LA-UR-12-24493
Approved for public release; distribution is unlimited.

Title: Software Users Manual (UM) for the FEHM Application Version 3.1 - 3.X

Author(s): Zyvoloski, George A.
Robinson, Bruce A.
Dash, Zora V.
Kelkar, Sharad
Viswanathan, Hari S.
Pawar, Rajesh J.
Stauffer, Philip H.
Miller, Terry A.
Chu, Shaoping

Intended for: User Manual for FEHM Software

Issued: 2015-12-14 (rev.2)

LA-UR-12-24493
Approved December 2015 for Public Release and Distribution

Version 3.3.0 December 03, 2015

User Manual for the FEHM Application Version 3.3.0

Contents:

Approval Signatures:

Software User Manual (UM) for the FEHM Application Version 3.1.0:

Author(s): Zyvoloski, George A.
Robinson, Bruce A.
Dash, Zora V.
Kelkar, Sharad
Viswanathan, Hari S.
Pawar, Rajesh J.
Stauffer, Philip H.

Appendix for V3.2.0 release: Prepared by Terry Miller and Shaoping Chu

Appendix for V3.3.0 release: Prepared by Shaoping Chu, Terry Miller and Zora Dash

SOFTWARE USERS MANUAL (UM)
for the
FEHM Application Version 3.1.0

Los Alamos National Laboratory

CHANGE HISTORY

Revision Number	Effective Date	Description of and Reason for Revision

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 FEHM V3.1.0 Users Manual 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.

Table II. Initial (Default) Values					
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	upwgta	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 35). 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 6.2.90 on page 185).

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 *fileid* ****`
	`.`
	`.`
	`.`
	`****----------****`
	`****  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 nonexistant file *filename*` `STOPPED trying to use optional input file`
Unable to open optional input file	`ERROR opening *filename*` `STOPPED trying to use optional input file`
Unable to determine file prefix for AVS output files	`FILE ERROR: nmfil2 file: *filename* unable to determine contour file prefix`

Unable to determine file prefix for pest output files	`FILE ERROR: nmfil15 file: *filename* unable to determine pest file name`
	`-or-`
	`FILE ERROR: nmfil16 file: *filename* unable to determine pest1 file name`

Unable to determine file prefix for streamline particle tracking output files	`FILE ERROR: nmfil17 file: *filename* unable to determine sptr1 file name`
	`-or-`
	`FILE ERROR: nmfil18 file: *filename* unable to determine sptr2 file name`
	`-or-`
	`FILE ERROR: nmfil19 file: *filename* unable to determine sptr3 file name`

Unable to determine file prefix for submodel output file	`FILE ERROR: nmfil24 file: *filename* 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 izone = izone` `please check icnl in macro CTRL`

Invalid AVS keyword read for macro **cont**	`ERROR:READ_AVS_IO`
	`unexpected character string (terminate program execution)`
	`Valid options are shown:`
	`.`
	`.`
	`.`
	`The invalid string was: string`

Invalid keyword or input order read for macro **boun**	`time change was not first keyword,stop`
	`-or-`
	`illegal keyword in macro boun, stopping`

Invalid keyword read for macro **subm**	`>>>> error in keyword for macro subm <<<<`
Invalid macro read	`**** error in input deck : char ****`

Invalid parameter values (macros using loop construct)	`Fatal error - for array number arraynum`
	`macro - macro`
	`Group number - groupnum`
	`Something other than a real or integer has been specified`
	`-or-`
	`Line number - line`
	`Bad input, check this line`
	`-or-`
	`Fatal error, too many`
	`real inputs to initdata2`
	`-or-`
	`Fatal error, too many`
	`integer inputs to initdata2`

Invalid streamline particle tracking parameter	`ist 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 (icns = -2), yet the Henrys Constant is computed as 0 for species number speciesnum and node number nodenum. If you want to simulate a vapor-borne species with no interphase transport, then you must specify a gaseous species (icns = -1).`
Invalid flag specified for diffusion coefficient calculation	`ERROR -- Illegal Flag to concadiff` `Code Aborted in concadiff`
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 *macroname* is *wrong_macroname* STOPPED trying to use optional input file
Option not supported	This option (welbor) 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 rd1dof. Stop in gencon
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 dpdp : stopping ************************************
Invalid parameters set	
Dual porosity	**** check fracture volumes,stopping**** **** check equivalent continuum VGs ****
Finite difference model (FDM)	>>>> dimension (icnl) not set to 3 for FDM: stopping <<<<
Maximum number of nodes allowed is less than number of equations	**** n0(*n0*) .lt. neq(*neq*) **** check parameter statements ***
Node number not in problem domain (macros dvel, flxo, node, nod2, nod3, zone, zonn)	**** Invalid input: macro *macro* ****' **** Invalid node specified, value is greater than n0 (*n0*): stopping ****
Noncondensible gas	cannot input ngas temp in single phase -or- ngas pressure lt 0 at temp and total press given max allowable temperature *temp* -or- ngas pressure gt total pressure i= *i* -or- ngas pressure lt 0.
Particle tracking	ERROR: Pcnsk in ptrk must be either always positive or always negative. Code aborted in set_ptrk.f
Relative permeabilities	cannot have anisotropic perms for rlp model 4 or rlp 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 *SRNAME* parameter number *I2* had an illegal value
Insufficient storage	
Boundary conditions	exceeded storage for number of models
Dual porosity	***** n > n0, stopping ****
Generalized dual porosity	In gdpm macro, ngdpmnodes must be reduced to reduce storage requirements A value of *ngdpm_actual* is required The current value set is *ngdpmnodes* -or- Fatal error in gdpm macro A value of *ngdpm_actual* is required' The current value set is *ngdpmnodes* Increase ngdpmnodes to *ngdpm_actual* 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 dpdp 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 *i1*
Error computing geometric coefficients	iteration in zone did not converge, izone = *zone_number* please check icnl 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, izone = *zone* please check icnl in macro CTRL
Unable to normalize matrix	cannot normalize
Singular matrix in LU decomposition	singular matrix in ludcmp

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` *`timestep_number current_timestep_size current_simulation_time`*
	`-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`*

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 Optional history plot files (ishis*)

5.22.1 Content

The optional history plot files contain 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, filen.trc, filen_param[.his, _his.dat, _his.csv, .trc, .dat, .csv]*. "*param*" will depend on the output parameters selected. The extension will depend on output format selected: tecplot (*.dat*), comma separated variables (*.csv*) or default (*.his, .trc*).

5.22.2 Use by Program

The FEHM application uses the optional history plot files for storing history data for selected parameters which include: pressure, head, temperature, water content, flow, saturation, humidity, enthalpy, density, viscosity, flux, mass, displacement, stress / strain, and concentration (node based) and global output. The basic history file, *filen.his,* will contain run information including which parameters were selected and

the output node and zone data. The basic history file, *filen.trc,* will contain output node data and numbers of solute species by type when concentrations are output. The parameter files are accessed throughout the program as the simulation steps through time.

5.22.3 Auxiliary Processing

These files may be used to produce history plots by external graphics programs.

5.23 Optional contour output files (Advanced Visual Systems [AVS], TECPLOT, and SURFER)

5.23.1 Content

The contour output files contain output data. for the entire grid or selected zones.The content will depend on output format (avs or avsx [*.avs*], tecplot [*.dat*], or surfer [comma separated variables, *.csv*]) and parameters selected (material, pressure or head, saturation, temperature, flux, permeability, saturation, porosity, velocity, displacement, stress / strain, and concentration). The geometry based data can be imported into Advanced Visual Systems (AVS) UCD (unstructured cell data), TECPLOT, or SURFER graphics routines.

The contour 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 root file name or contour output file prefix if defined, otherwise it is the input file prefix, *Number* is a value between 00001 and 99999, and *AVS_id* is a string denoting file content (see Table IV and Figure 1). In general, _head are header files (only used by AVS), _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. Contour File Content Tag

AVS_id	File purpose
_avs_log	Log file from contour output routines
_geo	Geometry output file containing coordinates and cell information (AVS UCD geometry file format)
_grid.dat	Geometry output file containing coordinates and element connectivity (Tecplot grid file format)
_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.

Table IV. Contour File Content Tag (Continued)	
AVS_id	File purpose
_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.

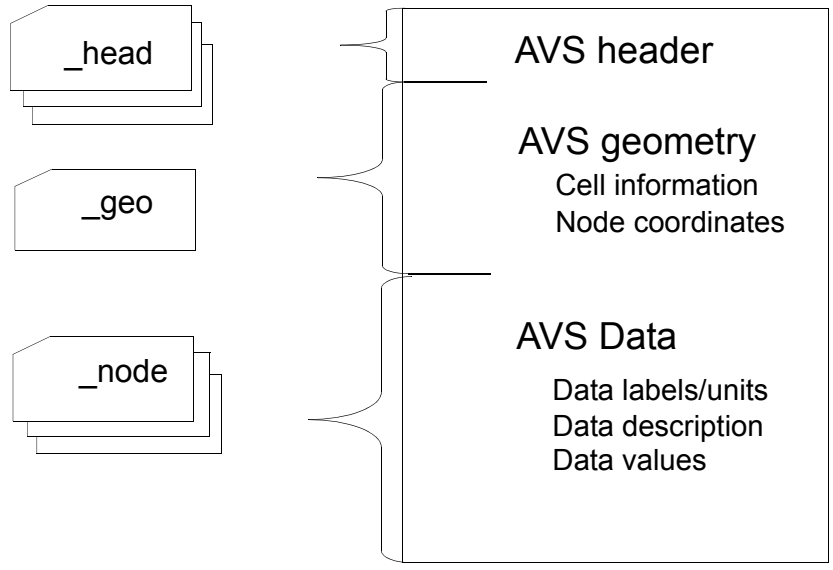

Figure 1. AVS UCD formatted FEHM output files.

5.23.2 Use by Program

The FEHM application uses the contour output files for storing geometry based data for material properties (permeabilities and porosities), temperature, saturation, pressure, velocities, and solute concentrations in a format readable by AVS, TECPLOT or SURFER 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 a specified format.

5.23.3 Auxiliary Processing

These files are used for visualization and analysis of data by AVS, TECPLOT or SURFER.

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 **fehm2avs** 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, it is possible to supply the name of the control file on the command line, otherwise input prompts are written to the screen preceded by a short description of the I/O files used by FEHM. It should be noted that a control file name entered on the command line will take precedence over the default control file. 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. It is also 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.msim* 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 or air	Isothermal air-water input
anpe	Anisotropic permeability
boun	Boundary conditions **(required for flow problem if macro flow is not used)**
bous	Boussinesq-type approximation
carb	CO_2 input
cden	Concentration-dependent density
cflx	Molar flow rate through a zone
cgdp	Rate-limited gdpm node
chea	Output in terms of head, not pressures (non-head problem)
cond	Thermal conductivity data **(required for non-isothermal problem)**
conn	Print number of connections for each node and stop
cont	Contour plot data
conv	Head input conversion for thermal problems
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	Simple equation of state data
exrl	Explicit evaluation of relative permeability
fdm	Finite difference grid generation **(required if macro coor and elem are not used)**
finv	Finite volume flow coefficients

Table V. Macro Control Statements for FEHM (Continued)	
Control Statement	Description
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)
floa	Alternate format for flow data (additive to previous flow definition)
flwt	Movable source or sink (wtsi only)
flxn	Write all non-zero source/sink internodal mass flows by node to an output file.
flxo	Internodal mass flow printout
flxz	Zone based mass flow output
fper	Permeability scaling factor
frlp	Relative permeability factors for residual air effect
ftsc	Flux correction for saturations over 1
gdkm	Generalized dual permeability model
gdpm	Generalized dual porosity model
grad	Gradient model input
hcon	Set solution to heat conduction only
head	Hydraulic head input
hflx	Heat flow input
hist	User selected history output
hyco	Hydraulic conductivity input **(required if macro perm is not used)**
ice or meth	Ice phase calculations (untested) methane hydrate input
impf	Time step control based on maximum allowed variable change
init	Initial value data **(required if macro pres or restart file is not used)**
intg	Set integration type
isot	Isotropic definition of control volume/finite element coefficients
iter	Iteration parameters
itfc	Flow and transport between zone interfaces
ittm	Sticking time for phase changes
itup	Iterations used with upwinding
iupk	Upwind transmissibility including intrinsic permeability

Table V. Macro Control Statements for FEHM (Continued)

Control Statement	Description
ivfc	Enable exponential fracture and volume model
mdnode	Enables extra connections to be made to nodes
mptr	Multiple species particle tracking simulation input
nfinv	Finite element instead of finite volume calculations
ngas	Noncondensible 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
nrst	Stop NR iterations on variable changes
para	Parallel FEHM (isothermal only)
perm	Permeability input **(required if macro hyco is not used)**
pest	Parameter estimation routine output
phys	Non-darcy well flow
ppor	Pressure and temperature dependent porosity and permeability
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
rest	Manage restart options
rflo	Read in flux values
rflx	Radiation source term
rich	Enable Richards' equation
rive or well	River or implicit well package
rlp	Relative permeability input **(required for 2-phase problem if macro rlpm is not used, otherwise optional)**
rlpm	Alternate style relative permeability input **(required for 2-phase problem if macro rlp is not used, otherwise optional)**
rock	Rock density, specific heat, and porosity input **(required)**
rxn	Chemical reaction rate model input

Table V. Macro Control Statements for FEHM (Continued)	
Control Statement	Description
sol	Solver specifications
sptr	Streamline particle tracking simulation input
stea	Steady state program termination
stop	Signals the end of input **(required)**
strs	enable stress solution
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
weli	Peaceman type well impedance
wgtu	Areas, weights (user-defined) for boundary conditions
wflo	Alternate submodel boundary output
wtsi	Water table, simplified
zeol	Zeolite water balance input
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.

A macro may be disabled (turned off or omitted from a run) by adding keyword "off" on the macro line and terminating the macro with an end statement of the form "**end*macro***" or "**end *macro***" (see Section 6.2.5).

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 202) 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 ZONE_DPADD + I is the default zone number for the second set of

nodes defined by Zone I, and Zone 2*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 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 207) 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 28). 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 file name parameters enumerated below [nmfil(2-13)], are entered in order one per line in the control file (excluding the control file name [nmfil(1)] and error file name [nmfil(14)]) 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 on the command line or 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.

In version 2.30 an alternate format for the control file has been introduced that uses keywords to identify which input and output files will be used. Please note that the file name input styles may not be mixed.

Group 1 NMFIL(*i*) (a file name for each *i* = 2 to 13)

-or-

Group 1 KEYWORD**:** NMFIL

Group 2 TTY_FLAG

Group 3 USUB_NUM

Unlike previous versions of the code, if a file name is not entered for the output file, check file, or restart file, the file will not be generated. An error output file will still be generated for all runs (default name *fehmn.err*). However, with the keyword input style the user has the option of naming the error file. File names that do not include a directory or subdirectory name, will be located in the current working directory. With keyword input a root filename may be entered for output files that use file name generation (**hist** macro output, **cont** macro avs, surfer or tecplot output, etc.). The data files are described in more detail in Section 5.0.

The following are examples of the input control file. The first example (left) uses keyword style input, while the second and third examples (right) use the original style control file input form. In the first example, four files are explicitly named, the input file, geometry file, tracer history output file and output error file. A root file name is also provided for file name generation. The "all" keyword indicates that all information should be written to the terminal and the ending "0" indicates that the user subroutine will not be called. In the second example in the center, 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 or restart output file, 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 third 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 and the ending "0" indicates that the user subroutine will not be called.

Input Variable	Format	Default	Description
keyword	character*5	No keywords, old style file format is used	Keyword specifying input or output file type. The keyword is entered followed immediately by a "**:**" with a "space" preceding the filename. Keywords, which must be entered starting in the first column, are: input - Main input file grid - Geometry data file; or grida or gridf - Ascii formatted geometry file; or gridu or gridb - Unformatted geometry file zone - Initial zone file outp - Output file rsti - Restart input file rsto - Restart output file hist - Simulation history output trac - Solute history output cont - Contour output dual, dpdp - Dual porosity, double porosity output stor - Coefficient storage file check -- Input check output file nopf - Symbolic factorization file colu - Column data file for free surface problems error - Error output file root - Root name for output file name generation co2i - CO_2 parameter data file (default co2_interp_table.txt) look - Equation of state data lookup table file (default lookup.in) Keyword file name input is terminated with a blank line. The keywords and file names may be entered in any order unlike the old style input.
nmfil	character*100		Input or output file name
		fehmn.files	nmfil(1) - Control file name (this file is not included in the old style control file) (optional)
		fehmn.dat	nmfil(2) - Main input file name (required)
		not used	nmfil(3) - Geometry data input file name (optional)
		not used	nmfil(4) - Zone data input file name (optional)
		not used	nmfil(5) - Main output file name (optional)
		not used	nmfil(6) - Restart input file name (optional)
		not used	nmfil(7) - Restart output file name (optional)
		not used	nmfil(8) - Simulation history output file name (optional)
		not used	nmfil(9) - Solute history output file name (optional)
		not used	nmfil(10) - Contour plot output file name (optional)
		not used	nmfil(11) - Dual porosity or double porosity / double permeability contour plot output file name (optional)
		not used	nmfil(12) - Coefficient storage file name (optional)
		not used	nmfil(13) - Input check output file name (optional)
		fehmn.err	nmfil(14) - Error output file name (this file is not included in the old style control file). The default name is used if not input.

Input Variable	Format	Default	Description
tty_flag	character*4	none	Terminal output flag: all, some, none
usub_num	integer	0	User subroutine call number

Files "*fehmn.files*":

input: /groupdir/c14-3
trac: c14-3.trc
grid: /groupdir/grid-402
root: c14-3
error: c14-3.err
all
0

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

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 or Windows bat file instructions: lines 2-n:lines which are written to a file called *fehmn.pre* (UNIX) *or* *fehmn.pre.bat* (Windows*)*, which is invoked before each realization using the following command: sh *fehmn.pre* $1 $2 (UNIX systems) or *fehmn.pre.bat* $1 $2 (Windows) 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 *fehmn.pre* script, and that the data for the post-processing script *fehmn.post* follows lines n+2 to N: lines which are written to a file called *fehmn.post* (UNIX) or *fehmn.post.bat* (Windows), which is invoked after each realization using the following command: sh *fehmn.post* $1 $2 (UNIX) or *fehmn.post.bat* $1 $2 (Windows) Thus, the files fehmn.pre and fehmn.post 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.

In the following (UNIX style) example, 100 simulations are performed with pre and post-processing steps carried out. File "*fehmn.msim"* contains the following:

```
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 $1`
/home/robinson/fehm/chun_li/ptrk/process1_fuj
rm np$curnum.output
mv results.output np$curnum.output
echo finishing the run again
```

The first line after the number of simulations demonstrates how the current and total number of simulations can be 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.

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.
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(value).

Input Variable	Format	Description
NORMAL_PARAM	real	Array of values for normalizing the parameters before performing the nearest neighbor or multi-curve interpolation. Each value is divided by NORMAL_PARAM(i) if LOG_FLAG is 0, and ln(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 curve_strucuture points nearest to the input values of the parameters. It is recommended that values no greater than 4 be used.
WEIGHT_FACTOR	real	Optional wieght factor used when 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 weight_factor. 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	*nump1* parameter 1 values defining transfer function curves.
NUMP2	integer	Number of parameter 2 values used to define transfer function curves.
PARAM2	real	*nump2* parameter 2 values defining transfer function curves.
NUMP3	integer	Number of parameter 3 values used to define transfer function curves.
PARAM3	real	*nump3* parameter 3 values defining transfer function curves.
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.

