed during data initialization and when it has been completed.

5.16.3 Auxiliary Processing

N/A

5.17 Submodel output file (isubm)

5.17.1 Content

The submodel output file contains “**flow**” macro data that represents boundary conditions for an extracted submodel (i.e., the output will use the format of the “**flow**” input macro). The naming convention is similar to that for the output file. The generated name is of the form *filen.subbc*.

5.17.2 Use by Program

The FEHM application uses the submodel output file for writing extracted boundary conditions. The submodel output file is accessed during data initialization and at the end of the simulation.

5.17.3 Auxiliary Processing

N/A

5.18 Output error file (ierr)

5.18.1 Content

The output error file contains any error or warning messages issued by the code during a run. The file is always named *fehmn.err* and will be found in the directory from which the problem was executed.

5.18.2 Use by Program

The FEHM application uses the output error file for writing error or warning messages issued by the code during a run. It may be accessed at any time.

5.18.3 Auxiliary Processing

N/A

5.19 Multiple simulations script files

5.19.1 Content

The multiple simulations script files contain instructions for pre- and post-processing input and output data during a multiple realization simulation. Pre-processing instructions are always written to a file named *fehmn.pre*, while post-processing instructions are always written to a file named *fehmn.post*, and will be found in the directory from which the program was executed.

5.19.2 Use by Program

The FEHM application uses the multiple simulations script files for writing UNIX shell script style instructions. They are generated from information contained in the multiple simulations input file at the beginning of the program. The pre-processing instructions are then executed (invoked as a shell script) prior to data input for each realization, and the post-processing instructions are executed at the completion of each realization. The following command is used to execute the scripts: `sh script_file $1 $2`, where \$1 is the current simulation number and \$2 is *nsim*, the total number of simulations.

5.19.3 Auxiliary Processing

N/A

5.20 PEST output files (ispest, ispst1)

5.20.1 Content

The PEST output files contain output data (pressure or head, saturations, and temperatures) in a format suitable for use by the Parameter Estimation Program (PEST) (Watermark Computing, 1994). The generated names are of the form *filen.pest* and *filen.pest1*, where *filen* is based on the file prefix for the general output file. If an output file is not defined the default names are *fehmn.pest* and *fehmn.pest1*.

5.20.2 Use by Program

The FEHM application uses the PEST output files for writing parameter values generated during a run. They may be accessed at any time throughout the program as the simulation steps through time, but only values at the final state are saved.

5.20.3 Auxiliary Processing

The primary file (*filen.pest*) is generated to provide input to the Parameter Estimation Program (PEST) (Watermark Computing, 1994). The second file is generated to provide a backup of general information for review purposes.

5.21 Streamline particle tracking output files (isptr1, isptr2, isptr3)

5.21.1 Content

The streamline particle tracking output files contain output data from a streamline particle tracking simulation. The generated names are of the form *filen.sptr1*, *filen.sptr2* and *filen.sptr3*, where *filen* is based on the file prefix for the tracer output file or the general output file. If those files are

not defined the default names are *fehmn.sptr1*, *fehmn.sptr2*, and *fehmn.sptr3*.

5.21.2 Use by Program

The FEHM application uses the streamline particle tracking output files for writing parameter values generated during a run. They may be accessed at any time throughout the program as the simulation steps through time.

5.21.3 Auxiliary Processing

These files may be used to produce streamline plots or breakthrough data plots by external graphics programs.

5.22 SURFER and TECPLOT output files

5.22.1 Content

The SURFER or TECPLOT output files contain output data (pressure or head, saturations, temperatures, fluxes, permeabilities, saturations, porosities, velocities, and concentrations) for selected locations or nodes. The generated names, *sur*.txt*, *trc*.txt*, and *vec*.txt* or *tec*.plt*, *trc*.plt*, and *vec*.plt*, contain a numeric string which identifies the output zone and contour time sequence.

5.22.2 Use by Program

The FEHM application uses the SURFER or TECPLOT output files for storing geometry based data for material properties (permeabilities and porosities), temperature, saturation, pressure or head, velocities, and solute concentrations in a format readable by SURFER or TECPLOT graphics. They are accessed at specified times throughout the program when contour data should be stored.

5.22.3 Auxiliary Processing

These files are used for visualization and analysis of data by SURFER or TECPLOT plotting programs.

5.23 Advanced Visual Systems (AVS) output files

5.23.1 Content

The AVS output files contain geometry based data that can be imported into Advanced Visual Systems (AVS) UCD (unstructured cell data) graphics routines. The AVS output files each have a unique file name indicating the section type, the data type and the time step the files were created. These file names are automatically generated by the code and are of the form *filen.NumberAVS_id*, where *filen* is common to the contour output file prefix if defined, otherwise it is the input file prefix, *Number* is a value between 10001 and 99999, and *AVS_id* is a string denoting file content (see Table IV). In general, *_head* are header files, *_geo* is the geometry file, and *_node* the data files. The following, *_mat*, *_sca*, *_vec*, *_con*, *_mat_dual*, *_sca_dual*, *_vec_dual*, or *_con_dual*, are pre-appended to *_head* and *_node* to further identify the data selected for output. Currently all properties are node based rather than cell based.

Table IV. AVS File Content Tag	
<i>AVS_id</i>	File purpose
_avs_log	Log file from AVS output routines
_geo	Geometry output file containing coordinates and cell information
_mat_head	AVS UCD header for material properties file.
_mat_dual_head	AVS UCD header for material properties file for dual or dpdp.
_sca_head	AVS UCD header for scalar parameter values file.
_sca_dual_head	AVS UCD header for scalar parameter values file for dual or dpdp.
_vec_head	AVS UCD header for vector parameter values.
_vec_dual_head	AVS UCD header for vector parameter values for dual or dpdp.
_con_head	AVS UCD header for solute concentration file.
_con_dual_head	AVS UCD header for solute concentration file for dual or dpdp.
_mat_node	Data output file with Material properties.
_mat_dual_node	Data output file with Material properties for dual or dpdp.
_sca_node	Data output file with Scalar parameter values (pressure, temperature, saturation).
_sca_dual_node	Data output file with Scalar parameter values (pressure, temperature, saturation) for dual or dpdp.
_vec_node	Data output file with Vector parameter values (velocity).
_vec_dual_node	Data output file with Vector parameter values (velocity) for dual or dpdp.
_con_node	Data output file with Solute concentration.
_con_dual_node	Data output file with Solute concentration for dual or dpdp.

5.23.2 Use by Program

The FEHM application uses the AVS output files for storing geometry based data for material properties, temperature, saturation, pressure, velocities, and solute concentrations in a format readable by AVS graphics. The log output file is created on the first call to the AVS write routines. It includes the code version number, date and problem title. When output for a specified time step has been completed, a line containing the file name prefix, time step, call number (the initial call is 1 and is incremented with each call to write AVS contour data) and problem time (days) is written.

The header files, one for each type of data being stored, and the single geometry file are written during the first call to the AVS output routines. The node data files are written for each call to the AVS write routines, at specified times throughout the program when contour data should be stored using AVS format.

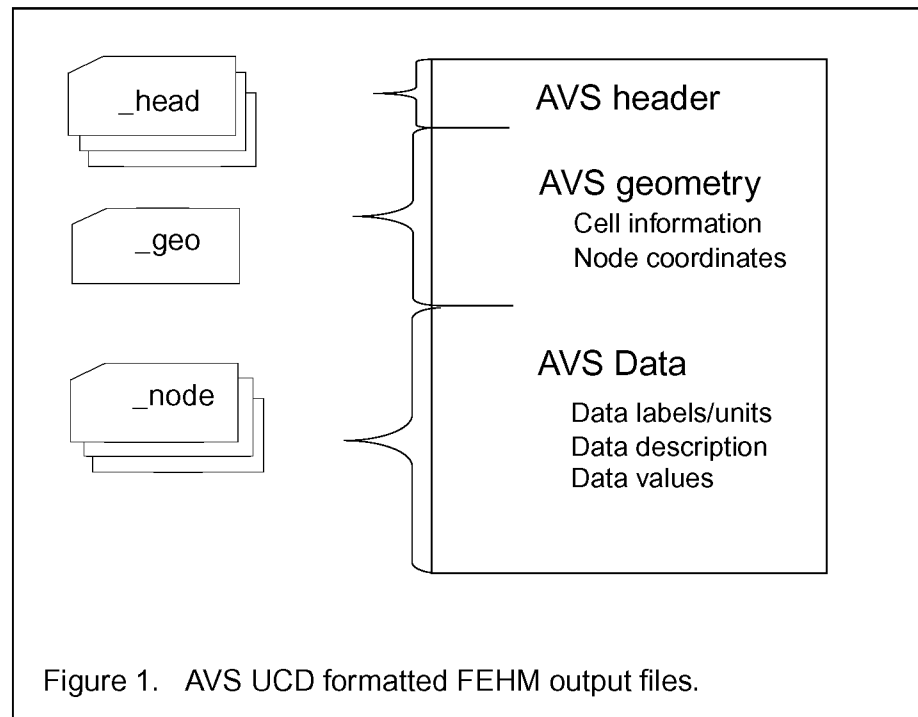
5.23.3 Auxiliary Processing

These files are used for visualization and analysis of data by AVS.

To use with AVS, the appropriate header file, geometry file, and data file for each node must be concatenated into one file of the form *filen.inp* (Fig. 1). This can be done with the script **feh2avs** for a series of files with the same root *filen* or manually, for example:

```
cat filen.10001_head filen.10001_geo filen.10001_mat_node > filen.10001.inp
```

Once header and geometry have been merged with data files into a single AVS file, the data can be imported into AVS using the read_ucd module.



6.0 INPUT DATA

6.1 General Considerations

6.1.1 Techniques

6.1.1.1 Control File or Terminal I/O Startup

The input/output (I/O) file information is provided to the code from an input control file or the terminal. The default control file name is *fehmn.files*. If a control file with the default name is present in the directory from which the code is being executed, no terminal input is required. If the default control file is not present, input prompts are written to the screen preceded by a short description of the I/O files used by FEHM. The descriptions of the I/O files are elaborated on in Section 5.0. The initial prompt asks for the name of a control file. If a control file name is entered for that prompt no further terminal input is required. If a control file is not used, the user is then prompted for I/O file names, the tty output flag, and user subroutine number. When the input file name is entered from the terminal the user has the option of letting the code generate the names for the remainder of the auxiliary files using the input file name prefix. The form of the input file name is *filen* or *filen.** where "*filen*" is the prefix used by the code to name the auxiliary files and ".*" represents an arbitrary file extension.

6.1.1.2 Multiple Realization Simulations

The code has an option for performing multiple simulation realizations (calculations) where input (e.g., porosity, permeability, saturation, transport properties or particle distributions) is modified for each realization but the calculations are based on the same geometric model. Multiple realizations are initiated by including a file called *fehmn.msimsim* in the directory from which the code is being run. If invoked, a set number of simulations are performed sequentially, with pre- and post-processing steps carried out before and after each simulation. This capability allows multiple simulations to be performed in a streamlined fashion, with processing to change input files before each run and post-processing to obtain relevant results after each run.

6.1.1.3 Macro Control Structure

The finite element heat and mass transfer code (FEHM) contains a macro control structure for data input that offers added flexibility to the input process. The macro command structure makes use of a set of control statements recognized by the input module of the program. When a macro control statement is encountered in an input file, a certain set of data with a prescribed format is expected and read from the input file. In this way, the input is divided into separate, unordered blocks of data. The input file is therefore a collection of macro control statements, each followed by its associated data block. Blocks of data can be entered in any order, and any blocks unnecessary to a particular problem need not be entered. The

macro control statements must appear in the first four columns of a line. The other entries are free format, which adds flexibility, but requires that values be entered for all input variables (no assumed null values).

As an aid to the user, the capabilities of FEHM summarized in Table I refer to applicable macro commands. Table V lists the macro control statements with a brief description of the data associated with each. A more detailed description of each macro control statement and its associated input are found in Section 6.2. Macro control statements may be called more than once, if, for example, the user wishes to reset some property values after defining alternate zones. Some statements are required, as indicated in Table V, the others are optional.

Table V. Macro Control Statements for FEHM	
Control Statement	Description
adif	Air-water vapor diffusion
airwater	Isothermal air-water input
boun	Boundary conditions (required for flow problem if macro flow is not used)
bous	Boussinesq-type approximation
cden	Concentration-dependent density
cond	Thermal conductivity data (required for non-isothermal problem)
cont	Contour plot data
coor	Node coordinate data (required if macro fdm is not used)
ctrl	Program control parameters (required)
dpdp	Double porosity/double permeability model input
dual	Dual porosity model input
dvel	Velocity printout (formerly macro velo)
elem	Element node data (required if macro fdm is not used)
eos	Equation of state data
exrl	Linearized relative permeability
fdm	Finite difference input (required if macro coor and elem are not used)
finv	Finite volume flow coefficients
flow	Flow data (required for flow problem if macro boun is not used)
flo2	Alternate format for flow data (input using 3-D planes)
flo3	Alternate format for flow data (defined for seepage faces)

Table V. Macro Control Statements for FEHM (Continued)

Control Statement	Description
floa	Alternate format for flow data (additive to previous flow definition)
flxo	Flux printout
flxz	Total flux through a zone printout
fper	Permeability scaling factors
frip	Relative permeability factors for residual air effect
gdpm	Generalized dual porosity model
grad	Gradient model input
head	Hydraulic head values
hflx	Heat flux input
hyco	Hydraulic conductivity input (required if macro perm is not used)
ice or meth	Ice phase calculations (untested)
impf	Time step control based on maximum allowed variable change
init	Initial value data (required if macro pres or restart file is not used)
isot	Isotropic definition of control volume/finite element coefficients
iter	Iteration parameters
itfc	Flow and transport between zone interfaces
itup	Iterations used with upwinding
iupk	Upwind transmissibility including intrinsic permeability
mdnode	Enables extra connections to be made to nodes
mptr	Multiple species particle tracking simulation input
ngas	Noncondensable gas (air) data
nobr	Don't break connection between nodes with boundary conditions
node	Node numbers for output and time histories
nod2	Node numbers for output and time histories, and alternate nodes for terminal output
nod3	Node numbers for output and time histories, alternate nodes for terminal output, and alternate nodes for variable porosity model information
perm	Permeability input (required if macro hyco is not used)
pest	Parameter estimation routine output
ppor	Pressure and temperature dependent porosity and permeability

Table V. Macro Control Statements for FEHM (Continued)

Control Statement	Description
pres	Initial pressure, temperature, and saturation data, boundary conditions specification (required if macro init or restart file is not used)
ptrk	Particle tracking simulation input
renu	Renumbers nodes
rflo	Read in flux values
rlp	Relative permeability input (required for 2-phase problem, otherwise optional)
rock	Rock density, specific heat, and porosity input (required)
rxn	Chemical reaction rate model input
sol	Solver specifications
sptr	Streamline particle tracking simulation input
stop	Signals the end of input (required)
subm	Submodel boundary condition output
svar	Enable pressure-enthalpy variables
szna or napl	Isothermal NAPL-water input
text	Text input to be written to output file
thic	Variable thickness input for two-dimensional problems
time	Time step and time of simulation data (required)
trac	Solute simulation input
user	User subroutine call
vapl	Vapor pressure lowering
vcon	Variable thermal conductivity input
wtsi	Water table, simplified
zone	Geometric definition of grid for input parameter assignment
zonn	Geometric definition of grid for input parameter assignment

Comments may be entered in the input file by beginning a line with a '#' symbol (the '#' symbol must be found in the first column of the line). Comments may precede or follow macro blocks but may not be found within a block.

Optional input files may be used by substituting a keyword and file name in the main input file (described in detail in Section 6.2.4). The normal macro input is then entered in the auxiliary file.

Many input parameters such as porosity or permeability vary throughout the grid and need to have different values assigned at different nodes. This is accomplished in two ways. The first uses a nodal loop-type definition (which is the default):

JA, **JB**, **JC**, PROP1, PROP2, . . .

where

JA -first node to be assigned with the properties PROP1, PROP2, . . .

JB -last node to be assigned with the properties PROP1, PROP2, . . .

JC -loop increment for assigning properties PROP1, PROP2, . . .

PROP1, PROP2, etc. - property values to be assigned to the indicated nodes.

In the input blocks using this structure, one or more properties are manually entered in the above structure. When a blank line is entered, that input block is terminated and the code proceeds to the next group or control statement. (Note that blank input lines are shaded in the examples shown in Section 6.2.) The nodal definition above is useful in simple geometries where the node numbers are easily found. Boundary nodes often come at regular node intervals and the increment counter **JC** can be adjusted so the boundary conditions are easily entered. To set the same property values at every node, the user may set **JA** and **JC** to 1 and **JB** to the total number of nodes, or alternatively set **JA** = 1, and **JB** = **JC** = 0.

For dual porosity problems, which have three sets of parameter values at any nodal position, nodes 1 to N [where N is the total number of nodes in the grid (see macro **coor**)] represent the fracture nodes, nodes N + 1 to 2N are generated for the second set of nodes, the first matrix material, and nodes 2N + 1 to 3N for the third set of nodes, the second matrix material. For double porosity/double permeability problems, which have two sets of parameter values at any nodal position, nodes 1 to N represent the fracture nodes and nodes N + 1 to 2N are generated for the matrix material.

For more complicated geometries, such as 3-D grids, the node numbers are often difficult to determine. Here a geometric description is preferred. To enable the geometric description the **zone** control statement (page 145) is used in the input file before the other property macro statements occur. The input macro **zone** requires the specification of the coordinates of 4-node parallelograms for 2-D problems or 8-node polyhedrons in 3-D. In one usage of the control statement **zone** all the nodes are placed in geometric zones and assigned an identifying number. This number is then addressed in the property input macro commands by specifying a **JA** < 0 in the definition of the loop parameters given above. For example if **JA** = -1, the

properties defined on the input line would be assigned to the nodes defined as belonging to geometric Zone 1 (**JB** and **JC** must be input but are ignored in this case). The control statement **zone** may be called multiple times to redefine geometric groupings for subsequent input. The previous zone definitions are not retained between calls. Up to 1000 zones may be defined. For dual porosity problems, which have three sets of parameter values at any nodal position, Zone $\text{ZONE_DPADD} + I$ is the default zone number for the second set of nodes defined by Zone I, and Zone $2 * \text{ZONE_DPADD} + I$ is the default zone number for the third set of nodes defined by Zone I. For double porosity/double permeability problems, which have two sets of parameter values at any nodal position, Zone $\text{ZONE_DPADD} + I$ is the default zone number for the second set of nodes defined by Zone I. The value of ZONE_DPADD is determined by the number of zones that have been defined for the problem. If less than 100 zones have been used ZONE_DPADD is set to 100, otherwise it is set to 1000. Zones of matrix nodes may also be defined independently if desired.

Alternatively, the **zonn** control statement (page 148) may be used for geometric descriptions. Regions are defined the same as for control statement **zone** except that previous zone definitions are retained between calls unless specifically overwritten.

6.1.1.4 GoldSim Interface

To interface with GoldSim, FEHM is compiled as a dynamic link library (DLL) subroutine that is called by the GoldSim code. When FEHM is called as a subroutine from GoldSim, the GoldSim software controls the time step of the simulation, and during each call, the transport step is carried out and the results passed back to GoldSim for processing and/or use as radionuclide mass input to another portion of the GoldSim system, such as a saturated zone transport submodel. The interface version of FEHM is set up only to perform particle tracking simulations of radionuclide transport, and is not intended to provide a comprehensive flow and transport simulation capability for GoldSim. Information concerning the GoldSim user interface may be found in the GoldSim documentation (Golder Associates, 2002).

6.1.2 Consecutive Cases

Consecutive cases can be run using the multiple realizations simulation option (see Section 6.1.1.2 on page 27). The program retains only the geometric information between runs (i.e., the grid and coefficient information). The values of all other variables are reinitialized with each run, either from the input files or a restart file when used.

6.1.3 Defaults

Default values are set during the initialization process if overriding input is not provided by the user.

6.2 Individual Input Records or Parameters

Other than the information provided through the control file or terminal I/O and the multiple realization simulations file, the main user input is provided using macro control statements in the input file, geometry data file, zone data file, and optional input files. Data provided in the input files is entered in free format with the exception of the macro control statements and keywords which must appear in the first four (or more) columns of a line. Data values may be separated with spaces, commas, or tabs. The primary input file differs from the others in that it begins with a title line (80 characters maximum) followed by input in the form of the macro commands. Each file containing multiple macro commands should be terminated with the **stop** control statement. In the examples provided in the following subsections, blank input lines are depicted with shading.

6.2.1 Control File or Terminal I/O Input

The parameters enumerated below, are entered in order one per line in the control file (excluding the control file name [nmfile(1)]) or in response to a prompt during terminal input. If there is a control file with the name *fehmn.files* in your local space (current working directory), FEHM will execute using that control file and there will be no prompts. If another name is used for the control file, it can be entered at the first prompt.

A blank line can be entered in the control file for any auxiliary files not required, for the “none” option for tty output, and for the “0” option for the user subroutine number. The code will always write an input check file and a restart file, so if names are not provided by the user the defaults will be used. If an output file name is not specified, the generalized output is written to the terminal. File names that do not include a directory or subdirectory name, will be located in the current working directory. The data files are described in more detail in Section 5.0.

Input Variable	Format	Opt/Req	Default	Description
nmfil(1)	character*100	Opt	fehmn.files	Control file name (this line is not included in the control file)
nmfil(2)	character*100	Req	fehmn.dat	Main input file name
nmfil(3)	character*100	Opt	not used	Geometry data input file name
nmfil(4)	character*100	Opt	not used	Zone data input file name
nmfil(5)	character*100	Opt	terminal	Main output file name
nmfil(6)	character*100	Opt	not used	Restart input file name
nmfil(7)	character*100	Opt	fehmn.fin	Restart output file name
nmfil(8)	character*100	Opt	not used	Simulation history output file name
nmfil(9)	character*100	Opt	not used	Solute history output file name
nmfil(10)	character*100	Opt	not used	Contour plot output file name. (Required if using avs option in cont macro.)

Input Variable	Format	Opt/Req	Default	Description
nmfil(11)	character*100	Opt	not used	Dual porosity or double porosity / double permeability contour plot output file name
nmfil(12)	character*100	Opt	not used	Coefficient storage file name
nmfil(13)	character*100	Opt	fehmn.chk	Input check output file name
tty_flag	character*4	Opt	none	Terminal output flag: all, some, none
usub_num	integer	Opt	0	User subroutine call number

The following are two examples of input control files. In the example on the left, all input will be found in the current working directory and output files will also be written to that directory. The blank lines indicate that there is no restart initialization file, the restart output file will use the default file name (*fehmn.fin*), a dual porosity contour plot file is not required, and the coefficient storage file is not used. The “some” keyword indicates that selected information is output to the terminal. The ending “0” indicates that the user subroutine will not be called. In the example on the right, input will be found in the “groupdir” directory, while output will be written to the current working directory. The “none” keyword indicates that no information should be written to the terminal.

Files “*fehmn.files*”:

```
tape5.dat
tape5.dat
tape5.dat
tape5.out

tape5.his
tape5.trc
tape 5.con

tape5.chk
some
0
```

```
/groupdir/c14-3
/groupdir/grid-402
/groupdir/c14-3
c14-3.out
/groupdir/c14-3.ini
c14-3.fin
c14-3.his
c14-3.trc
c14-3.con
c14-3.dp
c14-3.stor
c14-3.chk
none
0
```

6.2.2 Multiple Realization Simulations Input

The multiple realization simulations input file (*fehmn.msim*) contains the number of simulations to be performed and, on UNIX systems, instructions for pre- and post-processing input and output data during a multiple realization simulation. The file uses the following input format:

```
Line 1      nsim
Lines 2-N   single_line
```

In the following example, 100 simulations are performed with pre and post-processing steps carried out. The first line after the number of simulations demonstrates how the current and total number of simulations can be

Input Variable	Format	Description
nsim	integer	Number of simulation realizations to be performed
single_line	character*80	An arbitrary number of lines of UNIX shell script style instructions: lines 2-n: lines which are written to a file called <i>fehmn.pre</i> , which is invoked as a shell script (UNIX systems) that is performed before each realization using the following command: <code>sh fehmn.pre \$1 \$2</code> line n+1: the keyword 'post', placed in the first four columns of the input file, denotes that the previous line is the last line in the <i>fehmn.pre</i> script, and that the data for the post-processing script <i>fehmn.post</i> follows lines n+2 to N: lines which are written to a file called <i>fehmn.post</i> , which is invoked as a shell script (UNIX systems) that is performed after each realization using the following command: <code>sh fehmn.post \$1 \$2</code> Thus, the files <i>fehmn.pre</i> and <i>fehmn.post</i> are created by the code and are meant to provide the capability to perform complex pre- and post-processing steps during a multiple realization simulation. Script arguments \$1 and \$2 represent the current simulation number and nsim, the total number of simulations, respectively.

accessed in the *fehmn.pre* shell script. This line will write the following output for the first realization:

This is run number 1 of 100

The pre-processing steps in this example are to remove the *fehmn.files* file from the working directory, copy a control file to *fehmn.files*, copy a particle tracking macro input file to a commonly named file called *ptrk.input*, and write a message to the screen. The *fehmn.files* files should be used or else the code will require screen input of the control file name for every realization. One hundred particle tracking input files would have been generated previously, and would have the names *ptrk.np1*, *ptrk.np2*, . . . , *ptrk.np100*. Presumably, these files would all have different transport parameters, resulting in 100 different transport realizations. The post-processing steps involve executing a post-processor program for the results (*process1_fuj*). This post-processor code generates an output file called *results.output*, which the script changes to *np1.output*, *np2.output*, . . . , *np100.output*, for further processing after the simulation.

One other point to note is that the variable "curnum" in this example is defined twice, once in the pre-processor and again in the post-processor. This is necessary because *fehmn.pre* and *fehmn.post* are distinct shell scripts that do not communicate with one another.

File "*fehmn.msim*".

```
100
echo This is run number $1 of $2
rm fehmn.files
curnum=`expr $1`
cp control.run fehmn.files
rm ptrk.input
cp ptrk.np$curnum ptrk.input
echo starting up the run
post
curnum=`expr $1`
/home/robinson/fehmn/chun_li/ptrk/process1_fuj
rm np$curnum.output
mv results.output np$curnum.output
echo finishing the run again
```

6.2.3 Transfer function curve data input file

In the FEHM particle tracking models, diffusion of solute from primary porosity into the porous matrix is captured using an upscaling procedure designed to capture the small scale behavior within the field scale model. The method is to impart a delay to each particle relative to that of a nondiffusing solute. Each particle delay is computed probabilistically through the use of transfer functions. A transfer function represents the solution to the transport of an idealized system with matrix diffusion at the sub-grid-block scale. After setting up the idealized model geometry of the matrix diffusion system, a model curve for the cumulative distribution of travel times through the small-scale model is computed, either from an analytical or numerical solution. Then, this probability distribution is used to determine, for each particle passing through a given large-scale grid block, the travel time of a given particle. Sampling from the distribution computed from the small scale model ensures that when a large number of particles pass through a cell, the desired distribution of travel times through the model is reproduced. In FEHM, there are equivalent continuum and dual permeability formulations for the model, each of which call for a different set of sub-grid-block transfer function curves. These curves are numerical input to the FEHM, with a data structure described below. Optional input in macros **mptr**, **ptrk**, and **sptr** is used to tell the code when transfer function curves are required and whether 2 or 3 (numparams) parameter curves are to be used.

Input Variable	Format	Description
DUMMY_STRING	character*4	If keyword "free" is input at the beginning of the file, the code assumes an irregular distribution of transfer function parameters and performs either a nearest neighbor search to choose the transfer function curve, or a more computationally intensive multi-curve interpolation.
NCURVES	integer	Total number of transfer function curves input in the file when keyword "free" is used.

Input Variable	Format	Description
LOG_FLAG	integer	Array of flags to denote whether to take the natural log of the parameter before determining the distance from the input parameter to the values for each transfer function when keyword "free" is used. If 0, use the input value, if 1, use $\ln(\text{value})$.
NORMAL_PARAM	real	Array of values for normalizing the parameters before performing the nearest neighbor or multi-curve interpolation. Each value is divided by $\text{NORMAL_PARAM}(i)$ if LOG_FLAG is 0, and $\ln(\text{LOG_FLAG})$ if LOG_FLAG is 1.
CURVE_STRUCTURE	integer	Flag to denote the type of interpolation to be performed when keyword "free" is used. If 1, simple nearest neighbor search, if > 1 , a multi-curve interpolation is performed with <i>curve_structure</i> points nearest to the input values of the parameters. It is recommended that values no greater than 4 be used.
WEIGHT_FACTOR	real	Optional weight factor used when $\text{CURVE_STRUCTURE} > 1$. The default value is $1e-3$. When determining the interpolated value of time using a multi-curve interpolation there are occasions where the algorithm yields large values of the weights used to compute the particle residence time. In a few such cases numerical errors can make the scheme fail so that the interpolated values for time erroneously get very large. This occurs when the sum of the weights divided by any individual weight is small, that is, large weights of opposite sign cancelling one another out. To prevent this error in the scheme from affecting the results, the code reverts to a nearest neighbor approach to obtain the time. The criterion for this option is that the sum of the weights divided by any individual weight is less than <i>weight_factor</i> . Increasing this value to $1.e-2$ or higher can eliminate such occurrences. This parameter is very problem dependent, so this parameter is included for flexibility. It is recommended that the default of $1.e-3$ or a higher value of $1.e-2$ or so be used.
NUMP1	integer	Number of parameter 1 values used to define transfer function curves.
PARAM1	real	<i>nump1</i> parameter 1 values defining transfer function curves.
NUMP2	integer	Number of parameter 2 values used to define transfer function curves.
PARAM2	real	<i>nump2</i> parameter 2 values defining transfer function curves.
NUMP3	integer	Number of parameter 3 values used to define transfer function curves.
PARAM3	real	<i>nump3</i> parameter 3 values defining transfer function curves.

Input Variable	Format	Description
D4	integer	Fracture-matrix flow interaction flag ($d4 = 1, 4$). For the three-parameter option, the dual permeability model requires four transfer function curves for each set of parameters. Interactions can occur from fracture-fracture ($d4=1$), fracture-matrix ($d4=2$), matrix-fracture ($d4=3$), and matrix-matrix ($d4=4$).
NUMP_MAX	integer	Maximum number of delay time and concentration values for transfer function curves.
NUMP	integer	Number of delay time and concentration values in each transfer function curve ($nump1, nump2, nump3, d4$).
DTIME	real	Transfer function curve delay times ($nump1, nump2, nump3, d4, nump$).
CONC	real	Transfer function curve concentrations ($nump1, nump2, nump3, d4, nump$).
OUTPUT_FLAG	character*3	If optional keyword "out" is entered at the end of the file the code outputs information on the parameter space encountered during the simulation in the *.out file. See " mptr " macro for further discussion of the output option.

The transfer function curve data file uses the following format if a regular grid of parameters is input. Please note that parameter values for this format should be input from smallest to largest value:

```

nump1
param1 (i), i = 1 to nump1
nump2
param2 (j), j = 1 to nump2
If 2 parameter curves are being input
  nump_max
  For each i, j (nump3 = 1, d4 = 1)
    nump(i, j, 1, 1), param1(i), param2(j)
    followed by for each nump(i, j, 1, 1)
      time(i, j, 1, 1, nump), conc(i, j, 1, 1, nump)
Or if 3 parameter curves are being input
  nump3
  param3(k), k = 1 to nump3
  nump_max
  For each d4, i, j, k
    nump(i, j, k, d4), param1(i), param2(j), param3(k), d4
    followed by for each nump(i, j, k, d4)
      time(i, j, k, d4, nump), conc(i, j, k, d4, nump)
out_flag (optional) - keyword "out"

```

The transfer function curve data file uses the following format for the case in which the transfer functions are input without a regular grid of parameters:

```

dummy_flag - keyword "free"
log_flag(i), i = 1 to numparams
normal_param(i), i = 1 to numparams
curve_structure, weight_factor (optional)
ncurves
nump_max
For "free" form input of transfer function curves (nump1 = ncurves,
nump2 = 1, and nump3 = 1)
If 2 parameter curves are being input
  For each i = 1 to ncurve (d4 = 1)
    nump(i, 1, 1, 1), param1(i), param2(i)
    followed by for each nump(i, 1, 1, 1)
      time(i, 1, 1, 1, nump), conc(i, 1, 1, 1, nump)
Or if 3 parameter curves are being input
  For each d4 = 1 to 4, i = 1 to ncurve
    nump(i, 1, 1, d4), param1(i), param2(i), param3(i), d4
    followed by for each nump(i, 1, 1, d4)
      time(i, 1, 1, d4, nump), conc(i, 1, 1, d4, nump)
out_flag (optional) - keyword "out"

```

Please note that all fracture-fracture curves are input followed by fracture-matrix curves, followed by matrix-fracture curves, followed by matrix-matrix curves.

6.2.4 Optional Input Files

The data for any of the FEHM macros (with the exception of **coor** and **elem**, where use of a separate geometry input file is handled through control file input) may be entered in an alternate input file. To use this option the keyword 'file' must appear on the input line immediately following the control statement (macro name). The line immediately following this keyword will contain the name of the alternate input file. The contents of the alternate input file consist of the regular full macro description: the macro name followed by the data. Note that data from only one macro may be entered per auxiliary file. The entries in the optional input file may be preceded or followed by comments using the "#" designator (see discussion on page 30). As with regular macro input, comments may not be embedded within the data block.

Group 1 - LOCKEYWORD

Group 2 - LOCFILENAME

Input Variable	Format	Description
LOCKEYWORD	character*4	Keyword 'file' to designate an auxiliary input file is used.
LOCFILENAME	character*100	Name of the optional data input file.

The following illustrate the use of an optional input file and its contents. In this example, optional file "*rockfile*" is located in the current working

directory. Input for macro “rock” is described in Section 6.2.58 on page 104.

```
rock
file
rockfile
```

File “rockfile”:

```
# Auxiliary file used for rock macro input
rock
  1      140      1      2563.      1010.      0.3500
# End of rock macro input
```

6.2.5 Control statement adif (optional)

Air-water vapor diffusion. The air-water diffusion equation is given as Equation (21) of the “Models and Methods Summary” of the FEHM Application (Zyvoloski et al. 1999).

Group 1- TORT

Input Variable	Format	Description
TORT	real	Tortuosity for air-water vapor diffusion. If TORT > 0, τ of eqn 21, otherwise If TORT < 0, $abs(\tau\phi S_v)$ of the same equation.

The following is an example of **adif**. In this example the tortuosity (τ) for vapor diffusion is specified to be 0.8.

```
adif
  0.8
```

6.2.6 Control statement airwater (optional)

Isothermal air-water two-phase simulation.

Group 1 - ICO2D

Group 2 - TREF, PREF

Input Variable	Format	Description
ICO2D	integer	Determines the type of air module used. ICO2D = 1, 1 degree of freedom solution to the saturated-unsaturated problem is produced. This formulation is similar to the Richard’s Equation. ICO2D = 2, 1 degree of freedom solution is obtained assuming only gas flow with no liquid present. ICO2D = 3, full 2 degree of freedom solution. All other values are ignored. The default is 3.

Input Variable	Format	Description
TREF	real	Reference temperature for properties (°C).
PREF	real	Reference pressure for properties (MPa).

Several macros are affected if the air module is enabled. These are

- pres** - Because the air-water formulation is 2-phase at all times, care should be taken to insure that IEOSD is always specified to be 2. Likewise, saturations (not temperatures) are used.
- init** - This macro should not be used because the saturation values cannot be specified.
- flow** - A variety of different flow and boundary values are input with this macro when the macro **airwater** is also used. See description of control statement **flow**.

The following is an example of **airwater**. In this example, a full 2-degrees-of-freedom solution is specified with a reference temperature for property evaluation of 20 °C and a reference pressure of 0.1 MPa.

airwater
3
20. 0.1

6.2.7 Control statement **boun** (either **boun** or **flow** is required for a flow problem)

Implement boundary conditions and sources or sinks. Input may be time dependent and cyclic. Time step sizes may also be adjusted.

Group 1 - KEYWORD

The Group 1 KEYWORD 'model', which starts each model sequence, is followed immediately by a Group 2 KEYWORD of 'ti' or 'cy'.

Group 2 - KEYWORD

Group 3 - NTIMES, TIME(I), I=1,NTIMES

The Group 4 KEYWORDS define the various boundary condition parameters being entered. These KEYWORDS and associated data, Group 5, are repeated as needed for each model.

Group 4 - KEYWORD

Group 5 - VARIABLE(I), I=1,NTIMES

Additional models are entered by beginning again with Group 1. The MODEL_NUMBER is incremented each time a new model is read, and is used to assign boundary conditions to specified nodes or zones in Group 6.. After all models have been entered, the section is terminated with KEYWORD 'end' or a blank line.

Group 6 -JA, JB, JC, MODEL_NUMBER (JA, JB, JC-defined on page 31)

The following is an example of **boun**. In this example two models are defined. The first model cyclically changes the water source in a 1.e05 day cycle, i.e., the 'cy' keyword entry shows that the time cycle ends at 1.e05

Input Variable	Format	Description
KEYWORD	character*4	<p>Keyword specifying a model designation, time for boundary condition or source/sink changes, or actual variable or source change. Keywords, which must be entered starting in the first column, are:</p> <p>model- new model definition to follow</p> <p>Note: Descriptive text, such as the model number, may be appended after the 'model' keyword as long as it is contained on a single line, and begins after column four.</p> <p>ti - time sequence for changes to follow (days)</p> <p>cy - cyclic time sequence for changes to follow (days)</p> <p>Note: The keywords, 'ti' and 'cy', require the time to start at 0.0. This provides the initial boundary and source/sink information. If the input for 'ti' or 'cy' does not start at 0.0 time the code assumes boundary conditions and source/sinks are 0.0 at time 0.0. The 'cy' keyword involves a cyclic changing of conditions. In our procedure the cycle ends at the last specified time. Thus the code reverts to the first specified time values. Because of this, the boundary conditions and source/sinks for the last time change are always set to the first time values.</p> <p>sa - air source sequence for changes to follow (kg/s)</p> <p>sw - water source sequence for changes to follow (kg/s)</p> <p>swf - source water factor sequence for changes to follow (multiplier for existing flux)</p> <p>se - enthalpy source sequence for changes to follow (MW)</p> <p>sf - water seepage face sequence with pressures for changes to follow (MPa)</p> <p>sfh - water seepage face sequence with heads for changes to follow (m)</p> <p>fd - water drainage area sequence for changes to follow (m²)</p> <p>dsa - distributed air source sequence for changes to follow (kg/s)</p> <p>dsw - distributed water source sequence for changes to follow (kg/s)</p> <p>dse - distributed enthalpy source sequence for changes to follow (MW)</p> <p>Note: A distributed source (keywords 'dsa', 'dsw', and 'dse') is a source term divided over a group of nodes or a zone proportional to the nodal volume.</p> <p>s - fixed saturation sequence for changes to follow</p> <p>hd - fixed hydraulic head sequence for changes to follow (m)</p> <p>pw - fixed water pressure sequence for changes to follow (MPa)</p> <p>pa - fixed air pressure sequence for changes to follow (MPa)</p> <p>hdo - fixed hydraulic head sequence for changes to follow (m) (constrained to outflow only)</p> <p>pwo - fixed water pressure sequence for changes to follow (MPa) (constrained to outflow only)</p> <p>pao - fixed air pressure sequence for changes to follow (MPa) (constrained to outflow only)</p>

Input Variable	Format	Description
		en - fixed enthalpy sequence for changes to follow (MW) t - fixed temperature sequence for changes to follow (°C) h - fixed humidity sequence for changes to follow (must be used with van Genuchten relative permeability model) ft - fixed flowing temperature sequence for change to follow (°C). By flowing temperature we mean the temperature of the inflow stream for a specified source. If no influx source occurs where this condition is applied, it will be ignored. if - impedance factor for use with fixed water pressure boundary condition. If left out the impedance factor will be set to the volume of the grid cell. si - initial value saturation sequence for changes to follow pai - initial value air pressure sequence for changes to follow (MPa) pwi - initial value water pressure sequence for changes to follow (MPa) tmi - initial value temperature sequence for changes to follow (°C) Note: The keywords 'si', 'pai', 'pwi', and 'tmi' refer to changes for a variable that is NOT fixed. They are similar to specifying initial conditions in that regard but may be changed according to a time sequence. At present these 4 keywords only work with isothermal air-water calculations. chmo - model number sequence for changes to follow ts - timestep sequence for changes to follow (days) end - signifies end of keyword input, a blank line will also work.
NTIMES	integer	Number of time changes for boundary condition or source/sink specification.
TIME	real	NTIMES times for changes in boundary conditions or source/sinks.
VARIABLE	real	NTIMES new values for boundary conditions or source/sinks.
MODEL_NUMBER	integer	Boundary condition model to be associated with designated nodes or zones (the number corresponds to the numerical order in which the models were input, i.e., beginning with KEYWORD 'model')

days and at this time the cycle reverts to 0.0 days. Note that the water source at 1.e05 days equals that at 0.0 days. Also in model 1 the flowing temperature was alternated between 20°C and 50°C. The second model uses a time change that occurs at 1.e20 days. This effectively removes any time variance from model 2. Model 2 has a fixed water pressure and flowing temperature condition. The models are applied at nodes 26 and 27 in the last two lines. It should be noted that the model numbers included in the example (following KEYWORD 'model') are not part of the required input but are descriptive text used to enhance readability of the macro.

boun				
model 1				
cy				
4	0.0	1.e1	1.e2	1.e5
sw				
-1.e-4	-1.e-5	-1.e-3	-1.e-4	
ft				
20.0	50.0	50.0	20.0	
model 2				
ti				
2	0.0	1.e20		
pw				
0.1	0.1			
ft				
20.0	20.0			
end				
26	26	1	1	
27	27	1	2	

6.2.8 Control statement bous (optional)

Constant density and viscosity are used for the flow terms (Boussinesq approximation). NOTE: where the **bous** macro is used, the gravity term in the air phase is set to zero.