Input Variable	Format	Description
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 discusion 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 = 1to 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 auxilliary file. The entries in the optional input file may be preceded or followed by comments using the "#" designator (see discussion on page 32). 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 6.2.79 on page 141.

```
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 Option to disable a macro

The data from any input macro may be omitted from a simulation by including the "**off"** keyword on the macro line and terminating the macro with an end macro statement. This also allows the inclusion of an end macro statement for any input macro. The end macro will follow the last specified line of input. If a macro is normally terminated with an end keyword, the macro id is appended to that keyword (an additional end macro line is not added). This facilitates experimentation with input options without the need to comment out or remove unwanted macros from the input file.

In the following example the **perm** macro is turned off and the **hyco** macro is used in its place.

```
perm off
   1          0          0          1.e-12     1.e-12     1.e-12

end perm
hyco
   1          0          0          3.8e-3     3.8e-3     3.8e-3

end hyco
```

6.2.6 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. if TORT > 1, water-vapor diffusion coefficient is set equal to the value specified for the first vapor species defined in the **trac** macro.

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
```

6.2.7 Control statement airwater or air (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.
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
```

6.2.8 Control statement anpe

Anisotropic permeability input. Adds cross terms to the perm macro. Do not use with macros DPDP, GDKM or multiple porosity models.

Group 1 - JA, JB, JC, ANXY, ANXZ, ANYZ (JA, JB, JC - defined on page 33)

Input Variable	Format	Default	Description
ANXY	real	1.e-30	Anisotropic permeability in the xy-direction (m^2).
ANXZ	real	1.e-30	Anisotropic permeability in the xz-direction (m^2).
ANYZ	real	1.e-30	Anisotropic permeability in the yz-direction (m^2).

6.2.9 **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’, ‘ti_linear’, ‘cy’ or ‘cy_linear’.

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. Note that some keywords do not have associated variables.

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 33)

Input Variable	Format	Description
KEYWORD	character*4	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: model - new model definition to follow 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. ti - time sequence for changes to follow (days). The ‘ti’ keyword results in step function changes in boundary condition VARIABLE(i) at each TIME(i). ti_linear- time sequence for changes to follow (days). The ‘ti_linear’ keyword will apply a boundary condition that changes linearly with time. This option does not impose any control on time step size, so it is possible that a single time step can span an entire time interval and the linear change will not be seen. If time step size control is important it should be imposed in the **time** or **ctrl** macros. cy - cyclic time sequence for changes to follow (days). As with the ‘ti’ keyword, boundary condition changes are step functions. cy_linear-cyclic time sequence for changes to follow (days). As with the ti_linear keyword, boundary condition changes linearly with time. sec - Time sequence input is in seconds. min - Time sequence input is in minutes.

Input Variable	Format	Description
		day - Time sequence input is in days. (Default) year - Time sequence input is in years. Note: The keywords, 'ti', 'ti_linear', 'cy' and 'cy_linear', 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.The default units for time input is days. Input in seconds, minutes or years is converted to days. Time input units have no associated variable input. tran - Keyword to indicate boundary conditions will not be invoked until the steady state portion of the simulation is completed and a transient run is initiated. See macro **stea** for more details. sa - air source sequence for changes to follow (kg/s) sw - water source sequence for changes to follow (kg/s) swf - source water factor sequence for changes to follow (multiplier for existing mass flow rates) se - enthalpy source sequence for changes to follow (MW) sf - water seepage face sequence with pressures for changes to follow (MPa) sfh - water seepage face sequence with heads for changes to follow (m) fd - water drainage area sequence for changes to follow (m^2) dsa - distributed air source sequence for changes to follow (kg/s) dsw - distributed water source sequence for changes to follow (kg/s) dse - distributed enthalpy source sequence for changes to follow (MW) 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. wgt - Distributed source is weighted using the nodal control volume. wgtx - Distributed source is weighted using nodal area = control volume / x length scale. wgty - Distributed source is weighted using nodal area = control volume / y length scale. wgtz - Distributed source is weighted using nodal area = control volume / z length scale. wgtp - Distributed source is weighted using nodal control volume * permeability. wgtpx - Distributed source is weighted using nodal control volume * permeability / x length scale. wgtpy - Distributed source is weighted using nodal control volume * permeability / y length scale. wgtpz - Distributed source is weighted using nodal volume * permeability / z length scale.

Input Variable	Format	Description
		Note: The length scale term is the dimension of the control volume bounding box, xmax-xmin, ymax-ymin, zmax-zmin, depending upon the suffix, x,y,z. This option is useful when one wants to apply a distributed source on a mesh with variable size mesh cells and would like the source percentage to be allocated based on surface area of each node.
		wgtr - Distributed source is weighted using nodal volume * permeability * relative permeability
		wgtu - Distributed source is weighted with user specified values (See macro **wgtu**).
		wgww - Distributed source is weighted using nodal volume * permeability * relative permeability * exponentially weigthed distance from pump
		Note: The distributed source weighting options have no associated variable input.
		s - fixed saturation sequence for changes to follow
		hd - fixed hydraulic head sequence for changes to follow (m)
		pw - fixed water pressure sequence for changes to follow (MPa)
		pa - fixed air pressure sequence for changes to follow (MPa)
		hdo - fixed hydraulic head sequence for changes to follow (m) (constrained to outflow only)
		pwo - fixed water pressure sequence for changes to follow (MPa) (constrained to outflow only)
		pao - fixed air pressure sequence for changes to follow (MPa) (constrained to outflow only)
		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 source inflow occurs where this condition is applied, it will be ignored.
		kx - fixed X permeability sequence for changes to follow (m^2)
		ky - fixed Y permeability sequence for changes to follow (m^2)
		kz - fixed Z permeability sequence for changes to follow (m^2)
		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)

Input Variable	Format	Description
		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')

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 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
```

In the second example, a distributed water source is used to model a zone where production is turned on and off. Keyword 'kz' is used to specify a higher permeability when production is occurring. The 'ts' keyword is used to reset the time step to 1.0

days, at the beginning of each time interval. The model is applied to zone 100 in the last line.

```
boun
model 1
ti
     4              0.0            91.325          182.625         273.9375
ts
     1.0            1.0             1.0             1.0
dsw
   29.248           0.0            29.248           0.
kz
  8e-12           2e-12           8e-12           2e-12
end
  -100             0               0               1
```

6.2.10 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
```

6.2.11 Control statement **carb** (optional)

Macro **carb** is used to set up a CO_2 problem. Input following the problem type is grouped using sub keywords.

Group 1- IPRTYPE

Group 1 - KEYWORD

KEYWORD *"co2pres"*

JA, JB, JC, PHICO2, TCO2, ICES

KEYWORD *"co2flow"*

JA, JB, JC, SKTMP, ESKTMP, AIPED, IFLG_FLOWMAC

KEYWORD *"co2diff"*

JA, JB, JC, DIFF, TORTCO2

KEYWORD *"co2frac"*

JA, JB, JC, FW, FL, YC, CSALT, INICO2FLG

KEYWORD *"userprop"*

DENC, DENCP, DENCT, ENC, ENCP, ENCT, VISC, VISCP, VISCT

DENW, DENWP, DENWT, ENW, ENWP, ENWT, VISW, VISWP, VISWT

KEYWORD *"brine"*

Input is terminated with KEYWORD "*end carb*" or "*endcarb*"

Input Variable	Format	Default	Description
IPRTYPE	integer		IPRTYPE = 1, Water only problem (2 DOFs) IPRTYPE = 2, CO_2 only problem (2 DOFs) IPRTYPE = 3, CO_2-water problem, no solubility (3 DOFs) IPRTYPE = 4, CO_2-water problem, w/ soubility (4 DOFs) IPRTYPE = 5, CO_2-water-air w/ solubility (5 DOFs)
KEYWORD	character		Remaining input is grouped using sub-macro keywords.
KEYWORD *"end carb"* or *"endcarb"*			End of **carb** input.
KEYWORD *"co2pres"*			Set up the initial pressure (uses the same format as the **pres** macro)
PHICO2	real	0.	Initial CO_2 pressure (MPa).
TCO2	real	0.	Initial CO_2 temperature (°C)
ICES	integer	0	Initial guess for phase state of CO_2 (actual phase will be calculated internally): ICES = 1 for liquid ICES = 2 for two-phase liquid and vapor ICES = 3 for vapor ICES = 4 for super-critical CO_2.
KEYWORD *"co2flow"*			Set up co2 flow boundary conditions (similar to the **flow** macro used to set up water boundary conditions)
SKTMP	real	0.	If IFLG_FLOWMAC = 1, 2, 3, 4, 5 or 9 CO_2 flowing pressure (MPa). If SKTMP = 0 the initial value of pressure will be used for the flowing pressure. If IFLG_FLOWMAC = 6 or 7 CO_2 mass flow rate (kg/s). If IFLG_FLOWMAC = 8 CO_2 flowing saturation.
ESKTMP	real	0.	If IFLG_FLOWMAC = 1, 2, 3, 6, 7 or 9 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 (°C) and the enthalpy calculated accordingly. If IFLG_FLOWMAC = 4 or 5 CO_2 flowing saturation. if IFLG_FLOWMAC = 8 mass fraction of CO_2

AIPED	real	0.	If IFLG_FLOWMAC = 1, 2, 4 or 9 CO_2 impedance parameter. If IFLG_FLOWMAC = 5 or 6 value is ignored. If IFLG_FLOWMAC = 7 CO_2 flowing saturation. If IFLG_FLOWMAC = 8 Water mass flow rate (kg/s).
IFLG_FLOWMAC	integer	0	Flag specifying boundary condition type: IFLG_FLOWMAC =1 Constant pressure boundary condition with inflow or outflow allowed. AIPED is user specified IFLG_FLOWMAC =2 Constant pressure boundary condition with only outflow allowed. AIPED is user specified IFLG_FLOWMAC =3 Constant pressure boundary condition. AIPED is calculated in the code based on block geometric parameters. IFLG_FLOWMAC =4 Constant pressure and constant saturation boundary condition. AIPED is user specified IFLG_FLOWMAC =5 Constant pressure and constant saturation boundary condition. AIPED is calculated in the code based on block geometric parameters. IFLG_FLOWMAC =6 Constant free phase CO_2 mass flow rate boundary condition. IFLG_FLOWMAC =8 Constant source of water with specified mass fraction of CO_2 (kg/s) IFLG_FLOWMAC =9 Partial explicit update of nonlinear part of CO_2 constant pressure
KEYWORD *"co2diff"*			Read CO_2 diffusivity in water
DIFF	real	0.	Diffusion
TORTCO2	real	0.	Tortuosity for CO_2-water vapor diffusion.
KEYWORD *"co2frac"*			Read initial CO_2, air, water/brine saturation. FG, CO2/air-rich gas saturation (volume fraction), FG = 1 - FW - FL.
FW	real	0.	Water-rich liquid saturation (volume fraction).
FL	real	0.	CO_2-rich super-critical/liquid phase saturation (volume fraction).
YC	real	0.	Mass fraction of CO_2 in the CO_2-rich phase.
CSALT	real	0.	Initial salt concentration in water for brine (ppm) (only used if "brine" keyword is invoked.
INICO2FLG	integer	0	Flag to override CO_2 fractions read from restart file. If set to 1 the input values are used instead of those read from the restart file.
KEYWORD *"userprop"*			Read user defined properties for CO_2 and brine
DENC	real		CO_2 density (kg/m^3)
DENCP	real		Derivative of density with respect to pressure.
DENCT	real		Derivative of density with respect to temperature.
ENC	real		CO_2 enthalpy (MJ/kg).

ENCP	real	Derivative of enthalpy with respect to pressure.
ENCT	real	Derivative of enthalpy with respect to temperature.
VISC	real	CO_2 viscosity (Pa s)
VISCP	real	Derivative of viscosity with respect to pressure.
VISCT	real	Derivative of viscosity with respect to temperature.
DENW	real	Brine density (kg/m^3)
DENWP	real	Derivative of density with respect to pressure.
DENWT	real	Derivative of density with respect to temperature.
ENW	real	Brine enthalpy (MJ/kg).
ENWP	real	Derivative of enthalpy with respect to pressure.
ENWT	real	Derivative of enthalpy with respect to temperature.
VISW	real	Brine viscosity (Pa s)
VISWP	real	Derivative of viscosity with respect to pressure.
VISWT	real	Derivative of viscosity with respect to temperature.
KEYWORD *"brine"*		Invoke option for brine in the simulation. (salt-concentration dependent CO_2 solubility)

In the following example, zone 1 is injecting CO_2 dissolved water at 0.001 kg/s. The temperature is 20°C. The water has a dissolved CO_2 mass fraction of 0.3. The code will check internally whether the user specified mass fraction exceeds the equilibrium mass fraction calculated using the pressure and temperature values of the injection node. In case it does exceed that value, it is fixed at the equilibrium mass fraction. The user can specify a value of "zero" and the code will automatically fix the dissolved CO_2 mass fraction at the equilibrium value. Zone 2 is maintained at initial pressure using "aiped" calculated internally.

```
carb
    4
co2pres
    1      0      0      3.      20.      4
   -1      0      0     13       20.      4
   -2      0      0      .6      20.      4
```

```
co2frac
   1     0     0     1.0          0.0          0     100000     0
  -1     0     0     0.9465       .0535        0     0.         0.

co2flow
  -2     0     0     0            -20.         -1.e-1     3
  -1     0     0     -0.0001      -20.         0.         6

end carb
```

6.2.12 Control statement cden (optional)

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.
```

6.2.13 Control statement cflx (optional)

Total moles of liquid solute moving through a zone are output by choosing this control statement. Vapor solute molar flows are currently not available. When this macro is invoked, the following output is given at every solute time step:

- The sum of all solute source flow rates for each zone
- The sum of all solute sink rates for each zone
- The sum of all solute entering each zone
- he sum of all solute leaving each zone
- The net source/sink (boundary) solute flow for each zone

The following values can be included on the macro line to specify which solute flows should be output: *1 (source)*, *2 (sink)*, *3 (netin)*, *4 (netout)*, *5 (boundary)*. The default is to output all values.

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

Group 1 - CFLXZ

Group 2 - ICFLXZ(I), I = 1, CFLXZ

Input Variable	Format	Description
CFLXZ	integer	Number of zones for which output for solute flow through the zone is written.
ICFLXZ	integer	Zone numbers for which solute flow output is written (NFLXZ zones)

The following is an example of **cflx**. In this example solute flow through zones 1, 6 and 10 will be output.

```
cflx
    3
    1          6          10
```

6.2.14 Control statement cgdp (optional)

Assign rate-limited gdpm nodes [i.e. make the connection value large for those nodes that serve only to link gdpm nodes (diffusion only) to the flow field].

Group 1 - wdd1

Group 2 - JA, JB, JC, IGDPM_RATE_NODES

Input Variable	Format	Description
macro	character	Process to be modified: 'heat' 'tran'
IGDPM_RATE_NODES	integer	GDPM nodes for which rate should be limited.

6.2.15 Control statement chea (optional)

Convert output from pressure to head (non-head problems). The reference temperature and pressure are used to calculate density for the head calculation. For this macro the

data are entered on the macro line, and if omitted the specified default values are used. (Note that all five values must be entered to override the default values.)

Input Variable	Format	Default	Description
HEAD0	real	0.	Reference head (m)
TEMP0	real	20.	Reference temperature (°C)
PRES0	real	0.1	Reference pressure (MPa)
SAT_ICH	real	0.	Saturation adjustment after variable switch
HEAD_ID	real	0.	Output head identification for small saturations

The following is an example of **chea**. In this example pressures will be converted to heads for output using a reference pressure of 1 MPa and a reference temperature of 25 °C.

```
chea 0. 25. 1. 0. 0.
```

6.2.16 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 33)

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

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
```

6.2.17 Control statement conn (optional)

Print number of connections for each node (after simplification of connectivity based on porosity) and stop. This macro can be used to determine if isolated nodes are left active after the porosity has been set to zero to deactivate the flow solution in portions of the grid. Poorly connected nodes cause problems with the particle tracking algorithms in FEHM.

6.2.18 **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’, ‘avsx’, ‘surf’, or ‘tec’)

If CHDUM = ‘ zone’ that line is followed by

NSURF

IZONE_ISURF(I), I=1, NSURF

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. ‘surf’ produces a contour plot file compatible with the SURFER postprocessor. ‘tec’ produces a contour plot file compatible with the TECPLOT postprocessor.
NCNTR	integer	*Time step* interval for contour plots (number of timesteps). Output contour information each NCNTR timesteps.
CONTIM	real	*Time* 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 *altc* is “avs” or “avsx” or “sur” or “tec”)
CHDUM	character*72	Keyword specifying type of contour plot data files to be created in AVS UCD, AVS Express, SURFER or TECPLOT format. Keywords are entered one per line and terminated with ‘endcont’ or ‘end cont’. Valid keywords (case insensitive) are: (f)ormatted - output data in ASCII format. (m)aterial - output contour values for material properties. (l)iquid - output contour values for liquid phase. (va)por - output contour values for vapor phase. (dp)dp - output contour values for dual permeability nodes. (g)eo - output geometry values (coordinates and connectivity, avs style or old tecplot style). (gr)id - output grid geometry and connectivity in a tecplot grid file format. Parameter files will be output using tecplot variable format. (n)odit - do not output a contour file at each dit (see **time** macro).

Input Variable	Format	Description
		(c)oncentration - output solute concentration values. (ca)pillary - output capillary pressure values. (co2) - output saturation values (liquid/supercritical liquid and gas). (de)nsity - output density values. (di)splacement - output x, y, and z displacements for stress problem. When 'reldisp' is specified in the **strs** macro relative displacements are output. (fh)ydrate - output htdrate fraction. (fl)ux - output node flux (additional keywords 'net' 'volume' 'vwg') (fw)water - output water fraction. (h)ead - output head values. (hy)drate - output hydrate values. (pe)rmeability - output permeability values. (po)rosity - output porosity values. (p)ressure - output pressure values. (s)aturation - output saturation values. (so)urce - output source values. (stra)in - output strain for a stress problem. (stre)ss - output defined stresses (x, y, z, xy, xz, yz). (t)emperature - output temperature values. (ve)locity - output velocity values. (wt) - output water table elevation. (x)yz - output node coordinates (zi)d - output number of zone containing this node (as defined at end of input file) (z)one - output values for specified zones (entered on following lines) (e)ndcont - 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. The 'zone' and 'geo' keywords can not be used together. Geometry data will not be output if both keywords are specified.
NSURF	integer	Number of output zones (entered following 'zone' keyword).
IZONE_SURF	integer	List of *nsurf* zone numbers (entered following 'zone' 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 keyword output, if the *material* keyword is selected, the following material property values (at the initial time) 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*, *velocity, or density* must also be defined (otherwise, no data for these values will be written). *velocity* will result in vector values, other 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
```

6.2.19 Control statement conv (optional)

Convert input from head to pressure. Often used when converting a head-based isothermal model to a heat and mass simulation with pressure and temperature variables. The reference temperature and head are used to calculate density for the head calculation. It should be noted that this is an approximate method. Since the density is a nonlinear function of pressure and temperature this method will give slightly different answers than a calculation allowing a water column to come to thermal and mechanical equilibrium.

The reference head (head0) is converted to a pressure and added to the reference pressure (conv1) and this sum is used with the reference temperature to calculate a density. This density is used to convert the head to pressure for the identified zone. The option of adding a temperature gradient is provided as well.

The reference head (head0) is entered on the macro line:

conv HEAD0

Group 1 - NCONV

Group 2 – ZONE_CONV, ICONVF,CONV1,CONV2,CORDC,IDIRC,VARC

Group 2 is entered for each zone (nconv times).

Input Variable	Format	Description
HEAD0	real	Reference head (m)
NCONV	integer	Number of zones for variable conversion
ZONE_CONV	integer	Zone for variable conversion
ICONVF	integer	iconvf : 1- initial conditions, 2-boundary (fixed head)
CONV1	real	Reference pressure (MPa)
CONV2	real	Reference temperature (°C)

Input Variable	Format	Description
CORDC	real	Reference coordinate (m)
IDIRC	integer	Coordinate direction (1 - X, 2 - Y, 3 - Z)
VARC	real	Temperature gradient (°C/m)

The following is an example of **conv**. Here the density used to convert head to pressure is calculated with a reference head of 1000 m plus 0.1 MPa and 80 °C. The nodes in zone 45 are converted from heads to pressures at 80°C.

```
conv      1000.
   1
  45        1        0.1       80.       0.        0         0.0
```

6.2.20 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. Version2.30 added the ability to provide the coordinate data in a formatted or unformatted file.

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
```

6.2.21 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 33)

Group 3 - AAW, AGRAV, 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). Note - EPM gets overwritten by TMCH in ITER macro if that variable is defined. {1.e-5}
NORTH	integer		Number of orthogonalizations in the linear equation solver. Note - for more complicated problems, increase NORTH. For example, for fully coupled stress problme,s recommend using a value of 80 and gmre. {8 for gmre, 1 for bcgs}
MAXSOLVE	integer		Maximum number of solver iterations per Newton iteration allowed. {100}

Input Variable	Format	Default	Description
ACCM	character*4		Acceleration method for solver bcgs - Biconjugate gradient stabilized acceleration. Recommended for isothermal steady-state saturated flow problems. gmre - Generalized minimum residual acceleration. Recommended for all other types of problems.
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 {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. Set IAMM ≤ MAXIT {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)
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)

Input Variable	Format	Default	Description
LDA	integer	0	Parameter that specifies the external storage of geometric coefficients LDA = -2, element coefficients are calculated in the code and saved, unformatted, on file *filen.stor* LDA = -1, element coefficients are calculated in the code and saved on file *filen.stor* LDA = 0, element coefficients are calculated in the code and not saved LDA = +1, element coefficients are read from file *filen.stor* and no coefficients are calculated in the code LDA = +2, element coefficients are read, unformatted, from file *filen.stor* and no coefficients are calculated in the code LDA = +5, element coefficients are read from file *filen.stor* and no coefficients are calculated in the code. Coefficients are re-saved unformatted, on *filen_UNF.stor* LDA = +6, element coefficients are read from file *filen.stor* and no coefficients are calculated in the code. Coefficients are re-saved formatted, on *filen_FOR.stor* LDA = +7, element coefficients are read, unformatted, from file *filen.stor* and no coefficients are calculated in the code. Coefficients are re-saved unformatted, on *filen_UNF.stor* LDA = +8, element coefficients are read, unformatted, from file *filen.stor* and no coefficients are calculated in the code. Coefficients are re-saved formatted, on *filen_FOR.stor* It should be noted that if the coefficients are read from a file (LDA > 0) then the macro **nfinv** is ignored as well as information read from macros **elem** and **coor** since the coefficients are not being calculated.

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
```

6.2.22 Control statement **dpdp** (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 33)

Group 3 - JA, JB, JC, APUV1 (JA, JB, JC - defined on page 33)

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.	Half spacing between fractures (m). See TRANSFLAG in macros MPTR and PTRK.

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

```

6.2.23 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 33)

Group 3 - JA, JB, JC, VOLFD2 (JA, JB, JC - defined on page 33)

Group 4 - JA, JB, JC, APUVD (JA, JB, JC - defined on page 33)

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

```

6.2.24 **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 75). In the following example of **dvel**, a single internode velocity is calculated between nodes 101 and 102.

```
dvel
    1
  101       102
```

6.2.25 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 and2, 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
```

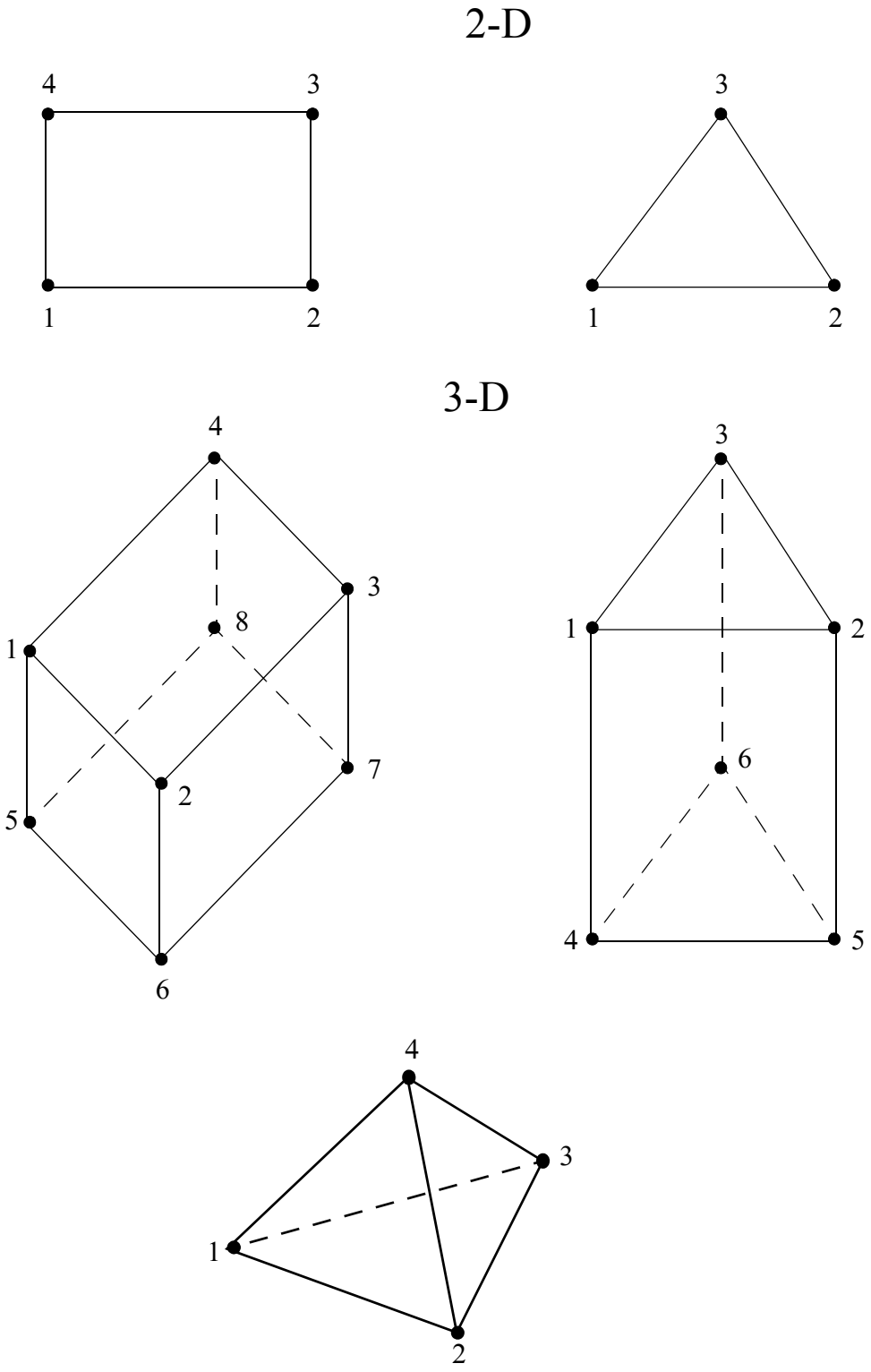

Figure 2. Elements available with FEHM in 2-D and 3-D problems showing nodal numbering convention.

6.2.26 **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 - EWI, EW2, EW3, EW4, EW5, EW6, EW7, EW8, EW9, EW10, EW11

Group 3 - EVI, 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.
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).

Input Variable	Format	Description
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/m^3, the derivative of density with respect to pressure is zero and with respect to temperature is -0.2 kg/m^3/°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/m^3. 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.     -      0.8     0.    4.2e-    9.e-     0.    0.
  0.1   20.    1.2     0.       0      8     0.      3       4      0.    0.
                  9                .    2.5            0.1    2.e-
                                   2                          4
                                0.
```

6.2.27 **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 ≥ 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
```

6.2.28 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, Z0 (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 divison from the previous to the current proportional to the assigned spacings.

In the following example of **fdm**, the "block" format is used.There is 1division 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        Þ

   1          1.0

   1          0.002
   -50        0.002

```

6.2.29 Control statement finv (optional)

Used so that FEHM calculates finite volume coefficients. No input or action is associated with this macro. This macro is on my default.

Note that in previous versions of FEHM the default was to generate finite element coefficients instead of finite volume coefficients for the flow calculations. This can be performed by using macro **nfinv**.

6.2.30 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 43).

Group 1 - JA, JB, JC, SKD, EFLOW, AIPED (JA, JB, JC - defined on page 33)

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 (°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.

Input Variable	Format	Default	Description
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 ≠ 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.
EFLOW	real	0.	a) EFLOW ≥ 0, EFLOW is the source liquid saturation, $Q_w = SKD \cdot EFLOW$ (kg/s), $Q_a = SKD \cdot (1 - EFLOW)$ (kg/s) b) EFLOW < 0, ABS(EFLOW) is the source air pressure (MPa) $Q_w = SKD$ (kg/s) $Q_a = 1.0 \cdot (P_a - ABS(EFLOW))$ (kg/s) 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.
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 pressure (MPa).
EFLOW	real	0.	a) EFLOW < 0, air only source. $Q_a = AIPED \cdot (P_a - SKD)$ (kg/s) b) 0 < EFLOW < 1, EFLOW is specified source liquid saturation, for SKD ≥ 0, 2-phase source, $Q_a = AIPED \cdot (P_a - SKD)$ (kg/s) $Q_w = AIPED \cdot (S_l - EFLOW) \cdot P_0$ (kg/s) when SKD < 0, water only source $Q_a = 0$. In the above relation S_l is the in-place liquid saturation. c) EFLOW = 1, water only source. $Q_w = AIPED \cdot (P_l - SKD)$ (kg/s)

Input Variable	Format	Default	Description
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.	Impedance parameter. $Q_w = ABS(AIPED) \cdot R_l / \mu_l (P_l - SKD)$ (kg/s) where R_l is the water relative permeability and μ_l is the water viscosity.

	PARAMETER	**SKD**		**EFLOW**	
Physics Model	**AIPED***	≥ 0	< 0	≥ 0	< 0
Non-isothermal	> 0	Flowing wellbore pressure (MPa)	Flowing wellbore pressure (MPa) - injection into rock mass	Enthalpy (MJ/kg)	Temperature (°C)
	$= 0$	Flow rate (kg/s) Heat only (MJ/s)	Flow rate (kg/s) Heat only (MJ/s) - injection into rock mass		
	< 0	Flowing wellbore pressure (MPa) - outflow only	N/A		
Isothermal	> 0	Specified source pressure (MPa)	Specified source pressure (MPa) - Water only source	Source liquid saturation	Air only source - (used as flag)
	$= 0$	Mass source strength (kg/s)	Mass source strength (kg/s) - injection into rock mass	Source liquid saturation	Source air pressure (MPa)
	< 0	Pressure above which outflow occurs (MPa)	N/A	N/A	

* Impedance parameter

The following are examples of **flow**. In the first example, at node 88, a mass flow 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 43), 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.

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

```

6.2.31 Control statement flo2 (optional)

Group 1 - JA, JB, JC, JD, SKD, EFLOW, AIPED (SKD, EFLOW, AIPED - defined on page 71 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:
JB	integer	
JC	integer	
JD	integer	

```
DO JK = JA, JB
      KL = JK - JA
      DO IJ = JA + KL, JC + KL, JD
            □□□□□□□□. . .
      ENDDO
ENDDO
```

6.2.32 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 33)

Input Variable	Format	Description
PFLOW	real	Ambient pressure (MPa) or flow area (m^2)
ESK	real	Not used

Input Variable	Format	Description
AIPED	real	Same as above for AIPED under keyword "**flow**"
KA	integer	Flag to indicate seepage face (-3) or unit gradient flux (-2)

6.2.33 Control statement floa (optional)

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

6.2.34 Control statement flxn (optional)

There is no input associated with this macro. It is a flag to FEHM to output all source and sink fluxes (non-zero values) by node to a file named source_sink.flux.

6.2.35 Control statement flxo (optional)

Mass flow 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
AIPED	real	Same as above for AIPED under keyword **flow**
NFLX	integer	Number of internode mass flows to be calculated.
IFLX1	integer	First node to be used in mass flow calculation.
IFLX2	integer	Second node to be used in mass flow calculation. If IFLX2 = 0, then the node connected to IFLX1 with the greatest internodal distance is used to calculate the mass flow.
X1	real	Coordinates of the first node to be used in mass flow calculation. Used only for those nodes where IFLX1 < 0.
Y1	real	
Z1	real	
X2	real	Coordinates of the second node to be used in mass flow 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.
```

6.2.36 Control statement **flxz** (optional)

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

The following keywords can be included on the macro line to specify which flows should be output: *liquid mass*, *vapor mass*, *thermal energy*. The default is to output any active quantity in the simulation.

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 mass flow through the zone is written.
IFLXZ	integer	Zone numbers for which water mass flow output is written (NFLXZ zones)

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

```
flxz water
   3
   1          6          10
```

6.2.37 Control statement **fper** (optional)

Assign permeability scaling factors.

Group 1 - JA, JB, JC, SCALEX, SCALEY, SCALEZ (JA, JB, JC - defined on page 33)

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
```

6.2.38 Control statement ftsc

Flag to allow distribution of fluxes (flux corr4ections) when saturation is greater than 1.

6.2.39 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 33)

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).

6.2.40 Control statement gdkm (optional)

Generalized dual permeability model.

The input structure for the **gdkm** module is the same as the **gdpm** input in FEHM.

Group 1 - GDKM_FLAG, NGDKMNODES

Group 2 - NGDKM_LAYERS(I), VFRAC_PRIMARY(I), (GDKM_X(I,J), J=1,NGDKM_LAYERS(I))

An arbitrary numbers of lines of input, terminated by a blank line.

Group 3 - JA, JB, JC, IGDKM (JA, JB, JC - defined on page 33)

Input Variable	Format	Description
GDKM_FLAG	integer	Flag to denote that the GDKM model option is being invoked. The default is 0 if GDKM is not being used. *At present the only model allowed is model 11*. This is a parallel fracture type model.

Input Variable	Format	Description
NGDKMNODES	integer	Total number of matrix nodes present in the simulation. The GDKM gridblocks, in contrast to GDPM gridblocks, are restricted to a single secondary node in a GDKM gridblock. Thus NGDPMNODES is equal to the number of gridblocks that have a GDKM model applied to them.
NGDKM_LAYERS	integer	The number of matrix nodes specified for this model number. This is always 1 for GDKM grid blocks. All primary nodes assigned to this model number (using the IGDPM input below) will have 1 matrix node.
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 assigned to the secondary porosity node.
GDKM_X	real	The matrix discretization distance for the matrix node associated with this model (units of meters). For the one secondary node allowed in the GDKM formulation, the average distance to the secondary node from the primary node.
IGDKM	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 permeability nodes at that primary nodes.

Input Variable Format Description

6.2.41 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.

Input Variable	Format	Description
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.
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 33). 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
```

6.2.42 **Control statement grad** (optional)

Gradient model input.

Group 1 - NGRAD

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

Group 2 is repeated (NGRAD times) for each gradient model being defined..

Input Variable	Format	Description
NGRAD	integer	Number of gradient models.
IZONE_GRAD	integer	Zone associated with ith model
CORDG	real	Reference coordinate 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 IGRADF = 4, Fixed boundary pressure IGRADF = 5, Fixed boundary temperature IGRADF = -5, Fixed boundary temperature, for inflow nodes IGRADF = 6, Methane pressure IGRADF = 7, Fixed Methane boundary pressure IGRADF = 8, Fixed boundary heat flow IGRADF = 9, CO_2 pressure IGRADF = 10, Fixed CO_2 boundary pressure IGRADF = 11, Pressure for matrix in gdkm or gdpm model IGRADF = 12, Temperature for matrix in gdkm or gdpm model
VAR0	real	Value of variable at reference point.
GRAD1	real	Gradient with distance.

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

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

6.2.43 Control statement hcon

Flag to set solution to heat conduction only.

6.2.44 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. Note that the value will be subtracted before output is written.