Group 1 - ICONS

Input Variable	Format	Description
ICONS	integer	Parameter to enable constant density and viscosity for flow terms ICONS ≠ 0 enabled. ICONS = 0 disabled (default).

The following is an example of **bous**. In this example the Boussinesq approximation is enabled.

```
bous
1
```

6.2.9 Control statement cden

Use concentration-dependent density for flow.

The following restrictions apply to the use of this macro: 1) It cannot be used with the macro "head", which assumes constant fluid density; 2) The updating of density is explicit, based on the concentration values at the previous time step. Therefore, accuracy of the solution must be tested by using a smaller time step and ensuring that the results have converged; 3) The fluid flow time steps should be small enough that only one or two solute time steps are carried out before the next fluid time step, because relatively small changes in the concentration field are required for accuracy; and 4) The heat and mass transfer solution must be kept on during the entire simulation for the results to be meaningful (see macro **trac**).

Group 1 - ISPCDEN

Group 2 - FACTCDEN

Input Variable	Format	Description
ISPCDEN	integer	The number of the chemical component in trac that is used for applying the concentration-dependent density.
FACTCDEN	real	The factor used in the following relationship for fluid density (kg/m^3): $density = density_water + FACTCDEN * C$ where $density_water$ = the density of pure water (kg/m^3), and C is the concentration of chemical component ISPCDEN

The following is an example of **cden**. In this example, component number 1 in **trac** is used. For concentrations of order 1, the density correction would be 100, of order 10% of the nominal value of water density of 1000 kg/m^3 .

```

cden
  1
100.

```

6.2.10 Control statement **cond** (required for non-isothermal problem)

Assign thermal conductivities of the rock.

Group 1 - JA, JB, JC, THXD, THYD, THZD (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
THXD	real	1.e-30	Thermal conductivity in the x-direction ($\frac{W}{m \cdot K}$).
THYD	real	1.e-30	Thermal conductivity in the y-direction ($\frac{W}{m \cdot K}$).
THZD	real	1.e-30	Thermal conductivity in the z-direction ($\frac{W}{m \cdot K}$).

The following is an example of **cond**. In this example all the nodes numbered 1 through 140 have thermal conductivities of $1 \left(\frac{W}{m \cdot K} \right)$ in the X and Y directions, and 0 in the Z direction.

```

cond
  1      140      1      1.00e-00      1.00e-00      0.00e-00

```

6.2.11 Control statement **cont** (optional)

Contour data output format, output timestep intervals, and time intervals.

Group 1 - NCNTR, CONTIM

An alternative form of input for macro **cont** is possible. This is

Group 1 - ALTC, NCNTR, CONTIM, KEYWORD

Group 2 - CHDUM (only input if ALTC is 'avs' or 'avsx')

-or-

Group 2 - NSURF, IZONE_ISURF(I), I=1, NSURF (only input if ALTC is 'sur' or 'tec')

Input Variable	Format	Description
ALTC	character*4	Keyword specifying the type of contour output wanted (avs, avsx, fehm, free, ment, ptrn): 'avs' produces contour plot files compatible with the AVS postprocessor. 'avsx' produces contour plot files compatible with the AVS Express postprocessor. 'fehm' produces a binary output file. The same contour plot file is produced using the first form of Group1 input. 'free' produces a free format contour plot file. 'ment' produces a contour plot file compatible with the MENTAT postprocessor. 'ptrn' produces a contour plot file compatible with the PATRAN postprocessor. 'sur' produces a contour plot file compatible with the SURFER postprocessor. 'tec' produces a contour plot file compatible with the TECPLOT postprocessor.
NCNTR	integer	<u>Time step</u> interval for contour plots (number of timesteps). Output contour information each NCNTR timesteps.
CONTIM	real	<u>Time</u> interval for contour plots (days). In addition to output each NCNTR timesteps, output contour information each CONTIM days.
KEYWORD	character*4	Optional keyword 'time', use time (days) in file name instead of number (used when <i>altc</i> is "avs" or "avsx" or "sur" or "tec")
NSURF	integer	Number of output zones used when <i>altc</i> is "sur" or "tec"
IZONE_SURF	integer	List of <i>nsurf</i> zone numbers used when <i>altc</i> is "sur" or "tec"
CHDUM	character*72	Keyword specifying type of AVS contour plot data files to be created in AVS UCD or AVS Express format. Keywords are entered one per line and terminated with 'endavs'. Valid keywords (case insensitive) are: (m)aterial - output contour values for material properties. (l)iquid - output contour values for liquid phase. (v)aapor - output contour values for vapor phase. (v)e)locity - output velocity values. (d)p)dp - output contour values for dual permeability nodes. (p)ressure - output pressure values. (h)ead - output head values.

Input Variable	Format	Description
		(t)emperature - output temperature values. (s)aturation - output saturation values. (c)oncentration - output solute concentration values. (f)ormatted - output data in ASCII format. (e)ndavs - last keyword entered. If a format keyword is not entered, the default is 'formatted'. In the current version of the code this is the only format option supported. The default for data keywords is "off" (no output). The letters given in () are sufficient to identify the keyword.

FEHM will automatically distinguish between the alternative input formats. When keywords are used they must be entered starting in the first column. The contour data will be output whenever either of the interval criteria are satisfied.

For AVS output, if the *material* keyword is selected, the following material property values will be written for each node: permeability in the x, y, and z directions, thermal conductivity in the x, y, and z directions, porosity, rock specific heat, capillary pressure, relative permeability model being used, and capillary pressure model being used. If *vapor* and/or *liquid* are selected, *pressure* or *velocity* must also be defined (otherwise, no data for these values will be written). *velocity* will result in vector values, *pressure* values will be scalar. If *concentration* is selected, values will be output only if *nspeci* is defined for tracer solutions. See the control statement **trac** for a description of *nspeci* for solutes.

The following are examples of **cont**. For the first example, FEHM binary format contour output files will be written every 100 timesteps and for each 1.e20 days. The second example invokes AVS contour output. AVS UCD formatted files will be written for every 100 time steps and 1.e20 days. The resulting files will include a log file, geometry file, plus header and data files for the following: material properties, solute concentrations, liquid velocities, pressures and temperatures.

cont		
100		1.e20

cont		
avs	100	1.e20
mat		
con		
liquid		
velocity		
pressure		
temp		
formatted		
endavs		

6.2.12 Control statement **coor** (required if macro **fdm** not used)

Node coordinate data. These data are usually created by a mesh generation program, then cut and copied into the input file or a separate geometry data input file. The mesh must be a right handed coordinate system. Note that X, Y, and Z coordinates must be entered even if a problem is not three-dimensional.

Group 1 - N

Group 2 - MB, CORD1, CORD2, CORD3

To end the control section a blank line is entered.

Input Variable	Format	Description
N	integer	Number of nodes in the grid
MB	integer	Node number. If MB < 0 then the difference between the absolute value of MB and the previously read absolute value of MB is used to generate intermediate values by interpolation.
CORD1	real	X-coordinate of node MB (m).
CORD2	real	Y-coordinate of node MB (m).
CORD3	real	Z-coordinate of node MB (m).

The following is an example of **coor**. In this example, there are 140 nodes in the grid. Node number 1 has X, Y, Z coordinates of 0., 200., and 0. meters respectively, node 2 has X, Y, Z coordinates of 12.5, 200., and 0. meters respectively, and so forth, with node number 140 having X, Y, Z coordinates of 300., 0., and 0. meters respectively.

coor			
140			
1	0.00000	200.00000	0.00000
2	12.50000	200.00000	0.00000
.	.	.	.
.	.	.	.
.	.	.	.
10	212.50000	200.00000	0.00000
.	.	.	.
.	.	.	.
.	.	.	.
140	300.00000	0.00000	0.00000

6.2.13 Control statement ctrl (required)

Assign various control parameters needed for equation solvers and matrix solver routines. Suggested values for the control parameters are shown in “{ }” in the table. For older input files where MAXSOLVE and ACCM were not input, the default is ACCM = gmre and MAXSOLVE = 3*NORTH.

Group 1 - MAXIT, EPM, NORTH, MAXSOLVE, ACCM

Group 2 - JA, JB, JC, NAR (JA, JB, JC - defined on page 31)

Group 3 - AAW, AGRV, UPWGT

Group 4 - IAMM, AIAA, DAYMIN, DAYMAX

Group 5 - ICNL, LDA

Input Variable	Format	Default	Description
MAXIT	integer		Maximum number of iterations allowed in either the overall Newton cycle or the inner cycle to solve for the corrections at each iteration. If MAXIT < 0 then the maximum number of iterations is ABS(MAXIT) but the minimum number of iterations is set to 2. {10}
EPM	real		Tolerance for Newton cycle (nonlinear equation tolerance). {1.e-5}
NORTH	integer		Number of orthogonalizations in the linear equation solver. {8 for gmre, 1 for bcgs}
MAXSOLVE	integer		Maximum number of solver iterations allowed
ACCM	character*4		Acceleration method for solve bcgs - Biconjugate gradient stabilized acceleration gmre - Generalized minimum residual acceleration
NAR	integer	1	The order of partial Gauss elimination {1 or 2 is recommended}. Larger values increase memory utilization but may be necessary for convergence.
AAW	real		Implicitness factor. {1} AAW ≤ 1, use standard pure implicit formulation. AAW > 1, use second-order implicit method.
AGRAV	integer		Direction of gravity AGRAV = 0, no gravity is used. AGRAV = 1, X-direction. AGRAV = 2, Y-direction. AGRAV = 3, Z-direction. A value for gravity of 9.81 m/s ² is used in the code when AGRAV ≠ 0. If AGRAV > 3, AGRAV is set equal to 3.
UPWGT	real		Value of upstream weighting {0.5 ≤ UPWGT ≤ 1.0}. If UPWGT < 0.5, UPWGT is set to 0.5 If UPWGT > 1.0, UPWGT is set to 1.0
IAMM	integer		Maximum number of iterations for which the code will multiply the time step size. If this number of time steps is exceeded at any time, the time step will not be increased for the next time {7-10}.
AIAA	real	1	Time step multiplier {1.2-2.0}
DAYMIN	real	1.0e-05	Minimum time step size (days)
DAYMAX	real	30.0	Maximum time step size (days)

Input Variable	Format	Default	Description
ICNL	integer		Parameter that specifies the geometry ICNL = 0, three-dimensional ICNL = 1, X - Y plane ICNL = 2, X - Z plane ICNL = 3, Y - Z plane ICNL = 4, X - Y radial plane, (radius is X) ICNL = 5, X - Z radial plane, (radius is X) ICNL = 6, Y - Z radial plane, (radius is Y)
LDA	integer	0	Parameter that specifies the external storage of geometric coefficients LDA = +2, element coefficients are read, unformatted, from file <i>filen.stor</i> and no coefficients are calculated in the code LDA = +1, element coefficients are read from file <i>filen.stor</i> and no coefficients are calculated in the code LDA = 0, element coefficients are calculated in the code and not saved LDA = -1, element coefficients are calculated in the code and saved on file <i>filen.stor</i> LDA = -2, element coefficients are calculated in the code and saved, unformatted, on file <i>filen.stor</i> It should be noted that if the coefficients are read from a file (LDA = 1) then the macro finv is ignored as well as information read from macros elem and coor .

The following is an example of **ctrl**. In this example, the maximum number of iterations allowed is 40, tolerance for solving the nonlinear equations using Newton iterations is 1.e-7, and the number of orthogonalizations in the linear equation solver is 8. The order of partial Gauss elimination for all nodes 1 through 140 is 1. A forward implicit formulation is used for the time derivative, there is no gravity, and full upstream weighting is used. The number of iterations for which the time step is increased is 40, the time step is increased by a factor of 1.2 at each iteration, the minimum time step size is 0.1 days, and the maximum time step size is 60 days. The geometry of the problem is 2-dimensional in the X-Y plane and the finite element coefficients are calculated during the run and not saved.

ctrl				
40	1.e-7	8	24	gmre
1	140	1	1	
1.0	0.0	1.0		
40	1.2	0.1	60.0	
1	0			

6.2.14 Control statement **dppd** (optional)

Double porosity / double permeability formulation. There are two sets of parameter values at any nodal position, for which property values must be defined. Nodes 1 to N (see macro **coor** for definition of N) represent the fracture nodes and nodes N + 1 to 2N the matrix material. When zones are

used with the **dpdp** macro, additional zones are automatically generated. See instructions for the macro **zone** for a more detailed description. The **dpdp** parameters are only defined for the first N nodes.

Group 1 - IDPDP

Group 2 - JA, JB, JC, VOLFD1 (JA, JB, JC - defined on page 31)

Group 3 - JA, JB, JC, APUV1 (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
IDPDP	integer		Solution descriptor for double porosity/double permeability solution. IDPDP = 0, information is read but not used. IDPDP ≠ 0, dpdp solution is implemented.
VOLFD1	real	1.	Volume fraction for fracture node.
APUV1	real	10.	Length scale for matrix nodes (m).

The volume fraction VOLFD1 is related to the total volume by

$$VOLFD1 + VOLFD2 = 1.0$$

where VOLFD2 is the volume fraction of the matrix node. If permeability model IRLP = 4 is selected in control statement **rlp**, VOLFD1 is calculated from RP15 (fracture porosity) in that control statement.

The following is an example of **dpdp**. In this example, the dual porosity/permeability solution is implemented for all nodes from 1 through 140. The fractional volume in the fractures (compared to the total volume) is 0.005 and the length scale for matrix nodes is 0.1 m.

dpdp			
1			
1	140	1	0.005
1	140	1	0.10

6.2.15 Control statement dual (optional)

Dual porosity formulation. There are three sets of parameter values at any nodal position, for which property values **must** be defined. Nodes 1 to N (see macro **coor** for definition of N) represent the fracture nodes, nodes N + 1 to 2N the first matrix material, and nodes 2N + 1 to 3N the second matrix material. When zones are used with the **dual** macro, additional zones are automatically generated. See instructions for the macro **zone** for a more detailed description. The **dual** parameters are only defined for the first N nodes.

Group 1 - IDUALP

Group 2 - JA, JB, JC, VOLFD1 (JA, JB, JC - defined on page 31)

Group 3 - JA, JB, JC, VOLFD2 (JA, JB, JC - defined on page 31)

Group 4 - JA, JB, JC, APUVD (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
IDUALP	integer		Solution descriptor for dual porosity solution. IDUALP = 0, information is read but not used IDUALP ≠ 0, dual porosity solution is implemented For the special case of IDUALP = 2, the permeabilities and conductivities are scaled by the volume fraction, i.e., $k = vf * k$.
VOLFD1	real	0.001	Volume fraction for fracture portion of the continuum.
VOLFD2	real	0.5	Volume fraction for the first matrix portion of the continuum.
APUVD	real	5.	Length scale for the matrix nodes (m).

The volume fractions VOLFD1 and VOLFD2 are related to the total volume by

$$VOLFD1 + VOLFD2 + VOLFD3 = 1.0$$

where VOLFD3 is the volume fraction of the second matrix node. If permeability model IRLP = 4 is selected in control statement **rlp**, VOLFD1 is calculated from RP15 (fracture porosity) in that control statement.

The following is an example of **dual**. In this example, the dual porosity solution is implemented for all nodes from 1 through 140. The volume fraction for the fracture is 0.006711409, the volume fraction for the first matrix portion is 0.335570470, and the length scale for the matrix nodes is 0.1 m.

dual				
1				
1	140	1		0.006711409
1	140	1		0.335570470
1	140	1		0.10

6.2.16 Control statement **dvel** (optional)

Velocity between two nodes is output by choosing this control statement. The input for this macro is identical to macro **flxo**, except that velocities instead of fluxes are calculated (see page 62). In the following example of **dvel**, a single internode velocity is calculated between nodes 101 and 102.

dvel		
1		
101	102	

6.2.17 Control statement **elem** (required if macro **fdm** not used).

Element connectivity data. These data are usually created by a mesh generation program, then cut and copied into the input file or a separate geometry data input file.

Group 1 - NS, NEI

Group 2 - MB, NELM (1), NELM (2), . . . , NELM (NS)

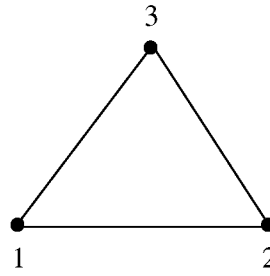
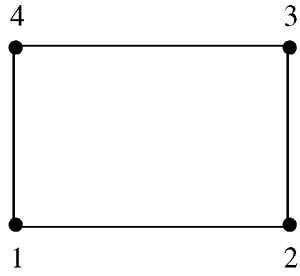
IF $NS < 0$ then ABS(NS) is interpreted as the number of nodes per element. $NS < 0$ signals the code to make rectangles (or bricks in three dimensions) a sum of triangles (or tetrahedrals). This provides more stability in nonlinear problems with a distorted mesh. Figure 2 shows available element types and the nodal numbering convention. To end the control section a blank line is entered.

Input Variable	Format	Description
NS	integer	Number of nodes per element.
NEI	integer	Number of elements
MB	integer	Element number. If MB < 0 then the difference between the absolute value of MB and the previous absolute value of MB is used to generate intermediate values by interpolation in the code.
NELM (1)	integer	First node of element MB
NELM (2)	integer	Second node of element MB
⋮	⋮	⋮
NELM (NS)	integer	Last node of element MB

The following is an example of **elem**. In this example there are 4 nodes per element, i.e., the elements are 2-dimensional quadrilaterals,. There are a total of 117 elements in the model, element number 1 is defined by nodes 15, 16, 2, and 1, element number 2 is defined by nodes 16, 17, 3 and 2, and so on.

elem				
4	117			
1	15	16	2	1
2	16	17	3	2
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
10	24	25	11	10
11	25	26	12	11
12	26	27	13	12
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
116	138	139	125	124
117	139	140	126	125

2-D



3-D

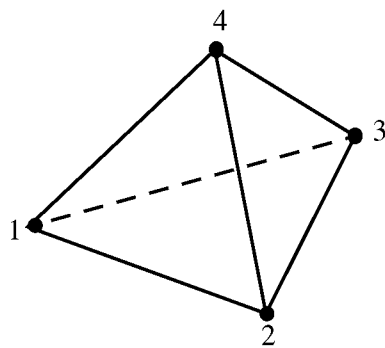
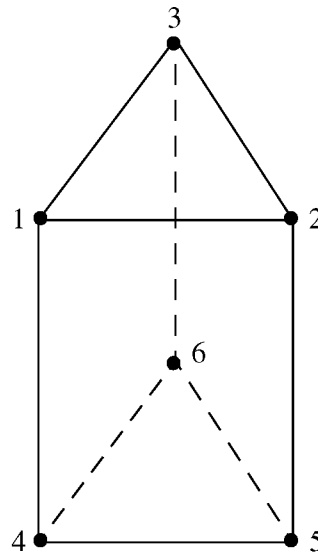
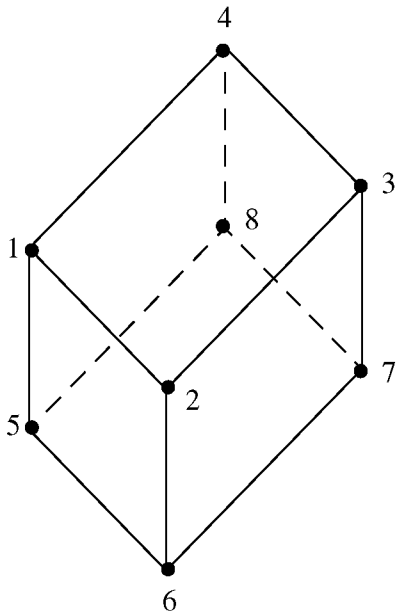


Figure 2. Elements available with FEHM in 2-D and 3-D problems showing nodal numbering convention.

6.2.18 Control statement eos (optional)

Equation of State. Provide the code with alternate thermodynamic properties for the liquid and/or vapor phases. (This is one way in which the code may be instructed to simulate nonisothermal, single phase air. It may also be used to make comparisons between the code and analytical solutions that use different equations of state.)

Group 1 - IIEOSD, IPSAT, ITSAT

Group 2 - EW1, EW2, EW3, EW4, EW5, EW6, EW7, EW8, EW9, EW10, EW11

Group 3 - EV1, EV2, EV3, EV4, EV5, EV6, EV7, EV8, EV9, EV10, EV11

For calculation of the simplified thermodynamic equations the above data is used to generate first order equations. The exception to this is the viscosity of the liquid and use of the ideal gas law. The viscosity of the liquid uses a $1/T$ term. For the calculation of vapor density and its derivatives, the ideal gas law is used instead of a linear relationship. Thus, EV4 and EV5 are not used, but are included so the format is the same as that for the liquid parameters in Group 2.

Input Variable	Format	Description
IIEOSD	integer	Equation of state reference number. When IIEOSD = 1 or 2 are used, they refer to the high and low pressure data sets, respectively, in FEHM. For these values the input in Group 2 and Group 3 will be ignored after it is entered. When any value other than 1 or 2 are used, the user-defined equation of state is used with Groups 2 and 3 for input.
IPSAT	integer	Parameter to set vapor pressure to zero. If IPSAT \neq 0 the vapor pressure is set to zero, otherwise the vapor pressure is calculated in the code.
ITSAT	integer	Parameter to adjust the saturation temperature. If ITSAT < 0, the saturation temperature is set to -1000°C. If ITSAT > 0, the saturation temperature is set to 1000°C. If ITSAT = 0, the calculated value is used.
EW1	real	Liquid reference pressure (MPa).
EW2	real	Liquid reference temperature (°C).
EW3	real	Liquid reference density (kg/m ³).
EW4	real	Derivative of liquid density with respect to pressure at reference conditions.
EW5	real	Derivative of liquid density with respect to temperature at reference conditions.
EW6	real	Liquid reference enthalpy (MJ/kg).
EW7	real	Derivative of liquid enthalpy with respect to pressure at reference conditions.
EW8	real	Derivative of liquid enthalpy with respect to temperature at reference conditions.

Input Variable	Format	Description
EW9	real	Liquid reference viscosity (Pa s).
EW10	real	Derivative of liquid viscosity with respect to pressure at reference conditions.
EW11	real	Derivative of liquid viscosity with respect to temperature at reference conditions.
EV1	real	Vapor reference pressure (MPa).
EV2	real	Vapor reference temperature (°C).
EV3	real	Vapor reference density (kg/m ³).
EV4	real	Not used, included only to maintain a similar format to Group 2. Density variation with pressure governed by ideal gas law.
EV5	real	Not used, included only to maintain a similar format to Group 2. Density variation with temperature governed by ideal gas law.
EV6	real	Vapor reference enthalpy (MJ/kg).
EV7	real	Derivative of vapor enthalpy with respect to pressure at reference conditions.
EV8	real	Derivative of vapor enthalpy with respect to temperature at reference conditions.
EV9	real	Vapor reference viscosity (Pa s).
EV10	real	Derivative of vapor viscosity with respect to pressure at reference conditions.
EV11	real	Derivative of vapor viscosity with respect to temperature at reference conditions.

The following is an example of **eos**. In this example, a user-defined equation of state is specified and the vapor pressure and saturation temperature are calculated in the code. For liquid properties, the reference pressure is 0.1 MPa, the reference temperature is 20 °C, and the reference density is 998. kg/m³, the derivative of density with respect to pressure is zero and with respect to temperature is -0.2 kg/m³/°C. The reference enthalpy is 0.88 MJ/kg, the derivative of enthalpy with pressure is zero, and the derivative with temperature is 4.2e-03 MJ/kg/°C. The reference viscosity is 9.e-04 Pa·s and the derivatives of viscosity with pressure and temperature are zero. For vapor properties, the reference pressure is 0.1 MPa, the reference temperature is 20 °C, and the reference density is 1.29 kg/m³. The reference enthalpy is 2.5 MJ/kg, the derivative of enthalpy with pressure is 0, and with temperature is 0.1 MJ/kg/°C. The reference viscosity is 2.e-4 Pa·s and its derivatives with pressure and temperature are zero.

eos										
3	0	0								
0.1	20.	998	0.	-0.2	0.88	0.	4.2e-3	9.e-4	0.	0.
0.1	20.	1.29	0.	0.	2.5	0.	0.1	2.e-4	0.	0.

6.2.19 Control statement **exrl** (optional)

Allows the user to choose linearized relative permeability. The linearized relative permeability is formed using the nonlinear value of relative permeability at the iteration number IEXRLP. After that iteration a relative permeability value based on a Taylor series expansion in saturation is used.

Group 1 - IEXRLP

Input Variable	Format	Description
IEXRLP	integer	If IEXRLP \geq 1, then linearized relative permeability. Otherwise not enabled.

In the following example of **exrl**, the user enables linearized relative permeability at the first iteration.

```
exrl
1
```

6.2.20 Control statement **fdm** (required if macro **coor** and **elem** not used)

Finite difference input.

Group 1 - KEYWORD

Group 2 - MODCHAR (only if KEYWORD is "modf")

or

Group 2 - NX, NY, NZ (if KEYWORD is "block" or "poin")

Group 3 - X0, Y0, N0 (only if KEYWORD is 'block')

Group 4 - MB, COORDINATE (X, Y or Z) (if KEYWORD is "poin")

or

Group 4 - MB, SPACING (DX, DY, DZ) (if KEYWORD is "bloc")

Group 4 is repeated for each NX, NY, and NZ, i.e., all X data are input, followed by Y data, followed by Z data for each division terminated by a blank line.

Input Variable	Format	Description
KEYWORD	character*4	Keyword indicating format of finite difference input to follow ("block", "poin", or "modf").
MODCHAR	character*132	If the keyword is "modf", the name of a modflow geometry data file is input and the finite difference input is read from that file and no other data is input.
NX	integer	Number of divisions in the x direction.
NY	integer	Number of divisions in the y direction.
NZ	integer	Number of divisions in the z direction.
X0	real	Coordinate of x origin point (m).
Y0	real	Coordinate of y origin point (m).
Z0	real	Coordinate of z origin point (m).
X	real	X coordinate (m).
Y	real	Y coordinate (m).
Z	real	Z coordinate (m).
DX	real	Node spacing in the x direction (m).
DY	real	Node spacing in the y direction (m).
DZ	real	Node spacing in the z direction (m).
MB	integer	Division number. If the division number is negative the code will space each division from the previous to the current proportional to the assigned spacings.

In the following example of **fdm**, the "block" format is used. There is 1 division in the X and Y directions, and 50 divisions in the Z direction. The spacing is 1 m in the X and Y directions, and 0.002 m for each division in the Z direction.

fdm			
block			
1	1	50	
0.	0.	0.	
1	1.0		
1	1.0		
1	0.002		
-50	0.002		

6.2.21 Control statement finv (optional)

No input is associated with this macro. When invoked, the code will perform finite volume calculations instead of finite element calculations for flow terms -- this may improve accuracy on non-orthogonal grid systems. Anisotropic properties (permeability, conductivity) are not supported with this macro. In this case, the values for permeability in the x-direction from control statement **perm** are used.

6.2.22 Control statement flow (either **boun or **flow** are required for a flow problem)**

Flow data. Source and sink parameters are input and may be used to apply boundary conditions. Note that the alternative definitions for isothermal models apply when **flow** is used in conjunction with control statement **airwater** (page 40).

Group 1 - JA, JB, JC, SKD, EFLOW, AIPED (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
Non-Isothermal model			
SKD	real	0.	Heat and mass source strength (kg/s), heat only (MJ/s). Negative value indicates injection into the rock mass.
EFLOW	real	0.	Enthalpy of fluid injected (MJ/kg). If the fluid is flowing from the rock mass, then the in-place enthalpy is used. If $EFLOW < 0$, then $ABS(EFLOW)$ is interpreted as a temperature ($^{\circ}C$) and the enthalpy (assuming water only) calculated accordingly. In heat only problems with $EFLOW < 0$, the node is in contact with a large heat pipe that supplies heat to the node through an impedance AIPED so as to maintain its temperature near $ABS(EFLOW)$. Large values (approximately 1000) of AIPED are recommended.
AIPED	real	0.	Impedance parameter. If AIPED is nonzero, the code interprets SKD as a flowing wellbore pressure (MPa) with an impedance $ABS(AIPED)$. If $AIPED < 0$, flow is only allowed out of the well. For heat only, AIPED is the thermal resistance. If $AIPED = 0$, SKD is flow rate. If $AIPED \neq 0$ and $SKD = 0$ the initial value of pressure will be used for the flowing pressure.
Note that If the porosity of the node is zero, then there is only a temperature solution, and the code forms a source proportional to the enthalpy difference. The source term is given by $Q = AIPED \cdot (E - EFLOW)$, where E is the in-place enthalpy and EFLOW is a specified enthalpy.			
Isothermal model			
Case 1: $AIPED = 0$ (Constant Mass Rate, 1- or 2-Phase Source or Sink)			
SKD	real	0.	Mass source strength (kg/s). Negative value indicates injection into the rock mass.

Input Variable	Format	Default	Description
EFLOW	real	0.	<p>a) $EFLOW \geq 0$, EFLOW is the source liquid saturation, $Q_w = SKD \cdot EFLOW \text{ (kg/s)},$ $Q_a = SKD \cdot (1 - EFLOW) \text{ (kg/s)}$</p> <p>b) $EFLOW < 0$, ABS(EFLOW) is the source air pressure (MPa) $Q_w = SKD \text{ (kg/s)}$ $Q_a = 1.0 \cdot (P_a - ABS(EFLOW)) \text{ (kg/s)}$</p> <p>In the above and following relations, Q_w is the source term for water, Q_a is the source term for air, and P_a is the in-place air pressure. The second case works well in situations where inflow is specified and it is desired to hold the air pressure at a constant value.</p>
AIPED	real	0.	Used only as flag to indicate Constant Mass Rate.
Case 2: AIPED > 0 (Constant Pressure, Constant Liquid Saturation Source or Sink)			
SKD	real	0.	Specified source air pressure (MPa).
EFLOW	real	0.	<p>a) $EFLOW < 0$, air only source. $Q_a = AIPED \cdot (P_a - SKD) \text{ (kg/s)}$</p> <p>b) $0 < EFLOW \leq 1$, EFLOW is specified source liquid saturation, for $SKD \geq 0$, 2-phase source, $Q_a = AIPED \cdot (P_a - SKD) \text{ (kg/s)}$ $Q_w = AIPED \cdot (S_l - EFLOW) \text{ (kg/s)}$ when $SKD < 0$, water only source $Q_a = 0$. In the above relation S_l is the in-place liquid saturation.</p>
AIPED	real		Impedance parameter. A large value is recommended ($10^2 - 10^6$) in order to create a flow term large enough to maintain constant pressure.
Case 3: AIPED < 0 (Outflow only, if $P_l > SKD$)			
SKD	real	0.	Pressure above which outflow occurs (MPa)
EFLOW	real	0.	Not used.
AIPED	real	0.	<p>Impedance parameter. $Q_w = ABS(AIPED) \cdot R_l / \mu_l (P_l - SKD) \text{ (kg/s)}$ where R_l is the water relative permeability and μ_l is the water viscosity.</p>

The following are examples of **flow**. In the first example, at node 88, a mass flux of 0.05 kg/s at 25 °C is being withdrawn from the model. Because fluid is being withdrawn the in-place temperature will actually be used. For every 14th node from node 14 to 140, the pressure and temperature are being held constant at 3.6 MPa and 160 °C, respectively. This represents a constant temperature recharge along one of the problem boundaries.

flow					
88	88	1	0.050	-25.0	0.
14	140	14	3.600	-160.0	1.

In the second example, the corresponding input for **airwater** (page 40), is included, indicating an isothermal air-water two-phase simulation is being run with a reference temperature for property evaluation of 20 °C and a reference pressure of 0.1 MPa. At nodes 26 and 52, water saturation is 100% and water is being injected at 2.e-3 kg/s. At nodes 1 and 27, there is an air only source, with a specified pressure of 0.1 MPa, and the air is being injected at the rate of 100*(Pa - 0.1) kg/s.

airwater					
2					
20.0	0.1				
flow					
26	52	26	-2.e-3	1.0	0.
1	27	26	0.1	-0.2	1.e2

6.2.23 Control statement flo2 (optional)

Group 1 - JA, JB, JC, JD, SKD, EFLOW, AIPED (SKD, EFLOW, AIPED - defined on page 59 under control statement **flow**)

Multiple lines of input may be used, terminated by a blank line.

Input Variable	Format	Description
JA	integer	Indices used to define planes in a 3-D simulation with a regular numbering pattern. The flow rates are defined within the inner loop of the do loops: DO JK = JA, JB KL = JK - JA DO IJ = JA + KL, JC + KL, JD ... ENDDO
JB	integer	
JC	integer	
JD	integer	
		ENDDO

6.2.24 Control statement flo3 (optional)

Alternate format for flow data for a seepage face.

Group 1 - JA, JB, JC, PFLOW, ESK, AIPED, KA (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
PFLOW	real	Ambient pressure (MPa) or flow area (m ²)
ESK	real	Not used
AIPED	real	Same as above for AIPED under keyword "flow"
KA	integer	Flag to indicate seepage face (-3) or unit gradient flux (-2)

6.2.25 Control statement floa (optional)

The input for this macro is identical to macro **flow**, except that it is relevant only to fluxes and the values are additive to existing fluxes for the defined nodes.

6.2.26 Control statement flxo (optional)

Mass flux between two nodes is output by choosing this control statement.

Group 1 - NFLX

Group 2 - IFLX1, IFLX2 (repeated NFLX times)

Group 3 - X1, Y1, Z1 (as needed)

Group 4- X2, Y2, Z2 (as needed)

If IFLX1 < 0, then after all IFLX1 and IFLX2 values are read, coordinates X1, Y1, and Z1 are read and the node nearest to these coordinates is used. If IFLX2 < 0, coordinates for the second node are read in on another line. The code cycles through each IFLX1 and IFLX2 in this manner, reading coordinates when needed. Results are written to the screen if tty output is enabled and to the output file **iout**.

Input Variable	Format	Description
NFLX	integer	Number of internode fluxes to be calculated.
IFLX1	integer	First node to be used in flux calculation.
IFLX2	integer	Second node to be used in flux calculation. If IFLX2 = 0, then the node connected to IFLX1 with the greatest internodal distance is used to calculate the mass flux.
X1	real	Coordinates of the first node to be used in flux calculation. Used only for those nodes where IFLX1 < 0.
Y1	real	
Z1	real	
X2	real	Coordinates of the second node to be used in flux calculation. Used only for those nodes where IFLX2 < 0.
Y2	real	
Z2	real	

The following are examples of **flxo**. In these examples, one internode flux is calculated. In the first case (left), from the node numbered 193 to the node

numbered 195, and in the second case (right) between nodes closest to coordinates 0., 0., 0. m and 20., 20., 20. m.

flxo
1
193 195

flxo
1
-1 -7
0. 0. 0.
20. 20. 20.

6.2.27 Control statement flxz (optional)

Total water flux through a zone is output by choosing this control statement. When this macro is invoked, the following output is given at every heat and mass transfer time step:

- The sum of all fluid source flow rates for each zone
- The sum of all fluid sink flow rates for each zone
- The net flux passing through each zone
- The net source/sink flux for each zone

Zones must be defined using macro **zone** prior to using this macro.

Group 1 - NFLXZ

Group 2 - IFLXZ(I), I = 1, NFLXZ

Input Variable	Format	Description
NFLXZ	integer	Number of zones for which output for water flux through the zone is written.
IFLXZ	integer	Zone numbers for which water flux output is written (NFLXZ zones)

The following is an example of **flxz**. In this example water flux through zones 1, 6 and 10 will be output.

flxz
3
1 6 10

6.2.28 Control statement fper (optional)

Assign permeability scaling factors.

Group 1 - JA, JB, JC, SCALEX, SCALEY, SCALEZ (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
SCALEX	real	1.0	Permeability scaling factor in the x-direction.
SCALEY	real	1.0	Permeability scaling factor in the y-direction.
SCALEZ	real	1.0	Permeability scaling factor in the z-direction.

The following is an example of **fper**. In this example, the values of the permeability (defined in a previous **perm** macro) are multiplied by 1.0 in the X direction, 0.5 in the Y direction, and 0.1 in the Z direction.

fper	1	140	1	1.0	0.5	0.1
------	---	-----	---	-----	-----	-----

6.2.29 Control statement **frlp** (optional)

Relative permeability factors for residual air effect.

Group 1 - JA, JB, JC, RLP_FAC (liquid), RLP_FAC (vapor) (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
RLP_FAC (liquid)	real	Residual liquid relative permeability value (i =1 to n0).
RLP_FAC (vapor)	real	Residual vapor relative permeability value (i = n0 + 1 to 2*n0).

6.2.30 Control statement **gdpm** (optional)

Data to define the parameters in the Generalized Dual Porosity model formulation.

Group 1 - GDPM_FLAG, NGDPMNODES

Group 2 - NGDPM_LAYERS(I), VFRAC_PRIMARY(I), (GDPM_X(I,J), J=1,NGDPM_LAYERS(I))- an arbitrary numbers of lines of input, terminated by a blank line.

Group 3 - JA, JB, JC, IGDPM (JA, JB, JC - defined on page 27)

Input Variable	Format	Description
GDPM_FLAG	integer	Flag to denote that the GDPM model option is being invoked. The default is 0 if GDPM is not being used. If 1, matrix node geometry is parallel fractures; if 2, matrix node geometry is spherical, with the fractured medium residing at the exterior of an idealized spherical block, and transport occurs into the block.
NGDPMNODES	integer	Total number of matrix nodes present in the simulation. Since this number may not be known at runtime, the code may be run once with a placeholder value for NGDPMNODES. If the number is incorrect, the code will report the appropriate value and stop. This value can then be entered and the simulation will proceed when the code is rerun.
NGDPM_LAYERS	integer	The number of matrix nodes specified for this model number. All primary nodes assigned to this model number (using the IGDPM input below) will have NGDPM_LAYERS matrix nodes.
VFRAC_PRIMARY	real	The fraction of the total control volume that is assigned to the primary porosity. Then, 1-VFRAC_PRIMARY is the fraction of the control volume that is divided among the dual porosity nodes.

Input Variable	Format	Description
GDPM_X	real	The matrix discretization distances for the matrix nodes associated with this model (units of meters). Grid points are placed at these values to discretize each matrix block. There must be NGDPM_LAYERS values, entered in ascending order. For the parallel plate geometry, the final value is the distance to the centerline between the fractures, and for the spherical geometry, the final value is the radius of the sphere.
IGDPM	integer	Model number for parameters defined in group 2. These values are assigned only for the primary nodes. The default is 0, which denotes that there are no dual porosity nodes at that primary node.

Based on the input in this macro, the code internally assigns node numbers, finite element coefficients, and reconstructs the connectivity array for the grid. The original nodes in the grid (the primary nodes) retain the node numbers 1 to NEQ_PRIMARY, where NEQ_PRIMARY is the number of nodes assigned in the macro **coor**. The matrix nodes are assigned numbers from NEQ_PRIMARY + 1 to NEQ_PRIMARY + NGDPMNODES. To assign matrix node numbers, the code loops through each primary node, and if GDPM nodes are specified, assigns the node numbers in increasing order within the matrix block of that primary node. [Note that the user is responsible for assigning rock, hydrologic, and transport properties for the matrix nodes as well as the primary nodes. For input using zones, this process is facilitated with the convention that zone numbers associated with matrix nodes are set to ZONE_DPADD + the zone number for the corresponding fracture node (see page 31). This convention is overwritten for any matrix node for which the user assigns a zone number using the 'nnum' option in the macro **zone**.]

For output, the code can report time-varying values in the ".out", ".his", and ".trc" files for both primary and matrix nodes, but fields written for the entire grid (for example, in the AVS output using the macro **cont**) are output only for the primary nodes.

The following is an example of **gdpm**. In this example the matrix node geometry is parallel to the fractures and there are 1479 matrix nodes distributed in 29 layers. A single model is defined which is applied to the entire problem domain.

gdpm							
1	1479						
29	.0001	.001	.002	.003	.004	.006	.009
.019	.02901	.03901	.04901	.05901	.09902	.19904	.29906
.39908	.49910	.59912	.69914	.79916	.89918	.99920	1.4993
1.9994	2.4995	2.9996	3.4997	3.9998	4.4999	5.0000	
1	0	0	1				

6.2.31 Control statement grad (optional)

Gradient model input.