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.
```

6.2.45 Control statement hflx (optional)

Heat flow input.

Group 1 - JA, JB, JC, QFLUX, QFLXM (JA, JB, JC - defined on page 33)

A negative heat flow indicates heat flow into the reservoir.

Input Variable	Format	Default	Description
QFLUX	real	0.	If QFLXM = 0, then QFLUX is the heat flow (MW). If QFLXM ≠ 0, then QFLUX is a temperature (°C) and the heat flow is calculated according to the formula: $Q_H = QFLXM \bullet (T - QFLUX)$ (MW).
QFLXM	real	0.	If QFLXM > 0, multiplier for heat flow 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 $Q_H = ABS(QFLXM) \bullet (S_l - QFLUX)$ (MW)

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

```
hflx
  401        410        1        -0.001      0.0
```

6.2.46 Control statement hist (optional)

History data output selection, output timestep intervals, and time intervals. Parameters will be output for the nodes specified in the **node** or **nod2** macro in individual files for each parameter selected. If output zones are defined (**node** macro) the output will be a volume weighted average for the zone. Currently zone averaged values can be output for pressure, head, temperature, and enthalpy. History files will be named using the root portion of the history file name (*root_name.his*) provided as input, e.g., pressure output would be in a file named: *root_name_pres.his*. The named history output file

will contain run information and the list of selected output nodes (with their coordinates) and zones (with node list).

Group 1 - CHDUM

or using optional input or keywords

Group 1 - CHDUM, NHIST, HISTIME

where CHDUM is ‘years’, ‘days’, ‘hrs’, or ‘seconds’

or

Group 1 - CHDUM, CHDUM1, . . . , CHDUMn

where CHDUM is mpa, pressure, density, viscosity, or global

Input Variable	Format	Description
CHDUM	character*80	Keyword specifying type of history plot data files to be created. Keywords are entered one per line and terminated with ‘end hist’ or a blank line. Keywords must be entered starting in the 1st column. Valid keywords (case insensitive) are: ‘tecplot’ - data will be output using tecplot style headers and format ‘csv’ or ‘surfer’ - data and parameter headers will be output as comma separated variables (‘csv’ format) Note; If a file format keyword is being used, it must immediately follow the macro name. Alternatively, it may be entered on the macro line. The default is for the headers and data to be output using plain text and spaces. ‘years’ - output time in years ‘days’ - output time in days ‘hrs’ - output time in hours ‘seconds’ - output time in seconds Note: If a time keyword (years, days, hrs, or seconds) is not entered, output time will be in days and data will be output for each timestep. The time output keywords may be used with optional input NHIST and HISTIME. ‘mpa’ or ‘pressure’ - output pressure in MPa ‘deg’ or ‘temperature’ - output temperature in °C ‘head’ or ‘meters’ - output head in meters ‘feet’ - output head in feet ‘saturation’ - output saturation ‘wco’ - water content ‘flow’ or ‘kgs’ - output flow in kg/s ‘enthalpy’ - output enthalpy in MJ/kg ‘efl’ or ‘mjs’ - output enthalpy flow (MJ/s)

Input Variable	Format	Description
		'density' - output density (kg/m^3)
		'humidity' - output relative humidity
		'viscosity' - output viscosity (Pa-s)
		'zflux' - output zone fluxes
		'concentration' - output species concentration (concentrations for each specie will be output in a separate file)
		'wt' - output water table elevation
		'co2s' - output CO_2 saturations (volume fractions) An 'l' or 'g' may be appended to co2s to specify that only liquid or gas saturation should be output.
		'co2m' - output total CO_2 mass (kg), free CO_2 mass fraction, and dissolved CO_2 mass fraction. Other wise use the form listed below to output specified quantity:
		'co2mt' - output total CO_2 mass (kg)
		'co2mf' - output free CO_2 mass fraction
		'co2md' - output dissolved CO_2 mass fraction
		'cfluxz' - output CO_2 zone fluxes
		'displacements' - output displacements (m), 'disx', 'disy' or 'disz' may be used to select only the x, y, or z displacement.
		'stress' - output stresses, 'strsx', strsy', 'strsz', 'strsxy', 'strsxz' or 'strsyz' may be used to select specific stress components.
		'strain' - output strain
		'rel' - output a table of relative permeability values for each input model.
		'global' - output global parameters
NHIST	integer	Optional: *Time step* interval for history plots (number of timesteps). Output history information each NHIST timesteps. If not entered NHIST = 1.
HISTIME	real	Optional: *Time* interval for history plots. In addition to output each NHIST timesteps, output history information each HISTIME. Units correspond to units specified by selected time output keyword (years, days, hours, or seconds). If not entered HISTIME = 1.e30.
CHDUM1 . . . CHDUMn	character*80	Optional keywords specifying selections for history plot data files to be created. Optional keywords are entered on the same line as primary keywords. If no optional keywords are used the code will determine what will be output based on problem input. Up to 3 optional keywords may be entered.

Input Variable	Format	Description
		Valid keywords (case insensitive) used with keyword ‘pressure’ or ‘mpa’ are: ‘total’ or ‘water’- output total or water pressure ‘air’ - output air pressure ‘capillary’ - output capillary pressure ‘co2’ - output CO_2 pressure Valid keywords (case insensitive) used with keyword ‘density’ or ‘viscosity’ are: ‘water’ - output water density or viscosity ‘air’ - output air/vapor density or viscosity ‘co2’ - output CO_2 liquid and gas density or viscosity. An ‘l’ or ‘g’ may be appended to co2 to specify that only liquid or gas density should be output. Valid keywords (case insensitive) used with keyword ‘global’ are: ‘mass’ - output mass balances only for problem (excluding steam) (used with keyword ‘global’) ‘water’ - output water balances only for problem (excluding steam) (used with keyword ‘global’) ‘steam’ - output mass/water balance only for problem including steam (used with keyword ‘global’) ‘air’ = output air / vapor balances only for problem (used with keyword ‘global’) ‘energy’ - output energy balances only for problem (used with keyword ‘global’) Note: If no optional keywords are used with the ‘global’ keyword the code will determine which balances will be output based on problem input (mass/ energy or water/air). Currently only 1 optional keyword may be used with global to specify a single balance type. Balance output includes: Total (mass, water, air in kg, energy in MJ) in system, total discharge, total input, current discharge, current input, and net discharge.

The following are examples of **hist**. For the first example, time will be output in years and temperatures in °C. Data will be output each 100000 timesteps or at time intervals Š 50 years. In the second example,□ pressures in MPa (water and air) and

temperatures in °C will be written each timestep and time will be output in days. The global mass balance for water will also be output at each time step.

```
hist
years        100000        50.
deg
end
```

```
hist
mpa          total         air
deg
global       mass
end
```

6.2.47 Control statement **hyco** (required if macro **perm** not usedl)

Hydraulic conductivity input.

Group 1 - JA, JB, JC, PNXD, PNYD, PNZD (JA, JB, JC - defined on page 33)

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
```

6.2.48 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 33)

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].

6.2.49 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
```

6.2.50 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 ≤ 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)

$$T = TIN1 + GRAD1 \times Z, 0 \le Z \le \text{DEPTH}$$

$$T = TIN2 + GRAD2 \times Z + QUAD \times Z^2, Z > \text{DEPTH}$$

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.
```

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.
```

6.2.51 Control statement intg

6.2.52 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.

6.2.53 Control statement iter (optional, but recommended)

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*SQRT(R**2), G2*R**2)))

where R**2 is the sum-squared of the equation residuals, and F0 is the 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 TMCH > 0. If satisfied by the residual norm, the Newton iteration is assumed to be complete. Newton-Raphson stopping criteria if TMCH < 0 (recommended). 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).

Input Variable	Format	Default	Description
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. {0} 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.
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 noncondensible gas, label the mass equation as 1, the heat equation as 2, and the noncondensible 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. {0}

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		

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. {0}

Input Variable	Format	Default	Description
ICOUPL	integer	0	Number of SOR iterations used in reduced degree of freedom methods. {0}
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
```

6.2.54 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

ITFCPORSIZE(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

Input Variable	Format	Description
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.
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
```

6.2.55 Control statement itup (optional)

Controls upstream direction. 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
```

6.2.56 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.

6.2.57 Control statement ivfc

Enable exponential fracture and volume model.

6.2.58 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.
I_ELIM	integer	0	IF I_ELIM Š 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
```

6.2.59 **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, MAX1D

Group 2 - POUT, PRNT_RST

or when PRNT_RST ≥ 20, selected output parameters

Group 2 - POUT, PRNT_RST, PRNT_VARNUM (1 . . . 6)

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, FFMAX

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, RSEED_RELEASE

Optional keyword “wtri” is input to indicate a water table rise calculation should be performed. It is followed by WATER_TABLE. For GoldSim the water table rise calculation is controlled by passing a new water table elevation to the code during the simulation and the keyword is not required.

KEYWORD ‘wtri’

WATER_TABLE

Group 4 - DAYCS, DAYCF, DAYHF, DAYHS

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 5, followed by the name of the file that the transport parameters reside in. The applicable transport parameters are defined in Group 5 and Group 9.

KEYWORD ‘file’

PFILENAME

The structure of the alternate parameter file is:

NINPUTS

PTRPARAM(I) I=1 to NINPUTS

.

.

.

[a line of parameters is present for each realization]

The method for assigning a given value of the particle tracking parameter (PTRPARAM) to a specific transport parameter, defined in Group 5 or Group 9, 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 11 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 5 or Group 9. 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 **mptr**. It should be noted that for the colloid diversity model, only K_REV need be negative to indicate values should be read from the parameter file, if K_REV is negative then all five parameters are read from the file, otherwise the equation parameters will be read from the **mptr** macro. This is to accommodate the fact that the SLOPE_KF may have negative values.

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 Group6.

Optional keyword “afm” indicates the Active Fracture Model input for determining fracture spacing should be used. Optional keyword “dfree” is input to indicate that a free water diffusion coefficient and tortuosity will be entered instead of the molecular diffusion coefficient.

KEYWORD ‘afm’

KEYWORD ‘dfre’

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 33)

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, ASTEP, CFRACTION

Optional keyword “size” is input to indicate that the colloid size distribution model option is being used. It is followed by PART_SIZE and PROBSIZE.

KEYWORD ‘size’

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

Optional keyword “dive” is input to indicate that the colloid diversity model is being used. It is followed by FLAG_COL_DAUGHTER, optional keyword “file” and the name of the file containing the CDF table or equation data (a description of the format for the file is provided with the second **mptr** example below), the TPRPFLAG with

optional SIMNUM, optional CDF equation parameters (when “file” is not used), and keyword “irreversible” or “reversible” with FLAG_LOG.

KEYWORD ‘dive’	or	KEYWORD ‘dive’
FLAG_COL_DAUGHTER		FLAG_COL_DAUGHTER
KEYWORD ‘file”		TPRPFLAG
CDFFILENAME		or
TPRPFLAG		TPRPFLAG, SIMNUM
or		K_REV, R_MIN,R_MAX, SLOPE_KF, CINT_KF
TPRPFLAG, SIMNUM		KEYWORD ‘irreversible’
KEYWORD ‘irreversible’		or
or		KEYWORD ‘reversible’, FLAG_LOG
KEYWORD ‘reversible’, FLAG_LOG		

Note that optional KEYWORDs “size” and “dive” are only used when colloid transport is enabled.

Group 8 - LAYERS

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

or for simulations using “dfree”

Group 9 - LAYER_I, TRANSFLAG, KD, RD_FRAC, H2O_DIFF, TORT_DIFF

or for simulations with colloid (TRANSFLAG < 0)

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

or for simulations with colloid using “dfree”

Group 9 - LAYER_I, TRANSFLAG, KD, RD_FRAC, H2O_DIFF, TORT_DIFF, KCOLL, RCOLL, FCOLL

Group 10 - NS

Group 11- JA, JB, JC, TMPCNSK (JA, JB, JC - defined on page 33)

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 39 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 ≤ 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
MAX1D	integer	Maximum 1-D array size for holding particle tracking information for all simulated species. The value of MAX1D depends on number of species, number of time steps, number of radionuclide release bins, number of species involved in ingrowth, and the length of the decay-ingrowth chain.
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", ".ptrk_fin", or ".ptrk" files: If PRNT_RST = 0, Particle information is not written to the output files. If PRNT_RST = 1, 11, 21, 31, 41 All particle information necessary for a restart is written to the ".fin" file. If PRNT_RST = -1, -11, -21, -31, -41 Only particle positions and ages are written to the ".fin" file. If ABS (PRNT_RST) = 2, 12, 22, 32, 42 Mass flux values are written to the ".fin" file followed by particle information. If 10 ≤ ABS(PRNT_RST) < 30 Particle exit locations and count are written to the ".ptrk_fin" file. If ABS(PRNT_RST) ≥ 20 Cumulative particle counts versus time are written to the ".ptrk" file, for variables specified by PRNT_VARNUM (the default is to output all variables). If ABS(PRNT_RST) ≥ 40, Cumulative mass output from a FEHM/ GoldSim coupled simulation will be written to file FEHM_GSM_Mass_balance.txt. Note that to track cumulative mass an additional array of size maxparticles*nspeci must be allocated so caution should be used when specifying this option to ensure sufficient system memory is available. 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 "-" options 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.
PRNT_VARNUM	integer	A list of integers specifying which particle counts should be output. For each value entered PRNT_VAR(PRNT_VARNUM) is set to true. If no values are entered the default is to print all variables. 1 – Number of particles that have entered the system 2 – Number of particles currently in the system 3 – Number of particles that have left the system 4 – Number of particles that have decayed 5 – Number of particles that have been filtered 6 – Number of particles that left this time interval

Input Variable	Format	Description
		Note: The data found in the ".ptrk" file was previously reported in the general output file. From version 2.25 of the code and forward that data will be reported in the optional, ".ptrk" file unless a coupled GoldSim-FEHM simulation is being run. In addition, the user has the option of selecting which statistics parameters are reported. The default is to report all statistics parameters.
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.
FFMAX	real	The maximum fracture flow fraction used in the transfer function curve data. Default value: 0.99.
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.
RSEED_RELEASE	integer	6-digit integer random number seed for particle release location calculation. If a value is not entered for RSEED_RELEASE it will be set equal to RSEED. Note that for GoldSim-FEHM coupled simulations the random seeds are controlled by GoldSIM and the values input in the **mptr** macro are not used.
KEYWORD	character*4	Optional keyword 'wtri" indicatiing a water table rise calculation should be performed.
WATER_TABLE	real	Water table elevation to be used for water table rise calculation. Note that for GoldSim-FEHM coupled simulations the water table rise calculations are controlled by GoldSIM and the values input in the **mptr** macro are not used and may be omitted.
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).

Input Variable	Format	Description
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.
KEYWORD	character*5	Optional keyword ‘dfree’ designates that the free water diffusion coefficient and tortuosity will be input instead of the molecular diffusion coefficient.
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.
APERTUR	real	Mean fracture aperture (m). The input value is ignored when matrix diffusion is turned off.
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.

Input Variable	Format	Description
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. 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. For both stand alone and GoldSim-FEHM coupling cases: If CONFACTOR > 0, the code assumes the input mass is in moles and uses the product of the CONFACTOR and the input mass to calculate the input number of particles at each time step. When CONFACTOR >0, FEHM may use an updated conversion factor from previous time step(s) as the input for the current time step instead of using the original input CONFACTOR for improved results. If CONFACTOR < 0, the code uses the product of the absolute value of CONFACTOR and the input mass (in moles) to calculate the input number of particles at each time step. A CONFACTOR updated from a previous time step will not be used.
NEWCONFACTOR	real	Replace the initial value of CONFACTOR with that specified by NEWCONFACTOR. If NEWCONFACTOR = 0, use automatic conversion factors. If NEWCONFACTOR > 0, then use the product of the CONFACTOR and the input mass (in moles) to calculate the input number of particles at each time step starting from CONFTIME. In this case, FEHM may use an updated conversion factor from previous time step(s) as a modification to CONFACTOR. If NEWCONFACTOR < 0, then FEHM uses the product of the absolute value of NEWCONFACTOR and the input mass (in moles) to calculate the input number of particles at each time step (CONFACTOR = -NEWCONFACTOR).
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.

Input Variable	Format	Description
P_FRACTION	real	The decay-ingrowth particle release factor (percentage of the maximum number of particles released for the current species). For decay-ingrowth simulations, P_FRACTION is used to reduce the number of particles released by parent or daughter species, thus, avoiding memory overflow in the daughter species due to parent decay-ingrowth where multiple parents decay to the same daughter species. The normal range of P_FRACTION is from 0 to 1. The default value is 0.25. A user should select an appropriate value based on the mass input of parent and daughter species, half-lives, importance of each species to the transport results, and simulation time period.
ASTEP	integer	Maximum length of array used to hold particle tracking information for the ith species. Its value depends on number of time steps, number of release bins, and number of parent species. The sum of ASTEP for all species should be equal to or smaller than MAX1D.
CFRACTION	real	The fraction of the user determined maximum number of particles (MAX_PARTICLES) to be assigned by mass, (1 – cfraction) will then be the fraction of particles assigned by time step.
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. Colloid size is only sampled once for each realization.
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.
KEYWORD	character*4	Optional keyword “dive” signifying that the specie being specified is either a colloid species using the colloid diversity model or a non-colloid daughter species of a colloid species.
FLAG_COL_DAUGHTER	integer	When FLAG_COL_DAUGHTER = 1 signals that the species being specified is a non-colloid species that can result as a daughter product of a colloid parent species. If the species is not a daughter product or the daughter product is a colloid, FLAG_COL_DAUGHTER = 0.
KEYWORD	character*4	Optional keyword ‘file’ designating the cumulative probability distribution function (CDF) retardation parameters for the colloid diversity model should be read from an external file.

Input Variable	Format	Description
CDF_FILENAME	character*80	Name of the file containing the cumulative probability distribution function (CDF) (entered if optional keyword 'file' follows keyword 'dive'). See below for file formats. If TPRPFLAG = 11 or 12, Table option If TPRPFLAG = 13 or 14, Equation option The following equations are used for $R_{min} \le R \le R_{max}$, $R = 1 + K_f / K_{rev}$, $\log_{10}(CDF) = b + m \cdot \log_{10}(K_f)$
TPRP_FLAG	integer	Values of TPRPFLAG between 11 and 14 signify that the colloid diversity model with equal weight sampling will be used: TPRPFLAG = 11: CDF vs retardation factor specified in a table TPRPFLAG = 12: similar to 11, but the SQRT(CDF) is used instead of CDF for sampling TPRPFLAG = 13: CDF vs K_f (Attachment rate constant) specified as a straight line equation in the log-log space TPRPFLAG = 14: similar to 13, but the SQRT(CDF) is used instead of CDF for sampling
SIMNUM	integer	Simulation number, used for selecting the table/equation from the colloid diversity file. For GoldSim-FEHM coupled simulations or FEHM runs using the 'msim' option this parameter is passed to the code. For non-coupled simulations it is an optional input. (Default value = 1)
K_REV	real	Detachment rate constant for reversible filtration of irreversible colloids.
R_MIN	real	Minimum value of the retardation factor for reversible filtration of irreversible colloids.
R_MAX	real	Maximum value of the retardation factor for reversible filtration of irreversible colloids
SLOPE_KF	real	Value of the slope (m) in the log-log space for the equation: $\log_{10}(CDF) = b + m \cdot \log_{10}(K_f)$
CINT_KF	real	Value of the intercept (b) in the log-log space for the above equation
KEYWORD	character	Keyword specifying whether the colloid species is 'irreversible' or 'reversible'.
FLAG_LOG	integer	For reversible colloids an average retardation factor is used: If FLAG_LOG = 0: a linear average of the distribution is used If FLAG_LOG = 1: a log-linear average of the distribution is used
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

Input Variable	Format	Description
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. For equivalent continuum solutions, the fracture spacing in the finite spacing model is determined using $SPACING = APERTURE/POROSITY$. For dual permeability models, the fracture spacing input parameter APUV1 in the **dpdp** macro is used as the half-spacing between fractures. 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.
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.
DIFFMFL	real	Molecular diffusion coefficient in the rock matrix (m^2/s). The input value is ignored unless matrix diffusion is invoked.
H2O_DIFF	real	Free water diffusion coefficient. The molecular diffusion coefficient is calculated as $H2O\_DIFF \times TORT\_DIFF$.
TORT_DIFF	real	Tortuosity

Input Variable	Format	Description
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 $DISTANCE$ is the length of the path between nodes.
NS		Number of spatial source terms for the ith species
TMPCNSK	real	Particle injection parameter assigned for nodes defined by JA, JB, and JC. Two options are available: 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. 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.
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.

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 (**mptr** or **ptrk**): 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, RSEED_RELEASE, 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 (Group 4 input) 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 starting time at the top of the ".ini" file to 0 or in the **time** macro 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 **mptr**. A multiple-species decay-chain ($\rightarrow 2 \rightarrow 3 \rightarrow$) 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

In the second example, transfer function data is used along with the active fracture and colloid diversity models. The the cumulative probability distribution function (CDF) retardation parameters for the colloid diversity model are entered in an external file using the table format. The format for the new input files associated with the colloid diversity model are:

For **ptrk/sptr/mptr** simulations with TPRP_FLAG = 11 or 12:

- Header line indicating the species number (always 1 for ptrk/sptr simulations)
- Multiple tables, each with the following format:

One line specifying the realization number

Multiple lines with two columns of data (real) representing rcdiv and probdiv. Note that probdiv should start at 0 and end with 1.

A blank line is used to specify the end of the distribution table.

- A blank line is used to specify the end of the file

For mptr, a distribution will need to be entered for each colloid species. Therefore, the header line and tables are repeated for each colloid species.

For **ptrk/mptr** simulations with TPRP_FLAG = 13 or 14:

- Header line containing comments
- Multiple lines, each line containing realization number, b, m, k_r, R_{min},R_{max}.
- A blank line is used to specify the end of the file

For **sptr** simulations with TPRP_FLAG = 13 or 14:

- Header line containing comments
- Multiple lines, each line containing realization number, b, m, k_r, R_{min}, R_{max}, α_L, α_{TH}, α_{TV}, DM, VRATIO

A blank line is used to specify the end of the file

Particle statistics data for the cumulative number of particles that have left the sytem and the number of particles that left during the current timestep are output.

```
mptr
3 100 500000 0                                                              Group 1
0 30 3 6                                                                    Group 2
tcurve
3
../colloid_cell/input/uz_tfcurves_nn_3960.in
244562                                                                      Group 3
0.     1.e20   0.     1.e20                                                 Group 4
afm
1.e-03  1.e-03  1.e-03  1.e-03        0.2       0.01   0.6 //layer 1  tcwm1 zone 1    Group 5
1.e-03  1.e-03  1.e-03  0.00e+00     0.2       0.00   0.0 //layer 2  tcwm2 zone 2    Group 5

1          10         1         1                                           Group 6
11         20         1         2                                           Group 6

1          1          0         -1        1         0         1.00E+15243   1.0       speci1   Group 7
diversity
0
file
../colloid_cell/input/rcoll_data.dat
11 1
irreversible
2                                                                           Group 8
1      -2       0.0  1.0e+00   1.00e-30      1e+20    1    1.00    #1 tcwM1      Group 9
2      -2       0.0  1.0e+00   1.00e-30      1e+20    1    1.00    #2 tcwM2      Group 9
```

```
1                                                                                                   Group 10
1       1       1       -1.                                                                         Group 11
100000  0.      0.01                                                                                Group 12

3       1       0       -1      1       0       1.00E+15243     1.0     speci3                      Group 7
diversity
0
file
../colloid_cell/input/rcoll_data.dat
11 3
irreversible
2                                                                                                   Group 8
1    -2     0.0  1.0e+00  1.00e-30    1e+20   1   1.00   #1 tcwM1                                  Group 9
2    -2     0.0  1.0e+00  1.00e-30    1e+20   1   1.00   #2 tcwM2                                  Group 9

1                                                                                                   Group 10
1       1       1       -1.                                                                         Group 11
0       0.      0.01                                                                                Group 12
```

With the file rcoll_data.dat as:

```
Test file for 1D importance sampling
1   0.0   0.0   0.0
1.0     0.0
2.0     0.125
3.0     0.25
4.0     0.375
5.0     0.5
6.0     0.625
7.0     0.75
8.0     0.875
9.0     1.0

.
.
.

3   0.0   0.0   0.0
1.0     0.0
2.0     0.125
.
.
.
9.0     1.0
```

6.2.60 Control statement **nfinv** (optional; not recommended)

No input is associated with this macro. When invoked, the code will generate finite element coefficients instead of finite volume coefficients for flow terms. Note that in previous versions of FEHM the default was to perform finite element calculations.

6.2.61 Control statement **ngas** (optional)

Noncondensible gas transport.

Group 1 - ICO2D

Group 2 - JA, JB, JC, PCO2 (JA, JB, JC - defined on page 33)

Group 3 - JA, JB, JC, CPNK (JA, JB, JC - defined on page 33)

Group 4 - JA, JB, JC, QCD (JA, JB, JC - defined on page 33)

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 noncondensible 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 noncondensible gas. If PCO2 < 0 then ABS (PCO2) is interpreted as a temperature and the partial pressure of the noncondensible 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 noncondensible 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 noncondensible gas pressure. There is no specified noncondensible gas source.

```
ngas
   3
   1        800        1        -20

   1        800        1         0.

   1        800        1         0.

```

6.2.62 Control statement **nobr** (optional)

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

6.2.63 Control statement **node** (optional)

Specify the node numbers for which detailed output is desired. In version 2.30 macro **node** has been modified to allow multiple instances of the macro to be used (results are cumulative) and to allow the definition of "output zones" which are used in conjunction with macro **hist**. Only a single input format / keyword can be used for each instance of the node macro.

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 33)

Input Variable	Format	Description
M	integer	Number of nodes for which information will be printed on the output (iout) and history plot (ishis, istrc) files. If M ≤ 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 ≤ 0.
MN	integer	M node numbers for which information will be printed on the output file (iout). If MN(I) < 0, then coordinates are used to define the print-out node, and the coordinate sets (X, Y, Z) for each MN(I) < 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	

Input Variable	Format	Description
KEYWORD	character*5	Keyword ‘block’ to invoke node specification by JA, JB, JC format. Keyword ‘azone’ to invoke output zone specification by JA, JB, JC format. This keyword allows a single node to be included in multiple output zones.

The following are examples of **node**. In the first example (top left), 2 nodes are specified for output, nodes 50 and 88. In the second example (top right), two nodes are specified for output, the node numbered 50 and the node closest to the coordinates X = 100. m, Y = 1000. m and Z = 0. m. In the third example (bottom left), 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**). In the fourth example (bottom right), output is specified for two zones, the first zone contains nodes 1, 11, 21, 31, 41, 51, 61, 71, 81, 91 and the second zone is made up of the nodes defined for zone 3 (previously specified using the zone macro).

```
node
   2
  50        88
```

```
node
   2
  50       -88
 100.     1000.        0.
```

```
node
block
   1       100        10
  -3         0         0
```

```
node
azone
   1       100        10
  -3         0         0
```

6.2.64 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 ≤ 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 ≤ 0.
M2	integer	Number of nodes for short list (terminal printout). If M2 ≤ 0, Group 4 is omitted.

Input Variable	Format	Description
MN	integer	M node numbers for which information will be printed on the output file (iout). If a MN(I) < 0, then coordinates are used to define that print-out node, and the coordinate sets (X, Y, Z) for each MN(I) < 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 ≠ 0. If MNI(I) < 0, then coordinates are used to define the terminal output nodes, and the coordinate sets (X, Y, Z) for each MNI(I) < 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. m, Y = 1000. m and Z = 0. m.

```
nod2
  2        1
 50       88
 50
```

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

6.2.65 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 ≤ 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 ≤ 0.

Input Variable	Format	Description
M2	integer	Number of nodes for short list (terminal printout). If M2 ≤ 0, Group 4 is omitted.
M3		Number of nodes for short list (variable porosity model information printout). If M3 ≤ 0, Group 6 is omitted.
MN	integer	M node numbers for which information will be printed on the output file (iout). If a MN(I) < 0, then coordinates are used to define that print-out node, and the coordinate sets (X, Y, Z) for each MN(I) < 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 ≠ 0. If MNI(I) < 0, then coordinates are used to define the terminal output nodes, and the coordinate sets (X, Y, Z) for each MNI(I) < 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. m, Y = 1000. m and Z = 0. m.

```
nod2
   2          1
  50         88
  50
```

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

6.2.66 Control statement nrst (optional)

Stop NR iterations on variable changes (not equation tolerance).

Group 1 VARD, STOPC

Input Variable	Format	Description
VARD	character	Variable to check.
STOPC	real	Value to check on variable change.

6.2.67 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. Permeabilties may be entered as log values.

Group 1 - JA, JB, JC, PNXD, PNYD, PNZD (JA, JB, JC - defined on page 33)

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. m^2 respectively.

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

6.2.68 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 (*fehmn.pest*) 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.
```

6.2.69 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 33)

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 MPa^{-1}.

```
ppor
    1
    1          0          0          1.e-2
```

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

Group 1 - JA, JB, JC, PHRD, TIND, IEOSD (JA, JB, JC - defined on page 33)

The initial values defined in control statement **pres** supersede all others. Note that the term “saturated” refered 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
```

6.2.71 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

Optional keyword “wtri” is input to indicate a water table rise calculation should be performed. It is followed by WATER_TABLE. For GoldSim the water table rise calculation is controlled by passing a new water table elevation to the code during the simulation and the keyword is not required.

KEYWORD ‘wtri’

WATER_TABLE

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

or when PRNT_RST ≥ 20, selected output parameters

Group 3 - TRAK_TYPE, HALF_LIFE, POUT, PRNT_RST PRNT_VARNUM (V1 … V6)

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, FFMAX

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
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 11 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**. It should be noted that for the colloid diversity model, only K_REV need be negative to indicate values should be read from the parameter file, if K_REV is negative then all five parameters are read from the file, otherwise the equation parameters will be read from the **ptrk** macro. This is to accommodate the fact that the SLOPE_KF may have negative values.

KEYWORD 'afm'

KEYWORD 'dfree'

Optional keyword "size" is input to indicate that the colloid size distribution model option is being used. It is followed by PART_SIZE and PROBSIZE.

KEYWORD 'size'

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

Optional keyword "dive" is input to indicate that the colloid diversity model is being used. It is followed by optional keyword "file" and the name of the file containing the CDF table or equation data (a description of the format for the file is provided with

the second **mptr** example), the TPRPFLAG with optional SIMNUM, optional CDF equation parameters (when "file" is not used), and keyword "irreversible" or "reversible" with FLAG_LOG.

KEYWORD 'dive'	or	KEYWORD 'dive'
KEYWORD 'file"		TPRPFLAG
CDFFILENAME		or
TPRPFLAG		TPRPFLAG, SIMNUM
or		K_REV, R_MIN,R_MAX, SLOPE_KF,
TPRPFLAG, SIMNUM		CINT_KF
KEYWORD 'irreversible'		KEYWORD 'irreversible'
or		or
KEYWORD 'reversible',		KEYWORD 'reversible', FLAG_LOG
FLAG_LOG		

Note that optional KEYWORDs "size" and "dive" are only used when colloid transport is enabled.

There are eight possible forms of input for Group 4, which depend on whether or not the Active Fracture Model is implemented (optional KEYWORD "afm"), the free water diffusion coefficient is used (optional KEYWORD "dfree"), and colloid transport is enabled (TRANSFLAG < 0). If colloid transport is enabled, then ABS(TRANSFLAG) is used to determine the transport mechanism.

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 with 'dfree"'

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), H2O_DIFF(ITRC), TORT_DIFF(ITRC), RD_FRAC(ITRC), MATRIX_POR(ITRC), APERTURE(ITRC)

or when 'afm' is implemented

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 'afm' is implemented with 'dfree'

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), H2O_DIFF(ITRC), TORT_DIFF(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 with 'dfree'

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), H2O_DIFF(ITRC), TORT_DIFF(ITRC), RD_FRAC(ITRC), MATRIX_POR(ITRC), APERTURE(ITRC), KCOLL(ITRC), RCOLL(ITRC), FCOLL(ITRC)

or when 'afm' is implemented

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)

or when 'afm' is implemented with 'dfree'

Group 4 - TRANSFLAG(ITRC), KD(ITRC), TCLX(ITRC), TCLY(ITRC), TCLZ(ITRC), H2O_DIFF(ITRC), TORT_DIFF(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 33)

Group 6 - JA, JB, JC, PCNSK, T1SK, T2SK (JA, JB, JC - defined on page 33)

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 33)

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.
RSEED_RELEASE	integer	6-digit integer random number seed for particle release location calculation. If a value is not entered for RSEED_RELEASE it will be set equal to RSEED. Note that for GoldSim-FEHM coupled simulations the random seeds are controlled by GoldSIM and the values input in the **ptrk** macro are not used.
MAX1D	integer	Maximum 1-D array size for holding particle tracking information for all simulated species. The value of MAX1D depends on number of species, number of time steps, number of radionuclide release bins, number of species involved in ingrowth, and the length of the decay-ingrowth chain.
KEYWORD	character*4	Optional keyword 'wtri" indicatiing a water table rise calculation should be performed.
WATER_TABLE	real	Water table elevation to be used for water table rise calculation. Note that for GoldSim-FEHM coupled simulations the water table rise calculations are controlled by GoldSIM and the values input in the **ptrk** macro are not used and may be omitted.

Input Variable	Format	Description
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.
GMOL	real	The molecular weight of the ith 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 (days). 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.

Input Variable	Format	Description
		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.
PRNT_RST	integer	Flag to specify whether particle information is written to the ".fin", ".ptrk_fin", or ".ptrk" files: If PRNT_RST = 0, Particle information is not written to the output files. If PRNT_RST = 1, 11, 21, 31, 41, All particle information necessary for a restart is written to the ".fin" file. If PRNT_RST = -1, -11, -21, -31, -41, Only particle positions and ages are written to the ".fin" file. If ABS (PRNT_RST) = 2, 12, 22, 32 or 42, Mass flux values are written to the ".fin" file followed by particle information. If 10 ≤ ABS(PRNT_RST) < 30, Particle exit locations and count are written to the ".ptrk_fin" file. If ABS(PRNT_RST) ≥ 20, Cumulative particle counts are written to the ".ptrk" file, for variables specified by PRNT_VARNUM (the default is to output all variables). If ABS(PRNT_RST) ≥ 40, Cumulative mass output from a FEHM/GoldSim coupled simulation will be written to file FEHM_GSM_Mass_balance.txt. Note that to track cumulative mass an additional array of size maxparticles*nspeci must be allocated so caution should be used when specifying this option to ensure sufficient system memory is available. 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.

Input Variable	Format	Description
		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.
PRNT_VARNUM	integer	A list of integers specifying which particle counts should be output. For each value entered PRNT_VAR(PRNT_VARNUM) is set to .true. If no values are entered the default is to print all variables. 1 – Number of particles that have entered the system 2 – Number of particles currently in the system 3 – Number of particles that have left the system 4 – Number of particles that have decayed 5 – Number of particles that have been filtered 6 – Number of particles that left this time interval Note: The data found in the ".ptrk" file was previously reported in the general output file. From version 2.25 of the code and forward that data will be reported in the optional, ".ptrk" file unless a coupled GoldSim-FEHM simulation is being run. In addition, the user has the option of selecting which statistics parameters are reported. The default is to report all statistics parameters.
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.
NINPUTS	integer	Number of inputs in each row of data in the alternate transport parameter input file (PFILENAME).

Input Variable	Format	Description
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*5	Optional keyword 'dfree' designates that the free water diffusion coefficient and tortuosity will be input instead of the molecular diffusion coefficient.
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.
KEYWORD	character*4	Optional keyword "dive" signifying that the specie being specified is either a colloid species using the colloid diversity model or a non-colloid daughter species of a colloid species.
KEYWORD	character*4	Optional keyword 'file' designating the cumulative probability distribution function (CDF) retardation parameters for the colloid diversity model should be read from an external file.
CDF_FILENAME	character*80	Name of the file containing the cumulative probability distribution function (CDF) (entered if optional keyword 'file' follows keyword 'dive'). See Section 6.2.59 for file formats. If TPRPFLAG = 11 or 12, Table option If TPRPFLAG = 13 or 14, Equation option The following equations are used for $R_{min} \le R \le R_{max}$, $R = 1 + K_f / K_{rev}$, $\log_{10}(CDF) = b + m \cdot \log_{10}(K_f)$
TPRP_FLAG	integer	Values of TPRPFLAG between 11 and 14 signify that the colloid diversity model with equal weight sampling will be used: TPRPFLAG = 11: CDF vs retardation factor specified in a table TPRPFLAG = 12: similar to 11, but the SQRT(CDF) is used instead of CDF for sampling TPRPFLAG = 13: CDF vs K_f (Attachment rate constant) specified as a straight line equation in the log-log space TPRPFLAG = 14: similar to 13, but the SQRT(CDF) is used instead of CDF for sampling

Input Variable	Format	Description
SIMNUM	integer	Simulation number, used for selecting the table/equation from the colloid diversity file. For GoldSim-FEHM coupled simulations or FEHM runs using the 'msim' option this parameter is passed to the code. For non-coupled simulations it is an optional input. (Default value = 1)
K_REV	real	Detachment rate constant for reversible filtration of irreversible colloids.
R_MIN	real	Minimum value of the retardation factor for reversible filtration of irreversible colloids.
R_MAX	real	Maximum value of the retardation factor for reversible filtration of irreversible colloids
SLOPE_KF	real	Value of the slope (m) in the log-log space for the equation: $\log_{10}(CDF) = b + m \cdot \log_{10}(K_f)$
CINT_KF	real	Value of the intercept (b) in the log-log space for the above equation
KEYWORD	character	Keyword specifying whether the colloid species is 'irreversible' or 'reversible'.
FLAG_LOG	integer	For reversible colloids an average retardation factor is used: If FLAG_LOG = 0: a linear average of the distribution is used If FLAG_LOG = 1: a log-linear average of the distribution is used
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. For equivalent continuum solutions, the fracture spacing in the finite spacing model is determined using $SPACING = APERTURE/POROSITY$. For dual permeability models, the fracture spacing input parameter APUV1 in the **dpdp** macro is used as the half-spacing between fractures. 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).

Input Variable	Format	Description
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 $DISTANCE$ is the length of the path between nodes.
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 [abs(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.

Input Variable	Format	Description
ITRC	integer	Model number for parameters defined in group 4. Default is 1.
PCNSK	real	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). 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. 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, abs(PCNSK) is proportional to the number of particles introduced at the node(s). Default is 0 for all unassigned nodes, meaning that no particles are injected at that node.
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.

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

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.

```
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 the second example, transfer function data is used along with the active fracture and colloid diversity models. Particle statistics data for the cumulative number of particles that have left the sytem and the number of particles that left during the current timestep are output.

```
ptrk
100000 244562 244562
0.      1.e20   0.      1.e20
1               0.230 3 6
tcurve
3
../colloid_cell/input/uz_tfcurves_nn_3960.in
afm
diversity
file
../colloid_cell/input/rcoll_equation.dat
13 5
irreversible
-2 0.0 1.e-03  1.e-03  1.e-03  1.00e-30 1.0e+00 0.2 1.e-03   1e+20    1   1.00 0.01   0.6 //layer 1
-2 0.0 1.e-03  1.e-03  1.e-03  1.00e-30 1.0e+00 0.2 0.00e+00 1e+20   1   1.00 0.00   0.0 //layer 2

   1       10       1        1
   11      20       1        2

   1       1        1        -1. 0.      0.01
```

With the file rcoll_equation.dat as:

```
test file for 3Dmptr equal weight sampling
1  1.  1.  1.  0.  0.
3  1.  5.  5.  0.  0.
4  1.  5.  5.  0.  0.
5  1.  1.  9.  1.  -0.9
6  1.  5. 85.  1.16667  -2.25
8  1.  1. 100000.  1.  -5.
```

6.2.72 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.

6.2.73 Control statement rest (optional)

Restart options for selecting flux output, file format, and content.

Group 1 - CHDUM

Input Variable	Format	Description
CHDUM	character*80	Keyword specifying format of restart file to be created and whether fluxes should be output. Keywords are entered one per line and terminated with 'end' or a blank line. Keywords must be entered starting in the 1st column. Valid keywords (case insensitive) are: 'ascii' - both read and write restart files are ascii 'binary' - both read and write restart files are unformatted 'rbinary' - unformatted read restart file 'wbinary' - unformatted write restart file Note: Default file format is ascii if not specified. 'noflux' - no flux output 'flux' - both liquid and vapor flux output 'lflux' - liquid flux output 'vflux' - vapor flux output Note: Default is to not write or read flux data to/from the restart files. 'old' - Use old style restart format (Version 2.3 or older). 'new' - Use new restart file output format (Version 3.0 or newer)

Input Variable	Format	Description
		Note: Default is to use the new style. The older style is maintained to facilitate use of post-processing applications written to use data as formatted in the original files. A description of the file formats is provided in Section 7.2. 'read' - A list of restart variables that should be read for the current simulation follows. : 'write' - A list of restart variables to be output to the restart file follows. Note: Default is to read all variables found in the restart file and to write all variables active in the current simulation. The variables are entered on the same line as the 'read' or 'write' keyword. The following variable keywords may be used: none all tem pres trac ptrk gasp pini saturation co2 mass disp (disx, disy, disz) strs or stre or strs (strx, stry, strz, stxy, stxz, styz)

The following is an example of **rest**. In this example restart data will be written to an unformatted file and liquid flux will be output. If a read restart file is used it will be in ascii format and if liquid flux data is present in the file it will be read.

```
rest
wbinary
lflux
```

6.2.74 Control statement rflo (optional)

No input is associated with this macro. This option is available for single continuum or double porosity/double permeability (dpdp) models only and is used in conjunction with transport models (**trac**, **ptrk, mptr**, or **sptr** macros). When this option is selected flow calculations are not done and it is assumed the flow field is at steady-state.