Group 1 - NGRAD

Group 2 - IZONE_GRAD, CORDG, IDIRG, IGRADF, VAR0, GRAD1

Input Variable	Format	Description
NGRAD	integer	Number of gradient models.
IZONE_GRAD	integer	Zone associated with ith model
CORDG	real	Fiducial point of gradient equation.
IDIRG	integer	Coordinate direction of gradient.
IGRADF	integer	Variable to which gradient is applied. IGRADF = 1, Pressure IGRADF = 2, Temperature IGRADF = 3, Saturation
VAR0	real	Value of variable at fiducial point.
GRAD1	real	Gradient with distance.

The following is an example of grad. A temperature gradient in the Y direction from the fiducial point of 0 will be applied to zone 1.

```
grad
1
1      0.      2      2      10.     -150.
```

6.2.32 Control statement head (optional)

Hydraulic head values are used for input and output instead of pressures. Use of this macro enables the Boussinesq approximation (**bous** macro) and isothermal air-water two-phase simulation (**airwater** macro) automatically. It affects the **pres** and **flow** macros by requiring head information where pressure values were previously required. The default is to have no input associated with this macro. However, an optional head increment can be given after the head macro keyword. This value will be added to all input head values to ensure a single phase fluid state.

Input Variable	Format	Default	Description
HEAD0	real	0.	An incremental value that will be added to all input heads (m).

The following is an example of **head**. In this example the optional head increment is included and a value of 1000. m is added to all input head values.

```
head      1000.
```

6.2.33 Control statement hflx (optional)

Heat flux input.

Group 1 - JA, JB, JC, QFLUX, QFLXM (JA, JB, JC - defined on page 31)

A negative heat flux indicates heat flow into the reservoir.

Input Variable	Format	Default	Description
QFLUX	real	0.	If QFLXM = 0, then QFLUX is the heat flux (MW). If QFLXM ≠ 0, then QFLUX is a temperature (°C) and the heat flux is calculated according to the formula: $Q_H = QFLXM \cdot (T - QFLUX) \text{ (MW)}$
QFLXM	real	0.	If QFLXM > 0, multiplier for heat flux equation given in QFLUX description (MW/°C). This must be large for large temperature gradients, or when a constant temperature must be maintained. If QFLXM < 0, then QFLUX is interpreted as a fixed saturation and $Q_H = ABS(QFLXM) \cdot (S_l - QFLUX) \text{ (MW)}$

The following is an example of **hflx**. In this example, at each node from 401 to 410, a heat flux of 0.001 MW is being injected into the model.

hflx				
401	410	1	-0.001	0.0

6.2.34 Control statement hyco (required if macro perm not used)

Hydraulic conductivity input.

Group 1 - JA, JB, JC, PNxD, PNYD, PNZD (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
PNX	real	1.e-30	Hydraulic conductivity in the x-direction (m/s).
PNY	real	1.e-30	Hydraulic conductivity in the y-direction (m/s).
PNZ	real	1.e-30	Hydraulic conductivity in the z-direction (m/s).

The following is an example of the **hyco** macro. In this example, nodes 1 through 140 are specified to have hydraulic conductivities in the X, Y, and Z directions of 1.0e-5, 1.0e-5, and 0. m/s respectively.

hyco					
1	140	1	1.00e-05	1.00e-05	0.00e-00

6.2.35 Control statement ice or meth (optional)

Ice phase calculations, not tested.

Group 1 - ICE, SIIN, TMELT

Group 2 - JA, JB, JC, SII (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
ICE	integer	Solution descriptor for ice solution. ICE = 0, information is read but not used. ICE ≠ 0, ice solution is implemented.
SIIN	real	Default value for ice saturation (used when ice saturation SII in Group 2 is set to 0 at any node).
TMELT	real	Freezing temperature of water (°C).
SII	real	Ice saturation. The default value is [0].

6.2.36 Control statement impf (optional)

Time step control based on maximum allowed variable change.

Group 1 - DELPT, DELTT, DELST, DELAT

Input Variable	Format	Description
DELPT	real	Maximum allowable pressure change for which time step will be increased. (MPa)
DELTT	real	Maximum allowable temperature change for which time step will be increased. (°C)
DELST	real	Maximum allowable saturation change for which time step will be increased.
DELAT	real	Maximum allowable air pressure change for which time step will be increased. (MPa)

The following is an examples of **impf**. In this example, pressure changes are limited to 0.5 MPa, temperature changes to 20 °C, saturation changes to 0.1, and air pressure changes to 0.05 MPa during a time step.

```
impf
  0.5   20.0   0.1   0.05
```

6.2.37 Control statement init (required if macro pres not used)

Set initial pressure and temperature at all nodes.

Group 1 - PEIN, TIN, TIN1, GRAD1, DEPTH, TIN2, GRAD2, QUAD

Note that the macro **pres** may overwrite some of the values that are set by macro **init**.

Input Variable	Format	Description
PEIN	real	Initial value of pressure (MPa). If initial values are read from the read file (iread), then this value is ignored. If gravity is present, this is the value of the pressure at node 1, and the other nodal pressures are adjusted by applying the hydraulic head. Absolute pressures are used. Pressure as a function of depth is calculated with $TIN < 0$.
TIN	real	Initial value of temperature (°C). If $TIN \leq 0$, then the initial temperatures are calculated using the temperature gradient formulas given below.
TIN1	real	Defined in formulas below (°C)
GRAD1	real	Defined in formulas below (°C/m)
DEPTH	real	Defined in formulas below (m)
TIN2	real	Defined in formulas below (°C)
GRAD2	real	Defined in formulas below (°C/m)
QUAD	real	Defined in formulas below (°C/m ²)
$T = TIN1 + GRAD1 \times Z, \quad 0 \leq Z \leq DEPTH$ $T = TIN2 + GRAD2 \times Z + QUAD \times Z^2, \quad Z > DEPTH$		

The following are examples of **init**. In the first example, the initial pressure is 3.6 MPa and the initial temperature is 240 °C over the entire range of depth for the model.

init								
3.6	0.0	240.	0.	0.	240.	0.	0.	0.

In the second example, the initial pressure is 5.0 MPa and the initial temperature field is defined using a surface temperature of 20 °C and linear gradient of 0.3 °C/m for depths ranging from 0 - 2500 m.

init								
5.0	0.0	20.	0.3	2500.	20.	0.3	0.	0.

6.2.38 Control statement isot (optional)

If used, assumes an isotropic geometry for area coefficients. The isotropic assumption results in a saving of up to 1/3 for coefficient storage. No input is associated with this macro.

6.2.39 Control statement iter (optional)

If the user is not familiar with the linear equation solver routines in FEHM (Zyvoloski and Robinson, 1995) control statement **iter** should not be used.

Group 1 - G1, G2, G3, TMCH, OVERF

Group 2 - IRDOF, ISLORD, IBACK, ICOUPL, RNMAX

The parameters G1, G2, and G3 are used to calculate the completion criteria for the linear equation solver. The equation for the stopping criteria is:

$$EPE = G3 * \max(TMCH, \max(F0, \min(G1 * \text{SQRT}(R^{**2}), G2 * R^{**2})))$$

where R^{**2} is the sum-squared of the equation residuals, and $F0$ is the $\text{SQRT}(R0^{**2}) * EPM$ for the first iteration (see macro **ctrl** for a definition of EPM). The other parameters are defined below.

Input Variable	Format	Default	Description
G1	real	1.e-6	Multiplier for the linear convergence region of the Newton-Raphson iteration.
G2	real	1.e-6	Multiplier for the quadratic convergence region of the Newton-Raphson iteration.
G3	real	1.e-3	Multiplier relating Newton Raphson residuals to stopping criteria for linear solver
TMCH	real	1.e-9	Machine tolerance. If satisfied by the residual norm, the Newton iteration is assumed to be complete. If $TMCH < 0$ then the $ABS(TMCH)$ is used as a tolerance for each equation at each node. Convergence is achieved if the residual of every equation at every node is $< ABS(TMCH)$.
OVERF	real	1.1	Over relaxation factor for passive nodes in adaptive implicit method.
IRDOF	integer	0	Enables the reduced degree of freedom method. If $IRDOF = 0$, reduced degrees of freedom are not required. When $IRDOF = 1$, a reduced degree of freedom from 3 to 2 or 3 to 1 is used. When $IRDOF = 2$, a reduced degree of freedom from 3 to 2 is used. If $IRDOF = 11$, then an air only solution is found for the isothermal air-water process model. If $IRDOF = -11$, then the residual for the air equation with the airwater macro is ignored. If $IRDOF = 13$, then a liquid only solution for the airwater macro is assumed. Examples of 1, 2, 3, 4 and 6 degrees of freedom models are: 1 - heat only or mass only. 2 - heat and mass, or air-water (isothermal) 3 - air-water with heat (non-isothermal) 4 - heat and mass, double permeability or air-water (isothermal), double permeability 6 - air-water with heat, double permeability See Tseng and Zyvoloski (2000) for more information on the reduced degree of freedom method.

Input Variable	Format	Default	Description																																			
ISLORD	integer	0	Reordering parameter. The value of ISLORD and the corresponding equation order is given below. The ordering has an effect on the speed of convergence of several solution algorithms, but will not affect most users. For problems of order 2 or greater, the ordering can be understood by labeling each equation. For example for a 3-degree of freedom problem with mass, heat, and noncondensable gas, label the mass equation as 1, the heat equation as 2, and the noncondensable gas equation as 3. In general mass (water), heat or air, air. For double permeability problems fracture equations precede matrix equations, i.e., for an air-water problem - mass water fracture, mass air fracture, mass water matrix, mass air matrix.																																			
<table border="1"> <thead> <tr> <th>ISLORD</th> <th>2 Degrees of Freedom</th> <th>3 Degrees of Freedom</th> <th>4 Degrees of Freedom</th> <th>6 Degrees of Freedom</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1, 2</td> <td>1, 2, 3</td> <td>1, 2, 3, 4</td> <td>1, 2, 3, 4, 5, 6</td> </tr> <tr> <td>1</td> <td>2, 1</td> <td>1, 3, 2</td> <td>1, 3, 2, 4</td> <td>1, 4, 2, 5, 3, 6</td> </tr> <tr> <td>2</td> <td></td> <td>2, 1, 3</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td></td> <td>2, 3, 1</td> <td></td> <td></td> </tr> <tr> <td>4</td> <td></td> <td>3, 1, 2</td> <td></td> <td></td> </tr> <tr> <td>5</td> <td></td> <td>3, 2, 1</td> <td></td> <td></td> </tr> </tbody> </table>				ISLORD	2 Degrees of Freedom	3 Degrees of Freedom	4 Degrees of Freedom	6 Degrees of Freedom	0	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4, 5, 6	1	2, 1	1, 3, 2	1, 3, 2, 4	1, 4, 2, 5, 3, 6	2		2, 1, 3			3		2, 3, 1			4		3, 1, 2			5		3, 2, 1		
ISLORD	2 Degrees of Freedom	3 Degrees of Freedom	4 Degrees of Freedom	6 Degrees of Freedom																																		
0	1, 2	1, 2, 3	1, 2, 3, 4	1, 2, 3, 4, 5, 6																																		
1	2, 1	1, 3, 2	1, 3, 2, 4	1, 4, 2, 5, 3, 6																																		
2		2, 1, 3																																				
3		2, 3, 1																																				
4		3, 1, 2																																				
5		3, 2, 1																																				
IBACK	integer	0	IRDOF parameter. If IBACK = 0, SOR iterations are not performed before call to solver. If IBACK = 1, SOR iterations are performed before call to solver. If IBACK = 2, SOR iterations are performed before call to SOLVER, and SOLVER is called twice.																																			
ICOUPL	integer	0	Number of SOR iterations used in reduced degree of freedom methods.																																			
RNMAX	real	1.0e+11	Maximum running time for problem before the solution is stopped (cpu minutes).																																			

The following is an example of **iter**. In this example, the tolerances for the linear and quadratic convergence regions for the Newton-Raphson method are specified to be 1.e-5 times the initial residual, tolerance for the adaptive-implicit method is 1.e-5, machine tolerance is 1.e-9, and over-relaxation factor is 1.2. The reduced degree of freedom method is enabled, reordering is not done, SOR iterations are not performed before calling the solver, two SOR iterations are used in the reduced degree of freedom method, and the solution procedure is terminated if not completed within 200 CPU minutes.

iter	1.e-5	1.e-5	1.e-5	1.e-9	1.2
1	0	0	2	200.0	

6.2.40 Control statement itfc (optional)

Data to define flow and transport parameters at interfaces between pairs of zones.

Group 1 - ZONE_PAIR(I,1), ZONE_PAIR(I,2), RED_FACTOR(I)- an arbitrary number of lines of input, terminated by a blank line.

Group 2 - (FILTER_FLAG(J), J= 1,NSPECI)

Group 3 - ZONEC_PAIR(K,1), ZONEC_PAIR(K,2), FTN_FACTOR(K)- an arbitrary number of lines of input, terminated by a blank line.

KEYWORD 'file'

SFILENAME

ITFCPORISIZE(I), ITFCPROBSIZE(I)- an arbitrary number of lines of input, terminated by a blank line.

Input Variable	Format	Description
ZONE_PAIR	integer	Zone number for the zones for which the code identifies the interface connections when applying the permeability reduction factor.
RED_FACTOR	real	Reduction factor multiplying the harmonically weighted saturated permeability for all connections at the interface identified by ZONE_PAIR
FILTER_FLAG	integer	FEHM has a provision to apply transport mechanisms for size exclusion or filtration at interfaces defined in the itfc macro. These provisions can be used to simulate conditions in which, for example, abrupt changes in properties occur at interfaces, or hydrologic conditions not explicitly incorporated in a model (a thin clay layer, for example) are thought to be present that affect transport across the interface. The means for specifying these interface transport conditions is the itfc macro. Thus, this parameter is a flag used to distinguish whether the size exclusion or filtration is to be implemented (a value 1) or not (a value 0) for each species identified in the trac , ptrk , or mptr macros. The default value is 0. See the definition of FTN_FACTOR below for details on how to invoke the size exclusion or filtration model.
ZONEC_PAIR	integer	Zone number for the zones for which the code identifies the interface connections when applying the transport filtration or size exclusion factors.

Input Variable	Format	Description
FTN_FACTOR	real	Filtration or size exclusion factor applied for all connections at the interface identified by ZONEC_PAIR. For the trac macro, a size exclusion model is implemented, where FTN_FACTOR = 0 (size exclusion) or 1 (no exclusion) are options. For ptrk or mptr , a filtration model is implemented, where the parameter is the probability of the particle passing through the interface (if 0, filtration is guaranteed; if 1, there is no filtration). For the particle tracking model, FTN_FACTOR < 0 denotes that the pore size distribution is being used. This option is used with the particle size distribution option in ptrk and mptr , so that each particle is assigned a size. The cumulative pore size distribution is then used as a probability distribution function, and when a particle encounters the interface, a pore size is randomly selected from the distribution. If the particle is larger than the pore, it is filtered. Note that filtered particles remain at that location in the model and are no longer transported.
KEYWORD	character*4	Optional keyword 'file' designating that the pore size distribution information is being input in a separate file. This input is entered only for interfaces in which FTN_FACTOR < 0 is used.
SFILENAME	character*80	Optional file name containing the pore size distribution table. This input is entered only for interfaces in which FTN_FACTOR < 0 is used.
ITFCPORSIZE	real	Pore size for this entry of the pore size distribution table (paired with a value of ITFCPROBSIZE). An arbitrary number of entries can be input, terminated with a blank line. These entries are located in the file SFILENAME if specified, or in the itfc input file if the alternate input file is not used. The code decides if particles are irreversibly filtered by comparing the particle size to the randomly selected pore size. This input is entered only for interfaces in which FTN_FACTOR < 0 is used.
ITFCPROBSIZE	real	Cumulative probability for the distribution of pore sizes (paired with a value of ITFCPORSIZE). See description of ITFCPORSIZE above for details. The final entry of the table must have ITFCPROBSIZE = 1, since the distribution is assumed to be normalized to unity. This input is entered only for interfaces in which FTN_FACTOR < 0 is used.

Note that data for each numbered group must be input. The other input is optional. If filtration is not implemented for any species, a single blank line is input for Groups 2 and 3, signaling the end of **itfc** input.

The following is an example of **itfc**. In this example, the permeability reduction factor of 0.1 is applied to all node connections at the interface between zones 6 and 10, or 6 and 11.

itfc			
6	10	0.1	
6	11	0.1	

6.2.41 Control statement itup (optional)

The use of the **itup** macro is sometimes useful in problems where the flow directions are changing rapidly. The parameter UPWGT (in macro **ctrl**) must be greater than 0.5 for this macro to have any effect.

Group 1 - IAD_UP

Input Variable	Format	Default	Description
IAD_UP	integer	1000	Number of iterations after which the upwind directions are held constant. {A value of 2 is suggested}

In the following example of **itup**, after 10 iterations the upwind directions are held constant.

```
itup
 10
```

6.2.42 Control statement iupk (optional)

No input is associated with this control statement. This macro enables upwinding, the technique of evaluating the non-linear equation coefficients using the direction of flow relative to the grid block. For example, if flow is moving from grid block j to i , the coefficients for block i , are evaluated at the "upwind" block j . When upwinding is enabled the full transmissibility term will be upwinded (including the intrinsic permeability). Otherwise the fluid and relative permeability part of the transmissibility will be upwinded and the intrinsic permeability will be harmonically averaged.

6.2.43 Control statement mdnode (optional)

Enables extra connections to be made to nodes. This is useful for simulating wellbore connections, faults, and flow across internal boundaries.

Group 1 - NUM_MD, MAX_CON, IELIM, SX_MULT

Group 2 - NODE, IPAR, NPAR (repeated NUM_MD times)

Input Variable	Format	Default	Description
NUM_MD	integer	0	Number of new connections to be entered.
MDMAX	integer	0	Maximum number of new connections to a given node. This does not include old connections. Thus, if a node was already connected to 5 neighboring nodes and two new connections were added to this node in this macro statement and this was the maximum number of connections added in this macro statement, then MDMAX = 2.

Input Variable	Format	Default	Description
I_ELIM	integer	0	IF I_ELIM = 0, then no action. IF I_ELIM < 0, then nodal connections are eliminated as needed if redundant.
SX_MULT	real*8	1.0	Multiplier for equilibrium conditions.
NODE	integer	0	Node to which new connection is established.
IPAR	integer	0	IPAR is not used at present. Its value is ignored. However the entered number must be an integer.
NPAR	integer	0	NPAR is the new connected node. If NPAR = NODE, no new connection is established.

The following are examples of **mdnode**. In the first example (left), 3 new connections are specified, node 10 is connected to node 15, node 100 is connected to node 106, and node 10 is connected to node 320. A maximum of 2 new connections are allowed per node. The multiplier for equilibrium conditions is set to 10. In the second example (right), 4 new connections are specified, node 1 is connected to node 16, node 2 is connected to node 1, node 4 is connected to node 1 and node 10 is connected to node 203. A maximum of 3 new connections are allowed per node. The multiplier for equilibrium conditions is set to 100.

mdnode			
3	2	0	10
10	0	15	
100	0	106	
10	0	320	

mdnode			
4	3	0	100
1	0	16	
2	0	1	
4	0	1	
10	0	203	

6.2.44 Control statement mptr (optional)

Multiple species ingrowth particle tracking. Note that data for each numbered group must be input. The other input is optional.

Group 1 - NSPECI, MAXLAYERS, MAX_PARTICLES, RIPFEHM

Group 2 - POUT, PRNT_RST

Optional keyword "tcurve" is input to indicate that transfer function curves should be input to model matrix diffusion. It is followed by NUMPARAMS and TFILENAME.

KEYWORD
NUMPARAMS
TFILENAME

Optional keyword "zptr" designates zones for breakthrough curves will be defined. It is followed by IPZONE and IDZONE.

KEYWORD 'zptr'
IPZONE
IDZONE(I) I = 1 to IPZONE

Group 3 - RSEED

Group 4 - DAYCS, DAYCF, DAYHF, DAYHS

Group 5 is used to define models in which identical transport parameters are assumed to apply. Group 5 data are read until a blank line is encountered. The model number ITRC is incremented by 1 each time a line is read. Model parameters defined in Group 5 are assigned to nodes or zones using Group 6.

KEYWORD 'file'

PFILENAME

For a discussion of optional file use and content see macro **ptrk** (page 91). The applicable transport parameters are defined in Group 5 and Group 9.

KEYWORD 'afm'

Group 5 - TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), APERTUR(ITRC),
MATRIX_POR(ITRC)

or when 'afm' is implemented

Group 5 - TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), APERTUR(ITRC),
MATRIX_POR(ITRC), SRESIDUAL(ITRC),
GAMMA_AFM(ITRC)

Group 6 - JA, JB, JC, ITRC (JA, JB, JC - defined on page 31)

The following groups (Group 7 - 12) are repeated for each species.

Group 7 - ITH_SPECI, TRAK_TYPE, HALF_LIFE, IDAUGHTER,
CONFACTOR, NEWCONFACTOR, CONFTIME, GMOL,
P_FRACTION

KEYWORD 'size'

PART_SIZE(I), PROBSIZE(I) - an arbitrary numbers of lines of
input, terminated by a blank line.

Note that optional KEYWORD 'size' is only used when colloid transport is enabled.

Group 8 - LAYERS

Group 9 - LAYER_I, TRANSFLAG, KD, RD_FRAC, DIFFMFL

or for simulations with colloid (TRANSFLAG < 0)

Group 9 - LAYER_I, TRANSFLAG, KD, RD_FRAC, DIFFMFL, KCOLL,
RCOLL, FCOLL

Group 10 - NS

Group 11- JA, JB, JC, TMPCNSK (JA, JB, JC - defined on page 31)

Note that because the number of source terms is controlled by the value entered for NS, Group 11 input is not terminated with a blank line.

Group 12 - PINMASS, T1SK, T2SK

For transient source terms, Group 12 is repeated for each time interval and terminated with a blank line. Groups 11 and 12 are repeated for each source term (from 1 to NS).

For decay-ingrowth calculations, when the particle injection period is too small (for example, 1.E-4 days) compared to the half-life of the radionuclides and the half-life is large (for example 1.E+9 days), numerical errors in the decay-ingrowth calculation may arise due to truncation error. To get better accuracy, the user should try to increase the length of the injection period.

For particle tracking simulations using the transfer function method (see Section 6.2.3 on page 36 for input file format), it is sometimes desirable to identify the parameter ranges over which the two- and three-parameter type curves are accessed, so that an assessment can be made regarding the density of transfer function curves in a given part of the parameter space. If the flag `output_flag` in the transfer function file is set to "out", the code writes the real*8 array `param_density` to the *.out file in the following format:

For regular parameter spacings, the output is:

```

i = 1, nump1
  j = 1, nump2
    k = nump3
      write(iout.*) param_density(i,j,k)
    end do
  end do
end do

```

For two-parameter models, only the *i* and *j* loops are used. The value of `param_density` is the number of times any particle passes through any node at those values of the parameters. This allows the user to identify regions in which a greater density of transfer functions may be required. For the option 'free' in which there is no structure to the parameter grid used for the transfer function curves, `nump1` is the total number of curves, and `nump2` and `nump3` are equal to 1.

Input Variable	Format	Description
NSPECI	integer	Number of species in the simulation.
MAXLAYERS	integer	Maximum number of property layers in the model. The actual number of layers used in the model must be \leq MAXLAYERS.
MAX_PARTICLES	integer	Maximum number of particles used for individual species.
RIPFEHM	integer	Flag to indicate if simulation is coupled with GoldSim. RIPFEHM = 0, FEHM standalone simulation RIPFEHM = 1, GoldSim-FEHM coupling simulation
POUT	integer	Flag to specify the concentration output format: 1 - Concentrations computed as number of particles per unit total volume (rock and fluid) 2 - Concentrations computed as number of particles per unit fluid volume (the fluid is liquid for TRAK_TYPE = 1 and gas for TRAK_TYPE = 2). 3 - Concentrations computed as number of particles at a given node point.

Input Variable	Format	Description
PRNT_RST	integer	Flag to specify whether particle information is written to the ".fin" file: 0 - Particle information is not written to ".fin" file. 1 - Particle information is written to the ".fin" file.
KEYWORD	character	Optional keyword "tcurve" indicating transfer function curve data should be input to model matrix diffusion. If the keyword is found then NUMPARAMS and FILENAME are entered, otherwise they are omitted.
NUMPARAMS	integer	Number of parameters that define the transfer function curves being used.
TFILENAME	character	Name of input file containing the transfer function curve data.
KEYWORD	character*4	Optional keyword 'zptr' designating zones for breakthrough curves will be defined. If no keyword is input, IPZONE and IDZONE are also omitted.
IPZONE	integer	Number of zones for which breakthrough curves are to be output
IDZONE	integer	A list of zones for which particle breakthrough data are required. The code outputs the number of particles that leave the system at each zone IDZONE at the current time step. This information is written to the ".out" file at each heat and mass transfer time step.
RSEED	integer	6-digit integer random number seed.
DAYCS	real	Time which the particle tracking solution is enabled (days).
DAYCF	real	Time which the particle tracking solution is disabled (days).
DAYHF	real	Time which the flow solution is disabled (days).
DAYHS	real	Time which the flow solution is enabled (days).
KEYWORD	character*4	Optional keyword 'file' designating alternate transport parameter file input for multiple simulation realizations.
PFILENAME	character*80	Name of file from which to read transport parameters.
KEYWORD	character*4	Optional keyword 'afm' designating the Active Fracture Model input for determining fracture spacing should be used.
TCLX	real	Dispersion in the x-direction (m). The input value is ignored when dispersion is turned off.
TCLY	real	Dispersion in the y-direction (m). The input value is ignored when dispersion is turned off.
TCLZ	real	Dispersion in the z-direction (m). The input value is ignored when dispersion is turned off.
APERTUR	real	Mean fracture aperture (m). The input value is ignored when matrix diffusion is turned off.

Input Variable	Format	Description
MATRIX_POR	real	Porosity of the rock matrix. Used to simulate diffusion and sorption in the rock matrix when matrix diffusion is invoked, otherwise the input value of MATRIX_POR is ignored.
SRESIDUAL	real	Residual saturation in the Active Fracture Model used for determining the spacing between active fractures. This parameter is only needed when the keyword 'afm' is included, in which case the input must be entered. However, the model is only used in dual permeability simulations at locations where the finite spacing matrix diffusion model is invoked.
GAMMA_AFM	real	Exponent in the Active Fracture Model used for determining the spacing between active fractures. See comments for SRESIDUAL above.
ITRC	integer	Model number for parameters defined in group 5.
ITH_SPECI	integer	Number index of the ith species.
TRAK_TYPE	integer	Flag to denote the fluid phase of the particles: 1 - liquid phase particles 2 - vapor phase particles
HALF_LIFE	real	Half-life for irreversible first order decay reaction(s) (days). Set HALF_LIFE = 0 for no decay.
IDAUGHTER	integer	Index of the daughter species (i.e., the index number of the species to which the current species decays) If IDAUGHTER = 0, there is no decay and no ingrowth, If IDAUGHTER = -1, there is decay but no ingrowth.
CONFACTOR	real	Initial conversion factor for GoldSim-FEHM coupling and FEHM standalone simulations (# of particles/mole). For FEHM stand alone simulations: If CONFACTOR = 0, no conversion is necessary. The input value of PINMASS is the number of particles. If CONFACTOR > 0, the code assumes the input mass is in moles and uses the CONFACTOR to convert it into number of particles. For GoldSim-FEHM coupling: If CONFACTOR = 0, at each time step, the code selects a conversion factor based on the available memory and the remaining simulation time (end time - current time). The code then uses the selected conversion factor to calculate the number of particles to be injected at the current time step. If CONFACTOR > 0, the code uses the product of the CONFACTOR and the input mass (in moles) to calculate the input number of particles at each time step.
NEWCONFACTOR	real	Replace the initial value of CONFACTOR with that specified by NEWCONFACTOR. If NEWCONFACTOR < 0, then CONFACTOR = -NEWCONFACTOR. If NEWCONFACTOR = 0, use automatic conversion factors.

Input Variable	Format	Description
CONFTIME	real	The time at which to change the CONFACTOR value to that specified by NEWCONFACTOR.
GMOL	real	The molecular weight of the ith species. The code uses GMOL and CONFACTOR to convert the mass from number of particles to grams in the final output for GoldSim-FEHM coupling.
P_FRACTION	real	The decay-ingrowth particle release factor (percentage of the maximum number of particles released for the parent species). Values are in the range 0. - 1. The default is 0.5.
KEYWORD	character*4	Optional keyword 'size' designating that the colloid size distribution model option is being used (combined with the interface filtration option in the itfc macro). If the keyword is not input, PART_SIZE and PROBSIZE are also omitted.
PART_SIZE	real	Colloid particle size for this entry of the particle size distribution table (paired with a value of PROBSIZE). An arbitrary number of entries can be input, terminated with a blank line. The code assigns each particle a size based on this distribution of particle sizes, and decides if particles are irreversibly filtered based on the pore size distribution assigned in the itfc macro.
PROBSIZE	real	Colloid cumulative probability for the distribution of sizes (paired with a value of PART_SIZE). See description of PART_SIZE above for details. The final entry of the table must have PROBSIZE = 1, since the distribution is assumed to be normalized to unity.
LAYERS	integer	Number of layers in which the transport properties of the ith species are to be modified. If no property is altered, then set layers=0.
LAYER_I	integer	The index number of the ith layer defined in group 5
TRANSFLAG	integer	Flag to specify which transport mechanisms apply [abs(TRANSFLAG)]: 1 - advection only (no dispersion or matrix diffusion) 2 - advection and dispersion (no matrix diffusion) 3 - advection and matrix diffusion, infinite fracture spacing solution (no dispersion) 4 - advection, dispersion, and matrix diffusion, infinite fracture spacing solution 5 - advection and matrix diffusion, finite fracture spacing solution (no dispersion) 6 - advection, dispersion, and matrix diffusion, finite fracture spacing solution 8 - use the the transfer function approach with 3 dimensionless parameters and type curves for handling fracture-matrix interactions. For TRANSFLAG < 0, transport simulations include colloids.

Input Variable	Format	Description
		<p>For equivalent continuum solutions, the fracture spacing in the finite spacing model is determined using</p> $SPACING = APERTURE/POROSITY.$ <p>For dual permeability models, the fracture spacing input parameter APUV1 in the dpdp macro is used as the half-spacing. If the Active Fracture Model (see keyword 'afm') is used, APUV1 is the geometric fracture half-spacing, and the additional terms SRESIDUAL and GAMMA_AFM are used to determine the spacing between active fractures (see below).</p>
KD	real	<p>Sorption coefficient (linear, reversible, equilibrium sorption). Units are kg-fluid / kg-rock (these units are equivalent to the conventional units of cc/g when the carrier fluid is water at standard conditions). This value applies to the medium as a whole when matrix diffusion is turned off, whereas for simulations invoking matrix diffusion, the value applies to the rock matrix. For the latter case, sorption in the flowing system (fractures) is modeled using the RD_FRAC variable.</p>
RD_FRAC	real	<p>Retardation factor within the primary porosity (fractures) for a matrix diffusion particle tracking simulation (use 1 for no sorption on fracture faces). The input value is ignored unless matrix diffusion is invoked.</p>
DIFFMFL	real	<p>Molecular diffusion coefficient in the rock matrix (m²/s). The input value is ignored unless matrix diffusion is invoked.</p>
KCOLL	real	<p>Colloid distribution parameter, the ratio of contaminant mass residing on colloids to the mass present in aqueous form. It is used to compute an effective aperture via the following:</p> $APWID = APERTURE \cdot (1 + KCOLL)$
RCOLL	real	<p>Colloid retardation factor. Used, in conjunction with kcoll, to adjust colloid retardation in fractures using the following formula:</p> $FRACRD = \frac{RD_FRAC + KCOLL \cdot RCOLL}{1 + KCOLL}$
FCOLL	real	<p>Colloid filtration parameter. Used to compute the probability a colloid will be irreversibly filtered along the path between two nodes using the following:</p> $PROBFILT = 1 - \exp(DISTANCE/FCOLL)$ <p>where <i>DISTANCE</i> is the length of the path between nodes.</p>
NS		<p>Number of spatial source terms for the ith species</p>
TMPCNSK	real	<p>Particle injection parameter assigned for nodes defined by JA, JB, and JC. Two options are available:</p>

Input Variable	Format	Description
		<p>TMPCNSK > 0. - particles are injected at each node in proportion to the source mass flow rate at the node. This boundary condition is equivalent to injecting a solute of a given concentration into the system. Note: the source flow rates used to assign the number and timing of particle injections are those at the beginning of the particle tracking simulation (time DAYCS). Transient changes in this source flow rate during the particle tracking simulation do not change the number of particles input to the system.</p> <p>TMPCNSK < 0. - particles are introduced at the node(s), regardless of whether there is a fluid source at the node. Default is 0. for all unassigned nodes, meaning that no particles are injected at that node.</p>
PINMASS	real	Input mass. If CONFACTOR = 0, PINMASS is the number of particles to be injected at locations defined by TMPCNSK. If CONFACTOR > 0, PINMASS is the input mass expressed in moles. The code uses CONFACTOR to convert PINMASS into number of particles.
T1SK	real	Time (days) when particle injection begins. Default is 0.
T2SK	real	Time (days) when particle injection ends. Default is 0.

The following is an example of **mptr**. A multiple-species decay-chain (1 → 2 → 3 → 4) is simulated, with decay half lives of the species equaling 10,000, 3,000, 10,000, and 4,000 years, respectively. In this simulation a maximum of 3 property layers are specified although only 1 layer is used, the maximum number of particles is specified to be 1100100, and FEHM is run in stand-alone mode. Concentrations will be computed as number of particles per unit fluid volume and no output will be written to the ".fin" file. Use of the 'zptr' keyword indicates that a single zone will be defined for breakthrough curve output which will be written to the ".out" file. The random number seed is defined to be 244562. The particle tracking solution is enabled at 0.1 days, and disabled at 3.65e8 days, while the flow solution is disabled at 38 days and re-enabled at 3.65e8 days. Dispersivity in the X-, Y-, and Z-directions are defined to be 0.005 m, the mean fracture aperture is 0.0001 m, and the matrix porosity is 0.3. Particles for species 1 are injected at a constant rate from 0 to 5,000 years, and species 2, 3, and 4 are formed through the decay reactions, with no input at the inlet. Advection and dispersion (without matrix diffusion) is being modeled. The retardation factors for the four species are 1, 1, 1.9, and 1, respectively (i.e. only species 3 sorbs).

mptr									
4	3	1100100	0						Group 1
2	0								Group 2
zptr									
1									
1									
244562									Group 3
0.1	3.65e8	38	3.65e8						Group 4
0.005	0.005	0.005	1.e-4	0.3					Group 5

1	0	0	1						Group 6
1	1	3.652485E6	2	1	-1	1.	1.	0.5	Group 7
1									Group 8
1	2	0.	1.					1.e-14	Group 9
1									Group 10
1	202	201	-1						Group 11
10000.	0.		365.25E2						Group 12
10000.	365.25E2		730.5E2						Group 12
10000.	730.5E2		1.09575E5						Group 12
10000.	1.09575E5		1.461E5						Group 12
.									.
.									.
10000.	1.7532E6		1.789725E6						Group 12
10000.	1.789725E6		1.82625E6						Group 12
2	1	1.095745E6	3	0	-1	1.	1.	0.5	Group 7
1									Group 8
1	2	0.	1.					1.e-14	Group 9
1									Group 10
1	202	201	-1						Group 11
0	0.		1.825E6						Group 12
3	1	3.652485E6	4	0	-1	1.	1.	0.5	Group 7
1									Group 8
1	2	0.108	1.					1.e-14	Group 9
1									Group 10
1	202	201	-1						Group 11
0	0.		1.825E6						Group 12
4	1	1.460972E6	-1	0	-1	1.	1.	0.5	Group 7
1									Group 8
1	2	0.	1.					1.e-14	Group 9
1									Group 10
1	202	201	-1						Group 11
0	0.		1.825E6						Group 12

6.2.45 Control statement ngas (optional)

Noncondensable gas transport.

Group 1 - ICO2D

Group 2 - JA, JB, JC, PCO2 (JA, JB, JC - defined on page 31)

Group 3 - JA, JB, JC, CPNK (JA, JB, JC - defined on page 31)

Group 4 - JA, JB, JC, QCD (JA, JB, JC - defined on page 31)

Note that all Group 2 values are entered first, followed by Group 3 values, followed by Group 4 values.

Input Variable	Format	Default	Description
ICO2D	integer	3	Solution descriptor for noncondensable gas transport. ICO2D = 1, the 3 degree of freedom solution will be reduced to a 1 degree of freedom problem. (See macro iter , the parameter ICOUPL is also set to 5 if ICO2D = 1.) ICO2D = 2, the 3 degree of freedom solution will be reduced to a 2 degree of freedom problem. (See macro iter , the parameter ICOUPL is also set to 5 if ICO2D = 2.) ICO2D = 3, full 3 degree of freedom.
PCO2	real	0.	Initial partial pressure of noncondensable gas. If PCO2 < 0 then ABS (PCO2) is interpreted as a temperature and the partial pressure of the noncondensable gas is calculated according to the formula: $PCO2 = P_T - P_{SAT}(T)$ where P_T is the total pressure and $P_{SAT}(T)$ is the water saturation pressure and is a function of temperature only.
CPNK	real	0.	If CPNK ≤ 0, then ABS (CPNK) is the specified noncondensable pressure and will be held at that value. If CPNK > 0, then CPNK is the specified relative humidity and the saturation, S_l , is calculated using the vapor pressure lowering formula and the capillary pressure formula: $Pcap(S_l) = \ln(h)\rho_l RT$ where $Pcap$ is the capillary function, h is the humidity, R is the gas constant, T is the temperature, and ρ_l is the liquid density. Once the formula is solved, S_l is held constant. The humidity condition is only enabled for the van Genuchten capillary function model. See macro rlp .
QCD	real	0.	Specified air source strength (kg/sec).

The following is an example of **ngas**. In this example, a full 3 degrees of freedom solution is specified. The initial temperature at nodes 1 to 800 is 20 °C and the code is asked to calculate the initial noncondensable gas pressure. There is no specified noncondensable gas source.

ngas				
3				
1	800	1	-20	
1	800	1	0.	
1	800	1	0.	

6.2.46 Control statement nobr (optional)

Do not break connection between nodes with boundary conditions. No input is associated with this macro.

6.2.47 Control statement node (optional)

Specify the node numbers for which detailed output is desired.

Group 1 - M

Group 2 - MN (1), MN (2), . . . , MN (M)

Group 3 - X, Y, Z (as needed)

or

Group 1 - KEYWORD

Group 2 - JA, JB, JC (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output (iout) and history plot (ishis, istr) files. If $M \leq 0$, pressure and temperature will be written on the output file for all nodes but no nodal parameter values will be printed in the history plot files. Group 2 is omitted if $M \leq 0$.
MN	integer	M node numbers for which information will be printed on the output file (iout). If $MN(l) < 0$, then coordinates are used to define the print-out node, and the coordinate sets (X, Y, Z) for each $MN(l) < 0$ are added after Group 2.
X	real	Coordinates of node for which information will be printed. One line for each $MN < 0$. The code finds the node closest to the coordinate given. For 2-D problems set $Z = 0$. No input if $MN > 0$.
Y	real	
Z	real	
KEYWORD	character*5	Keyword 'block' to invoke node specification by JA, JB, JC format.

The following are examples of **node**. In the first example (left), 2 nodes are specified for output, nodes 50 and 88. In the second example (center), two nodes are specified for output, the node numbered 50 and the node closest to the coordinates $X = 100. \text{ m}$, $Y = 1000. \text{ m}$ and $Z = 0. \text{ m}$. In the third example (right), output is specified for the block of nodes 1, 11, 21, 31, 41, 51, 61, 71, 81, 91 and for those nodes defined by zone 3 (see macro **zone**).

node
2
50 88

node
2
50 -88
100. 1000. 0.

node
block
1 100 10
-3 0 0

6.2.48 Control statement nod2 (optional)

Specify the node numbers for which detailed file output is desired and alternate nodes for terminal output.