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 simulation (see **rest** macro) or may be generated by an alternate flow code (see Ho, 1997).

In version 2.30 an additional option was added to the **rflo** macro. If the macro is entered as "**rflor**" the flux values will be negated and a reversed flow field will be used for the transport simulation.

6.2.75 **Control statement rich** (optional)

Invokes Richard's equation solution for unsaturated-saturated flow. A single phase approach that neglects air phase flow and assumes the movement of water is independent of air flow and pressure. Uses variable switching (Pressure, Saturation)

Group 1 - STRD_RICH, TOL_PHASE, PCHNG, SCHNG

Input Variable	Format	Description
STRD_RICH	real	Newton-raphson relaxation factor
TOL_PHASE	real	Tolerance for full saturation
PCHNG	real	Pressure adjustment after variable switch
SCHNG	real	Saturation adjustment after variable switch

The following is an example of rich.

```
rich
   0.95        1.e-5        1.e-3        1.e-3
```

6.2.76 **Control statement rive or well** (optional)

River or implicit well package.

Group 1 KEYWORD

KEYWORD *"wellmodel"*

NRIVER, IRIVER

If IRIVER = 1

INRIVERF, ISRIVERF, IFRIVERF,IWSP

KEYWORD *"end macro"* (required)

The input is terminated with keyword "*end rive*", "*endrive*", "*end well*" or "*endwell*", where the macro name should match the input macro name that was used.

Input Variable	Format	Description
KEYWORD	character	
KEYWORD "*wellmodel*"		
NRIVER	integer	Number of models defined for this call.

IRIVER	integer	Type of surface/well flow IRIVER = 0 no routing, ponding, groundwater connection IRIVER = 1 simple fluid routing, no groundwater connection IRIVER = 2 simple stream definition, no groundwater connection IRIVER = 3 simple stream definition, with groundwater connection IRIVER = 4 simple stream definition, with groundwater connection (with ponding)
INRIVERF		Section number (id) of ith section.
ISRIVERF		Number of layers for the ith section.
IFRIVERF		Number of coordinate points in the ith section.
IWSP		
KEYWORD "*endwell*"		

6.2.77 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 33)

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 numbers 4, 6, and 7 (the combined van Genuchten model), the permeability is isotropic and overwrites the input from macro **perm**. Macro **fper** can be used with models 4, 6, and 7 to introduce anisotropy.

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.
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.

Input Variable	Format	Description
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 $\frac{\partial}{\partial S} Pcap = 0$ and $\frac{\partial^2}{\partial S} Pcap = 0$ are forced at $S = 0$. If RP5 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet Pcap(S_{cutoff})$ is forced at $S = 0$.
RP6	real	Cutoff saturation used in fits described for RP5, must be greater than RP1.
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 [rlp(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.

Input Variable	Format	Description
RP4	real	Power n in van Genuchten formula, matrix rock material.
RP5	real	Low saturation fitting parameter, matrix rock material, 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 $\frac{\partial}{\partial S}Pcap = 0$ and $\frac{\partial^2}{\partial S}Pcap = 0$ are forced at $S = 0$. If RP5 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP5 \bullet Pcap(S_{cutoff})$ is forced at $S = 0$.
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	Low saturation fitting parameter, fracture material, multiple of cutoff capillary pressure assigned as maximum capillary pressure. 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. If RP11 = 0, a cubic fit is used. The slope at the cutoff saturation is matched and the conditions $\frac{\partial}{\partial S}Pcap = 0$ and $\frac{\partial^2}{\partial S}Pcap = 0$ are forced at $S = 0$. If RP11 > 0, a multiple of the value of the capillary pressure at the cutoff saturation, $RP11 \bullet Pcap(S_{cutoff})$ is forced at $S = 0$.
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 $RP13 \times RP15$. Can be made anisotropic with macro FPER.
RP14	real	Matrix rock saturated permeability (m^2). Can be made anisotropic with macro FPER.
RP15	real	Fracture volume fraction. Is equal to the fracture aperture divided by the fracture spacing (with same units). Sometimes called fracture porosity.

Input Variable	Format	Description
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 [rlp(S)] rather than capillary pressure.		
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 [rlp(S)] rather than capillary pressure.		
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.		
RP16	real	Fracture-matrix interaction term. If RP16 ≤ 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 $sl**RP16$ and $(1.-sl)**RP16$ is applied to the fracture-matrix interaction terms for the liquid and vapor phases, respectively, for dual permeability problems. Here, sl is the value of saturation at the given node.
Model 10: IRLP(i) = 10, linear relative permeability with minimum relative permeability values, linear capillary pressure (8 parameters required).		
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
```

6.2.78 **Control statement rlpm** (optional)

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

Group 1 - KEYWORD “group”, GROUP_NUMBER

The Group 1 KEYWORD “group”, which starts each model sequence.

Group 2 - PHASE, MODEL_TYPE, RLP_PARAM(i), i = 1, NUMP

An entry is made for each active phase in the model.

Group 3 - KEYWORD “cap”, COUPLING, MODEL_TYPE, CAP_PARAM(i), i = 1, NUMP

An entry is made for each phase-coupling in the model.

NUMP is the number of parameters needed for the selected model type (see parameter table).

Groups 1 to 3 are entered for each relative permeability and capillary pressure model being defined.

Alternatively, to enter relative permeability data in a table.

Group 2 - KEYWORD “table”, TBLNUM, NPARAMS, PHASE1, COUPLING

Group 3 - SATURATION, PHASE1 RELPERM, CAPILLARY PRESSURE

-or-

Group 2 - KEYWORD “table”, TBLNUM, NPARAMS, PHASE1, PHASE2, COUPLING

Group 3 - SATURATION, PHASE1 RELPERM, PHASE2 RELPERM, CAPILLARY PRESSURE

Group 3 is entered multiple times to cover the full range of wetting phase saturation (0. - 1.0).

Table input is terminated with KEYWORD ‘end’ or a blank line. Note input saturation is the saturation of the wetting phase, and PHASE1 is the wetting phase. nparams is the number of parameters found in the table (3 or 4).

-or-

Group 3 - KEYWORD “file”

TABLE_FILE

Table data will be read from the specified file.

Model input is terminated with KEYWORD ‘end’ or a blank line.

Group 4 -JA, JB, JC, GROUP_NUMBER (JA, JB, JC-defined on page 33)

Model	Input Parameter							
Relative permeability	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Linear	S_r	S_{max}						
Exponential	S_r	S_{max}	k					
Corey	S_{rl}	S_{rg}						
Brooks-Corey	S_{rl}	S_{rg}	λ					
Van Genuchten	S_{rl}	S_{max}	α_G	N				
Van Genuchten Fracture	S_{rl}	S_{max}	α_G	N	k_f	k_m	V_f	fmit
Capillary pressure								
Linear	Cp	S						
Brooks-Corey	S_{rl}	S_{rg}	λ	Cpe	Cut1	S_{cut}		
Van Genuchten	S_{rl}	S_{max}	α_G	N	Cut1	S_{cut}		

Parameter	Format	Description
S	real	Saturation at which capillary pressure goes to zero.
S_r	real	Residual (or minimum) saturation.
S_{rl}	real	Residual liquid saturation.
S_{rg}	real	Residual gas saturation.
S_{max}	real	Maximum saturation.
k	real	Exponent for exponential model.
λ	real	Exponent in Brooks-Corey model.
α	real	Inverse of air entry head, α_G (1/m)
N	real	Power ‘n’ in van Genuchten formula.
Cp	real	Capillary pressure at zero saturation (MPa)
Cpe	real	Capillary entry pressure (MPa)

Parameter	Format	Description
Cut1	real	Low saturation fitting parameter.
S_{cut}	real	Cutoff saturation
k_f	real	Fracture permeability (m^2)
k_m	real	Matrix permeability (m^2)
V_f	real	Fracture volume fraction
fmit	real	Fracture-matrix interaction term. (Can be omitted if not used)

Input Variable	Format	Description
KEYWORD	character	Keyword 'group' specifying the start of a relative permeability/ capillary pressure model
GROUP_NUMBER	integer	Identifier for the current model.
PHASE	character	Fluid state in the current model: 1 - water or h2o_liquid 2 - air 3 - co2_liquid or co2_sc 4 - co2_gas 5 - vapor or h2o_gas 6 - methane_hydrate 7 - oil 8 - gas
MODEL_TYPE	character	Relative permeability model type: 1 - constant 2 - linear (rlp) or linear_for (cap) 3 - exponential 4 - corey [Corey] 5 - brooks-corey [Brooks-Corey] 6 - vg [Van Genuchten (function of saturation)] 7 - vg_cap [Van Genuchten (function of capillary pressure)] 8 - same [Indicates that the model being used is the same as the last model entered with identical input parameters. Applies only to the current group. For the case of a Brooks-Corey or Van Genuchten capillary pressure model, this can be used for the first entry where the initial parameters are the same for both the rlp and cap models. In that case only the additional capillary pressure parameters need to be input.]
RLP_PARAM	real	Input parameters for the specified relative permeability model.

Input Variable	Format	Description
KEYWORD	character	Keyword ‘cap’ designating that associated capillary pressure model parameters are being input.
COUPLING	character	Phase coupling for the current capillary pressure model: 1 - air/water 2 - water/co2_liquid 3 - water/co2_gas 4 - co2_liquid/co2_gas 5 - water/vapor 6 - air/vapor
CAP_PARAM	real	Input parameters for the specified capillary pressure model.
KEYWORD	character	Keyword ‘table’ designating that relative perameability and capillary pressure will be input in a table.
TBLNUM	integer	Table identifier.
NPARAMS	integer	Number of parameters defined in the table (including saturation).
PHASE1	character	Fluid state of the wetting phase (see PHASE for phase ID)
PHASE2	character	Fluid state of the non-wetting phase (see PHASE for phase ID)

Examples

```
rlpm
group     1
water     corey     0.3       0.1
vapor     same
end
1         140       1         1
```

```
rlpm
group     1
water     linear    0.3       1.0
air       linear    0.3       1.0
cap       air/water linear_for 93.6       100.
end
1         0         0         1
```

```
rlpm
group     10
water     vg_cap    0.0001    1.0       3.0       3.0
air       same
air/water vg_cap    0.0001    1.0       3.0       3.0       2.        0.05
end
1         0         0         10
```

```
rlpm
group     1
water     vg_cap    0.0212    1.0       0.00715   1.62
fracture  0.03      1.0       12.05     3.00      4.06e-09  2.04e-18  2.93e-04  0.
air       same
cap       air/water vg_cap    0.0212    1.0       0.00715   1.62      2.        0.0312
fracture  0.03      1.0       12.05     3.00      20.0      0.0001
group     2
water     vg_cap    0.154     1.0       0.371     2.37
fracture  0.03      1.0       13.72     3.00      7.14e-09  2.51e-18  9.27e-05  0.
air       same
cap       air/water vg_cap    0.154     1.0       0.371     2.37      2.        0.164
fracture  0.03      1.0       13.72     3.00      20.0      0.0001
group     3
water     vg_cap    0.0453    1.0       0.0133    1.80
fracture  0.03      1.0       11.96     3.00      4.57e-09  2.09e-18  2.43e-04  0.
air       same
cap       air/water vg_cap    0.0453    1.0       0.0133    1.80      2.        0.0553
fracture  0.03      1.0       11.96     3.00      20.0      0.0001
group     4
water     vg_cap    0.0968    1.0       0.0273    2.46
fracture  0.03      1.0       11.96     3.00      6.53e-09  1.10e-16  1.11e-04  0.
air       same
cap       air/water vg_cap    0.0968    1.0       0.0273    2.46      2.        0.1068
fracture  0.03      1.0       11.96     3.00      20.0      0.0001
end
-1        0         0         1
-2        0         0         2
-3        0         0         3
-4        0         0         4
```

```
rlpm
group                1
water                corey          0.3          0.1
vapor                same
group                2
table                1              4            water        vapor        water/vapor
file
input/doe_rlpm.table
end
1                    140            1            2
```

For example, the file doe_rlpm.table would contain some header rows followed by saturation versus relative permeability and capillary pressure.

```
FEHM V3.00pgi64 10-10-20 QA:NA  10/20/2010  14:23:18
*** DOE Code Comparison Project, Problem 5, Case A ***
Relative permeability and Capillary pressure
"Saturation" "Liquid" "Vapor" "Capillary pressure"
0.00000000          0.00000000          1.00000000          0.00000000
0.500000000E-01     0.00000000          1.00000000          0.00000000
0.100000000         0.00000000          1.00000000          0.00000000
0.150000000         0.00000000          1.00000000          0.00000000
0.200000000         0.00000000          1.00000000          0.00000000
0.250000000         0.00000000          1.00000000          0.00000000
0.300000000         0.732682696E-64     1.00000000          0.00000000
0.350000000         0.482253086E-04     0.834442515         0.00000000
0.400000000         0.771604938E-03     0.675154321         0.00000000
0.450000000         0.390625000E-02     0.527343750         0.00000000
0.500000000         0.123456790E-01     0.395061728         0.00000000
0.550000000         0.301408179E-01     0.281201775         0.00000000
0.600000000         0.625000000E-01     0.187500000         0.00000000
0.650000000         0.115788966         0.114535108         0.00000000
0.700000000         0.197530864         0.617283951E-01     0.00000000
0.750000000         0.316406250         0.273437500E-01     0.00000000
0.800000000         0.482253086         0.848765432E-02     0.00000000
0.850000000         0.706066744         0.110918210E-02     0.00000000
0.900000000         1.00000000          0.00000000          0.00000000
0.950000000         1.00000000          0.00000000          0.00000000
1.00000000          1.00000000          0.00000000          0.00000000
```

6.2.79 **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 33)

Input Variable	Format	Description
DENRD	real	Rock density (kg/m^3).
CPRD	real	Rock specific heat $\left(\frac{MJ}{kg \cdot K}\right)$. If CPRD > 1 the code will assume the units are $\left(\frac{J}{kg \cdot K}\right)$ 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/m^3, a rock specific heat of 1010. J/(kg K) and a porosity of 0.35.

```
rock
  1      140      1      2563.     1010.     0.35
```

6.2.80 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, IMMPRT (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 - IDRXN

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

IDRXN = 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

IDRXN = 2: Langmuir kinetic reaction

Group 14 - IAQUEOUS, IIMMOBILE

Group 15 - DISTCOEEF

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

IDRXN = 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.

Input Variable	Format	Description
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*20	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.
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*20	For each aqueous complex, the name of the aqueous complex (e.g. H_2SO_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*20	For each immobile component, the name of the immobile component (e.g. Calcite, Co[adsorbed])

Input Variable	Format	Description
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	Fore each vapor component, the number identifying each vapor component (e.g. 1, 2, etc.)
VAPNAM	character*20	For each vapor component, the name of the vapor component (e.g. CO2[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) 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 $1x10^{-9}$ 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 143) 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 (°C)
LKEQ	real	Equilibrium constants for K function

Input Variable	Format	Description
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
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.

Input Variable	Format	Description
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
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. $[(\text{concentration units})^{p} \times s]^{-1}$, 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. $[(\text{concentration units})^{p} \times s]^{-1}$, 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		

Input Variable	Format	Description
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
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.

Input Variable	Format	Description
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)
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.

Input Variable	Format	Description
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.
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 192). In addition, the "Reactive Transport Example" (Section 9.5) and the Validation Test Plan for the FEHM Application Version 2.30 (10086-VTP-2.30-00) 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.

```
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]   **
2              FeEDTA[s]   0
3              Cobalt[s]   0
** IDVAP      VAPNAM      0                                           Group 6
                           VAPPRT
```

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

$$\sum_{j=1}^{N_c} a_{ij}\hat{C}_j \Leftrightarrow \hat{X}_i \qquad i = 1, \ldots, N_x \tag{1}$$

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 $\hat{N}_c$ is the number of aqueous components. Here is a simple example of an equilibrium speciation reaction.

$$aA + bB \Leftrightarrow A_aB_b \tag{2}$$

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:

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

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:

$$U_A = [A] + a[A_a B_b] \tag{4}$$

$$U_b = [B] + b[A_a B_b] \tag{5}$$

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

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, IDRXN = 1, is specified as occurring at each node in the problem (JA = 1, JB and JC = 0).

```
** IDRXN **                                                    Group 12
1
** JA          JB          JC **                               Group 13
1             0           0
```

IDRXN = 1: Linear Kinetic Reaction. The retardation of contaminants due to adsorption/desorption can be modeled with a linear kinetic sorption/desorption expression. The rate of adsorption/desorption of component *j* is given by:

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

where c_j denotes the uncomplexed aqueous concentration of *j* **{IAQUEOUS}**, m_j denotes the adsorbed concentration of species *j* **{IIMMOBILE}**, k_m is the mass transfer coefficient **{RATE}**, and K_D is the distribution coefficient **{KD}**. As $k_m \to \infty$, 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.

```
** IDRXN **                                                    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
```

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

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

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 $z_m \to \infty$, 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.

```
** IDRXN **                                                    Group 12
2
** JA          JB          JC **                               Group 13
1             0           0
** IAQ         IMMOBILE    **                                  Group 14
101           1
** DISTCO                                                      Group 15
**
2.0e5                                                         Group 16
** RATE **
0.05                                                          Group 17
** MAXCO
**
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:

$$\sum_{i=1}^{N_c+N_x+N_m+N_v} \eta_j \hat{z}_j \Leftrightarrow \sum_{k=1}^{N_c+N_x+N_m+N_V} \eta_k' \hat{z}_k \tag{8}$$

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:

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

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.

** IDRXN **				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		

IDRXN = 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:

$$\text{Substrate} + \text{Electron Acceptor} + H^+ + \text{Nutrients} \rightarrow \text{cells} + CO_2 + H_2O + NH_3$$

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

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

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.

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

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 NAQUEOUS = 2. In addition Group 29 data is left out since NBIOFRM = 0.

```
** IDRXN **                                                    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               **                                         Group 28
** NBIOFRM
0                                                             Group 29
** ICBIO **
```

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

$$A \rightarrow B \tag{12}$$

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.

```
** IDRXN **                                                    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 {IIMMOBILE} can be written in the following form:

$$m_p \Leftrightarrow \mu_{p1} c_1 + \mu_{p1} c_2 + \ldots + \mu_{p1} c_{N_c} \tag{13}$$

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:

$$Q_p = \prod_{i=1}^{N_c} c_j^{\mu_{pj}} \tag{14}$$

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

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

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 192). 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
1
** IDCPNT        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]           **
** IDVAP         VAPNAM          0                                           Group 6
                                 VAPPRT
** 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
**              **                                                          Group 14
IIMMOBLE
1                                                                          Group 15
** NAQSP **
1                                                                          Group 16
** IAQ **
1               **                                                         Group 17
** IMSTOIC
1               **                                                         Group 18
** AQSTOIC
1               **                                                         Group 19
** SOLPROD
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 37, Equation (79)]. For example, for a

species such as Cobalt from the multisolute problem (see Section 9.5, “Reactive Transport Example,” on page 257), the free ion concentration is simply the concentration of Cobalt in it’s uncomplexed state. The total Cobalt would be Free Ion Cobalt + all Cobalt Complexes (e.g. CoEDTA from the multisolute verification problem). Using IDRXN = 7 allows the total Cobalt to dissolve, while IDRXN= 8 allows only the free ion Cobalt to dissolve.

6.2.81 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
```

6.2.82 Control statement **sptr** (optional)

Streamline particle tracking is invoked with this control statement.

Group 1 - DTMX, IPRT, IPRTO, RSEED, TPLIM, DISTLIM, LINELIM, DIVD_WEIGHT

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

Keywords and their associated input are described below. These keywords may be entered in any order following Group 2, but must be directly followed by any associated data. All keywords are optional except “tprp” which flags input of the particle transport properties.

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 “tcurve”

NUMPARAMS

TFILENAME

Optional keyword “omr” is input to indicate the grid has octree mesh refinement. It is followed by the number of refined (OMR) nodes in the grid. An optional keyword, “file”, which if input is followed by the name of a file from which to read or to write omr initialization calculation arrays, and the file format. The “file” option should only be used for particle tracking runs using steady state flow. It should also be noted that

the file should be re-generated if parameter changes that affect velocity are made since velocities will not be recalculated if the file exists.

KEYWORD “omr”

OMR_NODES

KEYWORD “file”

OMRFILENAME

OMR_FORM

Optional keyword “tpor” is input to indicate tracer porosity will be input and is followed by tracer porosity values input using JA, JB, JC format or the keyword “file” and the name of a file containing the tracer porosity values.

KEYWORD “tpor”

JA, JB, JC, PS_TRAC (JA, JB, JC - defined on page 33)

-or-

KEYWORD “tpor”

KEYWORD “file”

TPORFILENAME

Optional keyword “wtdt” to indicate initial particle locations should be moved below the water table by distance DELTAWT.

KEYWORD “wtdt”, DELTAWT

Optional keyword “volum” to indicate control volumes associated with computation of sptr velocities should be written to an output file. These volumes are used with PLUMECALC to account for the approximate control volumes used for the velocity calculations on an OMR grid. The “volume” keyword may be followed by the file format.

KEYWORD “volum”

-or-

KEYWORD “volum”, SPTRX_FORMAT

Optional keywords “po”, “sa”, “pe”, “de”, “pr”, “te”, “zo”, “id” define parameters to be output. No data is associated with the parameter output flags. These parameters are output in the ‘*.sptr2” file in the order entered.

Optional keyword “xyz” indicates that coordinate data should be included in the abbreviated ‘*.sptr2’ output file (see IPRTO below).

The optional keyword “zbtc” indicates that breakthrough curves will be computed for specified zones. It is followed by NZBTC . . . and ZBTC. If keyword “alt” follows the “zbtc” keyword an alternate output format will be used where,

KEYWORD “zbtc”

-or-

KEYWORD “zbtc” “alt”

NZBTC, DTMN, PART_MULT, PART_FRAC, DELTA_PART, PART_STEPS, TIME_BTC

ZBTC

Optional keyword “cliff”

Optional keyword “corner”

Optional keyword “capture”

Optional keyword “spring”

Keyword (‘tprp’) specifies that the particle transport properties will be entered on subsequent lines. The transport properties input (Group 4) depends 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).

KEYWORD “tprp”

TPRP_FLAG = 1:

Group 3 - TPRP_FLAG, KD

TPRP_FLAG = 2:

ITENSOR = 1

Group 3 - TPRP_FLAG, KD, DM, A1, A2, A3, A4, ASX, ASY, VRATIO

ITENSOR = 2

Group 3 - TPRP_FLAG, KD, DM, AL, ATH, ATV, VRATIO

ITENSOR = 3

Group 3 - TPRP_FLAG, KD, DM, ALH, ALV, ATH, ATV, VRATIO

ITENSOR = 4

Group 3 - TPRP_FLAG, KD, DM, AL, AT, VRATIO

ITENSOR = 5

Group 3 - TPRP_FLAG, KD, AL, ATH, ATV, DM, VRATIO

TPRP_FLAG = 3:

Group 3 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX, APERTURE

TPRP_FLAG = 4:

ITENSOR = 1

Group 3 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX, APERTURE, DM, A1, A2, A3, A4, ASX, ASY, VRATIO

ITENSOR = 2

Group 3 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX, APERTURE, DM, AL, ATH, ATV, VRATIO

ITENSOR = 3

Group 3 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX, APERTURE, DM, ALH, ALV, ATH, ATV, VRATIO

ITENSOR = 4

Group 3 - 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 3 - TPRP_FLAG, KD, DIFM, RD_FRAC, POR_MATRIX, APERTURE, AL, ATH, ATV, DM, VRATIO

TPRP_FLAG =11 or TPRP_FLAG = 12:

Group 3- TPRP_FLAG, SIMNUM

KEYWORD 'file'

CDF_FILENAME

TPRP_FLAG =13 or TPRP_FLAG = 14:

Group 3 - TPRP_FLAG, SIMNUM

KEYWORD 'file'

CDF_FILENAME

or

Group 3 - TPRP_FLAG, K_REV, R_MIN, R_MAX, SLOPE_KF, CINT_KF, AL, ATH, ATV, DM, VRATIO

Group 4 - JA, JB, JC, MODEL_NUMBER

Group 3 is ended when a blank line is encountered. The *MODEL_NUMBER* is incremented each time a Group 3 line is read, and Group 4 lines refer to this parameter.

Group 5 - ITM, IST

-or-

Group 5 - ITM, IST, COUNT_STEPS_MAX, SPTR_FLAG

Group 6 - NX, NY, NZ

Group 7 - X10, Y10, Z10

Group 8 - XDIM, YDIM, ZDIM

Group 9 - IJKV(I), X1(I), Y1(I), Z1(I) for I = 1 to NUMPART

Group 9 input is terminated with a blank line.

-or-

Group 9 - KEYWORD "file"

SPTR_FILENAME

Restart runs for particle tracking may be accomplished by reading particle starting locations from a file. A particle restart file is generated by adding the optional SPTR_FLAG, keyword "save" to group 5.

Note when IST = 0 or 1, Group 10 is used and place holders are inserted for Groups 7-9 and NUMPART is equal to the number of particle starting locations that are entered;

however, when IST = 2, Group 10 is not implemented and Groups 7-9 are used followed by a blank line. NUMPART equals NX*NY*NZ in this case.

Input Variable	Format	Description
DTMX	real	Time step control (seconds). FEHM will account for all particles every 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 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 abs(IPRTO) = 1,output is formatted, abs(IPRTO) = 2 output is unformatted, and 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 466201.
TPLIM	real	Minimum amount of time (days) a particle should move before location is output. Default is 0 days.
DISTLIM	real	Minimum distance (m) a particle should move before location is output. Default is 0 meters.
LINELIM	integer	Maximum number of lines that will be written to sptr2 file.
DIVD_WEIGHT	real	Weight factor for the derivative of the dispersion tensor term. Default is 1. If a value of zero is entered, the derivative term is not used.
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.

Input Variable	Format	Description
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 Š 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. (2002) ITENSOR = 2, Axisymmetric form of the dispersion coefficient tensor of Burnett and Frind (1987) ITENSOR = 3, Modified form of the dispersion coefficient tensor of Burnett and Frind (1987). See Lichtner et al. (2002) 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 (1987) 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). 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.

Input Variable	Format	Description
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.
MAX_JUMP	integer	When using random walk, the maximum number of cells a particle is allowed to jump in a single step. (Default is 10).
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	Optional keyword “omr” to indicate the grid has octree mesh refinement. If the keyword is found then OMR_NODES is entered, and optionally keyword “file” with OMRFILENAME and OMR_FORM, otherwise they are omitted.
OMR_NODES	integer	Number of refined (omr) nodes in the grid.
KEYWORD	character	Optional keyword “file” indicating that the omr initialization calculation arrays should be written to or read from a file. This option should only be used with steday-state flow.
OMRFILENAME	character	Name of file from which to read or to write omr arrays.
OMR_FORM	character	Format of the omr file, formatted or unformatted.
KEYWORD	character	Optional keyword “tpor” to indicate tracer porosities should be read.
PS_TRAC	real	Tracer porosity
KEYWORD	character	Optional keyword “file” indicating that the tracer porosities should be read from a file.
TPORFILENAME	character	Name of file from which to read tracer porosity.
KEYWORD	character	Optional keyword “wtdt”
DELTAWT	real	Distance below the water table that particles should be started.
KEYWORD	character	Optional keyword “volum” to indicate control volumes should be output. Output is written to the “.sptrx” file.
SPTRX_FORMAT	character	File format for control volume output file “formatted” or “unformatted”. Default is formatted.

Input Variable	Format	Description
KEYWORD	character	Optional keywords “po” (porosity), “sa” (saturation), “pe” (permeability), “de” (density), “pr” (pressure), “te” (temperature), “zo” (zone number), “id” (particle identifier) 1 per line, indicating which parameters will be output along the particle path (written to “.sptr2” file). If no keywords are present no parameter ddata will be output. Note that in older versions of FEHM porosity and saturation were output if no keywords were entered.
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 11 - Colloid diversity model with importance sampling, CDF vs Retardation Factor specified as a table in the optional file specified by CDF_FILENAME 12- Similar to case 11 above , except the SQRT(CDF) is used instead of CDF for importance sampling 13- Colloid diversity model with importance sampling, CDF vs K_f (attachment rate constant) specified as a straight line equation in the log-log space either on this line or in the optional file specified by CDF_FILENAME 14- Similar to case 13 above, except the SQRT(CDF) is used instead of CDF for importance sampling
SIMNUM	integer	Simulation number, used for selecting the table/equation from the colloid diversity file.
KEYWORD	character*4	Optional keyword ‘file’ designating the cumulative probability distribution function (CDF) retardation parameters for the colloid diversity model should be read from an external file
CDF_FILENAME	character*80	Name of the file containing the cumulative probability distribution function (CDF) (entered if optional keyword ‘file’ follows keyword ‘dive’). See Section 6.2.59 for file formats. If TPRPFLAG = 11 or 12, Table option If TPRPFLAG = 13 or 14, Equation option. The following equations are used for $R_{min} \le R \le R_{max}$, $R = 1 + K_f/K_{rev}$, $\log_{10}(CDF) = b + m \cdot \log_{10}(K_f)$
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)

Input Variable	Format	Description
APERTURE	real	Fracture aperture (m)
AL	real	Longitudinal dispersivity, α_L (m). ITENSOR = 2, 4, or 5
ALH	real	Horizontal longitudinal dispersivity, α_{LH} (m). ITENSOR = 3
ALV	real	Vertical longitudinal dispersivity, α_{LT} (m). ITENSOR = 3
AT	real	Transverse dispersivity, α_T (m). ITENSOR = 4
ATH	real	Transverse horizontal dispersivity, α_{TH} (m). ITENSOR = 2, 3, or 5
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. (2002)
A2	real	Generalized dispersivity term α_2 (m) from Lichtner et al. (2002)
A3	real	Generalized dispersivity term α_3 (m) from Lichtner et al. (2002)
A4	real	Generalized dispersivity term α_4 (m) from Lichtner et al. (2002)
ASX	real	Direction cosine of the axis of symmetry from Lichtner et al. (2002)
ASY	real	Direction cosine of the axis of symmetry from Lichtner et al. (2002)
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. Note that DTMN, PART_MULT, PART_FRAC, DELTA_PART, PART_STEPS, and TIME_BTC are optional input. They must be entered in the order given. When not entered the default values will be used.
DTMN	real	Time step control (seconds). FEHM will account for all particles at time step intervals starting with dtmn seconds and write information to the ".sptr3" file if the "zbtc" keyword is present. This controls the output density for breakthrough curve information only. (Default DTMX)

Input Variable	Format	Description
PART_MULT	real	Time step multiplication factor. (Default 2.)
PART_FRAC	real	Fraction of particles that should break through before checking if time step should be increased. (Default 0.1*NUMPART)
DELTA_PART	real	Fraction of particles that should break through during a time step so that the time step is not increased
PART_STEPS	integer	Number of time steps that should be checked for DELTA_PART before increasing the time step
TIME_BTC	real	Time to start using small breakthrough time steps (DTMN) for late initial breakthrough (days).
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)
COUNT_STEPS_MAX	integer	Maximim number of steps a particle is allowed to take in a sptr run. (Default 1000000) Input of this value is optional. If this value is omitted, the default will be used. The value must precede SPTR_FLAG if being used.
SPTR_FLAG	character	Optional keyword "save" to signal that final particle locations and times should be written to a file, *.sptrs, for a particle restart run.
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})
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
KEYWORD	character	Optional keyword "file" indicating that the particle starting locations should be read from a file. If this file has been generated by the code using the "save" keyword in Group 6, particle starting times will also be read.
SPTRFILENAME	character	Name of file from which to read initial particle locations.

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                                    Group1
    0.25             0              5          0                                    Group 2
tprp
    2            0.0223715          100.       0.        0.        0.       0.     Group 3

    1                0              0          1                                    Group 4

zbtc
    1
    5
 1000                2                                                            Group 5
    1              100            1000                                             Group 6
    0.           -1500.              0.                                           Group 7
   10.            3000.           -12.5.                                          Group 8
                                                                                   Group 9
```

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.

```
sptr
    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

    1                0              0          1                                    Group 4

 1000                2                                                            Group 5
    1              100            1000                                             Group 6
    0.           -1500.              0.                                           Group 7
   10.            3000.           -12.5.                                          Group 8
                                                                                   Group 9
```

The third example uses the colloid diversity model with importance sampling specified as an equation using an external input file, using the third set of parameters in the rcoll_eqn.dat, with the file rcoll_eqn.dat as:

```
Colloid diversity model equation parameters
 1 1.5641426E-5 1.0 63933.785 0.7081742 0.0E+0 100. 10. 0.1 5.e-12 0.1
 2 1.1755084E-3 1.0 851.69573 0.7676392 0.0E+0 100. 10. 0.1 5.e-12 0.1
 3 1.0417102E-5 1.0 95996.984 0.7438557 0.0E+0 100. 10. 0.1 5.e-12 0.1
.
.
.
 100 2.0808208E-4 1.0 4806.7954 0.62846046 0.0E+0 100. 10. 0.1 5.e-12 0.1
```

```
sptr
1.728e8         0       0       0                       Group1
0.25            0       5       0                       Group 2
tprp
13              3                                       Group 3
rcoll_eqn.dat

1               0       0       1                       Group 4

zbtc
1
5
1000            2                                       Group 5
1               100     1000                            Group 6
0.              -1500.  0.                              Group 7
10.             3000.   -12.5.                          Group 8
                                                        Group 9
```

The fourth example uses the colloid diversity model with importance sampling specified as an equation with parameters input in the sptr macro.

sptr							
1.728e8	0	0	0				Group1
0.25	0	5	0				Group 2
tprp							
13	1.e-5	1.e-9	1.	5	1.e-9	200.	Group 3
(continued from previous line)			1.	0.01	5.e-12	0.01	
1	0	0	1				Group 4
zbtc							
1							
5							
1000	2						Group 5
1	100	1000					Group 6
0.	-1500.	0.					Group 7
10.	3000.	-12.5.					Group 8
							Group 9

6.2.83 Control statement stea (optional)

The macro **stea** is used to manage steady state simulations and is available with all physics modules in FEHM. The macro directs FEHM to monitor changes in variables from timestep to timestep and stop the steady state run when the changes are less than some prescribed tolerance. Alternatively the steady state run is directed to finish when the global “in” and “out” fluxes are less than a prescribed tolerance or the simulated time exceeds the input value.

After the steady state portion of the simulation is completed, a transient run may be performed. This is accomplished with the **boun** macro and the key word “tran”. See the description of the **boun** macro for details.

The user should be aware that when the “stea” macro is used, the parameters associated with the “time” macro pertain to the transient portion of the simulation if a transient part exists. Values for these parameters may be input using a keyword but if not entered will default to the values specified for the time macro.

Group 1 - KEYWORD, VALUE

Input Variable	Format	Description
KEYWORD	character	The following keywords are used with steady to specify the variables to be checked for steady state: shea - Head (m) spre - Pressure (MPa) stem - Temperature (°C) ssat - Saturation sair - Partial pressure of air/gas (MPa) sflu - Mass flux (kg/s) sent - Enthalpy (MJ/s) stim - Maximum time for steady state simulation (days) sday - Initial time step size for steady state simulation (days) smul - Time step multiplication factor smst - Minimum number of time steps to be used for steady state simulation snst - Maximum number of time steps to be used for steady state simulation shtl - Option to reduce the head_tol factor as the solution approaches steady-state stmc - Option to reduce the machine tolerancs factor (tmch) factor as the solution approaches steady-state sacc - Maximum change allowed in the accumulation term when flux is being checked sper - The tolerance is interpreted as a fractional change in the variable being checked [i.e., (new_value - old_value)/old_value]. Without this keyword it is an absolute change in the variable value. endstea- Signifies end of keyword input, a blank line will also work.
VALUE	real	Variable tolerance or time control parameter value.

In the following example a steady state solution is specified. The tolerance for head is specified to be 0.1 m and for flux 0.00001kg/s. The steady state solution will be allowed to run for a maximum of 1.e12 days and the time step multiplier is set to 2.