Group 1 - M, M2

Group 2 - MN (1), MN (2), . . . , MN (M)

Group 3 - X, Y, Z (as needed)

Group 4 - MNI(1), MNI(2), . . . , MNI(M2)

Group 5 - X, Y, Z (as needed)

Input Variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output file (iout). If $M \leq 0$, pressure and temperature will be written on the output file for all nodes but no nodal parameter values will be printed in the history plot files. Group 2 is omitted if $M \leq 0$.
M2	integer	Number of nodes for short list (terminal printout). If $M2 \leq 0$, Group 4 is omitted.
MN	integer	M node numbers for which information will be printed on the output file (iout). If a $MN(l) < 0$, then coordinates are used to define that print-out node, and the coordinate sets (X, Y, Z) for each $MN(l) < 0$ are added after Group 2.
MNI	integer	M2 node numbers for which information will be printed on the terminal (short list). This group exists only if $M2 \neq 0$. If $MNI(l) < 0$, then coordinates are used to define the terminal output nodes, and the coordinate sets (X, Y, Z) for each $MNI(l) < 0$ are added after Group 4.
X	real	Coordinates of node for which information will be printed. One line for each MN or MNI < 0 . The code finds the node closest to the coordinate given. For 2-D problems set $Z = 0$. No input if no MN or MNI < 0 .
Y	real	
Z	real	

The following are examples of **nod2**. In the first example (left), detailed output to the output file is specified for two nodes, the nodes numbered 50 and 88, and one node is specified for terminal output, node 50. In the second example (right), two nodes are specified for detailed output, the nodes numbered 50 and 88, and one node is specified for terminal output, the node closest to the coordinates $X = 100. \text{ m}$, $Y = 1000. \text{ m}$ and $Z = 0. \text{ m}$.

nod2
2 1
50 88
50

nod2
2 1
50 88
-88
100. 1000. 0.

6.2.49 Control statement nod3 (optional)

Specify the node numbers for which detailed file output is desired and alternate nodes for terminal output.

Group 1 - M, M2, M3

Group 2 - MN (1), MN (2), . . . , MN (M)

Group 3 - X, Y, Z (as needed)

Group 4 - MNI(1), MNI(2), . . . , MNI(M2)

Group 5 - X, Y, Z (as needed)

Group 6 - MNI(1), MNI(2), . . . , MNI(M3)

Group 7 - X, Y, Z (as needed)

Input Variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output file (iout). If $M \leq 0$, pressure and temperature will be written on the output file for all nodes but no nodal parameter values will be printed in the history plot files. Group 2 is omitted if $M \leq 0$.
M2	integer	Number of nodes for short list (terminal printout). If $M2 \leq 0$, Group 4 is omitted.
M3		Number of nodes for short list (variable porosity model information printout). If $M3 \leq 0$, Group 6 is omitted.
MN	integer	M node numbers for which information will be printed on the output file (iout). If a $MN(l) < 0$, then coordinates are used to define that print-out node, and the coordinate sets (X, Y, Z) for each $MN(l) < 0$ are added after Group 2.
MNI	integer	M2 node numbers for which information will be printed on the terminal (short list). This group exists only if $M2 \neq 0$. If $MNI(l) < 0$, then coordinates are used to define the terminal output nodes, and the coordinate sets (X, Y, Z) for each $MNI(l) < 0$ are added after Group 4.
X	real	Coordinates of node for which information will be printed. One line for each MN or MNI < 0 . The code finds the node closest to the coordinate given. For 2-D problems set $Z = 0$. No input if no MN or MNI < 0 .
Y	real	
Z	real	

The following are examples of **nod3**. In the first example (left), detailed output to the output file is specified for two nodes, the nodes numbered 50 and 88, and one node is specified for terminal output, node 50. In the second example (right), two nodes are specified for detailed output, the nodes numbered 50 and 88, and one node is specified for terminal output, the node closest to the coordinates $X = 100. \text{ m}$, $Y = 1000. \text{ m}$ and $Z = 0. \text{ m}$.

nod2		
2		1
50		88
50		

nod2			
2			1
50			88
-88			
100.	1000.		0.

6.2.50 Control statement perm (required if macro **hyco** not used)

Assign permeabilities of the rock. Permeabilities represent average values of a volume associated with a node. Note that using **rlp** models 4 or 6 to describe relative permeabilities causes these values to be overwritten.

Group 1 - JA, JB, JC, PNXD, PNYD, PNZD (JA, JB, JC - defined on page 31)

Input Variable	Format	Default	Description
PNXD	real	1.e-30	Permeability in the x-direction (m ²).
PNYD	real	1.e-30	Permeability in the y-direction (m ²).
PNZD	real	1.e-30	Permeability in the z-direction (m ²).

The following is an example of the **perm** macro. In this example, nodes 1 through 140 are specified to have permeabilities in the X, Y, and Z directions of 2.5e-14, 2.5e-14, and 0. m² respectively.

perm					
1	140	1	2.50e-14	2.50e-14	0.00e-00

6.2.51 Control statement pest (optional)

Output variable information for PEST parameter estimation routine.

Group 1 - MPEST

Group 2 - NPEST(I), I = 1, MPEST

Group 3 - X, Y, Z (as needed)

Input Variable	Format	Description
MPEST	integer	Number of nodes for PEST output. At present the code outputs only pressures (heads), saturations, and temperatures.
NPEST(I)	integer	Node numbers printed to the output file (<i>fehmn.pest</i>) with values of variables listed above. If NPEST(I) < 0 then the node numbers are determined from the coordinates.
X, Y, Z	real	Coordinates in grid if NPEST(I) < 0. The coordinates are used to find the node closest in distance to that point and that node is substituted for NPEST(I).

The following is an example of **pest**. In this example pest output is specified at 5 nodes, nodes numbered 21, 23, 35, and 47, and the node closest to the coordinates X=10. m, Y=15. m, Z=20. m.

pest				
21	23	35	47	-50
10.	15.	20.		

6.2.52 Control statement ppor (optional)

Group 1 - IPOROS

Group 2 - JA, JB, JC, POR1, POR2, POR3, POR4 (number of parameters entered depends on model type) (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
IPOROS	integer	Model type: IPOROS = 1, aquifer compressibility model IPOROS = -1, specific storage model (use only for isothermal conditions) IPOROS = -2, Gangi model (not available for air-water-heat conditions)
Model (1): IPOROS = 1,		Aquifer compressibility $\phi = \phi_o + \alpha_a(P - P_o)$ where α_a = aquifer compressibility (MPa ⁻¹) ϕ_o = initial porosity P_o = initial pressure (MPa)
POR1	real	Aquifer compressibility α (MPa ⁻¹)
Model (-1): IPOROS = -1,		Specific storage $S_s = \rho g(\alpha_a + \phi\beta)$ where ρ = liquid density (kg/m ³) g = gravity α_a = aquifer compressibility (MPa ⁻¹) ϕ = porosity β = liquid compressibility (MPa ⁻¹)
POR1	real	Specific storage S_s (m ⁻¹)
Model (-2): IPOROS = -2,		Gangi model with calculation of initial permeability and porosity. $\phi = \phi_o \left[1 - \left(\frac{P_c}{P_x} \right)^m \right]$ and $P_c = \sigma - P - \alpha E(T - T_o)$ where ϕ_o = initial porosity m = Gangi exponent P_x = fitted parameter (MPa) σ = in-situ stress (MPa) α = coefficient of thermal expansion (1/°C) E = Young's modulus (MPa) T = temperatures (°C) T_o = initial temperature (°C)
		Note: for the Gangi model the permeability is varied by $k = k_o \left(\frac{\phi}{\phi_o} \right)^3$
POR1	real	Exponent m in Gangi bed of nails model.
POR2	real	P_x parameter (MPa) in Gangi equation.

Input Variable	Format	Description
POR3	real	σ in-situ stress (MPa).
POR4	real	(αE) The product of the coefficient of thermal expansion for the rock and the Young's modulus (MPa/°C). Note: For isothermal simulations the thermal term does not apply.

In the following example of **ppor**, aquifer compressibility is modeled. All nodes in the model are assigned a compressibility of $1.e-2 \text{ MPa}^{-1}$.

ppor				
1				
1	0	0		1.e-2

6.2.53 Control statement **pres** (required if macro **init** not used)

Group 1 - JA, JB, JC, PHRD, TIND, IEOSD (JA, JB, JC - defined on page 31)

The initial values defined in control statement **pres** supersede all others. Note that the term "saturated" referred to in IEOSD, is **not** the groundwater hydrology definition (volumetric fraction of pore void that is filled with water) used elsewhere in this document. Saturated here indicates that vapor and liquid phases exist simultaneously. The superheated region means that all pore space is filled with gas.

Input Variable	Format	Default	Description
PHRD	real	PEIN	Initial pressure (MPa).
TIND	real		Initial temperature (°C) if IEOSD = 1 or 3, Initial saturation if IEOSD = 2
IEOSD	integer	1	Thermodynamic region parameter. IEOSD = 1, the compressed liquid region IEOSD = 2, the saturation region IEOSD = 3, the superheated region. If IEOSD < 0 then the code uses ABS (IEOSD) and fixes the values of PHRD and TIND to the values provided above.

The following is an example of **pres**. In this example, zones numbered 1, 2, 3, 4, 5 and 6, and nodes 1 through 800 have an initial pressure of 0.1 MPa and are located in the saturation region. The initial water saturation for zones 1, 2, and 4 is 0.1, for zone 3 is 0.003, for zones 5 and 6 is 0.11, and for nodes 1 through 800 is 0.5. In addition, for zone 6 the pressure and saturation are held constant throughout the run.

pres					
-1	0	1	0.1	0.1	2
-2	0	1	0.1	0.1	2
-3	0	1	0.1	0.003	2
-4	0	1	0.1	0.1	2
-5	0	1	0.1	0.11	2
-6	0	1	0.1	0.11	-2
1	800	1	0.1	0.5	2

6.2.54 Control statement ptrk (optional, cannot be used with trac)

Particle tracking simulation input. Note that data for each numbered group must be input. The other input is optional.

Group 1 - NPART, RSEED

KEYWORD 'rip'

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS

or if the optional keyword 'rip' follows Group 1

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS, RIPFEHM, CONFREAD, GMOL, P_FRACTION

Group 3 - TRAK_TYPE, HALF_LIFE, POUT, PRNT_RST

Optional keyword "tcurve" is input to indicate that transfer function curves should be input to model matrix diffusion. It is followed by NUMPARAMS and TFILENAME.

KEYWORD

NUMPARAMS

TFILENAME

Optional keyword "zptr" designates zones for breakthrough curves will be defined. It is followed by IPZONE and IDZONE.

KEYWORD 'zptr'

IPZONE

IDZONE(I) I = 1 to IPZONE

Group 4 is used to define models in which identical sorption and transport parameters are assumed to apply. Group 4 data are read until a blank line is encountered. The model number ITRC is incremented by 1 each time a line is read. Model parameters defined in Group 4 are assigned to nodes or zones using Group 5.

An optional, flexible input structure involving the assignment of transport parameters is implemented in the particle tracking input to allow multiple realization simulations to use different parameters for each realization. The user invokes this option using the keyword 'file' before Group 4, followed by the name of the file that the transport parameters reside in.

KEYWORD 'file'

PFILENAME

The structure of the alternate parameter file is:

NINPUTS
 heading or blank line
 PTRPARAM(I) I=1 to NINPUTS

.
 .
 .

[a line of parameters is present for each realization]

The method for assigning a given value of PTRPARAM to a specific transport parameter, defined in Group 4, is discussed below. There are an arbitrary number of input lines each representing a given realization of parameter values. In a multiple-realization scenario, the code enters the input file for each realization, and for this input, reads down the corresponding number of lines to obtain the parameters for that realization. For example, for realization number 10, the code reads down to the 10th line of data (line 12 in the file) and uses those parameter values.

Once these parameters are read in for a given realization, they must be assigned to specific transport parameters. This is done in the following way in Group 4. If any of the inputs other than TRANSFLAG are negative, the code takes the absolute value of the number and interprets it as the column number from which to assign the transport parameter. For example, if DIFFMFL = -5, then the diffusion coefficient is the fifth input number in the PTRPARAM array. In this way, any of the transport parameter inputs can be assigned through the alternate input file rather than the input line in **ptrk**.

There are four possible forms of input for Group 4, which depend on whether or not the Active Fracture Model is implemented (optional KEYWORD 'afm') and colloid transport is enabled (TRANSFLAG < 0). If colloid transport is enabled, then ABS(TRANSFLAG) is used to determine the transport mechanism. Note that optional KEYWORD 'size' is only used when colloid transport is enabled.

KEYWORD 'afm'

KEYWORD 'size'

PART_SIZE(I), PROBSIZE(I) - an arbitrary numbers of lines of input, terminated by a blank line.

If there is no colloid transport, TRANSFLAG > 0

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC),
 TCLZ(ITRC), DIFFMAT(ITRC), RD_FRAC(ITRC),
 MATRIX_POR(ITRC), APERTURE(ITRC)

or when 'afm' is implemented without colloid transport

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC),
 TCLZ(ITRC), DIFFMAT(ITRC), RD_FRAC(ITRC),
 MATRIX_POR(ITRC), APERTURE(ITRC), SRESIDUAL(ITRC),
 GAMMA_AFM(ITRC)

or when colloid transport is enabled, TRANSFLAG < 0

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC),
 TCLZ(ITRC), DIFFMAT(ITRC), RD_FRAC(ITRC),
 MATRIX_POR(ITRC), APERTURE(ITRC), KCOLL(ITRC),
 RCOLL(ITRC), FCOLL(ITRC)

or when 'afm' is implemented with colloid transport

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC),
TCLZ(ITRC), DIFFMAT(ITRC), RD_FRAC(ITRC),
MATRIX_POR(ITRC), APERTURE(ITRC), KCOLL(ITRC),
RCOLL(ITRC), FCOLL(ITRC), SRESIDUAL(ITRC),
GAMMA_AFM(ITRC)

Group 5 - JA, JB, JC, ITRC (JA, JB, JC - defined on page 31)

Group 6 - JA, JB, JC, PCNSK, T1SK, T2SK (JA, JB, JC - defined on
page 31)

If POUT = 5, an additional Group is included at the end of the ptrk input

Group 7 - JA, JB, JC, NODEPCONC (JA, JB, JC - defined on page 31)

The concentration output is written to the ".trc", ".out", and AVS
concentration output files. The ".fin" file is used only when specified (a non-
zero value is input for PRNT_RST).

Input Variable	Format	Description
NPART	integer	Number of particles in the simulation. Note: the actual number may be slightly less than the number specified by the user because when the code divides the particles among the starting nodes as specified in Group 7, the code must input an integer number of particles at each node.
RSEED	integer	6-digit integer random number seed.
KEYWORD	character*4	Optional keyword 'rip' designating this is a GoldSim coupled simulation and the alternate Group 2 data format should be used.
DAYCS	real	Time which the particle tracking solution is enabled (days).
DAYCF	real	Time which the particle tracking solution is disabled (days).
DAYHF	real	Time which the flow solution is disabled (days).
DAYHS	real	Time which the flow solution is enabled (days).
RIPFEHM	integer	Parameter for assigning the particle starting locations based on radionuclide flux input from the computer code GoldSim. Used when GoldSim is the driver program and FEHM is dynamically linked to perform particle tracking transport. If RIPFEHM = 1 Use input from GoldSim to assign particle starting locations. If RIPFEHM ≠ 1 Assign starting locations in the normal way.
CONFREAD	real	Initial conversion factor for GoldSim-FEHM coupling (# of particles/mole). If CONFREAD=0, at each time step, the code selects a conversion factor based on the available memory and the remaining simulation time (end time - current time). The code then uses the selected conversion factor to calculate the number of particles to be injected at the current time step. If CONFREAD >0, the code uses the product of the CONFREAD and the input mass (in moles) to calculate the input number of particles at each time step.

Input Variable	Format	Description
GMOL	real	The molecular weight of the <i>i</i> th species. The code uses GMOL and CONFREAD to convert the mass from number of particles to grams in the final output.
P_FRACTION	real	The decay-ingrowth particle release factor (percentage of the maximum number of particles released for the parent species). Values are in the range 0. - 1. If the value is omitted it will default to 0.5.
TRAK_TYPE	integer	Flag to denote the fluid phase of the particles: 1 - liquid phase particles 2 - vapor phase particles
HALF_LIFE	real	Half-life for irreversible first order decay reaction (s). Set HALF_LIFE = 0 for no decay.
POUT	integer	Flag to specify the concentration output: 0 - Concentration output is a running total of the number of particles which have left each node, divided by the fluid or vapor mass at that node depending on trak_type. 1 - Concentrations computed as number of particles per unit total volume (rock and fluid) 2 - Concentrations computed as number of particles per unit fluid volume (the fluid is liquid for TRAK_TYPE = 1 and gas for TRAK_TYPE = 2). 3 - Concentrations computed as number of particles at a given node point. 5 - If this option is invoked, the particles injected at a particular node are assigned concentrations according to the input concentration defined in the NODEPCONC array (Group 7). The code then outputs the mixed mean concentration at each node in the model based on the assumption of steady state flow. 6 - Used for C-14 radioactive decay particle mixing model (only liquid tracer). For meaningful results the particles must all be injected simultaneously in a pulse (give a very short duration of injection starting at time 0). The code contains data describing the function $f(t)$ vs. time where $f(t)$ is given as: $\int_0^t \exp(-kt) dt$ where t is the time the particle enters the system and k is the radioactive decay constant for C-14. The output is the final concentration after all the particles have left the system. -1, -2, -3, or -6 - Concentrations computed as specified above for abs(pout). The ".trc" file contains breakthrough output for the first node specified in the node macro. -7 -Output is written every time a particle leaves a cell. This output is particle number, cell number that the particle is leaving, zone number of the cell, and time the particle leaves the cell.

Input Variable	Format	Description
PRNT_RST	integer	<p>Flag to specify whether particle information is written to the “.fin” file:</p> <ul style="list-style-type: none"> 0 - Particle information is not written to “.fin” file. 1 - Particle information is written to the “.fin” file. 2 - Mass flux values are written to the “.fin” file followed by particle information. -1 - Particle positions and ages only are written to the “.fin” file. -2 - Mass flux values are written to the “.fin” file followed by particle positions and ages only. <p>When particle tracking data or mass fluxes are written to the “.fin” file, the arrays are written after all of the heat and mass simulation information. The mass fluxes can be read into the code in a subsequent ptrk or mptr simulation and the code can simulate transport on this steady state flow field (see macro rflo). The particle information written is sufficient to perform a restart of the particle tracking simulation and to post-process the data to compile statistics on the particle tracking run. However, for a large number of particles, this file can become quite large, so particle tracking information should only be written when necessary. Thus, 0 should be used for PRNT_RST unless restarting or post-processing to obtain particle statistics is required. Selecting the -1 option allows a subset of the full set of information needed for a restart (particle positions and ages) to be written. Restart runs that use this file as input will only be approximate, since the particle is assumed to have just entered its current cell. For restart runs, PRNT_RST = 1 is preferred, while PRNT_RST = -1 is appropriate for output of particle statistics for post-processing.</p>
KEYWORD	character	Optional keyword “tcurve” indicating transfer function curve data should be input to model matrix diffusion. If the keyword is found then NUMPARAMS and FILENAME are entered, otherwise they are omitted.
NUMPARAMS	integer	Number of parameters that define the transfer function curves being used.
TFILENAME	character	Name of input file containing the transfer function curve data.
KEYWORD	character*4	Optional keyword ‘zptr’ designating zones for breakthrough curves will be defined. If no keyword is input, IPZONE and IDZONE are also omitted.
IPZONE	integer	Number of zones for which breakthrough curves are to be output
IDZONE	integer	A list of zones for which particle breakthrough data are required. The code outputs the number of particles that leave the system at each zone IDZONE at the current time step. This information is written to the “.out” file at each heat and mass transfer time step.
KEYWORD	character*4	Optional keyword ‘file’ designating alternate transport parameter file input for multiple simulation realizations.
PFILENAME	character*80	Name of file from which to read transport parameters.

Input Variable	Format	Description
NINPUTS	integer	Number of inputs in each row of data in the alternate transport parameter input file (PFILENAME).
PTRPARAM	real	Array of transport parameter values in the alternate transport parameter input file (PFILENAME). The parameters that may be input are those entered for Group 4. Only those parameters being changed at each realization need be entered.
KEYWORD	character*4	Optional keyword 'afm' designating the Active Fracture Model input for determining fracture spacing should be used.
KEYWORD	character*4	Optional keyword 'size' designating that the colloid size distribution model option is being used (combined with the interface filtration option in the itfc macro). If the keyword is not input, PART_SIZE and PROBSIZE are also omitted.
PART_SIZE	real	Colloid particle size for this entry of the particle size distribution table (paired with a value of PROBSIZE). An arbitrary number of entries can be input, terminated with a blank line. The code assigns each particle a size based on this distribution of particle sizes, and decides if particles are irreversibly filtered based on the pore size distribution assigned in the itfc macro.
PROBSIZE	real	Colloid cumulative probability for the distribution of sizes (paired with a value of PART_SIZE). See description of PART_SIZE above for details. The final entry of the table must have PROBSIZE = 1, since the distribution is assumed to be normalized to unity.
TRANSFLAG	integer	<p>Flag to specify which transport mechanisms apply [abs(TRANSFLAG)]:</p> <ol style="list-style-type: none"> 1 - advection only (no dispersion or matrix diffusion) 2 - advection and dispersion (no matrix diffusion) 3 - advection and matrix diffusion, infinite fracture spacing solution (no dispersion) 4 - advection, dispersion, and matrix diffusion, infinite fracture spacing solution 5 - advection and matrix diffusion, finite fracture spacing solution (no dispersion) 6 - advection, dispersion, and matrix diffusion, finite fracture spacing solution 8 - use the the transfer function approach with 3 dimensionless parameters and type curves for handling fracture-matrix interactions. <p>For TRANSFLAG < 0, transport simulations include colloids.</p> <p>For equivalent continuum solutions, the fracture spacing in the finite spacing model is determined using</p> $SPACING = APERTURE / POROSITY .$

Input Variable	Format	Description
		For dual permeability models, the fracture spacing input parameter APUV1 in the dpdp macro is used as the half-spacing. If the Active Fracture Model (see keyword 'afm') is used, APUV1 is the geometric fracture half-spacing, and the additional terms SRESIDUAL and GAMMA_AFM are used to determine the spacing between active fractures (see below).
KD	real	Sorption coefficient (linear, reversible, equilibrium sorption). Units are kg-fluid / kg-rock (these units are equivalent to the conventional units of cc/g when the carrier fluid is water at standard conditions). This value applies to the medium as a whole when matrix diffusion is turned off, whereas for simulations invoking matrix diffusion, the value applies to the rock matrix. For the latter case, sorption in the flowing system (fractures) is modeled using the RD_FRAC variable.
TCLX	real	Dispersivity in the x-direction (m). The input value is ignored when dispersion is turned off.
TCLY	real	Dispersivity in the y-direction (m). The input value is ignored when dispersion is turned off.
TCLZ	real	Dispersivity in the z-direction (m). The input value is ignored when dispersion is turned off.
DIFFMAT	real	Molecular diffusion coefficient in the rock matrix (m ² /s). The input value is ignored unless matrix diffusion is invoked.
RD_FRAC	real	Retardation factor within the primary porosity (fractures) for a matrix diffusion particle tracking simulation (use 1 for no sorption on fracture faces). The input value is ignored unless matrix diffusion is invoked.
MATRIX_POR	real	Porosity of the rock matrix. Used to simulate diffusion and sorption in the rock matrix when matrix diffusion is invoked, otherwise the input value of MATRIX_POR is ignored.
APERTURE	real	Mean fracture aperture (m). The input value is ignored when matrix diffusion is turned off.
KCOLL	real	Colloid distribution parameter, the ratio of contaminant mass residing on colloids to the mass present in aqueous form. It is used to compute an effective aperture via the following: $APWID = APERTURE \cdot (1 + KCOLL)$
RCOLL	real	Colloid retardation factor. Used, in conjunction with kcoll, to adjust colloid retardation in fractures using the following formula: $FRACRD = \frac{RD_FRAC + KCOLL \cdot RCOLL}{1 + KCOLL}$
FCOLL	real	Colloid filtration parameter. Used to compute the probability a colloid will be irreversibly filtered along the path between two nodes using the following: $PROBFILT = 1 - \exp(DISTANCE/FCOLL)$ where <i>DISTANCE</i> is the length of the path between nodes.

Input Variable	Format	Description
SRESIDUAL	real	Residual saturation in the Active Fracture Model used for determining the spacing between active fractures. This parameter is only needed when the keyword 'afm' is included, in which case the input must be entered. However, the model is only used in dual permeability simulations at locations where the finite spacing matrix diffusion model is invoked [$\text{abs}(\text{TRANSFLAG}) = 5$ or 6].
GAMMA_AFM	real	Exponent in the Active Fracture Model used for determining the spacing between active fractures. See comments for SRESIDUAL above.
ITRC	integer	Model number for parameters defined in group 4. Default is 1.
PCNSK	real	<p>Particle injection parameter assigned for nodes defined by JA, JB, and JC. When multiple lines of input are given for Group 6, all PCNSK values must have the same sign (i.e. the two options, described below, cannot be invoked in the same simulation).</p> <p>PCNSK > 0 - particles are injected at each node in proportion to the source mass flow rate at the node. When multiple lines of input are given for Group 6, PCNSK is proportional to the particle injection concentration. This boundary condition is equivalent to injecting a solute of a given concentration into the system. Note: the source flow rates used to assign the number and timing of particle injections are those at the beginning of the particle tracking simulation (time DAYCS). Transient changes in this source flow rate during the particle tracking simulation do not change the input of particles to the system.</p> <p>PCNSK < 0 - particles are introduced at the node(s), regardless of whether there is a fluid source at the node. When multiple lines of input are given for Group 6, $\text{abs}(\text{PCNSK})$ is proportional to the number of particles introduced at the node(s).</p> <p>Default is 0 for all unassigned nodes, meaning that no particles are injected at that node.</p>
T1SK	real	Time (days) when particle injection begins. Default is 0.
T2SK	real	Time (days) when particle injection ends. Default is 0.
NODEPCONC	real	Input particle concentrations. The concentration associated with a particle entering the system at a specified node.

Notes on Restarting: As with all restart runs for FEHM, a ".ini" file is specified to be read to set the initial conditions upon restarting. However, there are two possibilities for restart calculations with particle tracking: 1) the heat and mass transfer solution is being restarted, but the particle tracking simulation is initiated during the restart run (it was not carried out in the simulation that generated the ".ini" file); or 2) the heat and mass transfer solution and the particle tracking simulation are both being restarted. If the code does not find the "ptrk" key word at the top of the ".ini" file, then the original run did not employ particle tracking, and Case 1 is assumed. A common example is a preliminary calculation that establishes a fluid flow steady state, followed by a restart simulation of transport.

If "ptrk" was written into the ".ini" file in the original run, the particle data in the ".ini" file are read and used to initialize the particle tracking simulation (Case 2). In this instance, the number of particles (NPART) must be set the same for the restart run as in the original run or the results will be unpredictable. When restarting a particle tracking simulation, certain input data are overwritten by information in the ".ini" file. These parameters include RSEED, PCNSK, T1SK, and T2SK. Other input parameters can be set to different values in the restart run than they were in the original run, but of course care must be taken to avoid physically unrealistic assumptions, such as an abrupt change in transport properties of Group 4 part way through a simulation.

A final note on restart calculations is in order. A common technique in FEHM restart calculations is to reset the time at the top of the ".ini" file to 0, so that the starting time of the restart simulation is arbitrarily 0, rather than the ending time of the original simulation. This is useful for the example of the steady state flow calculation, followed by a restart solute transport calculation. Although this technique is acceptable for particle tracking runs that are initiated only upon restart (Case 1), it is invalid when a particle tracking run is being resumed (Case 2). The reason is that all particle times read from the ".ini" file are based on the starting time of the original simulation during which the particle tracking simulation was initiated.

The following is an example of **ptrk**:

ptrk									
100000	122945								
10.	20.	10.	20.						
1	0	2	0						
4	0.	2.	2.	2.	5.e-11	1.	0.1	0.333	
4	3.	2.	2.	2.	1.e-10	1.	0.28	2.	
1	0	0	1						
-2	0	0	2						
-3	0	0	1.	10.	10.0001				

In this example, 100,000 nondecaying, liquid-borne particles are introduced as a sharp pulse (from time 10 to 10.0001 days) with the injection fluid in zone 3 (an injection well defined in the **zone** macro preceding **ptrk**). The particle tracking simulation starts as the heat and mass transfer simulation is turned off at day 10, after having established a fluid flow steady state. Two models are defined for assigning transport properties of the particles. All nodes are assigned to model 1, after which model 2 properties are assigned for zone 2. A combined advection, dispersion, and matrix diffusion model is used for all nodes. However, sorption in the matrix occurs only for model 2 (which is zone 2 in this simulation), and the matrix transport properties (porosity, fracture spacing, diffusion coefficient) differ for this model as well.

6.2.55 Control statement renu (optional)

Renumbers the nodes. This option should only be used by someone familiar with the linear equation solver routines and grid renumbering techniques.

Group 1 - IIRB(I) I = 1, NEQ

Input Variable	Format	Description
IIRB	integer	New node number for given node. A value is entered for each node.

6.2.56 Control statement rflo (optional)

This option is available for single continuum or double porosity/double permeability (dpdp) models only. No input is associated with this macro.

When invoked, the code will read mass flux values (kg/s) from the ".ini" file. Mass flux values are read for each connection of each node, starting with node 1. The mass flux values include sources and sinks for each node. Flow into a node is negative, and flow out of a node is positive. The mass flux values for the fracture domain are read first followed by the mass flux values in the matrix domain. The mass flux between fracture and matrix elements are read last. Flow from the fracture to the matrix is denoted as positive.

Mass flux values can be generated during a previous FEHM particle tracking simulation (see macro **ptrk**) or may be generated by an alternate flow code (see Ho, 1997).

6.2.57 Control statement rlp (optional)

Relative permeability and capillary pressure model. Several models are available.

Group 1 - IRLP(i), RP1, RP2, RP3, RP4, RP5, RP6, RP7, RP8, RP9, RP10, RP11, RP12, RP13, RP14, RP15, RP16 (number of parameters entered depends on model selected)

Group 2 - JA, JB, JC, I (JA, JB, JC - defined on page 31)

Only those parameters defined for a given model need to be input. Group 1 is ended when a blank line is encountered. The parameter *i* is incremented each time a Group 1 line is read. Group 2 lines will refer to this parameter. For model number 4 (the combined van Genuchten model), the permeability is isotropic and overwrites the input from macro **perm**.

Input Variable	Format	Description
IRLP(i)	integer	Relative permeability model type.
Model -1: IRLP(i) = -1, constant relative permeability, linear capillary pressure (4 parameters required).		
RP1	real	Liquid relative permeability (m ²).
RP2	real	Vapor relative permeability (m ²).
RP3	real	Capillary pressure at zero saturation (MPa).
RP4	real	Saturation at which capillary pressure goes to zero.

Input Variable	Format	Description
Model 1: IRLP(i) = 1, linear relative permeability, linear capillary pressure (6 parameters required).		
RP1	real	Residual liquid saturation.
RP2	real	Residual vapor saturation.
RP3	real	Maximum liquid saturation.
RP4	real	Maximum vapor saturation.
RP5	real	Capillary pressure at zero saturation (MPa).
RP6	real	Saturation at which capillary pressure goes to zero.
Model 2: IRLP(i) = 2, Corey relative permeability, linear capillary pressure (4 parameters required).		
RP1	real	Residual liquid saturation.
RP2	real	Residual vapor saturation.
RP3	real	Capillary pressure at zero saturation (MPa).
RP4	real	Saturation at which capillary pressure goes to zero.
Model 3: IRLP(i) = 3, van Genuchten relative permeability, van Genuchten capillary pressure (6 parameters required). In this model permeabilities are represented as a function of capillary pressure [rlp(h)].		
RP1	real	Residual liquid saturation.
RP2	real	Maximum liquid saturation.
RP3	real	Inverse of air entry head, α_G (1/m) [note some data is given in (1/Pa) convert using pressure = $\rho g \Delta h$].
RP4	real	Power n in van Genuchten formula.
RP5	real	Low saturation fitting parameter, multiple of cutoff capillary pressure assigned as maximum capillary pressure. If $RP5 < 0$ then a linear fit from the cutoff saturation (RP6) is used. The slope of the cutoff saturation is used to extend the function to saturation = 0. If $RP5 = 0$, a cubic fit is used. The slope at the cutoff saturation is matched and the conditions $\frac{\partial P_{cap}}{\partial S} = 0$ and $\frac{\partial^2 P_{cap}}{\partial S^2} = 0$ are forced at $S = 0$. If $RP5 > 0$, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet P_{cap}(S_{cutoff})$ is forced at $S = 0$.
RP6	real	Cutoff saturation used in fits described for RP5, must be greater than RP1.

Input Variable	Format	Description
Model 4: IRLP(i) = 4, van Genuchten relative permeability, van Genuchten capillary pressure, effective continuum (15 parameters required). In this model permeabilities are represented as a function of capillary pressure [rIp(h)].		
RP1	real	Residual liquid saturation, matrix rock material.
RP2	real	Maximum liquid saturation, matrix rock material.
RP3	real	Inverse of air entry head, α_G (1/m) [note some data is given in (1/Pa) convert using pressure = $\rho g \Delta h$], matrix rock material.
RP4	real	Power n in van Genuchten formula, matrix rock material.
RP5	real	<p>Low saturation fitting parameter, matrix rock material, multiple of cutoff capillary pressure assigned as maximum capillary pressure.</p> <p>If $RP5 < 0$ then a linear fit from the cutoff saturation (RP6) is used. The slope of the cutoff saturation is used to extend the function to saturation = 0.</p> <p>If $RP5 = 0$, a cubic fit is used. The slope at the cutoff saturation is matched and the conditions $\frac{\partial P_{cap}}{\partial S} = 0$ and $\frac{\partial^2 P_{cap}}{\partial S^2} = 0$ are forced at $S = 0$.</p> <p>If $RP5 > 0$, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet P_{cap}(S_{cutoff})$ is forced at $S = 0$.</p>
RP6	real	Cutoff saturation used in fits described for RP5, must be greater than RP1.
RP7	real	Residual liquid saturation, fracture material.
RP8	real	Maximum liquid saturation, fracture material.
RP9	real	Inverse of air entry pressure, α_G (1/m) [note some data is given in (1/Pa) convert using pressure = $\rho g \Delta h$], fracture material.
RP10	real	Power n in van Genuchten formula, fracture material.
RP11	real	<p>Low saturation fitting parameter, fracture material, multiple of cutoff capillary pressure assigned as maximum capillary pressure.</p> <p>If $RP11 < 0$ then a linear fit from the cutoff saturation (RP12) is used. The slope of the cutoff saturation is used to extend the function to saturation = 0.</p> <p>If $RP11 = 0$, a cubic fit is used. The slope at the cutoff saturation is matched and the conditions $\frac{\partial P_{cap}}{\partial S} = 0$ and $\frac{\partial^2 P_{cap}}{\partial S^2} = 0$ are forced at $S = 0$.</p> <p>If $RP11 > 0$, a multiple of the value of the capillary pressure at the cutoff saturation, $RP11 \bullet P_{cap}(S_{cutoff})$ is forced at $S = 0$.</p>

Input Variable	Format	Description
RP12	real	Cutoff saturation used in fits described for RP11, must be greater than RP7.
RP13	real	Fracture permeability (m^2). This is the permeability of the individual fractures. The bulk permeability of the fracture continuum is $RP13 \times RP15$.
RP14	real	Matrix rock saturated permeability (m^2).
RP15	real	Fracture volume fraction.
<p>Model 5: IRLP(i) = 5, van Genuchten relative permeability, van Genuchten capillary pressure (6 parameters required). This model and its input are the same as for Model 3 except that permeabilities are represented as a function of saturation [rIp(S)] rather than capillary pressure.</p>		
<p>Model 6: IRLP(i) = 6, van Genuchten relative permeability, van Genuchten capillary pressure, effective continuum (15 parameters required). This model and its input are the same as for Model 4 except that permeabilities are represented as a function of saturation [rIp(S)] rather than capillary pressure.</p>		
<p>Model 7: IRLP(i) = 7, van Genuchten relative permeability, van Genuchten capillary pressure, effective continuum with special fracture interaction term (16 parameters required). This model and its input are the same as for Model 6 except that the an additional term is included which represents the fracture-matrix interaction.</p>		
RP16	real	Fracture-matrix interaction term. If $RP16 \leq 0$, then an additional multiplying term equal to the relative permeability is applied to the fracture-matrix interaction term for dual permeability problems. If $RP16 > 0$, then an additional multiplying term equal to $sI^{**}RP16$ and $(1. - sI)^{**}RP16$ is applied to the fracture-matrix interaction terms for the liquid and vapor phases, respectively, for dual permeability problems. Here, sI is the value of saturation at the given node.
<p>Model 10: IRLP(i) = 10, linear relative permeability with minimum relative permeability values, linear capillary pressure (8 parameters required).</p>		
RP1	real	Residual liquid saturation.
RP2	real	Residual vapor saturation.
RP3	real	Maximum liquid saturation.
RP4	real	Maximum vapor saturation.
RP5	real	Minimum liquid permeability (m^2).
RP6	real	Minimum vapor permeability (m^2).
RP7	real	Capillary pressure at zero saturation (MPa).
RP8	real	Saturation at which capillary pressure goes to zero.

The following is an example of **rlp**. In this example, Corey type relative permeability is specified, with residual liquid saturation of 0.3, residual vapor saturation of 0.1, a base capillary pressure of 2 MPa, and capillary pressure goes to zero at a saturation of 1. This model is assigned to nodes numbered 1 through 140.

rlp	2	0.3	0.1	2.0	1.
	1	140	1	1	

6.2.58 Control statement rock (required)

Assign rock density, specific heat and porosity.

Group 1 - JA, JB, JC, DENRD, CPRD, PSD (JA, JB, JC - defined on page 31)

Input Variable	Format	Description
DENRD	real	Rock density (kg/m ³).
CPRD	real	Rock specific heat ($\frac{\text{MJ}}{\text{kg} \cdot \text{K}}$). If CPRD > 1 the code will assume the units are ($\frac{\text{J}}{\text{kg} \cdot \text{K}}$) and multiply by 10 ⁻⁶ .
PSD	real	Porosity. Special note on negative porosities. If the code encounters a negative porosity, the node at which the negative porosity occurs is effectively removed from the model. That is, the geometric connections from that node to other nodes in the model are removed. The volume associated with the node acts as a barrier to flow. For input purposes, the node may still be assigned properties, though they will have no effect on the simulation results.

The following is an example of **rock**. In this example the nodes numbered 1 through 140 are assigned a rock density of 2563. kg/m³, a rock specific heat of 1010. J/(kg K) and a porosity of 0.35.

rock	1	140	1	2563.	1010.	0.35
------	---	-----	---	-------	-------	------

6.2.59 Control statement rxn (optional)

Chemical reactions between components are invoked with this control statement. It is used in conjunction with control statement **trac**. For facilitating the construction of the **rxn** input, a header describing the input is required before each group whether data is entered for that group or not (see examples) unless otherwise noted. The header is an arbitrary text string that must be contained on a single line. Note that for components that do not react with other components, **rxn** is unnecessary. Specifically, conservative tracers or tracers that follow the equilibrium sorption

isotherms can be modeled with just the **trac** macro. Note the parameters NCPNT, NIMM, NVAP (used as indices for input) are determined by the code using information input for the **trac** macro. NCPNT is equal to the number of liquid components, NIMM is equal to the number of immobile components, and NVAP is equal to the number of vapor components specified in the **trac** macro.

Group 1 - NCPLX, NUMRXN

Group 2 - NGROUPS

GROUP (ICPNT), ICPNT = 1, NCPNT (repeated NGROUPS times, once for each group).

Group 3 - IDCPNT, CPNTNAM, IFXCONC, CPNTPRT, CPNTGS (repeated NCPNT times)

Group 4 - IDCPLX, CPLXNAM, CPLXPRT (repeated NCPLX times)

Group 5 - IDIMM, IMMNAM, IMMVRT (repeated NIMM times;)

Group 6 - IDVAP, VAPNAM, VAPPRT (repeated NVAP times)

Group 7 - ISKIP

Group 8 - RSDMAX

HEADING

Note that this is an additional heading line that precedes the LOGKEQ heading.

Group 9 - LOGKEQ

Group 10 - CKEQ, HEQ (NCPLX times)

or

Group 10 - KEYWORD, NUM_TEMPS

EQTEMP(I), I = 1, NUM_TEMPS

LKEQ(I), I = 1, NUM_TEMPS

For group 10, the keyword 'lookup' can be used in place of CKEQ and HEQ. LOOKUP allows a lookup table to be used to describe the equilibrium constant (K) as a function of temperature. After the keyword, the user must specify the number of values of temperature and K that will be used to describe K as a function of temperature. On the next lines the temperatures, and then the K values are entered. FEHM performs a piecewise linear interpolation between the values given.

Group 11 -STOIC(ICPNT), ICPNT = 1, NCPNT (repeated NCPLX times, once for each aqueous complex)

Input for groups 9, 10 and 11 is omitted if NCPLX, the number of aqueous complexes, is zero.

The remaining groups are entered as a unit for each kinetic reaction. If there are no kinetic reactions specified, none of the following groups (or their headers) are input. The input for groups 14 and on, depend on the kinetic reaction type specified in group 12. The input for each kinetic reaction type is described below.

Group 12 - IDRXXN

Group 13 - JA, JB, JC (JA, JB, JC - defined on page 22)

IDRXXN = 1: Linear kinetic reaction

Group 14 - IAQUEOUS, IIMMOBILE

Group 15 - KD

or

Group 15 - KEYWORD, NUM_TEMPS

EQTEMP(I), I = 1, NUM_TEMPS

TCOEFF(I), I = 1, NUM_TEMPS

NOTE: The distribution coefficient can be replaced with the keyword 'lookup' for temperature dependent coefficients. The keyword is described in Group 10.

Group 16 - RATE

IDRXXN = 2: Langmuir kinetic reaction

Group 14 - IAQUEOUS, IIMMOBILE

Group 15 - DISTCOEFF

or

Group 15 - KEYWORD, NUM_TEMPS

EQTEMP(I), I = 1, NUM_TEMPS

TCOEFF(I), I = 1, NUM_TEMPS

NOTE: The distribution coefficient can be replaced with the keyword 'lookup' for temperature dependent coefficients. The keyword is described in Group 10.

Group 16 - RATE

Group 17 - MAXCONC

IDRXXN = 3: General reaction

Group 14 - NIMMOBILE, NAQUEOUS, NVAPOR

Group 15 - KFOR, KREV

Group 16 - IIMMOBILE (I = 1, NIMMOBILE)

Group 17 - IMSTOIC (I = 1, NIMMOBILE)

Omit groups 16 and 17 (including headers) if NIMMOBILE is zero.

Group 18 - IAQUEOUS (I = 1, NAQUEOUS)

Group 19 - AQSTOIC (I = 1, NAQUEOUS)

Omit groups 18 and 19 (including headers) if NAQUEOUS is zero.

Group 20 - IVAPOR (I = 1, NVAPOR)

Group 21 - IVSTOIC (I = 1, NVAPOR)

Omit groups 20 and 21 (including headers) if NVAPOR is zero.

IDRXN = 4: Dual Monod kinetics biodegradation reaction

Group 14 - NAQUEOUS (must be >2 and < 5), NIMMOBILE

Group 15 - SUBSTRATE, ELECACC, COMP3, COMP4, COMP5

Note that the users choice of NAQUEOUS determines whether COMP3, COMP4, and COMP5 need to be entered. NIMMOBILE must be 1. NAQUEOUS also determines whether the COMP3, COMP4, and COMP5 stoichiometries are to be used by the code. However values must always be entered for groups 21-23 regardless of the value of NAQUEOUS.

Group 16 - BIOMASS

Group 17 - KS

Group 18 - KA

Group 19 - DECAY

Group 20 - ELECACCSTOIC

Group 21 - COMP3STOIC

Group 22 - COMP4STOIC

Group 23 - COMP5STOIC

Group 24 - PHTHRESH

Group 25 - QM

Group 26 - YIELD

Group 27 - XMINIT

Group 28 - NBIOFRM

Omit Group 29 if NBIOFRM = 0

Group 29 - ICBIO (I = 1, NBIOFRM)

IDRXN=5: Radioactive Decay Reaction

Group 14 - HALFLIFE

Group 15 - RXNTYPE

Group 16 - PARENT, DAUGHTER

IDRXN=6: Kinetic Henry's Law reaction

Group 14 - IAQUEOUS, IVAPOR

Group 15 - KH

Group 16 - RATE

IDRXN = 7: Kinetic precipitation/dissolution reaction for total component concentrations

Group 14 - IIMMOBILE

Group 15 - NAQUEOUS

Group 16 - IAQUEOUS (I = 1, NAQUEOUS)

Group 17 - IMSTOIC

Group 18 - AQSTOIC (I = 1, NAQUEOUS)

Group 19 - SOLUBILITY

or

Group 19 - KEYWORD, NUM_TEMPS

EQTEMP(I), I = 1, NUM_TEMPS

TCOEFF(I), I = 1, NUM_TEMPS

NOTE: Solubility can be replaced with the keyword 'lookup' for temperature dependent solubilities. The keyword is described in Group 10.

Group 20 - RATE

Group 21 - SAREA

IDRXN = 8: Kinetic precipitation/dissolution reaction for total component concentrations with rates based on free-ion concentrations

NOTE: The input is identical to that for reaction model 7.

Input Variable	Format	Description
NCPLX	integer	Number of aqueous complexes (equal to number of equilibrium reactions)
NUMRXN	integer	Number of kinetic reactions
NGROUPS	integer	Number of groups. See GROUP to determine how to set this parameter.
GROUP	integer	This variable controls the selective coupling solution method employed by FEHM. NCPNT values are entered for each line of input, and NGROUPS lines of input are required, one for each group. If a value is non-zero, then that aqueous component is present in the group. A value of zero denotes that the species is not present in the group. Grouping of aqueous components that take part in rapid kinetic reactions is required for convergence. However, memory requirements increase as the square of the maximum number of aqueous components in a group.
IDCPNT	integer	For each total aqueous component, the number identifying each total aqueous component (e.g. 1, 2, etc.)
CPNTNAM	character	For each total aqueous component, the name of the total aqueous component (e.g. Sulfate)
IFXCONC	integer	For each total aqueous component, the Flag denoting the type of total aqueous component 1 - total aqueous concentration is specified in TRAC macro 2 - specify log of free ion concentration in TRAC macro (use for pH). For example, if H+ is the component, IFXCONC of 2 allows for pH to be directly input in place of concentration values in the TRAC macro.

Input Variable	Format	Description
CPNTPRT	integer	For each total aqueous component, the Flag denoting which total aqueous component concentrations are printed to the ".trc" file and the AVS files. 0 - Print to file 1 - Do not print to file
CPNTGS	real	Guess for the initial uncomplexed component concentration used in speciation reactions. We recommend 1.0e-9. On rare occasions, the chemical speciation solver may have trouble converging. Choosing more representative values for CPNTGS will help convergence.
IDCPLX	integer	For each aqueous complex, the number identifying each aqueous complex. By convention, the first complex should be given the number 101, the second, 102, etc.
CPLXNAM	character	For each aqueous complex, the name of the aqueous complex (e.g. H ₂ SO ₄)
CPLXPRT	integer	For each aqueous complex, the Flag denoting which aqueous complex concentrations are printed to the ".trc" file and the AVS files. 0 - Print to file 1 - Do not print to file
IDIMM	integer	For each immobile component, the number identifying each immobile component (e.g. 1, 2, etc.)
IMMNAM	character	For each immobile component, the name of the immobile component (e.g. Calcite, Co[adsorbed])
IMMPRT	integer	For each immobile component, the Flag denoting which immobile component concentrations are printed to the ".trc" file and the AVS files. 0 - Print to file 1 - Do not print to file
IDVAP	integer	For each vapor component, the number identifying each vapor component (e.g. 1, 2, etc.)
VAPNAM	character	For each vapor component, the name of the vapor component (e.g. CO ₂ [gas])
VAPPRT	integer	For each vapor component, the Flag denoting which vapor component concentrations are printed to the ".trc" file and the AVS files. 0 - Print to file 1 - Do not print to file
ISKIP	integer	Flag denoting whether chemical speciation calculations should be done at nodes which have already converged in a previous transport iteration 0 - Do chemical speciation calculations at each node for every iteration (recommended option)

Input Variable	Format	Description
		1 - To save computational time, this options tells FEHM to do equilibrium speciation calculations only at nodes which have not converged during the previous transport iteration. Sometimes, this option can lead to mass balance errors (mass balances can be checked in the ".out" file to see if results are satisfactory). This option is only recommended for very large problems.
RSDMAX	real	The tolerance for the equilibrium speciation calculations. We recommend 1×10^{-9} for most problems.
HEADING	character	One line descriptive comment which precedes the LOGKEQ heading
LOGKEQ	integer	Flag denoting the whether K or log K is entered by the user 0 - constants are given as K 1 - constants are given as log K
CKEQ	real	For each aqueous complex, the equilibrium constant
HEQ	real	For each aqueous complex, the enthalpy of the equilibrium reaction. The Van Hoff equation is used to determine the value of the equilibrium constant as a function of temperature. Note the keyword 'lookup' (see page 105) can be used in the place of CKEQ and HEQ.
KEYWORD	character	Keyword 'lookup' designating a lookup table will be used to describe the equilibrium constant as function of temperature.
NUM_TEMPS	integer	Number of values of temperature and K that will be used to describe K as a function of temperature.
EQTEMP	real	Temperatures for K function ($^{\circ}\text{C}$)
LKEQ	real	Equilibrium constants for K function
STOIC	real	For each aqueous complex, the stoichiometry describing how to "make" the complex from the total aqueous components must be entered. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
IDRXN	integer	For each kinetic reaction, this parameter specifies kinetic reaction model. Currently, the following reaction models are available. Additional kinetic formulations can be added without significant code development. 1 - linear kinetic reaction 2 - langmuir kinetic reaction 3 - general kinetic reaction 4 - dual Monod biodegradation reaction 5 - radioactive decay reaction 6 - kinetic Henry's law reaction 7 - precipitation/dissolution reaction 8 - precipitation/dissolution reaction with rates based on free-ion concentration

Input Variable	Format	Description
JA, JB, JC	integer	JA, JB, JC are described on page 22. Here these parameters are used to specify the nodes at which the current kinetic reaction takes place. If the reaction takes place throughout the problem domain simply enter 1 0 0.
IDRXN = 1: Linear Kinetic Reaction		
IAQUEOUS	integer	The aqueous component number (e.g. 1, 2, etc.) or the aqueous complex number (e.g. 101, 102, etc.) which corresponds to the sorbing component
IIMMOBILE	integer	The immobile component number (e.g. 1, 2, etc.) which corresponds to the sorbed component
KD	real	Distribution coefficient (kg water / kg rock)
KEYWORD	character	Keyword 'lookup' designating a lookup table will be used to describe the distribution coefficient as a function of temperature.
NUM_TEMPS	integer	Number of values (temperatures and distribution coefficients) that will be used to describe KD as a function of temperature.
EQTEMP	real	Temperatures for distribution coefficient function (°C)
TCOEFF	real	Distribution coefficients corresponding to temperatures.
RATE	real	Reaction rate parameter (1/hr)
IDRXN = 2: Langmuir Kinetic Reaction		
IAQUEOUS	integer	The aqueous component number (e.g. 1, 2, etc.) or the aqueous complex number (e.g. 101, 102, etc.) which corresponds to the sorbing component
IIMMOBILE	integer	The immobile component number (e.g. 1, 2, etc.) which corresponds to the sorbed component
DISTCOEFF	real	Distribution coefficient (kg water/ moles)
KEYWORD	character	Keyword 'lookup' designating a lookup table will be used to describe the distribution coefficient as a function of temperature.
NUM_TEMPS	integer	Number of values (temperatures and distribution coefficients) that will be used to describe DISTCOEFF as a function of temperature.
EQTEMP	real	Temperatures for distribution coefficient function (°C)
TCOEFF	real	Distribution coefficients for DISTCOEFF as a function of temperature.
RATE	real	Reaction rate parameter (1/hr)
MAXCONC	real	Maximum concentration (moles/kg rock)
IDRXN = 3: General Kinetic Reaction		
NIMMOBILE	integer	The number of immobile components which participate in the reaction

Input Variable	Format	Description
NAQUEOUS	integer	The number of aqueous components and complexes which participate in the reaction
NVAPOR	integer	The number of vapor species which participate in the reaction
KFOR	real	The forward reaction rate parameter. $[(\text{concentration units})^p \times \text{s}]^{-1}$, where p is the sum of the exponents of all concentrations in the forward reaction minus 1. Thus the units of the reaction rate are (concentration units)/hr.
KREV	real	The reverse reaction rate parameter. $[(\text{concentration units})^p \times \text{s}]^{-1}$, where p is the sum of the exponents of all concentrations in the reverse reaction minus 1. Thus the units of the reaction rate are (concentration units)/hr.
IIMMOBILE	integer	The immobile component numbers which correspond to the immobile reactants and products in the reaction
IMSTOIC	real	The stoichiometry corresponding to each immobile component participating in the reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
IAQUEOUS	integer	The aqueous component or aqueous complex numbers which correspond to the aqueous reactants and products in the reaction
AQSTOIC	real	The stoichiometry corresponding to each aqueous component or aqueous complex participating in the reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
IVAPOR	integer	The vapor component numbers which correspond to the vapor reactants and products in the reaction.
IVSTOIC	real	The stoichiometry corresponding to each vapor component participating in the reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
IDRXN = 4: Dual Monod Biodegradation Reaction		
NAQUEOUS	integer	The number of aqueous species which participate in the reaction. At least 2 aqueous species must participate, the substrate (e.g. organic carbon) and the electron acceptor (e.g. Oxygen). Up to 5 aqueous species can participate. The third, fourth and fifth aqueous components are either reactants or products of the biodegradation reaction. The value entered for NAQUEOUS determines whether COMP3, COMP4, and COMP5 stoichiometries are to be used by the code.
NIMMOBILE	integer	The number of immobile components which participate in the reaction. For the biodegradation reaction this value is 1.
SUBSTRATE	integer	The aqueous component number which corresponds to the substrate (a.k.a the electron donor) for the biodegradation reaction

Input Variable	Format	Description
ELECACC	integer	The aqueous component number which corresponds to the electron acceptor for the biodegradation reaction
COMP3	integer	The aqueous component number which corresponds to a reactant or product in the biodegradation reaction (e.g. CO ₂ , NH ₃ , H ⁺ , etc.). Note that this parameter is optional. COMP3 should only be entered if NAQUEOUS>2.
COMP4	integer	The aqueous component number which corresponds to a reactant or product in the biodegradation reaction (e.g. CO ₂ , NH ₃ , H ⁺ , etc.). Note that this parameter is optional. COMP4 should only be entered if NAQUEOUS>3.
COMP5	integer	The aqueous component number which corresponds to a reactant or product in the biodegradation reaction (e.g. CO ₂ , NH ₃ , H ⁺ , etc.) Note that this parameter is optional. COMP5 should only be entered if NAQUEOUS>4.
BIOMASS	real	The solid component number which corresponds to the biomass (this is the immobile component).
KS	real	The Monod half-maximum-rate concentration for the substrate (moles/ kg water)
KA	real	The Monod half-maximum-rate concentration for the electron acceptor (moles/ kg water)
DECAY	real	First order microbial decay coefficient (hr ⁻¹)
ELECACCSTOIC	real	The stoichiometry corresponding to the electron acceptor. Note that the stoichiometry of the substrate is 1 by definition.
COMP3STOIC	real	The stoichiometry corresponding to COMP3. A value is always entered whether or not it is used.
COMP4STOIC	real	The stoichiometry corresponding to COMP4. A value is always entered whether or not it is used.
COMP5STOIC	real	The stoichiometry corresponding to COMP5. A value is always entered whether or not it is used.
PHTHRESH	real	In many systems, the biodegradation reaction will stop as the pH becomes either too acidic or basic. This parameter can be used to stop the biodegradation reaction once the simulated pH approaches a certain value. For example, if PHTHRESH = 10, the biodegradation reaction will cease if the pH is above 10 in the simulation. Note that PHTHRESH is an upper threshold for pH.
QM	real	The maximum specific rate of substrate utilization (moles/kg biomass/hr)
YIELD	real	The microbial yield coefficient (kg biomass/mole substrate)

Input Variable	Format	Description
XMINIT	real	In many systems, biomass does not decay below a certain concentration. The biomass concentration is not allowed to fall below XMINIT (moles/ kg rock).
NBIOFRM	integer	Depending on the problem setup, many aqueous complexes can be formed from a total aqueous component. Only some of these complexes may be biodegradable. This parameter is used to specify which forms (the total aqueous concentration, the free ion concentration, or various complex concentrations) of the component degrade. If NBIOFRM = 0, the total aqueous concentration of the component will degrade and the next parameter ICBIO should be omitted. If NBIOFRM = 2, then two forms of this component degrade. These forms are specified using ICBIO.
ICBIO	integer	Specify the aqueous component numbers (e.g. 1) or aqueous complex numbers (e.g. 101, 102, etc.) corresponding to the biodegradable form of the substrate.
IDRXN = 5: Radioactive Decay Reaction		
HALFLIFE	real	Half life (years)
RXNTYPE	integer	Flag denoting the type of component participating in the reaction: 0 - Solid 1 - Liquid -1 - Vapor
PARENT	integer	The number of the component which corresponds to the parent in the radioactive decay reaction
DAUGHTER	integer	The number of the component which corresponds to the daughter in the radioactive decay reaction. If the simulation does not model the daughter product set daughter = 0.
IDRXN = 7: Kinetic Precipitation/Dissolution Reaction		
IIMMOBILE	integer	The immobile component number (e.g. 1, 2, etc.) which corresponds to the dissolving mineral
NAQUEOUS	integer	The number of aqueous species which participate in the reaction
IAQUEOUS	integer	The aqueous component numbers which correspond to the aqueous components which enter into the solubility product expression. Note that the total aqueous concentration of the component will dissolve.
IMSTOIC	real	The stoichiometry corresponding to the immobile component participating in the reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.

Input Variable	Format	Description
AQSTOIC	real	The stoichiometry corresponding to the aqueous components participating in the reaction. If positive, the solute is a reactant; if negative, the solute is a product; and if 0, the solute is not present in the reaction.
SOLUBILITY	real	The solubility product. The units of the solubility product depend on the number of aqueous components participating in the reaction. For example, if there are two aqueous components participating the units would be (moles ² /[kg water] ²)
KEYWORD	character	Keyword 'lookup' designating a lookup table will be used to describe the solubility as a function of temperature.
NUM_TEMPS	integer	Number of values of temperature and solubility that will be used to describe solubility as a function of temperature.
EQTEMP	real	Temperatures for solubility function (°C)
TCOEFF	real	Solubilities for solubility as a function of temperature.
RATE	real	Reaction rate parameter (moles/[m ² s])
SAREA	real	Surface area of the mineral (m ² /m ³ rock)
IDRXN = 8: Kinetic Precipitation/Dissolution Reaction (rates based on free-ion concentrations)		
This model and its input are the same as for IDRXN = 7 except that the rates are based on the free-ion concentration instead of the total concentration.		

In general, the following examples illustrate only portions of the **rxn** macro and putting all of these example inputs together will not result in a working FEHM **rxn** macro. However, the dissolution example (the last example in this section) provides an example of a complete **rxn** macro and corresponds to the first example given for the **trac** macro (page 140). In addition, the "Reactive Transport Example" (Section 9.5) and the Validation Test Plan for the FEHM Application Version 2.21 (10086-VTP-2.21-00) include full example problems with input files which demonstrate the use of **rxn**. These input files can be used to see how **rxn** fits in with the other macros. Specifically, the information in **rxn** must be consistent with the **trac** macro. For example, if a linear kinetic sorption reaction is invoked by **rxn**, a liquid component and solid component must be specified in the **trac** macro.

General Reaction Parameters. In the following example two aqueous complexes and four kinetic reactions are specified. Three liquid components, Co, Fe, and EDTA, two complexes, CoEDTA and FeEDTA, and three immobile species, Co, CoEDTA and FeEDTA, are identified. Aqueous components 1 and 3 are coupled during solution while 2 is solved independently. Note that group 6 data has been omitted since there are no vapor species in this example.

Equilibrium Reaction Parameters. Equilibrium speciation reactions modeled by FEHM can be written in the following general form:

rxn						
** NCPLX	NUMRXN **					Group 1
2	4					
** GROUP **						Group 2
2						
1 0 1						
0 1 0						
** IDCPNT	CPNTNAM	IFXCONC	CPNTPRT	CPNTGS	**	Group 3
1	Co	0	0	1.0e-9		
2	Fe	0	0	1.0e-9		
3	EDTA	0	0	1.0e-9		
** IDCPLX	CPLXNAM	CPLXPRT	**			Group 4
101	CoEDTA	0				
102	FeEDTA	0				
** IDIMM	IMMNAM	IMMPRT **				Group 5
1	CoEDTA[s]	0				
2	FeEDTA[s]	0				
3	Cobalt[s]	0				
** IDVAP	VAPNAM	VAPPRT				Group 6

$$\sum_{j=1}^{N_c} a_{ij} \hat{C}_j \Leftrightarrow \hat{X}_i \quad i = 1, \dots, N_x \quad (1)$$

where \hat{C}_j is the chemical formula for the aqueous component j , and \hat{X}_i is the chemical formula for the aqueous complex i , a_{ij} is a stoichiometric coefficient **{STOIC}** giving the number of moles of component j in complex i , N_x is the number of aqueous complexes, and N_c is the number of aqueous components. Here is a simple example of an equilibrium speciation reaction.



where a and b are the stoichiometric coefficients of components A and B , respectively. At equilibrium, the concentrations of A , B , and A_aB_b must satisfy the law of mass action for this reaction:

$$K = \frac{[A_aB_b]}{[A]^a[B]^b} \quad (3)$$

where K is the equilibrium constant **{CKEQ}** for the reaction and $[X]$ is the molar concentration of species X . The total aqueous concentrations of components A and B are given by:

$$U_A = [A] + a[A_aB_b] \quad (4)$$

$$U_B = [B] + b[A_aB_b] \quad (5)$$

Therefore, the total aqueous concentrations, U , are functions of the uncomplexed component concentrations and the complex concentrations. Note that the kinetic reactions discussed in the next section are functions of the uncomplexed component concentrations and complex concentrations with the exception of the radioactive decay and precipitation/dissolution reaction in which the total aqueous concentration is used in the reaction rate law.

The following input describes the equilibrium speciation relations between the total aqueous components and aqueous complexes. Recall that the names of the components and aqueous complexes are given in Groups 3-6. The skip node option, ISKIP, is turned off and the tolerance for the speciation solver, RSDMAX, is set to $1e-9$ which is the recommended value. The stoichiometry, STOIC, is specified so that the components Co and EDTA make up the first complex CoEDTA, and the components Fe and EDTA make up the second complex FeEDTA. The equilibrium constants, CKEQ, for CoEDTA and FeEDTA are $1.e18$ and $6.31e27$, respectively. The enthalpy change is 0.

** ISKIP **							Group 7
0							
** RSDMAX **							Group 8
1e-9							
** Chemical Information **							
** LOGKEQ **							Group 9
0							
** CKEQ HEQ **							Group 10
1.0e18	0						
6.31e27	0						
** STOIC **							Group 11
1.0	0.0	1.0					
0.0	1.0	1.0					

Below is an example of using the 'lookup' keyword in place of CKEQ and HEQ. This example describes the temperature dependence of the equilibrium constant of complex HCO_3^- . Note that LOGKEQ = 1 in this example, so log K is specified in the input rather than K.

** LOGKEQ **							Group 9
1							
** KEYWORD NUM_TEMPS							Group 10
lookup	6						
0	25	60	100	150	200		
-6.58	-6.3447	-6.2684	-6.3882	-6.7235	-7.24		

General Kinetic Parameters. In FEHM, eight kinetic reaction models are supported. Additional kinetic subroutines can be added without significant code development. The following is an example of input for the general kinetic parameters. A linear kinetic reaction, IDRXXN = 1, is specified as occurring at each node in the problem (JA = 1, JB and JC = 0).

IDRXN = 1: Linear Kinetic Reaction. The retardation of contaminants due to adsorption/desorption can be modeled with a linear kinetic sorption/

** IDRXXN **				Group 12
1				
** JA	JB	JC **		Group 13
1	0	0		

desorption expression. The rate of adsorption/desorption of component j is given by:

$$R_j = -k_m \left(c_j - \frac{m_j}{K_D} \right) \quad (6)$$

where c_j denotes the uncomplexed aqueous concentration of j **{IAQUEOUS}**, m_j denotes the adsorbed concentration of species j **{IMMOBILE}**, k_m is the mass transfer coefficient **{RATE}**, and K_D is the distribution coefficient **{KD}**. As $k_m \rightarrow \infty$, this expression reduces to the linear equilibrium isotherm. The following example illustrates input for a linear kinetic reaction. In this kinetic reaction, aqueous component 1 adsorbs to form the immobile component 3. The K_D for the reaction is 5.07 and the mass transfer coefficient is 1.0.

** IDRXXN **				Group 12
1				
** JA	JB	JC **		Group 13
1	0	0		
** IAQ	IMMOBILE	**		Group 14
1	3			
** KD **				Group 15
5.07				
** RATE **				Group 16
1.0				

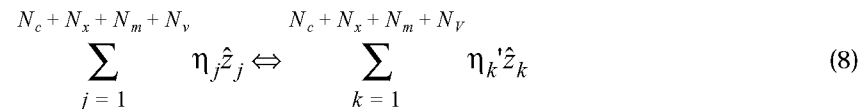
IDRXXN = 2: Langmuir Kinetic Reaction. The Langmuir kinetic reaction rate law is given by:

$$R_j = -k_m \frac{\rho}{\theta} (K_L c_j (m_j^{MAX} - m_j) - m_j) \quad (7)$$

where k_m is the rate constant for desorption **{RATE}**, ρ is the bulk rock density **{DENR}**, θ is the porosity **{POR}**. K_L is the distribution coefficient **{DISTCOEFF}**, and m_j^{MAX} is the maximum concentration that can adsorb onto the solid **{MAXCONC}**. As $k_m \rightarrow \infty$, this expression reduces to the Langmuir equilibrium isotherm. Example input for a Langmuir kinetic reaction follows. In this kinetic reaction, aqueous complex 101 sorbs to form immobile component 1. The distribution coefficient for the reaction is 2.e5 and the mass transfer coefficient is 0.05. The maximum sorbed concentration is 1.69e-5.

** IDRXXN **				Group 12
2				
** JA	JB	JC **		Group 13
1	0	0		
** IAQ	IMMOBILE	**		Group 14
101	1			
** DISTCO **				Group 15
2.0e5				
** RATE **				Group 16
0.05				
** MAXCO **				Group 17
1.69e-5				

IDRXN = 3: General Kinetic Reaction. Many reactions fall under the category of the general kinetic reaction. The reaction is described by a forward rate constant {KFOR}, a reverse rate constant {KREV}, and a set of stoichiometric coefficients. The form of the general reversible reaction is given by:



where N_c is the number of aqueous components, N_x is the number of aqueous complexes {NAQUEOUS = $N_c + N_x$ }, N_m is the number of immobile components {NIMMOBILE}, N_v is the number of vapor components {NVAPOR}, η_j are reactant stoichiometric coefficients, η_k' are product stoichiometric coefficients, and \hat{z}_i is the chemical formula for species i , which may be an uncomplexed aqueous component, aqueous complex, immobile component or vapor component. The rate law for a general reversible reaction is given by the following expression:

$$R(z_i) = (\eta_i - \eta_i') \left[k_f \prod_{j=1}^{N_c+N_x+N_m+N_v} z_j^{\eta_j} - k_r \prod_{k=1}^{N_c+N_x+N_m+N_v} z_k^{\eta_k'} \right] \quad (9)$$

where z_i is the concentration of species i . The following is example input for a general kinetic reaction. Solid component 1 reacts to form solid components 2 and 3. The forward reaction rate is 1.26e-2 and the reverse reaction rate is 0. Therefore, this is an irreversible reaction. Note also that only solid components are reacting so groups 18-21 have been omitted.

** IDRZN **				Group 12
3				
** JA	JB	JC **		Group 13
1	0	0		
** NIMM	NAQSP	NVAPOR	**	Group 14
3	0	0		
** KFOR	KREV **			Group 15
1.26e-2	0.0			
** IIMM **				Group 16
1	2	3		
** IMSTOIC	**			Group 17
1.0	-1.0	-1.0		

IDRZN = 4: Dual Monod Biodegradation Reaction. Biodegradation is an irreversible process in which bacteria oxidize an organic substrate to produce energy and biomass. In addition to biomass, the biodegradation process requires the presence of an electron acceptor (e.g. oxygen, nitrate, etc.) and nutrients (e.g. nitrogen and phosphorous). An example of a biodegradation reaction is given by the following reaction:



FEHM models the rate of biodegradation of a substrate with a multiplicative Monod model, which is given by:

$$R_s = -q_m m_b \frac{[S]}{K_s + [S]} \frac{[A]}{K_A + [A]} \quad (10)$$

where $[S]$ is the aqueous concentration of substrate (a.k.a the electron donor) **{SUBSTRATE}**, $[A]$ is the aqueous concentration of the electron acceptor **{ELECACC}**, and m_b is the concentration of the immobile biomass **{BIOMASS}**. The parameter q_m is the maximum specific rate of substrate utilization **{QM}**, which represents the maximum amount of substrate that can be consumed per unit mass of bacteria per unit time. The parameters K_S **{KS}** and K_A **{KA}** are the Monod half-maximum-rate concentrations for the electron donor and electron acceptor, respectively. The rate of microbial growth is given by the synthesis rate (which is proportional to the rate of substrate degradation) minus a first-order decay rate.

$$R_{cells} = -YR_s - b(m_b - m_{b,init}) \quad (11)$$

where Y is the microbial yield coefficient **{YIELD}** and b is the first-order microbial decay coefficient **{DECAY}**. In the above equation, the assumption is made that the background conditions are sufficient to sustain a microbial population of a given size; therefore, the biomass concentration is not allowed to fall below its initial background concentration ($m_{b,init}$) **{XMINIT}**. In the following example of input for a dual monod biodegradation reaction, aqueous component 3 is the substrate and aqueous component 4 is the electron acceptor. Note that there are only two entries in Group 15 and Groups 21-23 are omitted since $\text{NAQUEOUS} = 2$. In addition Group 29 data is left out since $\text{NBIOFRM} = 0$.

** IDRZN **				Group 12
4				
** JA	JB	JC **		Group 13
1	0	0		
** NAQSP	NIMMOBILE	**		Group 14
2	1			
** SUBSTRA	ELECACC **			Group 15
3	4			
** BIOMASS	**			Group 16
4				
** KS **				Group 17
0.201e-3				
** KA **				Group 18
0.00625e-3				
** DECAY **				Group 19
0.0020833				
** EASTOIC	**			Group 20
3.10345				
** COMP3STOIC **				Group 21
0				
** COMP4STOIC **				Group 22
0				
** COMP5STOIC **				Group 23
0				
** PHTHRSH **				Group 24
8				
** QM **				Group 25
8.0226 e-4				
** YIELD **				Group 26
44.8732				
** XMINIT **				Group 27
0.0				
** NBIOFRM	**			Group 28
0				
** ICBIO **				Group 29

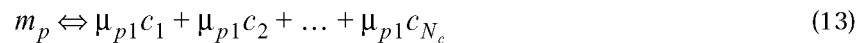
IDRXN = 5: Radioactive Decay Reaction. Radioactive decay is a simple first order decay process given by:



where A is the parent **{PARENT}** and B is the daughter product **{DAUGHTER}**. The half life of the reaction is defined as the time it takes for the concentration of A to decrease by a factor of 2. In the following example of input for a radioactive decay reaction, aqueous component 2, the parent, reacts to form aqueous component 3, the daughter product. The half life for the reaction is 432.0 years.

** IDRZN **			Group 12
5			
** JA	JB	JC **	Group 13
1	0	0	
** HALFLIFE	**		Group 14
432.0			
** RXNTYPE	**		Group 15
1			
** PARENT	DAUGHTER	**	Group 16
2	3		

IDRXN = 7: Kinetic Precipitation/Dissolution Reaction. A general reaction describing the precipitation/dissolution of a mineral p {IMMOBILE} can be written in the following form:



where the c_j are the aqueous concentrations {IAQUEOUS} and the μ_{pj} are stoichiometric coefficients {AQSTOIC}. The equilibrium constant for this reaction is known as the solubility product. Since the activity of a pure solid is equal to one, the reaction quotient Q_p is defined as follows:

$$Q_p = \prod_{j=1}^{N_c} c_j^{\mu_{pj}} \quad (14)$$

At equilibrium, Q_p is equal to the solubility product. The surface-controlled rate of precipitation/dissolution of a mineral is given by:

$$R(m_p) = \text{sign}\left(\log \frac{Q_p}{K_{sp}}\right) A_p k_p \left| \left(\frac{Q_p}{K_{sp}}\right) - 1 \right| \quad (15)$$

where A_p is reactive surface area of the mineral {AREA}, k_p is the precipitation rate constant {RATE}, and K_{sp} is the solubility product {SOLUBILITY}. Currently, this precipitation/dissolution subroutine only allows for the total aqueous concentration of a component to dissolve. The dissolution of uncomplexed aqueous concentration and complex concentrations is not currently supported.

The following is example input for a kinetic precipitation/dissolution reaction (calcite dissolution). This example corresponds to the example used for the **trac** macro (page 140). A single kinetic reaction is specified with no aqueous complexes. The liquid and immobile components are identified. Aqueous component 1 participates in the precipitation/dissolution reaction with immobile component 1. No data is input for groups 4, 6, 9, 10, and 11. The solubility product for the reaction is 6.26e-5 and the reaction rate constant is 100.

rxn						
** NCPLX	NUMRXN **					Group 1
0	1					
** GROUP **						Group 2
1						
** IDCNT	CPNTNAM	IFXCONC	CPNTPRT	CPNTGS	**	Group 3
1	Ca[aq]	0	0	1.0e-9		
** IDCPLX	CPLXNAM	CPLXPRT	**			Group 4
** IDIMM	IMMNAM	IMMPRT **				Group 5
1	Ca[s]	0				
** IDVAP	VAPNAM	VAPPRT				Group 6
** ISKIP **						Group 7
0						
** RSDMAX	**					Group 8
1e-13						
** Chemical	reaction	information	**			
** LOGKEQ	**					Group 9
** CKEQ	HEQ **					Group 10
** STOIC **						Group 11
** IDRXN **						Group 12
7						
** JA	JB	JC **				Group 13
1	0	0				
** IIMMOBLE	**					Group 14
1						
** NAQSP **						Group 15
1						
** IAQ **						Group 16
1						
** IMSTOIC	**					Group 17
1						
** AQSTOIC	**					Group 18
1						
** SOLPROD	**					Group 19
6.26e-5						
** RATE **						Group 20
100						
** AREA **						Group 21
1.0						

IDRXN = 8: Kinetic Precipitation/Dissolution Reaction (rates based on free-ion concentrations). The reaction modeled is analogous to that for IDRXN =7, except that rates are based on the uncomplexed (free-ion) concentration of the species. The total concentration is equal to the free ion concentration + all of the complex concentrations. For a more detailed discussion of the differences between total aqueous and free-ion concentration, see the "Models and Methods Summary" of the FEHM Application [Zyvoloski et al. 1999, page 39, Equation (79)]. For example, for a species such as Cobalt from the multisolute problem (see Section 9.5, "Reactive Transport Example," on page 182), the free ion concentration is simply the concentration of Cobalt in its uncomplexed state. The total Cobalt would be Free Ion Cobalt + all Cobalt Complexes (e.g. CoEDTA from

the multisolute verification problem). Using IDRXXN = 7 allows the total Cobalt to dissolve, while IDRXXN= 8 allows only the free ion Cobalt to dissolve.

6.2.60 Control statement **sol** (optional)

Group 1 - NTT, INTG

Input Variable	Format	Default	Description
NTT	integer	1	Parameter that defines the type of solution required NTT > 0 coupled solution NTT ≤ 0 heat transfer only solution
INTG	integer	-1	Parameter that defines element integration type INTG ≤ 0 Lobatto (node point) quadrature is used, recommended for heat and mass problems without stress. INTG > 0 Gauss quadrature is used, recommended for problems requiring a stress solution.

The following is an example of **sol**. In this example, a coupled heat-mass solution using Lobatto quadrature is specified.

```
sol
1          -1
```

6.2.61 Control statement **sptr** (optional)

Streamline particle tracking is invoked with this control statement.

Group 1 - DTMX, IPRT, IPRTO, RSEED

Optional keyword "tcurve" is input to indicate that transfer function curves should be input to model matrix diffusion. It is followed by NUMPARAMS and TFILENAME.

```
KEYWORD
NUMPARAMS
TFILENAME
```

Optional keywords "po", "sa", "pe", "de", "pr", "te", "zo" define parameters to be output.

```
KEYWORD
```

Group 2 - COURANT_FACTOR, IPRTR, ITENSOR, IREVERS,
FREEZ_TIME

Group 3 - KEYWORD

The Group 3 keyword ('tprp') specifies that transport properties will be entered on subsequent lines. The transport properties input (Group 4) depend on the value of TPRP_FLAG, the first value input for that group. For TPRP_FLAG = 2 or 4, format of Group 4 input also depends on the form of the dispersion coefficient tensor, as selected using the flag ITENSOR (Group 2).

TPRP_FLAG = 1:

Group 4 - TPRP_FLAG, KD

TPRP_FLAG = 2:*ITENSOR = 1*

Group 4 - TPRP_FLAG, KD, DM, A1, A2, A3, A4, ASX, ASY, VRATIO

ITENSOR = 2

Group 4 - TPRP_FLAG, KD, DM, AL, ATH, ATV, VRATIO

ITENSOR = 3

Group 4 - TPRP_FLAG, KD, DM, ALH, ALV, ATH, ATV, VRATIO

ITENSOR = 4

Group 4 - TPRP_FLAG, KD, DM, AL, AT, VRATIO

ITENSOR = 5

Group 4 - TPRP_FLAG, KD, AL, ATH, ATV, DM, VRATIO

TPRP_FLAG = 3:Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE***TPRP_FLAG = 4:****ITENSOR = 1*Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE, DM, A1, A2, A3, A4, ASX, ASY, VRATIO*ITENSOR = 2*Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE, DM, AL, ATH, ATV, VRATIO*ITENSOR = 3*Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE, DM, ALH, ALV, ATH, ATV, VRATIO*ITENSOR = 4*Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE, DM, AL, AT, VRATIO*ITENSOR = 5* (not recommended, included for compatibility with older
versions of the code)Group 4 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX,
APERTURE, AL, ATH, ATV, DM, VRATIO

Group 5 - JA, JB, JC, MODEL_NUMBER

Group 4 is ended when a blank line is encountered. The *MODEL_NUMBER*
is incremented each time a Group 4 line is read, and Group 5 lines refer to
this parameter.The optional keyword 'zbtc' indicates that breakthrough curves will be
computed for specified zones. It is followed by NZBTC and ZBTC.

KEYWORD

NZBTC

ZBTC

Group 6 - ITM, IST

Group 7 - NX, NY, NZ

Group 8 - X10, Y10, Z10

Group 9 - XDIM, YDIM, ZDIM

Group 10 - IJKV(I), X1(I), Y1(I), Z1(I) for I = 1 to NUMPART

Group 10 input is terminated with a blank line.

Note when IST = 0 or 1, Group 10 is used and place holders are inserted for Groups 7-9; however, when IST = 2, Group 10 is not implemented and Groups 7-9 are used.

Input Variable	Format	Description
DTMX	real	Time step control (seconds). FEHM will account for all particles every $\text{abs}(dtmx)$ seconds and write information to the ".sptr3" file if the "zbtc" keyword is present. This controls the output density for breakthrough curve information only. If you are not using/creating breakthrough curves, set DTMX very large (e.g. $1e20$). If DTMX is negative, the time step for streamline calculations is forced to be $\text{abs}(DTMX)$ seconds.
IPRT	integer	Flag to denote whether individual particle positions are written at specified intervals to the ".sptr1" file. The particle coordinate positions are used to get a snapshot of the particle plume at various times during the simulation. IPRT = 0, No output IPRT > 0, Output is written to the ".sptr1" file every IPRT time steps.
IPRTO	integer	Flag to denote if particle streamline information is written to the ".sptr2" file. The information is used to draw complete particle streamlines (for a relatively small number of particles). IPRTO = 0, No output IPRTO > 0, Extended output is written to the ".sptr2" file. IPRTO < 0, Abbreviated output is written to the ".sptr2" file. If $\text{abs}(IPRTO) = 1$, output is formatted, $\text{abs}(IPRTO) = 2$ output is unformatted, and $\text{abs}(IPRTO) = 3$ output is in binary format.
RSEED	integer	Random number seed for the random number generator. For compatibility with earlier versions of FEHM in which this input did not exist, if no value of RSEED is input, the code assigns a value of 0.
KEYWORD	character	Optional keyword "tcurve" indicating transfer function curve data should be input to model matrix diffusion. If the keyword is found then NUMPARAMS and FILENAME are entered, otherwise they are omitted.

Input Variable	Format	Description
NUMPARAMS	integer	Number of parameters that define the transfer function curves being used.
TFILENAME	character	Name of input file containing the transfer function curve data.
KEYWORD	character	Optional keywords "po" (porosity), "sa" (saturation), "pe" (permeability), "de" (density), "pr" (pressure), "te" (temperature), "zo" (zone number), 1 per line, indicating which parameters will be output along the particle path (written to ".sptr2" file). If no keywords are present, porosity and saturation will be output.
COURANT_FACTOR	integer	Fraction of the distance through a cell that a particle moves in a single time step. This is used to ensure that the particle, on average, traverses less than one cell before a random-walk dispersion step is performed. For example, a factor of 0.25 indicates that the particle should take at least 4 time steps to move through a cell.
IPRTR	integer	Flag for choosing the method for computing concentrations in cells based on the particle tracking information that will be written to the ".trc" or AVS output files. IPRTR = 0, particle concentrations are computed as number of particles residing in the cell divided by the fluid mass in the cell. IPRTR < 0, an integral of the particle concentration specified above is made and reported. This integral is the normalized cumulative concentration, which for a steady state flow field is equivalent to the response to a step change in particle concentration (note that the particles are input as a pulse).
ITENSOR	integer	Flag indicating the mathematical form of the dispersion coefficient tensor to be selected. ITENSOR = 0, No dispersion. ITENSOR = 1, Generalized form of the axisymmetric tensor, from Lichtner et al. (2000) ITENSOR = 2, Axisymmetric form of the dispersion coefficient tensor of Burnett and Frind (1982) ITENSOR = 3, Modified form of the dispersion coefficient tensor of Burnett and Frind (1982). See Lichtner et al. (2000) for details ITENSOR = 4, Isotropic form of the dispersion coefficient tensor of Tompson et al. (1987) ITENSOR = 5, Original form of the Burnett and Frind (1982) tensor as implemented in FEHM Version 2.10. Note: for Version 2.10 and earlier, the variable ITENSOR did not exist. For compatibility with these earlier versions, when ITENSOR is omitted from the input file, the code uses the ITENSOR = 5 formulation and the pre-existing input format. It is recommended that new simulations use one of the other tensor formulations (ITENSOR = 1 to 4).

Input Variable	Format	Description
		In addition, the sign of ITENSOR is used as a switch as follows: if $ITENSOR < 0$, $abs(ITENSOR)$ is the flag, but the $\nabla \bullet D$ term is not included in the computation of particle displacements. Under normal circumstances, an approximation of the term $\nabla \bullet D$ is used in the particle tracking algorithm to obtain accurate solutions in cases where there are gradients in D .
IREVERS	integer	Flag indicating if reverse particle tracking should be performed. If omitted, forward tracking is performed. IREVERS = 0, Standard forward tracking IREVERS = -1, Forward tracking only after exiting the time loop (this is needed for comparing results with reverse tracking) IREVERS = +1, Reverse tracking. Note: When using reverse tracking, turn off the dispersion, $ITENSOR = 0$, as it does not make sense to try to reverse the random part of the displacement. The value for ITENSOR must be entered to use this option.
FREEZ_TIME	real	If greater than zero, time (days) at which flow solution is frozen and only particle transport is computed after that time. If omitted, the flow solution continues for the entire simulation. Values for ITENSOR and IREVERS must be entered to use this option.
KEYWORD	character	Keyword 'tprp' specifying that transport properties are to follow on subsequent lines.
TPRP_FLAG	integer	Flag indicating what type of transport property information is to follow on the line 1 - KD only 2 - KD and 5 terms of dispersivity tensor 3 - (Dual porosity) - Matrix KD, diffusion coefficient, retardation factor in fracture, and fracture aperture. No dispersion 4 - (Dual porosity) - Matrix KD, diffusion coefficient, retardation factor in fracture, fracture aperture, and 5 terms of dispersivity tensor
KD	real	Matrix sorption coefficient
DIFM	real	Diffusion coefficient applying to matrix diffusion submodel (m^2/s)
RD_FRAC	real	Retardation factor in fracture media
POR_MATRIX	real	Matrix porosity (fracture volume fraction is specified in rock macro)
APERTURE	real	Fracture aperture (m)
AL	real	Longitudinal dispersivity, α_L (m). $ITENSOR = 2, 4, \text{ or } 5$
ALH	real	Horizontal longitudinal dispersivity, α_{LH} (m). $ITENSOR = 3$
ALV	real	Vertical longitudinal dispersivity, α_{LV} (m). $ITENSOR = 3$
AT	real	Transverse dispersivity, α_T (m). $ITENSOR = 4$
ATH	real	Transverse horizontal dispersivity, α_{TH} (m). $ITENSOR = 2, 3, \text{ or } 5$

Input Variable	Format	Description
ATV	real	Transverse vertical dispersivity, α_{TV} (m). ITENSOR = 2, 3, or 5
A1	real	Generalized dispersivity term α_1 (m) from Lichtner et al. (2000)
A2	real	Generalized dispersivity term α_2 (m) from Lichtner et al. (2000)
A3	real	Generalized dispersivity term α_3 (m) from Lichtner et al. (2000)
A4	real	Generalized dispersivity term α_4 (m) from Lichtner et al. (2000)
ASX	real	Direction cosine of the axis of symmetry from Lichtner et al. (2000)
ASY	real	Direction cosine of the axis of symmetry from Lichtner et al. (2000)
DM	real	Molecular diffusion coefficient (m^2/s)
VRATIO	real	Parameter to control the movement of particles into low velocity cells via random walk. Used to restrict the artificial migration of particles into low permeability zones due to dispersion. The value of VRATIO is used as a ratio for determining if random walk into a new cell is allowed. If the ratio of the average velocity in the new cell divided by the velocity in the previous cell is less than VRATIO, then the particle is not allowed to migrate into the new cell. It is returned to its previous location, and a new random walk is computed and applied. Up to 10 attempts at a random walk are allowed, after which the particle location is left at the current location for the next advective step.
MODEL_NUMBER	integer	Number of model (referring to the sequence of models read) to be assigned to the designated nodes or zone.
KEYWORD	keyword	Optional keyword 'zbtc' specifying that zone breakthrough curves will be computed. Output will be written to the ".sptr3" file. If 'zbtc' is omitted, so are NZBTC and ZBTC. Note that the zones must be specified in a zone macro preceding the sptr macro in the input file before they are invoked using the keyword 'zbtc'.
NZBTC	integer	Number of zones for which breakthrough curves will be computed.
ZBTC	integer	NZBTC zone numbers of the zone(s) for which breakthrough curves will be computed.
ITM	integer	Maximum number of time steps to accomplish the FEHM time step 'day'
IST	integer	Flag to specify type of input for particles IST = 0, local position and corresponding element number (Group 10) IST = 1, global position (Group 10) IST = 2, specify a zone of particles (Groups 7-9)
NX	integer	Number of divisions in the x-direction
NY	integer	Number of divisions in the y-direction
NZ	integer	Number of divisions in the z-direction
X10	real	X-coordinate of the origin (x_{min})

Input Variable	Format	Description
Y10	real	Y-coordinate of the origin (y_{min})
Z10	real	Z-coordinate of the origin (z_{min})
XDIM	real	Length of X-direction
YDIM	real	Length of Y-direction
ZDIM	real	Length of Z-direction
IJKV(I)	integer	Node or element number
X1(I)	real	Starting X-coordinate for a particle
Y1(I)	real	Starting Y-coordinate for a particle
Z1(I)	real	Starting Z-coordinate for a particle

The following are examples of **sptr**. In the first example 10000 particles are inserted at the inlet within a single cell, and the breakthrough curve at a downstream location (defined in a call to **zone**) is recorded for the case of longitudinal dispersion with a dispersivity of 100 m and sorption with a KD of 0.0223715. Breakthrough concentration is output every 1.728e8 seconds to the ".sptr3" file.

sptr	1.728e8	0	0	0				Group 1
	0.25	0	5	0				Group 2
tprp								Group 3
	2	0.0223715	100.	0.	0.	0.	0.	Group 4
	1	0	0	1				Group 5
zbtc								
	1							
	5							
	1000	2						Group 6
	1	100	1000					Group 7
	0.	-1500.	0.					Group 8
	10.	3000.	-12.5.					Group 9
								Group 10

In the second example, both longitudinal and transverse dispersion are invoked, but no sorption. The solute is input as a patch on the inlet face of the model. The dimensions of the patch will be 3,000 m in the y -direction and 12.5 m in the vertical direction, starting at the surface, and 100000 particles are injected. Data to generate a steady state concentration plume is output in the ".trc" file.

spr	2.88e7	0	0	0				Group1
	0.25	0	5	0				Group 2
tprp	2	0.	100.	0.1	0.1	0.	-1.e-10	Group 3
								Group 4
	1	0	0	1				Group 5
1000	2							Group 6
1	100	1000						Group 7
0.	-1500.	0.						Group 8
10.	3000.	-12.5.						Group 9
								Group 10

6.2.62 Control statement stop (required)

No input is associated with this control statement. It signals the end of input, and as such it always appears as the last line of an input deck.

6.2.63 Control statement subm (optional)

Create a new **flow** macro to represent boundary conditions on an extracted submodel.

Group 1 - KEYWORD, IZONE1, IZONE2

Input Variable	Format	Description
KEYWORD	character*4	Keyword "flux", "head", or "pres" to specify type of boundary condition to output.
IZONE1	integer	Zone defining submodel nodes.
IZONE2	integer	Zone defining nodes outside of the submodel (optional).

In the following example a submodel containing all nodes in zone 2 is defined, and all nodes in zone 1 are excluded.

subm		
pres	2	1

The zone macro used for this example is as follows:

zone				
1				
0.	100.0	100.0	0.	
0.	100.0	100.0	0.	
0.	0.	100.0	100.0	
0.	0.	100.0	100.0	
100.0	100.0	100.0	100.0	
0.	0.	0.	0.	
2				
20.0	30.0	20.0	30.0	
20.0	30.0	20.0	30.0	
20.0	20.0	30.0	30.0	
20.0	20.0	30.0	30.0	

100.0	100.0	100.0	100.0
99.0	99.0	99.0	99.0
3			
60.0	90.0	90.0	60.0
60.0	90.0	90.0	60.0
60.0	60.0	90.0	90.0
60.0	60.0	90.0	90.0
1.0	1.0	1.0	1.0
0.0	0.0	0.0	0.0

The code produces the following “flow” macro output for this example:

flow	Boundary	Conditions	Output: FEHM V2.21	sun 03-07-26	7/29/2003	10:47:50			
336	336	1	1.43112776	1.0	1.0E+02	#	22.5000	27.5000	100.000
337	337	1	1.42620726	1.0	1.0E+02	#	27.5000	27.5000	100.000
358	358	1	1.43081773	1.0	1.0E+02	#	22.5000	22.5000	100.000
359	359	1	1.42601577	1.0	1.0E+02	#	27.5000	22.5000	100.000

6.2.64 Control statement svar (optional)

Group 1 - IVAR

Input Variable	Format	Description
IVAR	integer	To enable pressure-enthalpy variables in place of pressure temperature for water/water vapor physics IVAR = 1. Default is disabled.

6.2.65 Control statement szna or napl (optional)

Group 1 - ICO2D

Group 2 - TREF, PREF

Group 3 - DENNAPL, VISCNAPL

Input Variable	Format	Description
ICO2D	integer	Determines the type of air module used. ICO2D = 1, 1 degree of freedom solution to the saturated-unsaturated problem is produced. This formulation is similar to the Richard's Equation. ICO2D = 2, 1 degree of freedom solution is obtained assuming only gas flow with no liquid present. ICO2D = 3, full 2 degree of freedom solution. All other values are ignored. The default is 3.
TREF	real	Reference temperature for properties (°C).
PREF	real	Reference pressure for properties (MPa).
DENNAPL	real	NAPL density (kg/m ³).
VISCNAPL	real	NAPL viscosity (Pa s).

The following is an example of **szna**. In this example, a full 2-degrees-of-freedom solution is specified with a reference temperature for property evaluation of 20 °C and a reference pressure of 0.1 MPa. The NAPL density is 800 kg/m³ and the viscosity is 5.e-3 Pa·s.

szna
3
20. 0.1
800. 5.e-3

6.2.66 Control statement text (optional)

Group 1- WDD1

Input Variable	Format	Description
WDD1	character*80	Line of text. A maximum of 80 characters per line are entered. Text is input until a blank line is inserted to signal the end of the control statement. This text is written to the output file (iout).

The following is an example of **text**

text
This 2-d model of the PACE problem will be used to study thermal effects.
user # = -20 to get waste packages

6.2.67 Control statement thic (optional)

Input for variable thickness for two-dimensional problems.

Group 1 - JA, JB, JC, THIC (JA, JB, JC - defined on page 31).

Input Variable	Format	Description
THIC	real	Thickness of the model domain in the third dimension (m). Default is 1.

The following is an example of **thic**. In this example, the thickness for all nodes is set to 10 m, after which the nodes defined by zone 2 are set to 5 m. Thus, the thickness is 10 m everywhere except zone 2, where thickness is 5 m.

thic
1 0 0 10.
-2 0 0 5.

6.2.68 Control statement time (required)

Time step and time of simulation data.

Group 1 - DAY, TIMS, NSTEP, IPRTOUT, YEAR, MONTH, INITTIME

Group 2 - DIT1, DIT2, DIT3, ITC, DIT4 (as needed)

DAY should be larger than DAYMIN defined in control statement **ctrl**. The code proceeds to the next control statement when a blank line is encountered for Group 2. Group 2 can be used to generate output at specific times (with multiple Group 2s). Contour plot output will be written at each DIT1 regardless of the input in control statement **cont**. The restart file will be written (or rewritten if one already exists) at each DIT1. If DIT4 is omitted (for compatibility with older input files where DIT4 was not input) the maximum time step defined in the control statement **ctrl** will be used.

Input Variable	Format	Description
DAY	real	Initial time step size (days).
TIMS	real	Final simulation time (days).
NSTEP	integer	Maximum number of time steps allowed.
IPRTOU	integer	Print-out interval for nodal information (pressure, enthalpy etc.), as set up under control statement node . (i.e., number of time steps).
YEAR	integer	Year that simulation starts.
MONTH	integer	Month that simulation starts.
INITTIME	real	Initial time of simulation (days). For compatibility with older versions, if this parameter is absent the initial time of simulation will be 0 if no restart file is used, or the time in the restart file if one is used.
DIT1	real	Time (days) for time step change.
DIT2	real	New time step size (days). If $DIT2 < 0$ then $ABS(DIT2)$ is the new time step multiplier.
DIT3	real	Implicitness factor for new time step. $DIT3 \leq 1.0$ backward Euler. $DIT3 > 1.0$ for second-order implicit scheme.
ITC	integer	New print-out interval.
DIT4	real	Maximum time step size for next time interval (days).

The following is an example of **time**. In this example, the initial time step size is 30 days, the final simulation time is 3650 days, the number of time steps allowed is 20, nodal information is printed out for every 5th time step, the simulation starts in the 10th month of 1989, and the initial time of simulation is assigned a value of 0. The time step multiplier is changed after 1 day, and the new time step multiplier is 1.2, backward Euler is used from this time on and the printout interval is every 10th time step. The maximum time step size for the next interval is omitted so the default value entered in the **ctrl** macro will be used.

time						
	30.0	3650.0	20	5	1989	10
	1.0	-1.2	1.0	10		0.0

6.2.69 Control statement trac (optional)

Group 1 - KEYWORD 'userc', ANO, AWC, EPC, UPWGTA

or

Group 1 - ANO, AWC, EPC, UPWGTA

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS

Group 3 - IACCMX, DAYCM, DAYCMM, DAYCMX, NPRTRC

Group 4 - KEYWORD 'tpor'

Group 5 - JA, JB, JC, PS_TRAC (JA, JB, JC - defined on page 31)

Tracer porosity is entered only if the Group 4 keyword ('tpor'), which specifies tracer porosity input, is present, otherwise Groups 4 and 5 are omitted.

Group 6 - NSPECI

Group 7 - KEYWORD 'ldsp'

The Group 7 keyword ('ldsp') specifies longitudinal / transverse dispersion should be used. If X, Y, Z dispersion is desired Group 7 is omitted, and dispersivities are input in X, Y, Z order in Group 9 or Group 12. When longitudinal / transverse dispersion is invoked the Z-components of dispersivity are omitted from the Group 9 or Group 12 input, and X and Y represent longitudinal and transverse dispersion respectively. Note that an "L" or "V" added to the Group 9 or Group 12 variable names (SEHDIFF, THETARESID, TCX, TCY, TCZ, IADSF, A1ADSF, A2ADSF, BETADF, DIFFM) indicates the value is for the liquid or vapor phase, respectively.

Group 8 - KEYWORD 'dspl' or 'dspv' or 'dspb'

The Group 8 keyword specifies that the same diffusion coefficient and dispersivities are to be used for all species of the same type (liquid and/or vapor). This will make the calculations more efficient and thus should be used if applicable. If Group 8 is omitted, Groups 9 and 10 are also omitted, and input resumes with Group 11.

If only liquid species are present (keyword 'dspl') or only vapor species are present (keyword 'dspv') with no longitudinal / transverse dispersion,

Group 9 is defined as follows: If $SEHDIFF \geq 0$

Group 9- SEHDIFF, TCX, TCY, TCZ

else if $SEHDIFF < 0$

Group 9 - SEHDIFF, THETARESID, TCX, TCY, TCZ

Otherwise if both liquid and vapor are present (keyword 'dspb'), parameters for both must be entered. If $SEHDIFFL \geq 0$

Group 9- SEHDIFFL, TCLX, TCLY, TCLZ, SEHDIFFV, TCVX, TCVY, TCVZ

else if $SEHDIFFL < 0$

Group 9 - SEHDIFFL, THETARESIDL, TCLX, TCLY, TCLZ, SEHDIFFV, THETARESIDV, TCVX, TCVY, TCVZ

Group 10 - JA, JB, JC, ITRCDSP (JA, JB, JC - defined on page 31)

Group 11 - ICNS

There are two options for group twelve. If the same diffusion coefficient and dispersivities are to be used for all species of the same type (liquid and/ or vapor - keyword 'dspl', 'dspv', or 'dspb') only sorption parameters are input:

Group 12 -IADSF, A1ADSF, A2ADSF, BETADF

or for a Henry's Law Species (both liquid and vapor)

Group 12 - IADSFL, A1ADSFL, A2ADSFL, BETADFL, IADSFV,
A1ADSFV, A2ADSFV, BETADFV

In the absence of a Group 8 keyword ("dspl", 'dspv', or 'dspb') the following input (for liquid or vapor) which includes the sorption and dispersion parameters is used: If $\text{DIFFM} \geq 0$

Group 12 - IADSF, A1ADSF, A2ADSF, BETADF, DIFFM, TCX, TCY, TCZ
else if $\text{DIFFM} < 0$

Group 12 - IADSF, A1ADSF, A2ADSF, BETADF, DIFFM, THETARESID,
TCX, TCY, TCZ

For a Henry's Law Species (both liquid and vapor) if $\text{DIFFML} \geq 0$

Group 12 - IADSFL, A1ADSFL, A2ADSFL, BETADFL, DIFFML, TCLX,
TCLY, TCLZ, IADSFV, A1ADSFV, A2ADSFV, BETADFV,
DIFFMV, TCVX, TCVY, TCVZ

else if $\text{DIFFML} < 0$

Group 12 - IADSFL, A1ADSFL, A2ADSFL, BETADFL, DIFFML,
THETARESIDL, TCLX, TCLY, TCLZ, IADSFV, A1ADSFV,
A2ADSFV, BETADFV, DIFFMV, THETARESIDV, TCVX,
TCVY, TCVZ

Group 13 - JA, JB, JC, ITRCD (JA, JB, JC - defined on page 31)

Group 14 - HENRY_MODEL, HAWWA(1), HAWWA(2), HAWWA(3),
HAWWA(4), HAWWA(5) (only input for a Henry's Law species,
otherwise omitted)

Group 15 - JA, JB, JC, ANQO (JA, JB, JC - defined on page 31)

Group 16 - JA, JB, JC, CNSK, T1SK, T2SK (JA, JB, JC - defined on
page 31)

Groups 11, 12, 13, 14, 15, and 16 are entered as a unit for each solute. However, for a solid species, only groups 11, 15, and 16 are entered (groups 12, 13, and 14 are not applicable for a solid species). Groups 12 and 13 are used to define transport models for which sorption and dispersion parameters are identical. For a liquid or vapor species, only one set of Group 12 parameters should be entered per region. However, for a Henry's Law species, two sets of parameters per region must be entered. For this case, the liquid sorption parameters should be entered on the first line and the vapor sorption parameters on a second line or as a continuation of the first line. Group 12 is read in until a blank line is encountered. The model number is incremented by 1 each time a line is read. Group 13 then assigns a transport model number to every node.

Injection nodes must be specified in control statement **flow**.

Input Variable	Format	Description
KEYWORD	character*5	Keyword for invoking a solute transport user subroutine. If the word 'userc' is placed in this position, then the code invokes a solute transport user subroutine at each time step. Omit this key word if there is no solute user subroutine for the simulation.
ANO	real	Initial solute concentration, set at all nodes for all species unless overwritten by a restart file input or values in group 14 below (moles/kg fluid).
AWC	real	Implicitness factor for solute solution. AWC > 1.0 gives 2nd order solution AWC ≤ 1.0 gives 1st order solution
EPC	real	Equation tolerance for solute solution. When the square root of the sum of the squared residuals is lower than EPC, the solution is assumed to be converged.
UPWGTA	real	Upstream weighting term for the solute solution. UPWGTA < 0.5 UPWGTA is set to 0.5 UPWGTA > 1.0 UPWGTA is set to 1.0
DAYCS	real	Time which the solute solution is enabled (days).
DAYCF	real	Time which the solute solution is disabled (days).
DAYHF	real	Time which the flow solution is disabled (days).
DAYHS	real	Time which the flow solution is enabled (days).
IACCMX	integer	Maximum number of iterations allowed in solute solution if time step multiplier is enabled
DAYCM	real	Time step multiplier for solute solution
DAYCMM	real	Initial time step for solute solution (days)
DAYCMX	real	Maximum time step for solute solution (days)
NPRTTRC	integer	Print-out interval for solute information. Data for every NPRTTRC solute time step will be written to the ".trc" file. If this parameter is omitted (for compatibility with old input files) the default value is 1. Note that the first and last solute time step within a heat and mass transfer step automatically get printed.
KEYWORD	character*4	Keyword 'tpor' specifying optional tracer porosity should be input. If group 4 is omitted, porosities assigned in macro rock are used.
PS_TRAC	real	Tracer porosity
NSPECI	integer	Number of solutes simulated.

Input Variable	Format	Description
KEYWORD	character*4	Keyword ' <i>ldsp</i> ' specifying longitudinal / transverse dispersion. If x, y, z dispersion is desired group 7 is omitted, and dispersivities are input in x, y, and then z order (group 9 or group 12). Otherwise, if longitudinal / transverse dispersion is desired the keyword ' <i>ldsp</i> ' is entered and dispersivities are instead input in longitudinal and then transverse order with values for the third dimension omitted.
KEYWORD	character*4	Keyword specifying the same diffusion coefficient and dispersivities are to be used for all species of the same type (liquid and/or vapor). ' <i>dsp</i> ' indicates that only liquid species exist. ' <i>dspv</i> ' indicates that only vapor species exist. ' <i>dspb</i> ' indicates that both liquid and vapor species exist.
ICNS	integer	Phase designation for the ith solute -2 - Henry's Law species (input and output concentration values are gas concentrations). -1 - Vapor species. 0 - Solid species 1 - Liquid species 2 - Henry's Law species (input and output concentration values are liquid concentrations)
SEHDIFF	real	Molecular diffusion coefficient (m^2/s) If SEHDIFF > 0, the molecular diffusion coefficient is a constant. If SEHDIFF < 0, this indicates the model of Conca and Wright (1992) for diffusion coefficient as a function of volumetric water content is used, and ABS(SEHDIFF) is the diffusion coefficient at saturation equal to one, and THETARESID is the water content at which the diffusion coefficient reaches a low value.
THETARESID	real	Residual water content for Conca and Wright function.
TCX	real	Dispersivity in x-direction (m)
TCY	real	Dispersivity in y-direction (m)
TCZ	real	Dispersivity in z-direction (m)
ITRCDSP	integer	Region number for dispersion parameters given in group 9 (keyword). Default is 1.
IADSF	integer	Adsorption model type for the ith species, ith region 0 - conservative solute 1 - linear sorption isotherm 2 - Freundlich sorption isotherm 3 - Modified Freundlich sorption isotherm 4 - Langmuir sorption isotherm
A1ADSF	real	α_1 parameter in adsorption model
A2ADSF	real	α_2 parameter in adsorption model
BETADF	real	β parameter in adsorption model

Input Variable	Format	Description
DIFFM	real	Molecular diffusion coefficient (m^2/s) See discussion for SEHDIFF.
ITRCD	integer	Region number for group 12 sorption parameters or for sorption and dispersion parameters (no keyword). Default is 1.
HENRY_MODEL	integer	Flag denoting which model is to be used for defining the temperature dependence of the Henry's law constant 1 - van't Hoff model 2 - Multi-parameter fit to experimental data (used for carbonate system).
HAWWA(1)	real	Term in Henry's Law temperature dependence model: For model 1 - parameter value is A_H For model 2 - parameter value is $A_{H,1}$
HAWWA(2)	real	Term in Henry's Law temperature dependence model: For model 1 - parameter value is ΔH_H For model 2 - parameter value is $A_{H,2}$
HAWWA(3)	real	Term in Henry's Law temperature dependence model: For model 1 - not used For model 2 - parameter value is $A_{H,3}$
HAWWA(4)	real	Term in Henry's Law temperature dependence model: For model 1 - not used For model 2 - parameter value is $A_{H,4}$
HAWWA(5)	real	Term in Henry's Law temperature dependence model: For model 1 - not used For model 2 - parameter value is $A_{H,5}$
ANQO	real	Initial concentration of tracer, which will supersede the value given in group 1. Note that if initial values are read from a restart file, these values will be overwritten. Units are moles per kg vapor or liquid for a liquid, vapor, or Henry's law species, and moles per kg of solid for a solid species. Default is 0.
CNSK	real	Injection concentration at inlet node (moles per kg liquid or vapor). If fluid is exiting at a node, then the in-place concentration is used. If $CNSK < 0$, then the concentration at that particular node will be held at a concentration of $abs(cnsk)$ (default is 0 for all unassigned nodes).
T1SK	real	Time (days) when tracer injection begins. Default is 0.

Input Variable	Format	Description
T2SK	real	Time (days) when tracer injection ends. Default is 0. If T2SK < 0, the absolute value of T2SK is used for this parameter, and the code interprets the negative value as a flag to treat the node as a zero-solute-flux node for cases in which a fluid sink is defined for that node. For this case, the solute will stay within the model at the node despite the removal of fluid at that location. If a fluid <i>source</i> is present at the node, CNSK is the concentration entering with that fluid, as in the normal implementation of a solute source. Note that the code cannot handle the case of T2SK < 0 and CNSK < 0 (fixed concentration), as these are incompatible inputs. Therefore, the code prints an error message and stops for this condition.

In the following example of **trac**, calcite dissolution is simulated. The input groups are given to the right of the table to facilitate review of the example. The initial solute concentration is set to 0 (but is later overwritten by group 14 input), the implicitness factor is 1 resulting in a 1st order solution, the equation tolerance is 1.e-7, and the upstream weighting is set to 0.5. The solute transport solution is turned on as the heat and mass solution is turned off at day 1. The heat and mass solution resumes on day 1000. Two solutes are simulated in this example. Solute 1 is a nonsorbing (conservative) liquid species ($\alpha_1 = \alpha_2 = 0.$, $\beta = 1.$) with a molecular diffusion coefficient of 1.e-9 m²/s, and dispersivity of 0.0067 m in the X-direction. This transport model applies over the entire problem domain. The initial solute concentration for solute 1 is 6.26e-5 mol/kg-water. Solute 2 is a solid species with an initial solute concentration of 2.e-5 mol/kg-solid. There is no solute source for either solute. The corresponding data for the **rxn** macro, that would complete this example, is given on page 122.

trac										
0	1	1.e-7	.5							Group 1
1.	1.e20	1.	1000.							Group 2
50	1.2	1.1574e-6	1.1574e-3	1						Group 3
2										Group 6
1										Group 11
0	0.	0.	1.	1.e-9	0.0067	0.	0.			Group 12
1	0	0	1							Group 13
1	0	0	6.26e-5							Group 15
										Group 16
0										Group 11
1	0	0	2.e-5							Group 15
										Group 16

The second example of **trac** is modified to illustrate tracer porosity input and use of longitudinal and transverse dispersion.

The third example of **trac**, is modified to illustrate keyword use for assigning the same diffusion coefficient and dispersivities for each solute in the simulation.

trac							
0	1	1.e-7	.5				Group 1
1.	1.e20	1.	1000.				Group 2
50	1.2	1.1574e-6	1.1574e-3	1			Group 3
tpor							
1	0	0	0.2				Group 4
							Group 5
2							Group 6
ldsp							
1							Group 7
0	0.	0.	1.	1.e-9	0.0067	0.0067	Group 11
							Group 12
1	0	0	1				Group 13
1	0	0	6.26e-5				Group 15
0							Group 16
1	0	0	2.e-5				Group 11
							Group 15
							Group 16

trac							
0	1	1.e-7	.5				Group 1
1.	1.e20	1.	1000.				Group 2
50	1.2	1.1574e-6	1.1574e-3	1			Group 3
2							Group 6
dspl							
1.e-9	0.0067	0.0067	0.0067				Group 8
1	0	0	1				Group 9
							Group 10
1							Group 11
0	0.	0.	1.				Group 12
1	0	0	1				Group 13
1	0	0	6.26e-5				Group 15
0							Group 16
1	0	0	2.e-5				Group 11
							Group 15
							Group 16

The final example of **trac** illustrates use of the “**userc**” subroutine and applies only to the version of this subroutine that is included/compiled with the current version of FEHM. The user subroutine is invoked from the **trac** macro using the keyword “**userc**” in Group 1. When invoked, the code looks for an additional data file “*userc_data.dat*” in the current working directory, in which the following input is required:

- Group 1 - USROPTION (integer)
The only option currently supported is option 2: Time-varying solute mass flux input at prescribed nodes.

- Group 2- N_POINTS (integer)
Number of flux-time points in the transient input curve.
- Group 3- Required header line (read in by code but not used)
- Group 4 - USERFLUX(J), J = 1, NSPECI
This group has N_POINTS lines of mass flux values (moles/s), one for each point in the time-varying input curve. Each line has NSPECI entries, one for each species in the simulation (NSPECI is input in the **trac** macro Group 6). Positive mass flux values are solute mass injected into the model (note: this is the opposite of the convention for fluid flow sources and sinks).
- Group 5 - Required header line (read in by code but not used)
- Group 6 - USERTIME
This group has N_POINTS lines of times (seconds), one for each point in the time-varying input curve.

The code uses a special flag to decide in which nodes to inject the mass flux. In the **trac** macro in the source/sink input (Group 15), the value for CNSK is set to -9876 as a flag to denote that the mass flux is input at that node. If multiple nodes are flagged, the mass flux value is input at each of these nodes (as opposed to the entire mass flux being divided evenly between these nodes).

In this example, there are 3 species defined in **trac**, so 3 mass flux inputs are required in USERFLUX (Group 4 of "*userc_data.dat*"). The first solute has a linearly increasing mass flux from 1.e-3 to 5.e-3 over the time from 0. to 4.e5 seconds. The second species is a solid, and the 0. values denote that no mass flux is input. The third species has a constant mass flux input of 1.e-5 moles/s for all times.

The **trac** macro has the flag of -9876 for CNSK in all nodes of zone 3 (zone 3 would be defined in a previous invocation of the zone macro) for species 1 and 3. The mass flux is therefore input in all nodes of zone 3. Since the flag was not used for the solid species 2, the 0. values input in USERFLUX are actually irrelevant because no nodes are being flagged with the -9876 value in **trac**. Thus, for mass flux to be input using this method, the values in USERFLUX must be non-zero and the nodes must be flagged in **trac**. Note that there is no restriction on the phase of the species: solid or gas species can be added at a prescribed mass flux if desired (though this is not done in this example).

trac									
userc	0	1	1.e-6	1.					Group 1
0.	1.e20	1.	1.e30						Group 2
10	1.2	1.e2	1.e3	1					Group 3
3									Group 6
1									Group 11
0	0	0	1	1.e-10	1.	1.	1.		Group 12
1	0	0	1						Group 13
1	0	0	0.						Group 15
-3	0	0	-9876.	0.	1.e20				Group 16

0									Group 11
1	0	0	0.						Group 15
1									Group 16
0	0	0	1	1.e-10	1.	1.	1.		Group 11
									Group 12
1	0	0	1						Group 13
1	0	0	0.						Group 15
-3	0	0	-9876.	0.	1.e20				Group 6

File "userc_data.dat":

2					Group 1
5					Group 2
Mass flux (moles/s)					Group 3
1.e-3	0.			1.e-5	Group 4
2.e-3	0.			1.e-5	
3.e-3	0.			1.e-5	
4.e-3	0.			1.e-5	
5.e-3	0.			1.e-5	
Time (seconds)					Group 5
0.					Group 6
1.e5					
2.e5					
3.e5					
4.e5					

A final point needing explanation is the use of times T1SK and T2SK in Group 15 of **trac**. These values are times (in days, as opposed to the seconds used for USERTIME in Group 6 of userc_data.dat) defining the interval over which source/sink values are used. The normal logic of the code is still used to decide whether a source or sink is employed, namely that the source/sink term is only present between T1SK and T2SK. Therefore, the time-varying input must fall between T1SK and T2SK: any portion of the simulation outside of this range will have no source/sink regardless of the input in "userc_data.dat". Practically speaking, setting the values of T1SK and T2SK to 0. and 1.e20 days (very large), respectively, will ensure that the entire span of times in userc_data.dat is used for the mass flux input, and the source is not shut off during the simulation.

6.2.70 Control statement user (optional)

Group 1 - KK

Input Variable	Format	Description
KK	integer	Integer number passed to subroutine user for user defined input parameters. This user subroutine call differs from the one invoked in the control file in that whereas that subroutine is called at every time step, this one is called only at the beginning of the simulation to set parameters that do not change later in the simulation.

The following is an example of user. In this example, the number 5 is passed to the user subroutine.

```
user
5
```

6.2.71 Control statement vapl (optional)

Enable vapor pressure lowering. For more information see the “Models and Methods Summary” of the FEHM Application [Zyvoloski et al. 1999, Equation (127), page 62]. No input is associated with this macro.

6.2.72 Control statement vcon (optional)

Variable thermal conductivity information.

Group 1 - IVCON(i), VC1F(i), VC2F(i), VC3F(i)

Group 2 - JA, JB, JC, IVCND (JA, JB, JC - defined on page 31)

The parameter (i) is incremented each time Group 1 is read. Group 2 lines will refer to this parameter. Group 1 is ended with a blank line.

Input Variable	Format	Description
IVCON(i)	integer	Model type for ith conductivity model. IVCON(i) = 1, linear variation of thermal conductivity with temperature. IVCON(i) = 2, square root variation of thermal conductivity with liquid saturation.
VC1F(i)	real	Reference temperature (°C) for IVCON(i) = 1. Conductivity $\left(\frac{W}{m \cdot K}\right)$ at liquid saturation = 1 for IVCON(i) = 2.
VC2F(i)	real	Reference conductivity $\left(\frac{W}{m \cdot K}\right)$ for IVCON(i) = 1. Conductivity $\left(\frac{W}{m \cdot K}\right)$ at liquid saturation = 0 for IVCON(i) = 2.
VC3F(i)	real	Change in conductivity with respect to temperature for IVCON(i) = 1. Not used for IVCON(i) = 2.
IVCND	integer	Number referring to the sequence of models read in Group 1. The default is 1.

The following is an example of **vcon**. In this example, a linear conductivity model is defined and applied at each node. The reference temperature is 20°C, the reference conductivity is $1 \frac{W}{m \cdot K}$, and the change in conductivity with temperature is 0.01.

vcon			
1	20.0	1.0	1.e-2
1	0	0	1

6.2.73 Control statement wtsi (optional)

Water table, simplified.

Group 1 - NFREE, IFREE_IM_EX, IZONE_FREE(I), I = 1 to NFREE

Input Variable	Format	Description
NFREE	integer	Number of zones to apply water table condition.
IFREE_IM_EX	integer	Update parameter (0 - explicit update, 1 - implicit update).
IZONE_FREE	integer	Zone number.

The following is an example of **wtsi**.

wtsi		
1	1	1

6.2.74 Control statement zone (optional)

Geometric definition of grid for input parameter assignment, the default is input by nodes.

Group 1 - IZONE

Group 2 - X1, X2, X3, X4 (for 2-D) or X1, X2, X3, X4, X5, X6, X7, X8 (for 3-D)

Group 3 - Y1, Y2, Y3, Y4 (for 2-D) or Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8 (for 3-D)

Group 4 - Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8 (for 3-D problems only)

The following alternate form of input may be used (starting with Group 2):

Group 2 - MACRO

Group 3 - XG, YG (for 2D) or XG, YG, ZG (for 3-D) [used with 'list' option]
or

Group 3 - NIN, NODE(1), . . . , NODE(NIN) [used with 'nnum' option]

or

Group 3 - TOL_ZONE, ZXY_MIN, ZXY_MAX [used with 'xyli' option]

Group 4 - XG, YG [used with 'xyli' option]

Input Variable	Format	Description
IZONE	integer	Zone identification number for geometric input.
X1-X8	real	X coordinates defining zone IZONE (m).
Y1-Y8	real	Y coordinates defining zone IZONE (m).
Z1-Z8	real	Z coordinates defining zone IZONE (m).
MACRO	character*4	String denoting alternate input format MACRO = "list", read a list of X, Y, Z - coordinates, one set per line until a blank line is encountered. The nodes corresponding to these coordinates make up the zone. MACRO = "nnum", read the number of specified nodes, followed by the node numbers. These comprise the zone. MACRO = "xyli", read a column radius, followed by a list of X, Y - coordinates, one set per line until a blank line is encountered. The nodes contained in columns centered on each x, y pair and extending to the defined boundaries in the Z direction make up the zone. The column radius is necessary because there are (usually) slight variations in the Z direction of nodes above and below the prescribed X, Y coordinates.
XG	real	X coordinate of node to be included in IZONE (m).
YG	real	Y coordinate of node to be included in IZONE (m).
ZG	real	Z coordinate of node to be included in IZONE (m).
NIN	integer	Number of nodes in IZONE.
NODE(i)	integer	NIN node numbers of the nodes to be included in IZONE.
TOL_ZONE	real	Column radius (m).
ZXY_MIN	real	Minimum Z coordinate for XY list (m).
ZXY_MAX	real	Maximum Z coordinate for XY list (m).

The geometric zone description is implemented by defining geometric regions. The coordinates given in Group 2, 3, and 4 refer to node positions shown in Fig. 2. All properties defined by node (JA, JB, JC) in any control statements may be defined by **zone**. In the previous macro descriptions if $JA < 0$, then the zone $IZONE = ABS(JA)$ is referenced.

It is a good policy to refer to the input check file to insure that node assignments have been made as expected. When X, Y, Z coordinates are used to define zones, boundaries of those zones may be slightly different than specified. This is due to the inclusion of volume from elements adjoining included nodes.

When macro statements **dpdp** and **dual** are used, additional zone definitions are automatically generated. These are identified by zones 101-200 for the first set of matrix nodes and 201-300 for the second set of

matrix nodes. For example, Zone 101 corresponds to the matrix material occupying the same space occupied by the fracture identified by Zone 1. Furthermore, Zone 201 refers to the second matrix layer in the **dual** control statement. Zones for the **dmdp** and **dual** matrix nodes may be explicitly defined and will not be superseded by automatically generated zones.

The macro **zone** must precede the usage of a ZONE reference. **zone** is ended with a blank line. **zone** can be called more than once and regions redefined. When this is done, all previous zone definitions are eliminated. A node may be included in only a single zone at a time.

The following are examples of **zone**. In the first example, 8 zones (for a 2-D problem) are defined, zone 1 is all the nodes that lie within the area bounded by the four points with X, Y coordinates given by (0.,1075.), (1000.,1074.), (1000.,1079.) and (0.,1080.) and similarly for zones 2 - 6. Zone 7 is defined using the "nnum" keyword and consists of a single node, 100. Zone 8 is defined using the "list" keyword and consists of the single node closest to coordinates (0., 0.).

zone				
1				
0.00	1000.00	1000.00	0.00	
1075.00	1074.00	1079.00	1080.00	
2				
0.000	1000.00	1000.00	0.00	
870.000	869.000	1074.00	1075.00	
3				
0.000	1000.00	1000.00	0.000	
860.000	859.000	869.000	870.000	
4				
0.000	1000.00	1000.00	0.000	
780.000	779.000	859.000	860.000	
5				
0.000	1000.00	1000.00	0.000	
730.000	730.000	779.000	780.000	
6				
0.000	1000.00	1000.00	0.000	
700.000	700.000	730.000	730.000	
7				
nnum				
1	100			
8				
list				
0.	0.			

In the second example 2 zones of a 3D problem are defined using "xyli". The columns have a radius of 0.5 m extending from 0 - 1000 m in Z, and are centered around the nodes closest to the given coordinates.

zone			
1			
xylist			
0.5	0.	1000.	
0.	0.		
0.	1.0		
0.	1.5		
0.	2.0		
0.	3.0		
<hr/>			
2			
xylist			
0.5	0.	1000.	
1.0	0.		
1.0	1.0		
1.0	1.5		
1.0	2.0		
1.0	3.0		

6.2.75 Control statement zonn (optional)

The input for this macro is identical to macro **zone**, except that previous zone definitions are retained between calls unless specifically overwritten.

7.0 OUTPUT

Output is found in the code generated files (output file, write file, history plot file, solute plot file, contour plot file, contour plot file for dual or dpdp, stiffness matrix data file, input check file, submodel output file, PEST output files, streamline particle tracking output files, and AVS output files) described in Section 5.0.

Macro commands (input options) dealing with output control are **cont** (page 45), **ctrl** (page 48), **nod2** (page 85), **node** (page 85), **mptr** (page 75), **pest** (page 88), **ptrk** (page 91), **sptr** (page 124), and **time** (page 133): **cont** is used to specify output format and time intervals for contour data output (*fehmn.con*, *fehmn.dp*); **ctrl** is used to specify if element coefficients calculated in the code should be saved (*fehmn.stor*); **node** and **nod2** are used to provide nodal or coordinate positions for which general information and history data will be output (*fehmn.out*, *fehmn.his*, *fehmn.trc*, and terminal output); **mptr** has an option to specify whether or not particle tracking information is written to the restart file (*fehmn.fin*); **pest** is used to specify PEST parameter estimation routine output format (*fehmn.pest*, *fehmn.pest1*); **ptrk** has an option to specify whether or not particle tracking information is written to the restart file (*fehmn.fin*) and what information to output; **sptr** has options to specify what streamline particle tracking information will be output (*fehmn.sptr1*, *fehmn.sptr2*, *fehmn.sptr3*); **subm** is used to specify nodes and boundary conditions should be output for an extracted submodel region; and **time** provides input on the time printout interval for nodal information (*fehmn.out* and terminal output).

The code itself provides no graphical capabilities. History plots of the energy source, source strength, temperature, pressure, capillary pressure, and saturation are made from the *filen.his* FEHM output files. Data from the *filen.trc* files is used to make history tracer plots of the 10 species concentrations. Contour plots can be made from the *filen.con*, *filen.dp*, and AVS FEHM output files.

AVS provides tools for visualization and analysis of volumetric scalar and vector data. Contour plots using 2-d quad grids and 3-d hex grids for material properties, temperature, saturation, pressure, velocities, and solute concentrations can be made. The plots can be rotated, zoomed, scaled, translated, and printed. Axis values and the color bar can be customized. AVS FEHM output files are available for the following node data: material properties, liquid and vapor phase values, velocity and pressure values, temperature, saturation, concentration, and for the dual and dpdp models. The AVS output files from FEHM are written in an ASCII format that can be imported into AVS UCD graphics routines for viewing.

Additional information on the data found in the output files is given below.

7.1 Output file (*filen.out*)

Information contained in the general output file is mostly self explanatory. The file starts with the code version, date, and time followed by the user input problem title. A summary of the I/O files used, macro control statements read, and array storage follow. Variable information for user specified nodes at user selected time intervals is written. The file ends with a summary of simulation time, number of time steps in the problem, the number of iterations taken, and total cpu time.

7.2 Write file (*filen.fin*)

The write file contains the final values of pressure, temperature, saturation, and simulation time for the run. The final version of the file is generally written at the end of the simulation. This information is also written if the minimum user supplied time step has been realized and execution is terminated. If the write file

has not been specified at startup the code will use *fehmn.fin*. The primary use of the write file is as a restart file. The write file contains the following:

- Code version number, date, time
- Problem title
- Simulation time (days)
- Gas flag [h20 (default), ngas, air]
- Tracer flag [trac, ptrk, ntra (default - no output)]
- Stress flag [strs, nstr (default - no output)]
- Dpdp flag [dpdp, ndpd (default - no output)]
- Dual flag [dual, ndua (default - no output)]

If the gas flag is 'h20' (neither air or ngas are set), followed by

- Final temperature (°C) at each node
- Final saturation (dimensionless) at each node
- Final pressure (MPa) at each node

Or if 'ngas' flag is set, followed by

- Final temperature (°C) at each node
- Final saturation (dimensionless) at each node
- Final pressure (MPa) at each node
- Final capillary pressure (MPa) at each node

Or if 'air' flag is set, followed by

- Final saturation (dimensionless) at each node
- Final pressure (MPa) at each node

If 'trac' flag is set followed by

- Number of species
- Species concentration (vapor or liquid, dimensionless) for each node for each species

Or if 'ptrk' flag is set followed by

If (ABS (PRNT_RST) = 2) (see PRNT_RST description in the PTRK macro)

Label: mass flux values

Number of mass flux values

Mass flux value (kg/s) for each connection of each node, starting with node 1. Note: mass flux values for the fracture domain are listed first followed by the mass flux values in the matrix domain. The mass flux between fracture and matrix elements are listed last.

Number of particles, final random number seed

Final node position for each particle. If the value is negative, the particle left the model domain at a fluid sink at that node.

Fractional time remaining at current node for each particle.

Multiplier to the plug flow residence time for each particle at the current node position, accounting for dispersion, sorption, and matrix diffusion effects.

Age for each particle, i.e. the time since the particle entered the system. However, if the particle has left the system, this value is the time that the particle left the system.

If the random number seed in the file is negative, the arrays for the fractional time remaining and the multiplier to the plug flow time have been omitted using the PRNT_RST = -1 or PRNT_RST = -2 option (see PRNT_RST description in the PTRK macro). A restart simulation using this input file will only approximate the behavior of particles since each particle will be assumed to have just entered the node. It is preferable to restart a particle tracking simulation using a file that contains the full restart information.

If strs (not implemented in this version)

If 'dmdp' or 'dual' flag is set

The above information includes dual porosity/dual permeability nodes.

7.3 History plot file (*filen.his*)

The history plot file contains the following:

Code version number, date, time

Problem title

Gas flag ('ngas', 'airw', or blank)

Tracer flag ('trac' or blank)

Stress flag ('strs' or blank)

Number of nodes for which data are output

Node number and X, Y, and Z coordinate (m) of each node for which data are output

'headings'

Depending on problem flags, 2 lines with field descriptors, Case 1 (default):

"node flow enthalpy(Mj/kg) flow(kg/s) temperature (deg C) total pressure (Mpa)"

"capillary pressure (Mpa) saturation (kg/kg)"

or Case 2 (hydraulic head):

"node flow enthalpy(Mj/kg) flow(kg/s) temperature (deg C) hydraulic head (m)"

"total pressure (Mpa) saturation (kg/kg)"

or Case 3 (ngas):

"node flow enthalpy(Mj/kg) flow(kg/s) temperature (deg C) total pressure (Mpa)"

"air pressure (Mpa) capillary pressure (Mpa) saturation (kg/kg) relative humidity"

And for each time step

Time (days) followed by

For Case 1:

Node number, Energy source (MJ/s), Source strength (kg/s), Temperature (°C), Total pressure (MPa), Capillary pressure (MPa), and Saturation (dimensionless) for each specified output node.

For Case 2 (hydraulic head):

Node number, Energy source (MJ/s), Source strength (kg/s), Temperature (°C), Hydraulic head (m), Total pressure (MPa), and Saturation (dimensionless) for each specified output node.

For Case 3 (ngas):

Node number, Energy source (MJ/s), Source strength (kg/s), Temperature (°C), Total pressure (MPa), Air pressure (MPa), Capillary pressure (MPa), Saturation (dimensionless), and Relative humidity for each specified output node

7.4 Solute plot file (*filen.trc*)

Solute data is output for the same nodes used for the history plot file. The solute plot file contains:

Code version number, date, time

Problem title

Number of nodes for which data are output

Node number and X, Y, and Z coordinate (m) of each node for which data are output

Number of different species/components for tracer solution, Number of liquid components, Number of immobile components, Number of vapor components, and Number of aqueous complexes

and for each time step and each species

Time (days), species number followed by

Species concentration (dimensionless) for each specified output node.

When particle tracking is used, the concentration can be output in several different forms (number of particles, number per fluid mass, or number per total volume). The choice of which form to use is input in the **ptrk** macro.

7.5 Contour plot file (*filen.con*)

The contour plot file contains:

Code version number, date, time

Problem title

Tracer ('trac') solution or blank

Stress ('strs') solution or blank

Number of nodes for which data are output

X, Y, and Z coordinate (m) of each node for which data are output

Number of nodes per element, total number of elements

Nodal connectivity information for each node of each element

X, Y, Z permeability (m^2) for each node

X, Y, Z thermal conductivity ($\frac{W}{m \cdot K}$) for each node

Porosity, Rock specific heat ($\frac{MJ}{kg \cdot K}$), Capillary pressure (MPa) for each node

Number of degrees of freedom per node for the current problem, Direction of gravity in problem, Value of gravity

If tracer solution is present

Number of species

and for each specified time

Time (days), injection phase (0 liquid, < 0 vapor) followed by

If injection phase is liquid

Liquid transmissibility / density, Liquid density (kg/m^3), Pressure - Capillary Pressure (MPa), Temperature ($^{\circ}C$)

and if tracer solution is present

Species concentration of liquid phase

Or if injection phase is vapor

Vapor transmissibility / density, Vapor density (kg/m^3), Pressure (MPa), Temperature ($^{\circ}C$)

and if tracer solution is present

Species concentration of vapor phase.

7.6 Contour plot file for dual or dpdp (*filen.dp*)

The contour plot file for dual or dpdp contains the same information as the regular contour plot file only the parameter values are for the dual porosity / dual permeability nodes.

7.7 Stiffness matrix data (*filen.stor*)

The stiffness matrix data file is used to store the finite element coefficients for each node. It eliminates the need for the code to recompute the coefficients for subsequent runs. It contains the following:

Code version number, date, time

Problem title

Number of storage locations needed to store geometric input types, Number of nodes, Size of connectivity array

Volume associated with each node

Nodal connectivity information for each connection

Position of geometric coefficient for each connection

Diagonal position in connectivity array for each node

Finite element geometric coefficient for each storage location

If stress solution is enabled

Finite element geometric coefficient for each storage location for the stress module.

7.8 Input check file (*filen.chk*)

This file contains a summary of input information that may be of use in debugging or memory management of the code. The positions of maximum and minimum values of input parameters and derived quantities are given. Also provided is an analysis of array storage requirements.

7.9 Submodel output file (*filen.subbc*)

The submodel output file contains “**flow**” macro data that represents boundary conditions for an extracted submodel (i.e., the output will use the format of the “**flow**” input macro). The file contains:

Heading: “flow Boundary Conditions Output:”, code version number, date, time

for each submodel node, if boundary type is head or pressure,

Node number, Node number, 1, Head (m) or Pressure (MPa), Impedance parameter, #, X coordinate, Y coordinate, Z coordinate

or if boundary type is flux

Node number, Node number, 1, Flux (kg/s), 0.0d00, #, X coordinate, Y coordinate, Z coordinate

A blank line to signal end of flow macro input followed by the file termination macro

stop

An example is provided with **subm** input on page 131.

7.10 Error output file (*fehmn.err*)

This file contains the code version number, date, and time followed by any error or warning messages issued by the code during a run.

7.11 Multiple simulations script files (*fehmn.pre*, *fehmn.post*)

The multiple simulations script file *fehmn.pre* contains UNIX shell script style instructions for pre-processing input data, while the script file *fehmn.post* contains UNIX shell script style instructions for post-processing data.

7.12 PEST output files (*filen.pest*, *filen.pest1*)

The PEST output file is used to output data in a format suitable for use by the Parameter Estimation Program (PEST) (Watermark Computing, 1994). The first file (*filen.pest*) contains:

Heading: “PEST Output:”, code version number, date, time

First parameter label: “pressures” or “heads”

node number and pressure (MPa) or head (ft) for each specified output node

Second parameter label: "saturations"

node number and saturation for each specified output node

Third parameter label: "temperatures"

node number and temperature (°C) for each specified output node

Fourth parameter label: "permeabilities"

node number and x, y, and z permeability (m²) for each specified output node

Heading: "Total Flux (kg/s) Leaving Zone (flxz macro option)"

"Zone Number Source Sink Net Boundary"

zone number, source flux, sink flux, net flux, boundary flux

The second file (*flen.pest1*) contains:

Heading: "PEST Output:", code version number, date, time

Parameter label: "pressures" or "heads"

node number, relative permeability model used, pressure (MPa) or head (ft), saturation, and temperature (°C) for each specified output node

7.13 Streamline particle tracking output files (*flen.sptr1*, *flen.sptr2*, *flen.sptr3*)

The streamline particle tracking output files contain information generated during a streamline particle tracking simulation. Depending on output options selected (macro **sptr**) zero, one, two or three output files are generated.

When option IPRT ≥ 1, the first file (*flen.sptr1*) contains:

Code version number, date, time

Problem title

"days=", Current time of streamline particle tracking iteration (days), streamline particle tracking timestep number for current iteration, and

For each particle:

Particle number, X coordinate of particle, Y coordinate of particle, Z coordinate of particle, Element or node number where the particle is located

When option IPRTO ≥ 1, the second file (*flen.sptr2*) contains:

Code version number, date, time

Problem title

Heading: "particle_number x y z time porosity saturation permeability rock_density pressure temperature zone old_node new_node" (Note that the heading only includes property titles for the default properties or those properties specified by keyword.)

For each particle:

Particle number, X coordinate of particle, Y coordinate of particle, Z coordinate of particle, Current time that the particle has reached (days), Property value of the unit the particle is residing in for each specified keyword [in the following order, if specified: porosity,

saturation, permeability (m^2), density (kg/m^3), pressure (MPa),
temperature ($^{\circ}C$), zone number], Element or node number where the
particle is located, Previous node number

If option "zbtc" is invoked, the third file (*filen.sptr3*) contains:

Code version number, date, time

Problem title

Heading: " Time (days) Zone1 Particles . . ."

Time (days), Cumulative number of particles that have arrived at each
specified zone for breakthrough curve output

7.14 SURFER and TECPLOT output files (*surzone_number.txt*, *veczone_number.txt*, *trczone_number.txt* or *teczone_number.plt*, *veczone_number.plt*, *trczone_number.plt*)

The SURFER and TECPLOT output files contain output data (pressure or head,
saturations, temperatures, fluxes, permeabilities, saturations, porosities,
velocities, and concentrations) for selected locations or nodes in a format suitable
for use by the SURFER (Countouring and 3D Mapping) or TECPLOT plotting
programs. The file data content is the same for either plot program with the
exception that the initial heading line is omitted from the SURFER files and
heading formats are slightly different. The file names are generated using the
output zone number (+ 1000) and contour plot number (+ 10000), i.e.
sur1001_10001.txt or *tec1001_10001.plt*. The first file (*sur*.txt*, or *tec*.plt*)
contains:

Heading (tecpot files only): "TITLE =", Code version number, date, time

For pressure output

Heading: VARIABLES = "X" "Y" "Z" "pres" "temp" "px" "py" "pz" "sat"
"node"

ZONE T =", zone number, "Simulation time", simulation time (days)

and for each node

X coordinate (m), Y coordinate(m), Z coordinate(m), Pressure (MPa),
Temperature ($^{\circ}C$), X permeability (m^2), Y permeability (m^2), Z permeability
(m^2), Saturation, Node number

or for head output

Heading: VARIABLES = "X" "Y" "Z" "head" "sat" "source" "px" "py" "pz"
"por" "node"

ZONE T =", zone number, "Simulation time", simulation time (days)

and for each node

X coordinate (m), Y coordinate(m), Z coordinate(m), Head (m), Saturation,
Source flux (kg/s), X permeability (m^2), Y permeability (m^2), Z
permeability (m^2), Porosity, Node number

The second file (*vec*.txt*, or *vec*.plt*) contains:

Heading (tecpot files only): "TITLE =", Code version number, date, time

Heading: VARIABLES = "X" "Y" "Z" "Vx" "Vy" "Vz" "node"

ZONE T =", zone number, "Simulation time", simulation time (days)

and for each node

X coordinate (m), Y coordinate(m), Z coordinate(m), X velocity (m²/s), Y velocity (m²/s), Z velocity (m²/s), Node number

The third file (*trc*.txt*, or *trc*.plt*) contains:

Heading (tecplot files only): "TITLE =", Code version number, date, time

Heading: VARIABLES = "X" "Y" "Z" "anl" "anv" "node"

ZONE T =", zone number, "Simulation time", simulation time (days)

and for each node

X coordinate (m), Y coordinate(m), Z coordinate(m), Liquid concentration, Vapor concentration, Node number

7.15 AVS log output file (*filen.10001_avs_log*)

The AVS log output file is identical for AVS and AVS Express output. It contains:

Code version number, date

AVS log identifier

Problem title

and for each specified time

AVS output file prefix, Call number, and Time (days)

7.16 AVS header output files (*filen.number_type_head*)

The AVS ASCII (formatted) header files are identical for AVS and AVS Express output. The data types, "*mat*", "*sca*", "*vec*" or "*con*", are described in Section 7.18. The header files contain:

20 lines of text with information about the FEHM AVS output files. The text is followed by a one line AVS UCD file header containing:

number of nodes

number of cells

number of data components for the nodes

number of data components for the cells (currently 0)

number of data components for the model (currently 0)

7.17 AVS geometry output file (*filen.10001_geo*)

The ASCII (formatted) geometry file for AVS contains the following:

Node id and X, Y, Z coordinates for each node

Cell id, Material id, Cell type, and the list of Cell vertices

The ASCII (formatted) geometry file for AVS Express contains one additional line of data at the beginning of the file, followed by the data specified above:

Number of nodes, Number of elements, 0, 0, 0

7.18 AVS data output files (*filen.number_type_node*)

All the ASCII (formatted) node data files for AVS contain the following headers:

Number of data components and size of each component

A label/unit string for each data component

followed by for each node

the associated node data (described by data type below).

All the ASCII (formatted) node data files for AVS Express contain the following headers, on a single line delimited by “ : ” :

Current simulation time (with format “nodes at *time* days”)

A label/unit string for each data component

followed by for each node

the associated node data (described by data type below), delimited by “ : ”.

7.18.1 Material properties (*_mat* and *_mat_dual*)

These data will consist of 11 fields. The order of the fields are:

Permeability in X, Y, and Z direction (m^2),

Thermal conductivity in X, Y, and Z direction ($\frac{W}{m \cdot K}$),

Porosity,

Rock specific heat ($\frac{MJ}{kg \cdot K}$),

Capillary pressure (MPa),

Relative permeability model, and

Capillary pressure model.

The dual or dpdp values for each of these fields will be written to a file with “mat_dual_node” appended to the file name.

7.18.2 Scalar parameters (*_sca* and *_sca_dual*)

These data files will contain scalar data including

Saturation,

Temperature ($^{\circ}C$),

Liquid pressure(MPa), and

Vapor pressure(MPa).

If dual values are calculated, they can be written to the sca_dual output file.

7.18.3 Vector parameters (*_vec* and *_vec_dual*)

These data files contain the vector values for

Liquid and Vapor velocities (m/s).

If idualp is defined, dual porosity values for the vapor phase will be written to the _vec_dual file. If idpdp is defined, double porosity/double permeability values for the liquid phase will be written.

7.18.4 Solute concentrations (*_con* and *_con_dual*)

Up to 20 fields per node can be written for solute concentrations. The number written is determined by the number of species. The dual counterparts to each will be written to the *_con_dual* file.

8.0 SYSTEM INTERFACE

8.1 System-Dependent Features

In addition to standard intrinsic math routines only two system routines are required by the FEHM code. The code uses a system call to get the date (subroutine dated) and a system routine to get the CPU clock time (subroutine timing).

8.2 Compiler Requirements

FEHM Version 2.21 is written for Fortran 90. FEHM has been successfully compiled and run on SUN and PC computers.

8.3 Hardware Requirements

No special hardware features or environments are required by the software. The code will run on SUN workstations running Solaris 7 or later (UNIX) and PC workstations running Windows 2000 or later or Linux 2.4.18 or later. Memory requirements depend on the problem being modeled (based on the number of nodes). It is suggested that the system being used have a minimum of 128 MB of memory.

8.4 Control Sequences or Command Files

None.

8.5 Software Environment

N/A

8.6 Installation Instructions

8.6.1 Creating the FEHM binary from source (UNIX)

On the system where FEHM is to be installed, make an installation directory, with subdirectories src and objects:

```
mkdir fehm
```

```
mkdir src objects
```

Copy all fehm source files (i.e., extract them from a tar file -- fehm_src.tar) into the src directory:

```
cd fehm/src
```

```
tar xvf fehm_src.tar
```

A Makefile is included and should be placed in your objects directory. To compile and link FEHM, change into the objects directory and compile the code:

```
cd fehm/objects
```

```
make -OR- make -f Makefile
```

The makefile creates an executable called:

```
xfehm_v2.21
```


It should be noted that FEHM uses the GZSOLVE Application (ECD-97) reuse components, solve_new, solve_r dof, and slvesu. The GZSOLVE subroutines are compiled directly into this version of FEHM.

8.6.2 Installation Verification and Validation

A series of test scripts have been developed to automate the validation procedure for FEHM. They are described in more detail in the APPENDIX: FEHM VALIDATION SCRIPTS, of the Validation Test Plan for the FEHM Application Version 2.21 (10086-VTP-2.21-00). See the FEHM VTP for a discussion of the tests performed and their results.

8.6.3 FEHM for YMP

For official use of the FEHM code on the YMP project, an executable version should be obtained from the project configuration management group. For binary installation instructions, refer to the Installation Test Plan for the FEHM Application Version 2.21 (10086-ITP-2.21-00).

9.0 EXAMPLES AND SAMPLE PROBLEMS

The following describes execution of the FEHM code. Section 9.1 discusses the construction of an input file. Section 9.2 illustrates the entire procedure for executing the FEHM code using terminal input. Example 1 describes the setup and results from a simple 2-D heat conduction simulation. The remaining sections provide more complex example problems and deal only with problem setup and expected results.

9.1 Constructing an Input File

FEHM is a very general simulation code. Thus it is preferable to discuss the construction of an input file from a problem oriented point of view. In what follows the needs of the physical problem (initial conditions, boundary conditions, etc.) will be addressed in terms of the macro statements.

Initial conditions. These are needed for every problem, even if it is a steady state simulation. If the simulation is comprised of fully saturated water flow or heat conduction only, then the appropriate control statement would be **init** (page 68). The use of **init** also allows the specification of initial temperature and pressure (gravity) gradients. If two phase flow is prescribed (thermal or isothermal) then entering the initial conditions through the control statement **pres** (page 90) is more convenient. Initial values for noncondensable gas are handled in the **ngas** (page 83) control statement. It should be remembered that if a restart file is present, those values will have precedence over values input in control statement **init** but not over values input in control statement **pres**. Solute initial conditions are prescribed through the control statement **trac** (page 135).

Boundary conditions. Fluid and heat flow boundary conditions can be prescribed through control statements **pres**, **boun** (page 41), **flow** (page 59), and **hflx** (page 67). Boundary conditions are entered with **pres** by specifying a negative phase state designation (the code will actually use the absolute value of the phase state designation). In this case the code will keep the variable values constant at whatever value was prescribed in **pres**. Flowing pressures are input with the **boun** or **flow** control statement. Solute boundary conditions are prescribed through the control statement **trac**.

Material and Energy Balance Equations. The choice of the coupled system equations is made in control statements **sol** (page 124), **ngas**, and **air** (page 40).

Rock or Media Properties. These are found in the **rock** (page 104) and **perm** (page 87) control statements.

Fluid Properties. These are found in control statement **eos** (page 55), which is optional. If **eos** is not invoked, then the properties of water and air included in the code are used. Relative permeabilities, depending on both the fluid and media type, are found in control statement **rlp** (page 100).

Mesh Geometry and Nodal Coordinates. This geometry information is found in control statements **coor** (page 47) and **elem** (page 52). This information is usually created with a mesh generation program.

Simulation Time. The time stepping information including printout intervals and time step sizing is found in control statement **time** (page 133).

Numerics. Convergence criteria, upwinding parameters, fill-in for the preconditioned conjugate gradient solver and geometry type (2-D, 3-D, radial) are entered with control statement **ctrl** (page 48).

Advanced Iteration Control. Reduced degree of freedom methods are invoked with the **iter** (page 69) control statement. One important quantity entered with this statement is the maximum time for the job to run on the computer.

Sources and Sinks. These are input with the control statements **boun** or **flow**. Care must be taken as the parameters have different meanings for different physical models.

The following table lists the input macros which should be used to formulate various types of problems.

Problem Type : Heat Conduction		Problem Type : Water / Water Vapor / Heat Equivalent Continuum, Dual Porosity*, Dual Permeability**	
Required Macros	Optional Macros	Required Macros	Optional Macros
title	cont	title	cden
boun or flow or hflx	finv	boun or flow or hflx	cont
cond	flo2	cond	eos
coor	flxo or flxz	coor	exrl
ctrl	iter	ctrl	finv
elem	node or nod2	elem	flo2
init or pres	renu	init or pres	flxo or flxz
rock	rflx	perm	fper
sol	text or comments (#)	rlp	gdpm
time	user	rock	hflx
stop	vcon	sol	iter
	zone or zonn	time	node or nod2
		stop	ppor
			renu
		dual (* only)	rflx
		dpdp (** only)	rxn
			text or comments (#)
			trac
			user or userc
			vcon
			velo
			zone or zonn

Table VI. Required and Optional Macros by Problem Type (Continued)			
Problem Type : Air / Water / Water Vapor / Heat or Gas / Water / NAPL / Heat Equivalent Continuum, Dual Porosity* Dual Permeability**		Problem Type : Air / Water / No Heat Equivalent Continuum, Dual Porosity* Dual Permeability**	
Required Macros	Optional Macros	Required Macros	Optional Macros
title	adif	title	bous
boun or flow or hflx	cden	airwater	cont
cond	cont	boun or flow	eos
coor	eos	coor	exrl
ctrl	finv	ctrl	finv
elem	flo2	elem	flo2
init or pres	flxo	init or pres	flxo
ngas	fper	node or nod2	fper
perm	gdpm	perm	gdpm
rlp	iter	rock	head
rock	node or nod2	sol	iter
sol	ppor	time	ppor
time	renu	stop	pres
stop	rflx		renu
	rxn	dual (*only)	rlp
dual (*only)	szna	dpdp (**only)	rxn
dpdp (**only)	text or comments (#)		text or comments (#)
	trac		trac
	user or userc		user or userc
	vapl		vapl
	vcon		velo
	velo		zone or zonn
	zone or zonn		

9.2 Code Execution

To run FEHM, the program executable file name is entered at the system prompt:

```
<PROMPT> xfehm_v2.21
```

Note that executable names may vary depending on the system being used.

The I/O file information is provided to the code from an input control file or the terminal. The default control file name is *fehmn.files*. If a control file with the default name is present in the directory from which the code is being executed, no terminal input is required. If the default control file is not present, input prompts are written to the screen. A short description of the I/O files used by FEHM precedes the initial prompt. The following assumes the default control file was not found in the execution directory (for this example /home/fehm/heat2d).

After the command **xfehm_v2.21** is given, the code queries the user regarding the input files, as follows:

```
Enter name for iocntl -- default file name: not using
```

```
[(name/na or not using), RETURN = DEFAULT]
```

This query asks for a control file name. If a control file name is entered no further terminal input is required. Figure 3 shows the control file that would produce the same results as the terminal responses discussed below and illustrated in Fig. 4.

```

/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/output/heat2d.out

/home/fehm/heat2d/output/heat2d.fin
/home/fehm/heat2d/output/heat2d.his

/home/fehm/heat2d/output/heat2d.chk
some
0

```

Figure 3. Input control file for heat conduction example.

Files that are not needed for output can be represented with a blank line. If names are not provided for the write file and/or the data check file, the code will use the following defaults: *fehmn.fin* and *fehmn.chk*. Following the file names is the flag that controls terminal output. The last line of the file is the user subroutine number. Omitting these values results in no terminal output and no user subroutine call. For now, we assume a carriage return <cr> is entered and a control file is not being used. The following query will appear

```
Enter name for inpt -- default file name: fehmn.dat
```

```
[(name/na or not using), RETURN = DEFAULT]
```

This query asks for an input file name. If a <cr> is given, the default *fehmn.dat* is used for the input file. We shall assume that the input file name entered is

```
input/heat2d.in
```

Note that a subdirectory containing the file is also given. If the file did not exist, the code would repeat the prompt for an input file. Next the code would query to determine if the prefix of the input file name (the portion of the name preceding the final "." or first space) should be used for code generated file names.

```
Do you want all file names of the form input/heat2d.* ? [(y/n), RETURN = y]
*** Note: If "y" incoor and inzzone will equal inpt ***
```

A <cr> will produce files with identical prefixes, including the subdirectory. If the response is negative, the code will query for the names of all required files.

Assume we enter "n".

```
Enter name for incoor -- default file name: input/heat2d.in
```

```
[(name/na or not using), RETURN = DEFAULT]
```

(See Fig. 4 for the remaining file name queries.)

Next a query for terminal output appears.

```
tty output -- show all reference nodes, selected reference nodes, or none:
[(all/some/none), RETURN = none]
```

An "all" reply prints out the primary node information to the terminal at every time step. A "some" reply prints a selected subset of the node information. A reply of "none" suppresses all tty output with the exception of error messages printed if code execution is terminated abnormally or when maximum number of iterations are exceeded. Assume we enter "some".

The next query concerns the subroutine USER. This subroutine is used for special purposes and is not available to the general user.

```
user subroutine number (provided to subroutine USER before every time step):
[RETURN = none]
```

Assume a <cr> is entered.

The code will then print a summary of the I/O files to be used.

The final query regards the acceptance of the file set just created. A "yes" reply denotes that the user has accepted the file set and the code proceeds with calculations. A "no" reply starts the query sequence again so I/O file names may be reentered or modified. A "stop" reply stops the current computer job.

```
If data is OK enter yes to continue, no to restart terminal input,
or stop to end program: [(yes/no/stop), RETURN = yes]
```

Screen output for this example execution using terminal input and a previous version of the code is shown in Fig. 4. The only difference in the output is that the code version identifier and date are updated for the current version. User responses are shown in *italics*.

```
<PROMPT> xfehm_v2.21sun
version FEHM V2.21sun 03-09-15 QA:QA    09/15/2003 11:08:14

**** Default names for I/O files ****

control file                : fehmn.files
input file                  : filen.*
geometry data file         : filen.*
zone data file             : filen.*
output file                : filen.out
read file (if it exists)   : filen.ini
write file (if it exists)  : filen.fin
history plot file         : filen.his
tracer history plot file   : filen.trc
contour plot file         : filen.con
dual or dpdp contour plot file : filen.dp
stiffness matrix data read/write file : filen.stor
input check file          : filen.chk

**** where ****
"filen.*" may be 100 characters maximum. If a name is not entered
when prompted for, a default file name is used. "fehmn.dat" is the
default used for the input file name.

**** note ****
A save file and input check file are always written. If you do not
provide a name for these files, the following defaults will be used:
fehmn.fin, fehmn.chk

Enter name for iocntl -- default file name: not using

[(name/na or not using), RETURN = DEFAULT]
<cr>

Enter name for inpt -- default file name: fehmn.dat

[(name/na or not using), RETURN = DEFAULT]
input/heat2d.in

Do you want all file names of the form input/heat2d.* ? [(y/n), RETURN = y]
*** Note: If "y" incoor and inzone will equal inpt ***
n

Enter name for incoor -- default file name: input/heat2d.in

[(name/na or not using), RETURN = DEFAULT]
<cr>
```

Figure 4. Terminal query for FEHM example run.

Enter name for inzone -- default file name: input/heat2d.in

[(name/na or not using), RETURN = DEFAULT]
<cr>

Enter name for iout -- default file name: input/heat2d.out

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.out

Enter name for iread -- default file name: input/heat2d.ini

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for isave -- default file name: input/heat2d.fin

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.fin

Enter name for ishis -- default file name: input/heat2d.his

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.his

Enter name for istrc -- default file name: input/heat2d.trc

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for iscon -- default file name: input/heat2d.con

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for iscon1 -- default file name: input/heat2d.dp

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for isstor -- default file name: input/heat2d.stor

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for ischk -- default file name: input/heat2d.chk

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.chk

Figure 4. Terminal query for FEHM example run. (Continued)


```
tty output -- show all reference nodes, selected reference nodes, or none:  
[(all/some/none), RETURN = none]  
some
```

```
user subroutine number (provided to subroutine USER before every time step):  
[RETURN = none]  
<cr>
```

First reference output node will be written to tty

File purpose - Variable - Unit number - File name

control	- iocntl	- 0	- not using
input	- inpt	- 11	- input/heat2d.in
geometry	- incoor	- 11	- input/heat2d.in
zone	- inzone	- 11	- input/heat2d.in
output	- iout	- 14	- output/heat2d.out
initial state	- iread	- 0	- not using
final state	- isave	- 16	- output/heat2d.fin
time history	- ishis	- 17	- output/heat2d.his
time his.(tr)	- istrc	- 18	- not using
contour plot	- iscon	- 19	- not using
con plot (dp)	- iscon1	- 20	- not using
fe coef stor	- isstor	- 21	- not using
input check	- ischk	- 22	- output/heat2d.chk

Value provided to subroutine user: not using

```
If data is OK enter yes to continue, no to restart terminal input,  
or stop to end program: [(yes/no/stop), RETURN = yes]  
<cr>
```

Figure 4. Terminal query for FEHM example run. (Continued)

9.3 Heat Conduction in a Square

This simple 2-D problem is used to illustrate input file construction and basic output. Heat conduction in a 1 meter square with an initial temperature, $T_0 = 200\text{ }^\circ\text{C}$, is modeled after a surface temperature, $T_s = 100\text{ }^\circ\text{C}$, is imposed at time, $t = 0$ (Fig. 5). The input parameters used for the heat conduction problem are defined in Table VII. The finite element mesh for this problem is shown in Fig. 6. Only a quarter of the square needs to be modeled because of problem symmetry.

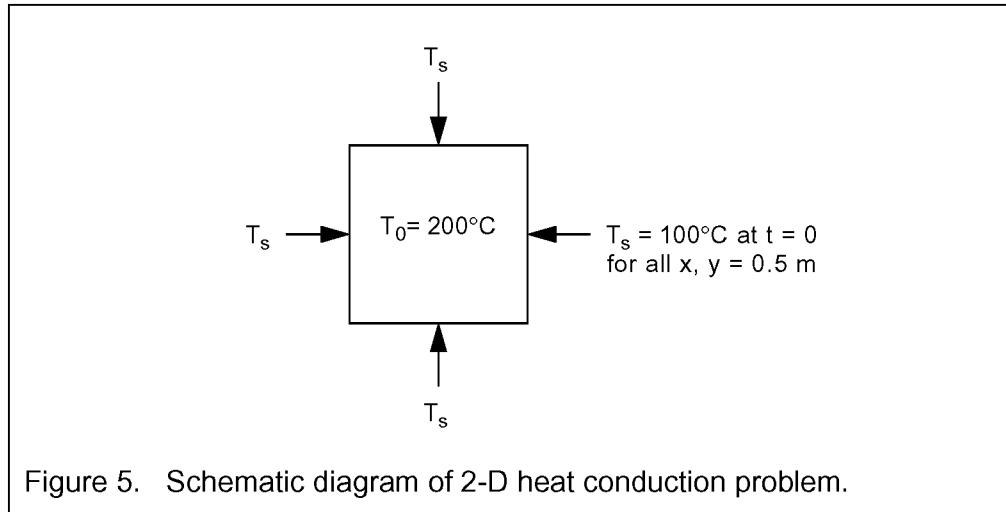
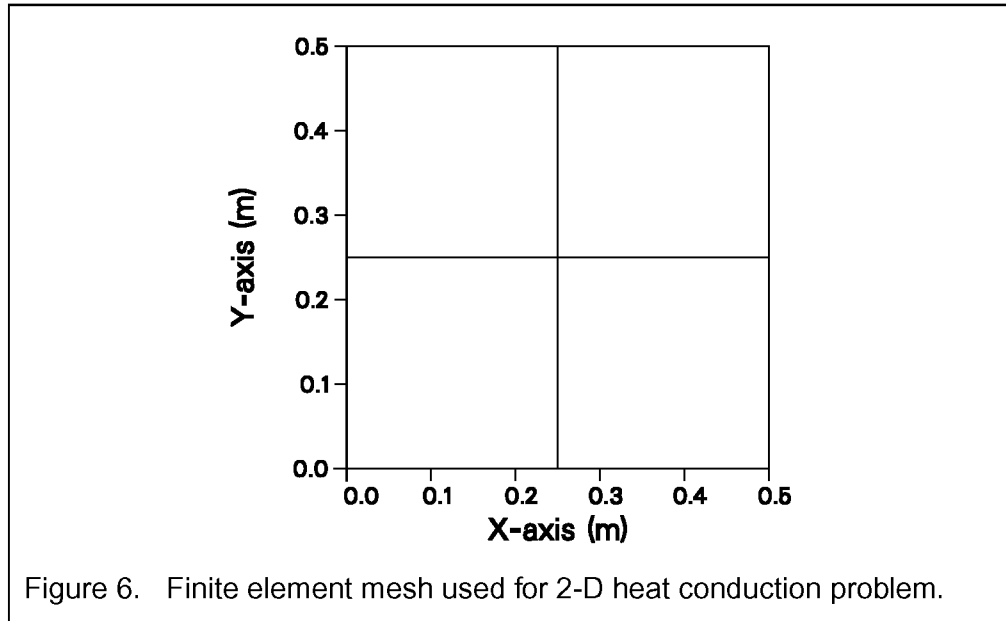


Table VII. Input Parameters for the 2-D Heat Conduction Problem

Parameter	Symbol	Value
Rock thermal conductivity	κ_r	$2.7 \frac{\text{W}}{\text{m} \cdot \text{K}}$
Rock density	ρ_r	2700 kg/m^3
Rock specific heat	C_r	$1000 \frac{\text{J}}{\text{kg} \cdot \text{K}}$
Width	a	0.5 m
Length	b	0.5 m
Initial temperature	T_0	200 °C
Surface temperature for all $x, y = 0.5 \text{ m}$	T_s	100 °C
Rock thermal diffusivity	$\kappa = \frac{\kappa_r}{\rho_r C_r}$	



The input file (see Fig. 7) uses optional macro control statement **node** (output nodes) and the required macro control statements **sol** (solution specification - heat transfer only), **init** (initial value data), **rock** (rock properties), **cond** (thermal conductivities), **perm** (permeabilities), **time** (simulation timing data), **ctrl** (program control parameters), **coor** (node coordinates), **elem** (element node data), and **stop**. For this problem macro control statement **flow** is also used to set the temperature boundary conditions. A portion of the output file is reproduced in Fig. 8.

The analytical solution for 2-D heat conduction (Carslaw and Jaeger, 1959) is given by

$$T = T_s + \frac{16(T_0 - T_s)}{\pi^2} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n}}{(2m+1)(2n+1)} \cos \frac{(2m+1)\pi x}{2a} \cos \frac{(2n+1)\pi y}{2b} e^{-\alpha_{m,n} t}$$

where $\alpha_{m,n} = \frac{\kappa\pi^2}{4} \left[\frac{(2m+1)^2}{a^2} + \frac{(2n+1)^2}{b^2} \right]$ and the region is taken to be

$$-a < x < a, -b < y < b.$$

Figure 9 shows a plot of the simulation results compared to the analytical solution for the selected output nodes at $x = 0$, m and $x = 0.25$ m.

```

***** 2-D Heat Conduction Model (2X2 rectangles) *****
node
  2
  7 5
sol
  -1 -1
init
  10. 0. 200. 0. 0. 200. 0. 0.
rock
  1 9 1 2700. 1000. 0.
cond
  1 9 1 2.7e-00 2.7e-00 2.7e-00
perm
  1 9 1 1.e-30 1.e-30 1.e-30
flow
  1 3 1 10.00 -100.00 1.e03
  3 9 3 10.00 -100.00 1.e03
time
  0.005 4.00 1000 10 1994 02
ctrl
  40 1.e-04 08
  1 9 1 1
  1.0 0.0 1.0
  10 1.0 0.00005 0.005
  1 0
coord Feb 23, 1994 11:39:40
  9
  1 0. 0.50 0.
  2 0.25 0.50 0.
  3 0.50 0.50 0.
  4 0. 0.25 0.
  5 0.25 0.25 0.
  6 0.50 0.25 0.
  7 0. 0. 0.
  8 0.25 0. 0.
  9 0.50 0. 0.
elem
  4 4
  1 4 5 2 1
  2 5 6 3 2
  3 7 8 5 4
  4 8 9 6 5
stop

```

Figure 7. FEHM input file for heat conduction example (heat2d.in).

```
FEHM V2.10 00-06-28 08/07/2000 13:25:08
```

```
***** 2-D Heat Conduction Model *****
```

```
File purpose - Variable - Unit number - File name
```

```
control      - iocntl - 0 - not using
input        - inpt   - 11 - heat2d.in
geometry     - incoor - 11 - heat2d.in
zone         - inzone - 11 - heat2d.in
output       - iout   - 14 - heat2d.out
initial state - iread  - 0 - not using
final state  - isave  - 16 - fehmn.fin
time history - ishis  - 17 - heat2d.his
time his.(tr) - istrc - 0 - not using
contour plot - iscon  - 0 - not using
con plot (dp) - iscon1 - 0 - not using
fe coef stor - isstor - 0 - not using
input check  - ischk  - 22 - fehmn.chk
```

```
Value provided to subroutine user: not using
```

```
**** input title : coor **** incoor = 11 ****
**** input title : elem **** incoor = 11 ****
**** input title : stop **** incoor = 11 ****
**** input title : node **** inpt = 11 ****
**** input title : sol **** inpt = 11 ****
**** input title : init **** inpt = 11 ****
**** input title : rock **** inpt = 11 ****
**** input title : cond **** inpt = 11 ****
**** input title : perm **** inpt = 11 ****
**** input title : flow **** inpt = 11 ****
**** input title : time **** inpt = 11 ****
**** input title : ctrl **** inpt = 11 ****
**** input title : stop **** inpt = 11 ****
BC to BC connection(s) found(now set=0.0)
BC to BC connection(s) found(now set=0.0)
```

Figure 8. FEHM output from the 2-D heat conduction example.

```

pressures and temperatures set by gradients
>>>reading nop from file nop.temp.....
>>>reading nop was succesful....

storage needed for ncon          43 available    43
storage needed for nop           43 available    46
storage needed for a matrix      33 available    33
storage needed for b matrix      33 available    46
storage needed for gmres         81 available    81
storage available for b matrix resized to 33<<<<<<

time for reading input, forming coefficients 0.204E-01

**** analysis of input data on file fehmn.chk          ****

*****
Time Step      1
Timing Information
  Years      Days      Step Size (Days)
  0.136893E-04  0.500000E-02  0.500000E-02
Cpu Sec for Time Step = 0.8081E-03 Current Total = 0.2650E-02

Equation Performance
Number of N-R Iterations:      1
Avg # of Linear Equation Solver Iterations: 3.0
Number of Active Nodes:      9.
Total Number of Newton-Raphson Iterations:      1 , Solver:      3
Largest Residuals
EQ1 R= 0.1660E-07 node= 5 x=0.2500 y=0.2500 z= 1.000
Node Equation 1 Residual Equation 2 Residual
  7  0.111444E-07  0.185894E-01
  5  0.165983E-07  0.135450E+01

Nodal Information (Water)
source/sink          source/sink
Node p(MPa) e(MJ) l sat temp(c) (kg/s) (MJ/s)
  7  10.000  0.00  0.000  199.981  0.  0.
  5  10.000  0.00  0.000  198.645  0.  0.

Global Mass & Energy Balances
Total mass in system at this time:      0.000000E+00 kg
Total mass of steam in system at this time: 0.000000E+00 kg
Total enthalpy in system at this time: 0.105123E+03 MJ

Water discharge this time step: 0.000000E+00 kg (0.000000E+00 kg/s)
Water input this time step: 0.000000E+00 kg (0.000000E+00 kg/s)
Total water discharge: 0.000000E+00 kg (0.000000E+00 kg/s)
Total water input: 0.000000E+00 kg (0.000000E+00 kg/s)

```

Figure 8. FEHM output from the 2-D heat conduction example. (Continued)

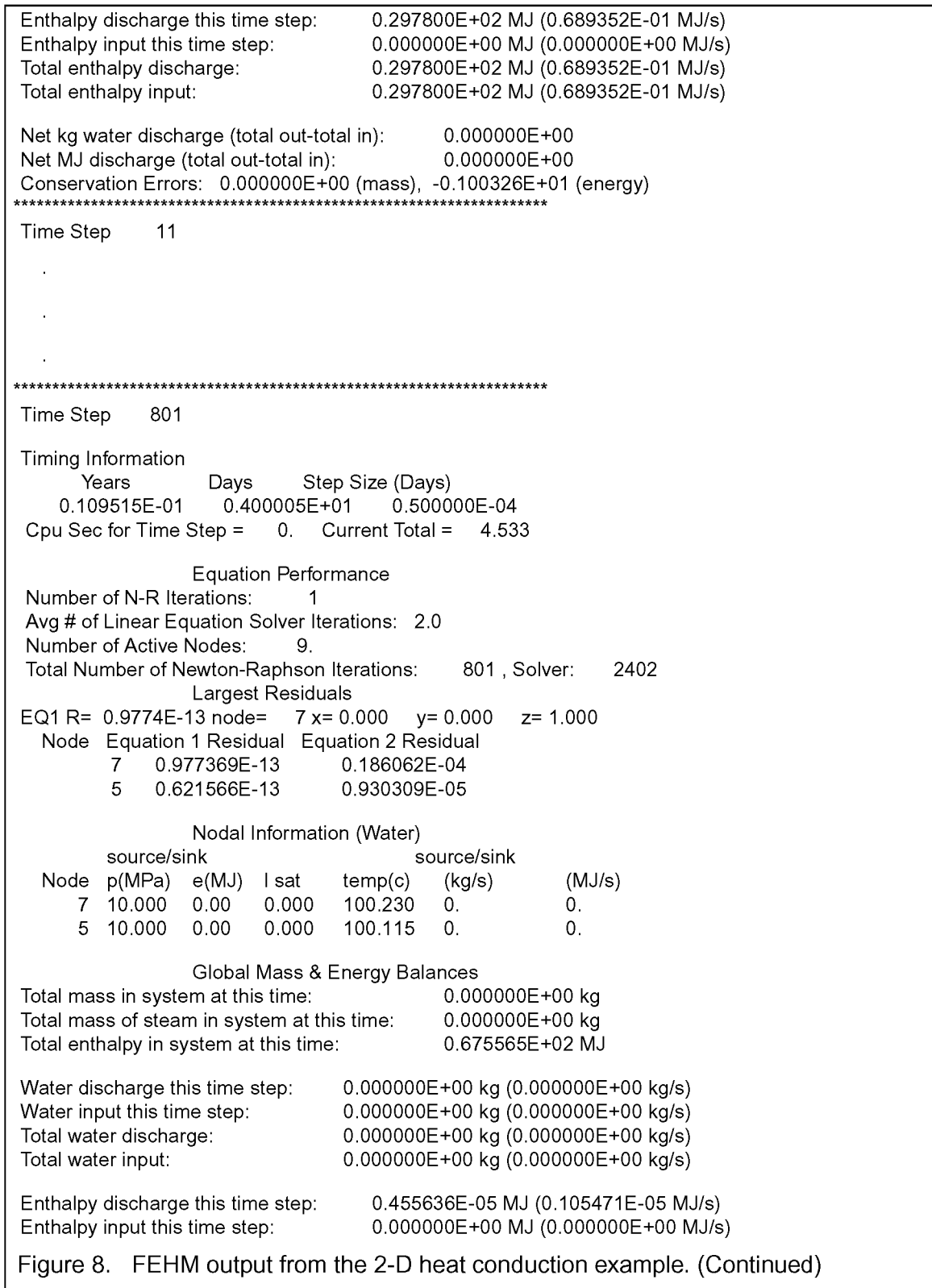


Figure 8. FEHM output from the 2-D heat conduction example. (Continued)

```

Total enthalpy discharge:      0.673463E+02 MJ (0.155894E+02 MJ/s)
Total enthalpy input:         0.673463E+02 MJ (0.155894E+02 MJ/s)

Net kg water discharge (total out-total in):0.000000E+00
Net MJ discharge (total out-total in):0.000000E+00
Conservation Errors:  0.000000E+00 (mass), -0.100144E+01 (energy)

simulation ended: days  4.00  timesteps  801

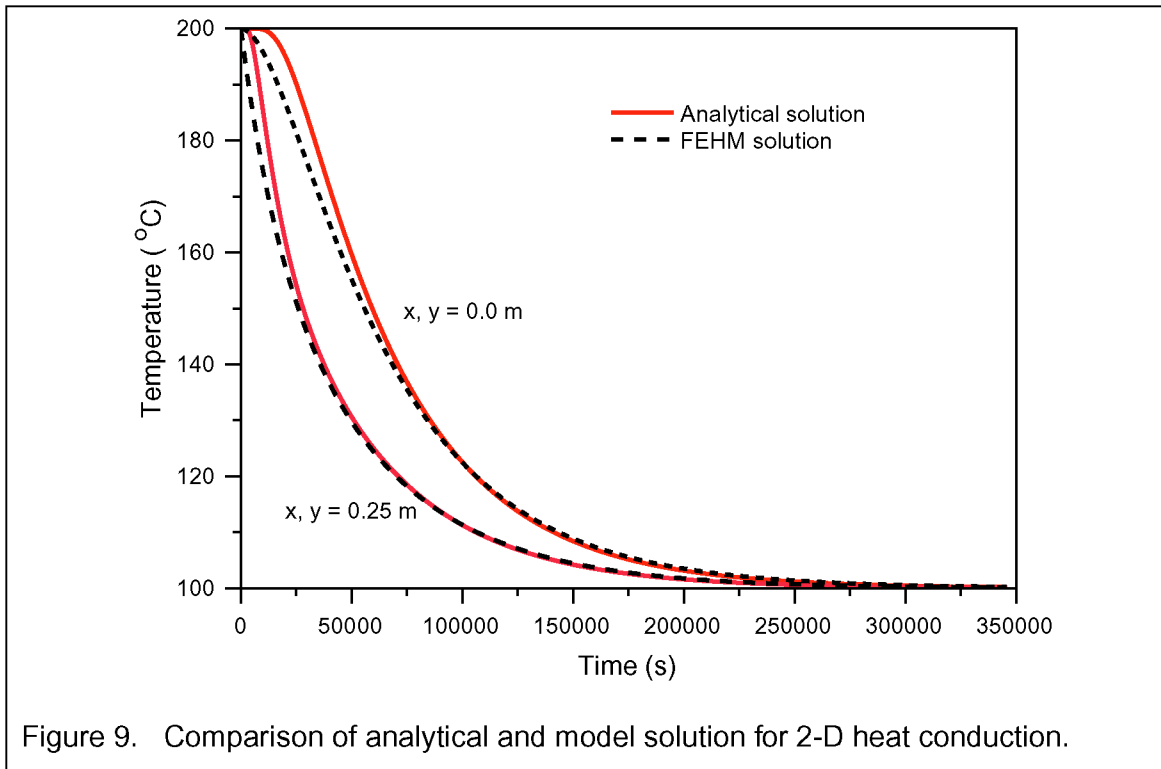
total N-R iterations =    801
total solver iterations =  2402

total code time(timesteps) =  0.526277

**** -----
**** This program for
**** Finite Element Heat and Mass Transfer in porous media
**** -----
**** Version      : FEHM V2.10 00-06-28
**** End Date     : 08/07/2000
**** Time        : 13:25:08
**** -----
****

```

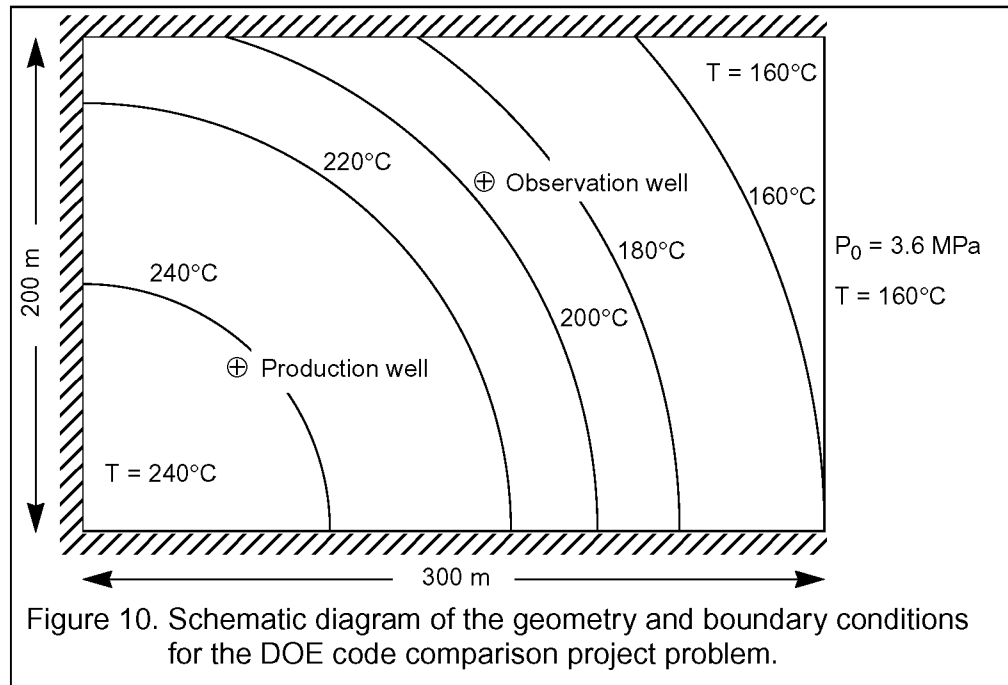
Figure 8. FEHM output from the 2-D heat conduction example. (Continued)



9.4 DOE Code Comparison Project, Problem 5, Case A

This problem involves multiphase flow in a 2-D horizontal reservoir. The problem is characterized by a moving two-phase region, i.e., the fluid produced at the production well is replaced by cold water recharge over one of the outer boundaries. The problem parameters are given in Table VIII and the geometry and boundary conditions are shown in Fig. 10. Of particular note are the variable initial temperature field, provided to the code through a read file (see Section 5.6 on page 18), and the prescribed pressure and temperature on the right boundary. A partial listing of the input file is provided in Fig. 11. In addition to the required macros, macro **flow** is used to specify the pressure and temperature boundary condition and the production flow rate. Macro **rlp** is used to set the residual liquid and gas saturations.

Table VIII. Input Parameters for the DOE Code Comparison Project Problem		
Parameter	Symbol	Value
Reservoir permeability	k	$2.5 \times 10^{-14} \text{ m}^2$
Reservoir porosity	ϕ	0.35
Rock thermal conductivity	κ_r	$1 \frac{\text{W}}{\text{m} \cdot \text{K}}$
Rock density	ρ_r	2563 kg/m^3
Rock specific heat	C_r	$1010 \frac{\text{J}}{\text{kg} \cdot \text{K}}$
Reservoir length	x	300 m
Reservoir thickness	y	200 m
Liquid residual saturation	s_{lr}	0.3
Gas residual saturation	s_{gr}	0.1
Reservoir discharge	q_m	$0.05 \frac{\text{kg}}{\text{m} \cdot \text{s}}$
Initial Pressure	P_o	3.6 MPa
Production well coordinates:	$x = 62.5 \text{ m}, y = 62.5 \text{ m}$	
Observation well coordinates:	$x = 162.5 \text{ m}, y = 137.5 \text{ m}$	
Initial temperature distribution (T in $^{\circ}\text{C}$, r in m):		
$T(x, y, 0) = \begin{cases} 240 & r \leq 100 \\ 240 - 160\left(\frac{r-100}{200}\right)^2 + 80\left(\frac{r-100}{200}\right)^4 & 100 < r < 300 \\ 160 & r \geq 300 \end{cases}$		
where $r = \sqrt{x^2 + y^2}$		

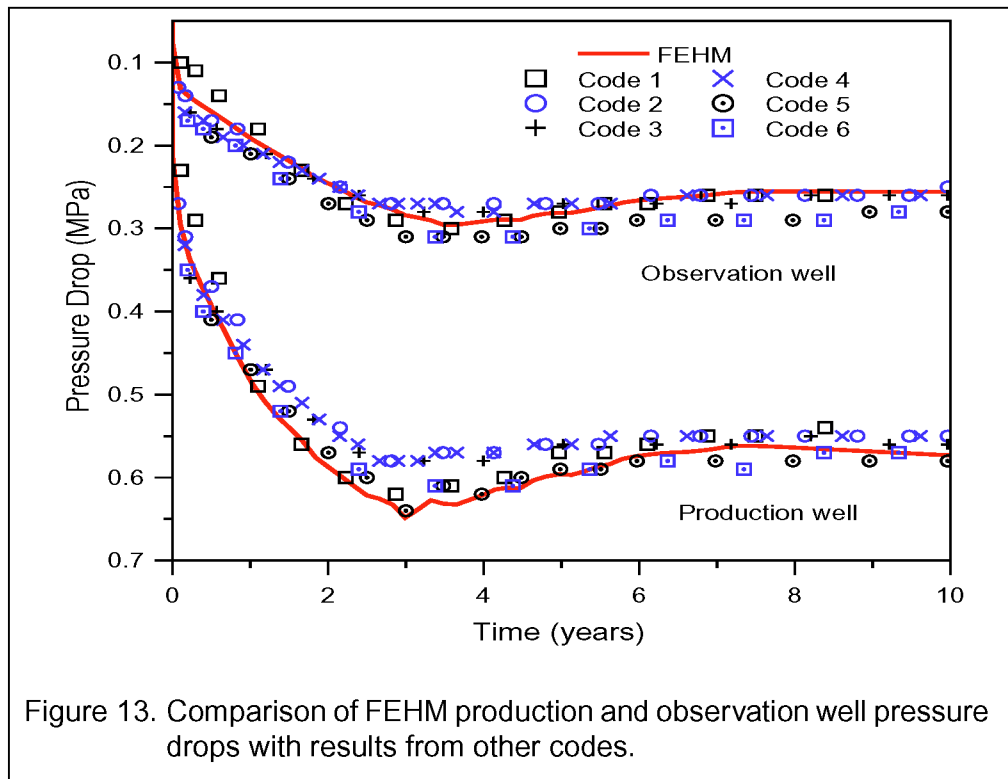
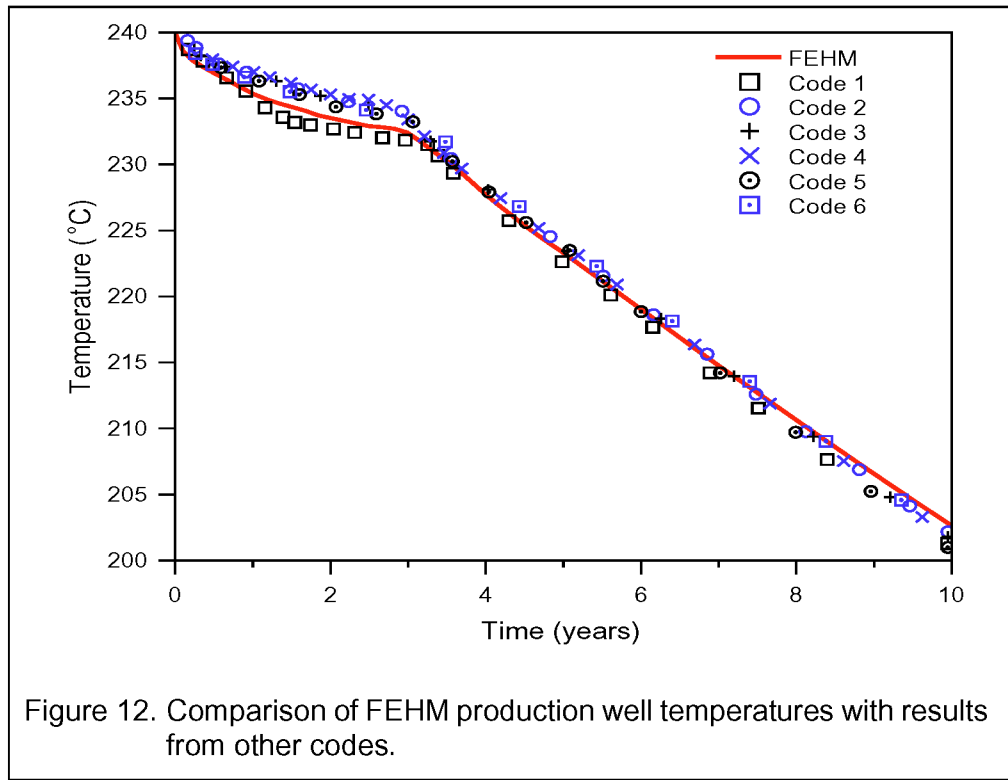


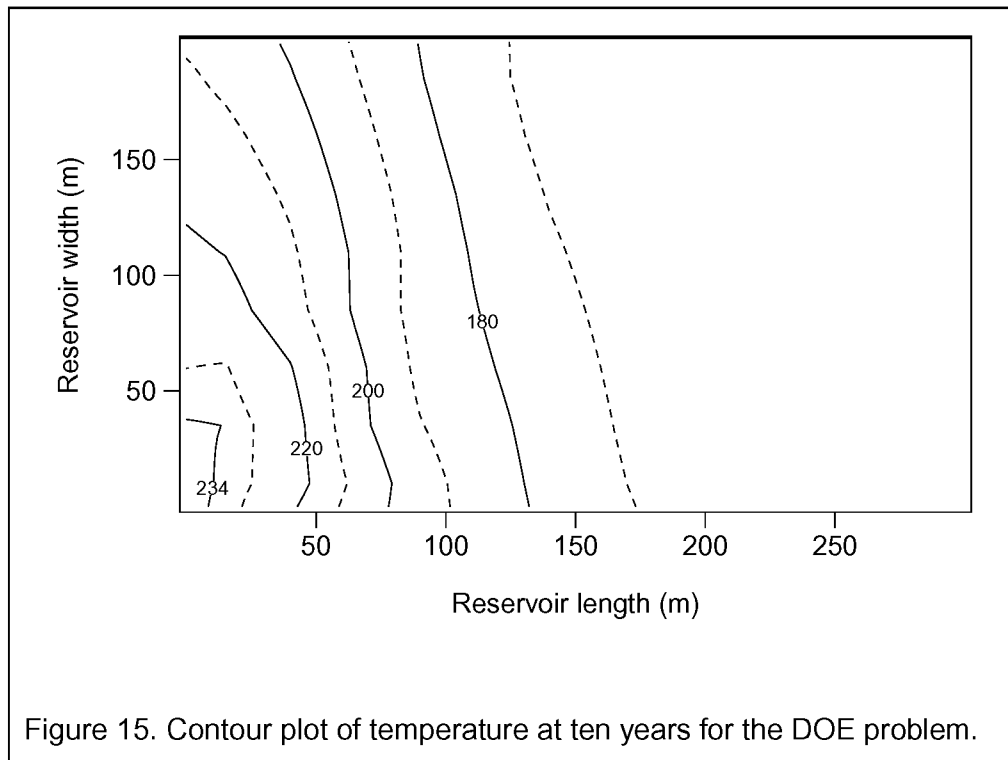
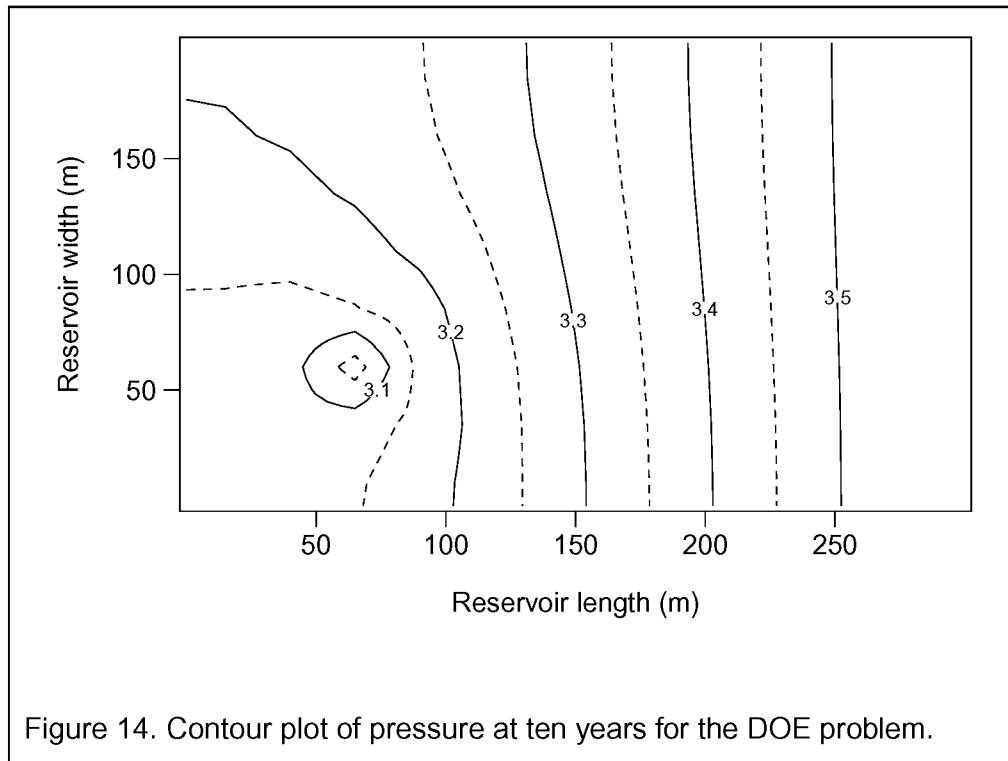
There is no analytical solution for this problem, but six researchers produced results for the DOE code comparison project (Molloy, 1980). The reader is referred to this reference for a more detailed discussion of this problem and the code comparison. Results from this problem are compared to those for the other codes, obtained from Molloy (1980), as a check on FEHM. The results for the outlet temperature, shown in Fig. 12, are in excellent agreement with the other codes. The results for the outlet pressure and pressure at an observation well 125 m distant, Fig. 13, are also in good agreement with the other codes. Contour plots of pressure and temperature at the end of the simulation were also generated for this problem and are shown in Fig. 14 and Fig. 15.

```

*** DOE Code Comparison Project, Problem 5, Case A ***
node
  2
  50  88
sol
  1  1
init
  3.6  0.  240.  0.  0.  240.  0.  0.
rlp
  2  0.3  0.1  0.0  0.0
  1  140  1  1
rock
  1  140  1  2563.  1010.  0.35
cond
  1  140  1  1.00e-00  1.00e-00  1.00e-00
perm
  1  140  1  2.5e-14  2.5e-14  0.e-00
flow
  88  88  1  0.050  -25.00  0.
  14  140  14  3.600  -160.00  1.
time
  30.0  3650.  10000  1000  1994  03
ctrl
  40  1.e-07  08
  1  140  1  1
  1.0  0.0  1.0
  40  1.2  0.1  60.
  1  0
coor
  140
  .
  .
  .
elem
  4  117
  .
  .
  .
stop
Figure 11. FEHM input file for DOE problem.

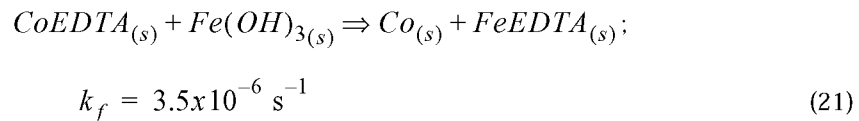
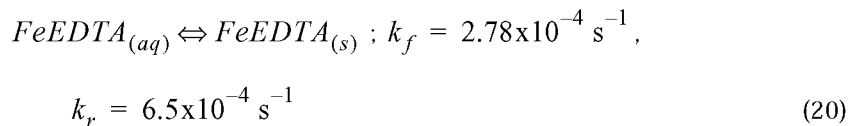
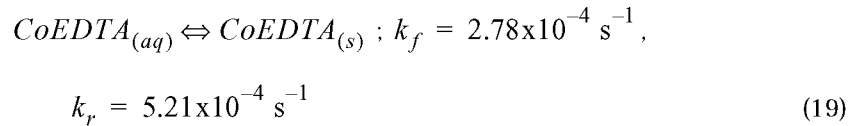
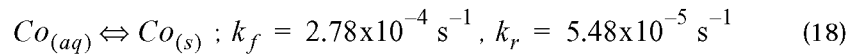
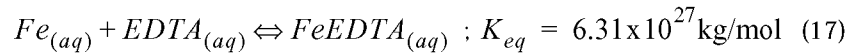
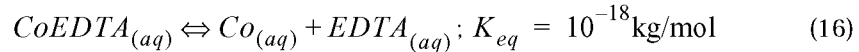
```





9.5 Reactive Transport Example

This one-dimensional example demonstrates the use of the reactive transport module of FEHM. The application of this simulation is the transport of cobalt (Co) in groundwater. Radioactive cobalt is present in the subsurface at several DOE sites. Although its presence as a divalent cation implies that it should sorb strongly to most soils, its migration rate has been shown to be greater than expected due to complexation with EDTA, a decontaminating agent also found in the subsurface of these sites. Much experimental work has gone into studying the transport of Co as CoEDTA, a much less strongly sorbed species. The chemical reactions and equilibrium or rate constants used to perform this simulation are:



Thus the transport system consists of 8 species and six reactions, with reactions specified as either equilibrium or kinetically controlled. $Fe(OH)_3$ is so prevalent in the sand that its concentration is assumed to be constant. In addition, it does not act as a true species in the simulation. A list of relevant input parameters and conditions for the simulation are given in Table IX. Figure 16 illustrates the transport problem. The flow system is represented by a one-dimensional flow path of 202 nodes (101 x 2). A partial listing of the input file is provided in Fig. 17.

FEHM results for this problem are compared to those of PDREACT (Valocchi et al., 1994), a two-dimensional, isothermal, saturated-zone flow and transport code in Figs. 18 and 19.

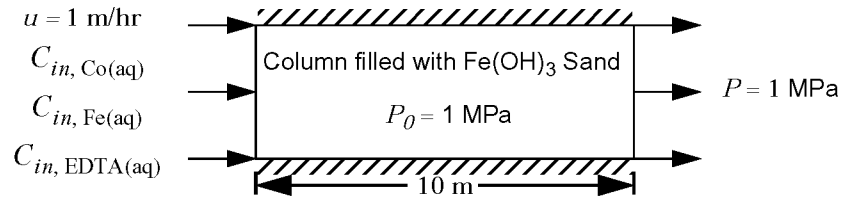


Figure 16. Schematic drawing of the geometry and boundary conditions for the cobalt transport problem.

Table IX. Input Parameters for the Reactive Transport Test Problem

Parameter	Symbol	Value
Reactor Length	L	10 m
Node spacing	Δl	0.1 m
Fluid Density	ρ_f	1000 kg/m ³
Bulk Rock Density	ρ_b	1500 kg/m ³
Porosity	ϕ	0.4
Pore Water Velocity	u	1 m/hr
Dispersivity	α	0.05 m
Time step (tracer)	Δt	0.09 - 360 s
Total elapsed time	t	7.25 days
Pressure	P_0	1.0 MPa
Co Inlet Concentration	$C_{in, Co}$	3.1623×10^{-5} M
Fe Inlet Concentration	$C_{in, Fe}$	0 M
EDTA Inlet Concentration	$C_{in, EDTA}$	3.1623×10^{-5} M
Boundary conditions:	At $l = 0$ At $l = 1$	$u = 1$ m/hr $P = 1$ MPa
‡Flow rate $q = u\rho_f\phi/2$ nodes = 0.05556 kg/s		

```

COMPARE FEHMN and PDREACT: Linear Sorption w/ Surface Exchange
cond
1 202 1 2.7 2.7 2.7

ctrl
50 1e-6 8
1 202 1 2

1 0 0.5
25 2. 1.e-6 1.e-1
1 0
flow
1 202 101 -0.05556 -25 0
101 202 101 1. -25 -1

init
1. 25 25 0 1000 25 0 0
node
1
202
perm
1 202 1 5.0e-13 5.0e-30 5.0e-30

rock
1 202 1 1500 1000 0.4

sol
1 -1
time
1.e-6 7.25 1000 10 92 11

# solute 1: Total Cobalt Concentration
# solute 2: Total Iron Concentration
# solute 3: Total EDTA Concentration
# solute 4: CoEDTA adsorbed concentration
# solute 5: Co adsorbed concentration
# solute 6: FeEDTA adsorbed concentration
trac
0.0 1.0 1.e-6 0.5
1. 2000 1.0 2000
5 5.0 1.e-6 4.1667e-3
6
1
1 0. 0. 1. 1.e-9 .05 1.e-34 1.e-34

1 202 1 1

1 202 1 0.

1 202 101 3.1623e-5 1.0 4.16667

```

Figure 17. FEHM input file for reactive transport problem .


```

1
1 0. 0. 1. 1.e-9 .05 1.e-34 1.e-34

1 202 1 1

1 202 1 0.

1 202 101 1.e-13 1.0 4.16667

1
1 0. 0. 1. 1.e-9 .05 1.e-34 1.e-34

1 202 1 1

1 202 1 0.

1 202 101 3.1623e-5 1.0 4.16667

0
1 202 1 0.

0
1 202 1 0.

0
1 202 1 0.0

rxn
** NCPLX, NUMRXN
2,4
** Coupling of the aqueous components (dRi/dUj)
2
1 0 1
0 1 0
** IDCPLX(IX), CPNTNAM(IX), IFXCONC(IX), CPNTPRT(IX) (comp,name,cond.; NCPNT
rows)
1 Cobalt[aq] 0 0 1.e-9
2 Iron[aq] 0 0 1.e-9
3 EDTA[aq] 0 0 1.e-9
** IDCPLX(IX), CPLXNAM(IX), CPLXPRT(IX) (ID # and name of complex, NCPLX rows)
101 Co-EDTA[aq] 0
102 Fe-EDTA[aq] 0
** IDIMM(IM), IMMNAM(IM), IMPRT(IM)(ID # and name of immobile spec, NIMM rows)
1 Co-EDTA[s] 0
2 Fe-EDTA[s] 0
3 Cobalt[s] 0
** IDVAP(IV), VAPNAM(IM), VAPPRT(IV) (ID # and name of vapor spec, NVAP rows)
** Skip nodes
0

```

Figure 17. FEHM input file for reactive transport problem (Continued).

```

** RSDMAX
1.0e-10
**** Chemical reaction information for equilibrium reactions ****
** LOGKEQ (=0 if stability constants are given as K, =1 if given as log(K))
    0
** CKEQ(IX) ,HEQ(IX) (Stability constants and Enthaplys, NCPLX rows)
1.0e+18  0
6.31e+27  0
** STOIC(IX,IC) (Stoichiometric coeff: NCPLX rows, NCPNT columns)
1.0    0.0    1.0
0.0    1.0    1.0
** LINEAR KINETIC REACTION (type 1) **
    1
** Where does the reaction take place? **
1 0 0

** Aqueous Component/Complex #, Solid Component #
    101  1
** Distribution coeffienct (kg water/ kg rock) **
    0.533
** Mass transfer coefficient (1/hr) **
    1.0
** LINEAR KINETIC REACTION (type 1) **
    1
** Where does the reaction take place? **
1 0 0

** Aqueous Component/Complex #, Solid Component #
    1  3
** Distribution coeffienct (kg rock/ kg water) **
    5.07
** Mass transfer coefficient (1/hr) **
    1.0
** LINEAR KINETIC REACTION (type 1) **
    1
** Where does the reaction take place? **
1 0 0

** Aqueous Component/Complex #, Solid Component #
    102  2
** Distribution coeffienct (kg rock/ kg water) **
    0.427
** Mass transfer coefficient (1/hr) **
    1.0
** GENERAL EXCHANGE REACTION (type 3) **
    3
** Where does the reaction take place? **
1 0 0

** # of solid, liquid and vapor species **
    3 0 0

```

Figure 17. FEHM input file for reactive transport problem (Continued).

```

** forward and reverse rate constants (1/hr) **
1.26e-2 0
** Solid Species in reaction **
1 2 3
** Stoichiometry **
1.0 -1.0 -1.0
coord n/a
202
1 0.00000 1.00000 0.00000
2 0.10000 1.00000 0.00000
3 0.20000 1.00000 0.00000
.
.
.
200 9.80000 0.00000 0.00000
201 9.90000 0.00000 0.00000
202 10.00000 0.00000 0.00000

elem
4 100
1 102 103 2 1
2 103 104 3 2
3 104 105 4 3
.
.
.
98 199 200 99 98
99 200 201 100 99
100 201 202 101 100

stop

```

Figure 17. FEHM input file for reactive transport problem (Continued).

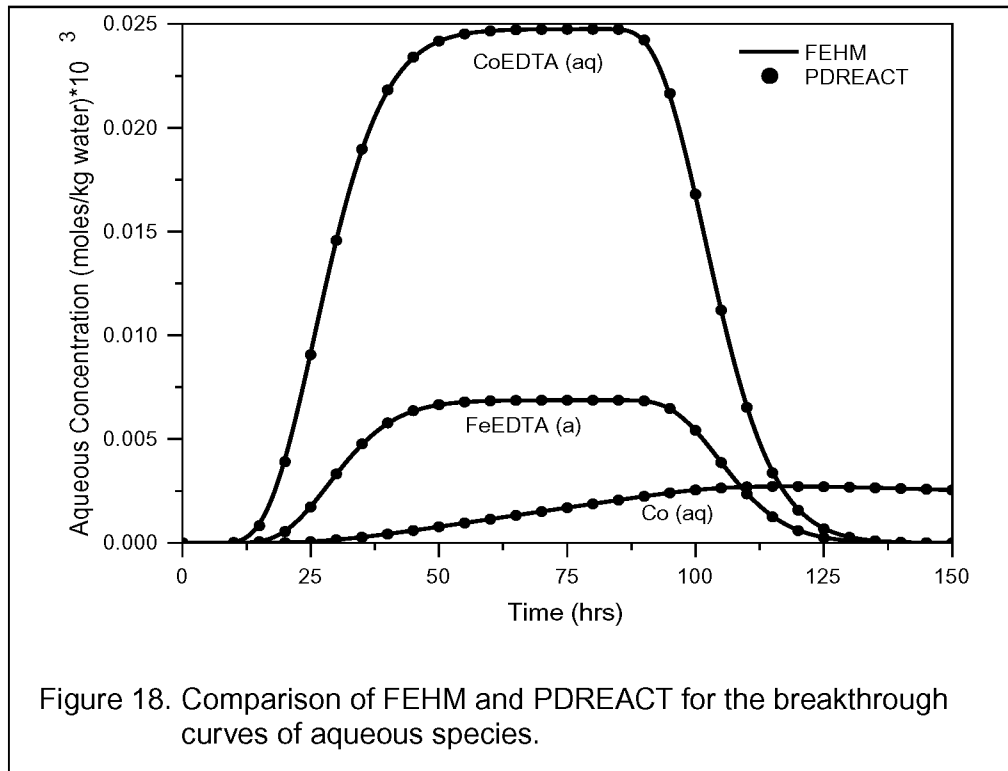


Figure 18. Comparison of FEHM and PDREACT for the breakthrough curves of aqueous species.

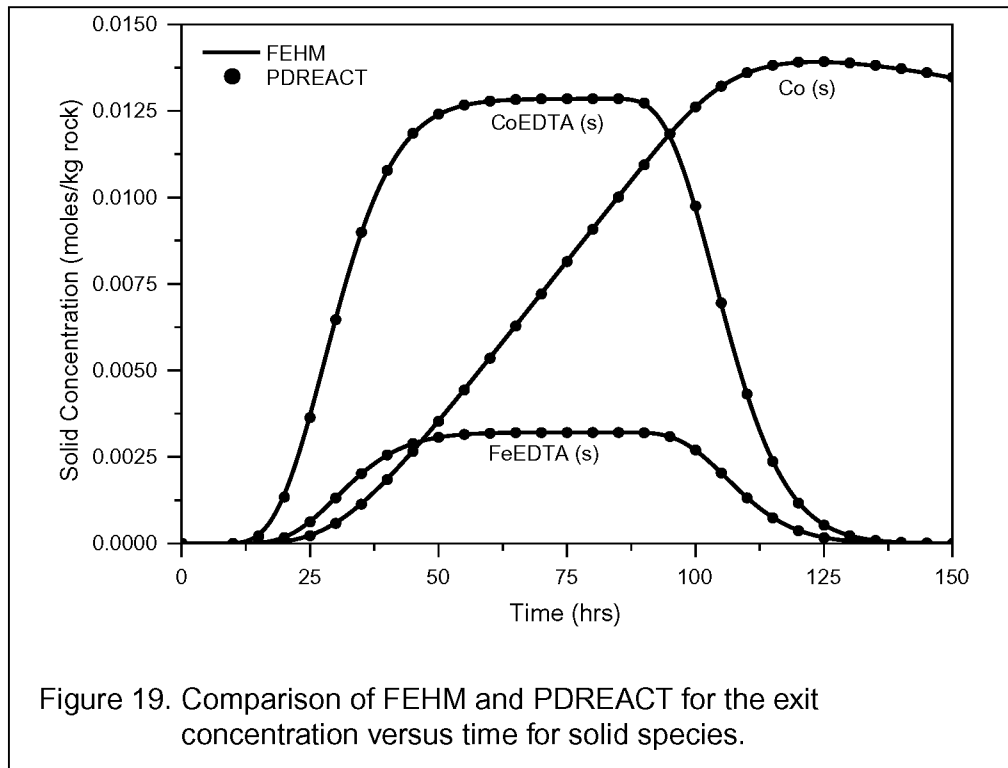


Figure 19. Comparison of FEHM and PDREACT for the exit concentration versus time for solid species.

10.0 USER SUPPORT

Licensing and installation support can be received from:

Zora Dash
zvd@lanl.gov
505-667-1923

Technical support can be received from:

George Zyvoloski
gaz@lanl.gov
505-667-1581

Bruce Robinson
robinson@lanl.gov
505-667-1910