```
stea
shead       1.d-1
stime       1.e12
smult       2.
sflux       1.d-5
end
```

6.2.84 **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.

6.2.85 **Control statement strs** (optional)

The solid mechanical deformation module is invoked with this control statement.

Group 1- ISTRS, IHMS

Group 2- KEYWORD

The remaining input is entered in subgroups defined by additional keywords. These keywords are all optional unless otherwise noted, but the user should be careful to ensure the problem is completely defined with the keywords selected.

Input associated with KEYWORDS is shown below and described in more detail in the following table. Unless otherwise specified, blank lines are not permitted.

KEYWORD *"excess_she"*

FRICTION_OUT, STRENGTH_OUT, PP_FAC_OUT

KEYWORD *"permmodel"*

IPERM, SPMF1, SPMF2, . . . SPMF13

(as many models as needed,one per line, terminated by a blank line)

JA, JB, JC, MODEL_NUMBER (JA, JB, JC-defined on page 33)

KEYWORD *"elastic*"

JA, JB, JC, ELASTIC_MOD, POISSON

KEYWORD *"nonlinear"*

NONLIN_MODEL_FLAG

If the value of NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young's modulus and Poisson's ratio:

E_INI, DEDT, POISSON_INI, DNUEDT

Else, if the value of NONLIN_MODEL_FLAG = 91then a table lookup is used:

YOUNG_TEMP_FILE

KEYWORD *"plastic"*

NUMPLASTICMODELS

The following are repeated NUMPLASTICMODELS times

PLASTICMODEL, MODULUS, NU, [PLASTICPARAM1, PLASTICPARAM2]

JA, JB, JC, MODELNUMBER

KEYWORD *"biot*"

JA, JB, JC, ALPHA, PP_FAC

KEYWORD *"stressboun"*

SUB-KEYWORD '*distributed*' or '*lithostatic*' (optional)

or

SUB-KEYWORD '*lithograd*' SDEPTH GDEPTH (optional)

JA, JB, JC, BOUNVAL, KQ

KEYWORD *"tolerance* (required)

STRESS_TOL

KEYWORD *"end strs"* (required)

The input is terminated with keyword "*end strs*" or "*endstrs*".

Input Variable	Format	Default	Description
ISTRS	integer	0	State of stress ISTRS = 0 - skip stress solution ISTRS = 1 - plain strain and 3-D solution ISTRS = 2 - plain stress solution (must be 2-D)
IHMS	integer		Identify the amount and frequency of coupling between TH and M parts of the code IHMS = -1 - stress solved only at the end of the TH (flow) simulation IHMS = -2 - stress solved at the beginning and end of the TH (flow) simulation (useful for establishing a lithostatic load IHMS = -3 - stress solved after each timestep of the TH (flow) simulation IHMS = -4 - stress solved after a timestep of the TH (flow) simulation as determined automatically by the code (not fully implemented) IHMS = 1 - stress solved fully coupled with the TH (flow) simulation IHMS = 2 - stress solved sequentially coupled with the TH (flow) simulation
KEYWORD *"end strs"* or *"endstrs"*			End of **strs** input.
KEYWORD *"initcalc"*			Initiate an initial stress calculation that is useful for establishing lithostatic stress.
KEYWORD *"bodyforce"*			Sets a body force if gravity is non zero. Force is calculated using the rock density information provided in the ***rock*** macro.
KEYWORD *"reldisp"*			Use relative displacement in the calculation of volume strains, permeability models, and output.
KEYWORD *"stresspor"*			Explicitly update the porosity after each time step.
KEYWORD *"fem"*			Use the Finite Element modules for forming displacement equations, and calculating stresses. Although optional, it is strongly recommended that this keyword be included.
KEYWORD *"principal"*			For stress output to the files generated by the **cont** macro, output the principal stress values and the orientation of the axis.
KEYWORD *"strainout"*			Create a file, strain.out, containing x, y, z, node number, ε_{xx}, ε_{yy}, ε_{zz}, ε_{xy}, ε_{xz}, ε_{yz}

Input Variable	Format	Default	Description
KEYWORD "*excess_shear"*			For stress output to the files generated by the **cont** macro, output the excess shear stress and the direction of the failure plane given in the equation below, as well as the Young's modulus $\tau_{excess}(max) = \frac{1}{2}(\sigma_1^{eff} - \sigma_2^{eff})(\mu^2+1)^{1/2} - \frac{1}{2}\mu(\sigma_1^{eff} + \sigma_2^{eff}) - S_0$ and $\sigma^{eff} = \sigma - PP\_FAC \cdot PORE\ PRESSURE$. Where τ_{excess} is the excess shear, σ_1 and σ_2 are the maximum and minimum principal stresses, μ is the coefficient of friction, and S_0 is the shear strength. The angle β between this plane and the orientation of the maximum principal stress is given by $\tan(2\beta) = -\frac{1}{\mu}$.
FRICTION_OUT	real		Coefficient of friction
STRENGTH_OUT	real		Cohesion
PP_FAC_OUT	real		Pore pressure factor similar to Biot's coefficient in the 'biot' macro.
KEYWORD *"zone*"			The format and inputs for this are described in the **zone** macro. Inclusion of **zone** macros within the strs macro are allowed to facilitate input associated with the following keywords.
KEYWORD "*permmodel"*			This keyword identifies the stress or displacement dependent permeability model. The permeability model can be invoked in a fully coupled, sequentially coupled, or explicitly coupled manner.
IPERM	integer	1	Specifies the type of permeability model used, input parameters specified on this line change depending on the model selected.
IPERM = 1	Equivalent to no stress permeability model		
IPERM = 2	Stress permeability model dependent on tensile stress in the coordinate directions. Changes are linear in stress up to the prescribed maximum change. Tensile stress in a given coordinate direction affects the permeabilities in the other two directions. Input: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f,spm9f		
SPM1F	real		Minimum tensile stress (x direction) for damage to occur.
SPM2F	real		Minimum tensile stress (y direction) for damage to occur
SPM3F	real		Minimum tensile stress (z direction) for damage to occur
SPM4F	real		Damage factor for elastic modulus in x direction.
SPM5F	real		Damage factor for elastic modulus in y direction.

Input Variable	Format	Default	Description
SPM6F	real		Damage factor for elastic modulus in z direction.
SPM7F	real		Maximum factor for x-permeability.
SPM8F	real		Maximum factor for y-permeability.
SPM9F	real		Maximum factor for z-permeability.
IPERM = 22			Mohr-coulomb failure criteria on the plane that maximizes the excess shear. Here z-prime is along the normal to the plane of failure, and y-prime is along the plane of median principal stress. Input: iperm,spm1f,spm2f,spm3f,spm4f,spm5f,spm6f,spm7f, spm8f,spm9f,spmf10
SPM1F	real		Friction coefficient of shear in the fault plane.
SPM2F	real		Shear strength of the fault plane.
SPM3F	real		Factor in effective stress calculation where $effectivestress = \sigma - (ppfac \cdot porepressure)$
SPM4F	real		Range of excess shear stress over which damage is ramped
SPM5F	real		Maximum multiplier for young's modulus in x-prime direction.
SPM6F	real		Maximum multiplier for young's modulus in y-prime direction
SPM7F	real		Maximum multiplier for young's modulus in z-prime direction
SPM8F	real		Maximum multiplier for permeability x-prime direction.
SPM9F	real		Maximum multiplier for permeability y-prime direction.
SPM10F	real		Maximum multiplier for permeability z-prime direction.
IPERM = 91			Table input from a file
FILENAME	character		Name of the file with permeability model factors. The file has the following format: Line 1: # of rows in the table Lines 2 through (# of rows)+1: stress, x-factor, y-factor, z-factor
KEYWORD *"elastic"*			For linear elastic material.
ELASTIC_MOD	real		Young's modulus. (MPa)
POISSON	real		Poisson's ratio.
KEYWORD *"nonlinear"*			

Input Variable	Format	Default	Description
NONLIN_MODEL_FLAG	integer		If NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young's modulus and Poisson's ratio. Input: E_INI, DEDT, POISSON_INI, DNUEDT If NONLIN_MODEL_FLAG= 91 then a table lookup is used:. Input: YOUNG_TEMP_FILE
E_INI	real		Value of Young's modulus at the reference temperature (MPa).
DEDT	real		Derivative of Young's modulus with respect to temperature (MPa/^{o}C)
POISSON_INI	real		Value of Poisson's ratio at the reference temperature.
DNUEDT	real		Derivative of Poisson's ratio with respect to temperature (per ^{o}C)
YOUNG_TEMP_FILE	character		Name of the file with nonlinear model values. The file has the following format: Line 1: # of rows in the table (nentries_young) Lines 2 through (# of rows)+1: temperature, young's modulus, poisson's ratio
KEYWORD *"plastic"*			
NUMPLASTICMODELS	integer		Number of plastic models.
PLASTICMODEL	real		Plastic model number If PLASTICMODEL = 1 Isotropic, linear elastic solid If PLASTICMODEL = 2 von Mises model PLASTICPARAM1 and PLASTICPARAM2 are only entered for the von Mises model.
MODULUS	real		Young's modulus in the elastic region (MPa).
NU	real		Poisson's ratio in the elastic region.
PLASTICPARAM1	real		Yield stress for von Mises model (MPa).
PLASTICPARAM2	real	0.	Currently not used.
KEYWORD *"biot*"			
ALPHA	real	0	Volumetric coefficient of thermal expansion (per ^{o}C)
PP_FAC	real		Factor multiplying the pore pressure coupling term in the stress-strain relations, given by $\bar{\bar{\sigma}} = \overline{\overline{D^{ep}}} \cdot (\bar{\bar{\varepsilon}} - \alpha \cdot \Delta T \bar{\bar{I}} - PP\_FAC \cdot \Delta P \bar{\bar{I}})$. Where the symbols have the usual meanings.
KEYWORD *"stressboun"*			Enter boundary conditions for the mechanical deformation equations. These can be a combination of specified values of displacements, stresses, or forces.

Input Variable	Format	Default	Description
SUB-KEYWORD '*distributed*'			Distribute the applied force in proportion to areas of the members of the zone to which the force is applied.
SUB-KEYWORD '*lithostatic*'			BOUNVAL and KQ are interpreted as multipliers of the lithostatic stress and the stress direction. The lithostatic stress is always in the vertical (downward) direction. The z-axis is taken to be positive upwards. In the **ctrl** macro the direction of gravity must be set to 3.
SUB-KEYWORD '*lithograd*'	SDEPTH GDEPTH		BOUNVAL and KQ are interpreted as the stress gradient and stress direction. The parameters sdepth and gdepth are read on the same line as lithograd, and the KQ^{th} diagonal component of the stress at any node is calculated as follows, where z is the vertical coordinate of the node (see Figure 3) $STRESS(KQ) = (SDEPTH + GDEPTH - Z) \cdot BOUNVAL$
SDEPTH	real		Depth (m) of the reference level from the free surface of the earth corresponding to the level specified by GDEPTH, i.e., model elevation of GDEPTH meters is equivalent to SDEPTH meters depth.
GDEPTH	real		In the coordinate system of the model, the z coordinate of the reference level.
BOUNVAL	real	0	This is a fixed displacement, specified stress, or specified force depending on the value of KQ and optional keywords. No keyword, and kq > 0 : prescribed displacement (m) in the kq direction No keyword and kq < 0 : applied stress (MPa) in the kq direction Keyword = 'lithograd' and kq > 0 : the stress gradient (MPa/m) in the kq direction Keyword = 'distributed' and kq < 0 : prescribed force (MN) in the kq direction.
KQ	integer	0	Parameter that determines the type of boundary condition kq = 1 or -1: prescribed value in the x direction kq = 2 or -2: prescribed value in the y direction kq = 3 0r -3: prescribed value in the z direction
KEYWORD *"tolerance* (required)			
STRESS_TOL	real	0	The tolerance for solution of the stress equations STRESS_TOL > 0 STRESS_TOL is the reduction of initial residual of the stress equations STRESS_TOL < 0 STRESS_TOL is the required absolute value of the residual of the normalized equations

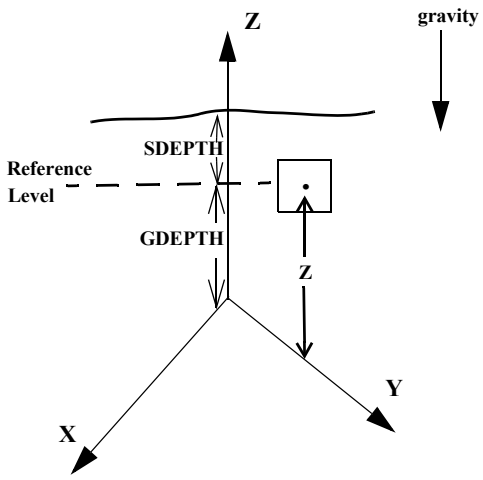

Figure 3. Schematic illustrating varaibles for ‘lithograd’ option.

In the 3D example below, the option to explicitly couple stress with heat-mass equations is invoked. Initial stresses and displacements are calculated, a body force due to gravity is applied, optional strain output is activated, computations are performed using the finite element module, material is specified to be elastic, with temperature dependence of Young's modulus and Poisson's ratio specified in a file called "EvsT.txt", linear coefficient of thermal expansion 1.4e-5/0C, Biot's coefficient equal to 0. Zone 3 is pinned in all 3 directions, zones 4 And 5 are constrained in the X direction, and zones 6 and 7 are constrained in the Y direction. Tolerance for the stress solution is set to 1.e-3.

```
strs
1 -3
initcalc
bodyforce
strainout
fem
elastic
1 0 0 1.59e4 0.25

nonlinear
91
EvsT.txt
biot
1 0 0 5.4e-5 0.

zone
2  ! top,Z=300
    -1.e+15  +1.e+15  +1.e+15  -1.e+15 -1.e+15  +1.e+15  +1.e+15  -1.e+15
      +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
          300.01 300.01 300.01 300.01 299.99 299.99 299.99 299.99
```

```
 3  ! bottom, Z=0
   -1.e+15  +1.e+15  +1.e+15  -1.e+15 -1.e+15  +1.e+15  +1.e+15  -1.e+15
     +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
       0.1 0.1 0.1 0.1 -0.1 -0.1 -0.1 -0.1

stressboun
-3 0 0   0.  3

stressboun
-3 0 0   0.  2

stressboun

-3 0 0   0.  1

zone
 4  ! back X=20
   19.99 20.01 20.01 19.99 19.99 20.01 20.01 19.99
     +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
        300.01 300.01 300.01 300.01 -1. -1. -1. -1.
 5  ! front X=0
    -0.01 +0.01 +0.01 -0.01 -0.01 +0.01 +0.01 -0.01
     +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
        300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-4 0 0   0.  1

stressboun
-5 0 0   0.  1

zone
 6  ! right, Y=0
  -1.e+15  +1.e+15  +1.e+15  -1.e+15 -1.e+15  +1.e+15  +1.e+15  -1.e+15
    0.01  0.01  -0.01  -0.01 0.01  0.01  -0.01  -0.01
        300.01 300.01 300.01 300.01 -1. -1. -1. -1.
 7  ! left, Y=60.
   -1.e+15  +1.e+15  +1.e+15  -1.e+15 -1.e+15  +1.e+15  +1.e+15  -1.e+15
      60.01 60.01 59.99 59.99 60.01 60.01 59.99 59.99
        300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-6 0 0   0.  2

stressboun
-7 0 0   0.  2
```

```
tolerance
-1.e-3
end stress
```

6.2.86 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 V2.21sun 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
```

6.2.87 **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/m^3 and the viscosity is 5.e-3 Pa·s.

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

6.2.88 **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

```

6.2.89 **Control statement thic** (optional)

Input for variable thickness for two-dimensional problems.

Group 1 - JA, JB, JC, THIC (JA, JB, JC - defined on page 33).

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.

```

6.2.90 **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.
IPRTOUT	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 ≤ 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          0.0
   1.0        -1.2         1.0        10
```

6.2.91 **Control statement trac** (optional)

Group 1 - KEYWORD ‘userc’, ANO, AWC, EPC, UPWGTA

Optional keyword “file” is used to specify the name of the data file that will contain input for the userc subroutine.

KEYWORD ‘file’

USERC_FILENAME

or

Group 1 - ANO, AWC, EPC, UPWGTA

Group 2 - DAYCS, DAYCF, DAYHF, DAYHS

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

Group 4 - KEYWORD ‘tpor’

Group 5 - JA, JB, JC, PS_TRAC (JA, JB, JC - defined on page 33)

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 (MFLAG, SEHDIFF, 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:

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

Otherwise if both liquid and vapor are present (keyword ‘dspb’), parameters for both must be entered.

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

Groups 9 is used to define transport models for which diffusion and dispersion parameters are identical. Group 9 is read in until a blank line is encountered. The model number is incremented by 1 each time a line is read.

Group 10 - JA, JB, JC, ITRCDSP (JA, JB, JC - defined on page 33)

Group 11 - ICNS [SPNAM]

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:

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

For a Henry's Law Species (both liquid and vapor) if DIFFML ≥ 0

Group 12 - IADSFL, A1ADSFL, A2ADSFl, BETADFL, MFLAGL, DIFFML, TCLX, TCLY, TCLZ, IADSFV, A1ADSFV, A2ADSFV, BETADFV, MFLAGV, DIFFMV, TCVX, TCVY, TCVZ

Group 13 - JA, JB, JC, ITRCD (JA, JB, JC - defined on page 33)

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 33)

Group 16 - JA, JB, JC, CNSK, T1SK, T2SK (JA, JB, JC - defined on page 33)

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, diffusion 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.
KEYWORD	character*4	Optional keyword 'file' designating the name for the user subroutine input transport parameter file will be input. If this keyword and the following line are omitted, the default name will be 'userc_data.dat'.
USERC_FILENAME	character*80	Name of file from which to read transport parameters for optional user subroutine.
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).

Input Variable	Format	Description
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.
KEYWORD	character*4	Keyword '*ldsp*' 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 '*ldsp'* 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). '*dspl'* indicates that only liquid species exist. '*dspv*' indicates that only vapor species exist. *'dspb'* indicates that both liquid and vapor species exist.

Input Variable	Format	Description
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)
SPNAM	character*20	For each species, the name of the species (e.g. Sulfate). This is an optional identifier that may be input when macro **rxn** is not being used.
MFLAG	integer	Flag denoting type of diffusion model to be used 0 - the molecular diffusion coefficient is a constant. 1 - Millington Quirk diffusion model for liquid or vapor. 2 - Conca and Wright diffusion model for liquid, alternate Millington Quirk diffusion model for vapor. 3 - vapor diffusion coefficient is calculated as a function of pressure and temperature using tortuosity from **adif** macro, Equation (21) of the "Models and Methods Summary" of the FEHM Application (Zyvoloski et al. 1999). FEHM calculates liquid contaminant flux as J = (Water Content)x(D*)x(GradC) and vapor contaminant flux as J = (Air Content)x(D*)x(GradC) where D* is the diffusion coefficient input in this macro. Water content is defined as porosity x saturation and air content is defined as porosity x (1 - saturation). For more explanation on the Millington Quirk and Conca/Wright models see Stauffer, PH, JA Vrugt, HJ Turin, CW Gable, and WE Soll (2009) Untangling diffusion from advection in unsaturated porous media: Experimental data, modeling, and parameter uncertainty assessment. Vadose Zone Journal, 8:510-522, doi:10.2136/vzj2008.0055. SEHDIFF real Molecular diffusion coefficient (m2/s) When MFLAG = 0, the input diffusion coefficient is used directly in the contaminant flux equations presented above. However, MFLAG = 1 or 2, the free air or free water diffusion coefficient is input and the correct porous diffusion is calculated within FEHM. For MFLAG = 3, the code assumes a free air diffusion coefficient of 2.33e-5 m2/s for water vapor in air as described in the Models and Methods Summary Eq. 21.
SEHDIFF	real	Molecular diffusion coefficient (m^2/s)
TCX	real	Dispersivity in x-direction (m)
TCY	real	Dispersivity in y-direction (m)
TCZ	real	Dispersivity in z-direction (m)

Input Variable	Format	Description
ITRCDSP	integer	Region number for dispersion parameters given in group 9 (keyword *dspl*, *dspv*, or *dspv*). 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
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) 3 - Henry's model uses water vapor pressure ($H = P_{wv}$)
HAWWA(1)	real	Term in Henry's Law temperature dependence model: For model 1 or 3 - parameter value is A_H For model 2 - parameter value is $A_{H,1}$ For model 3 - not used
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 $A_{H,2}$ For model 3 - Henry's constant modifier, $H = P_{wv} \cdot \Delta H_H$
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 $A_{H,5}$

Input Variable	Format	Description
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.
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 *source* 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 m^2/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 159.

```
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.      0   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.

```
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                                                                                Group 4
    1       0       0           0.2                                                 Group 5

    2                                                                               Group 6
ldsp                                                                                Group 7
    1                                                                               Group 11
    0       0.      0.          1.          0   1.e-9   0.0067  0.0067              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 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
2					Group 6
dspl					Group 8
0	1.e-9	0.0067	0.0067	0.0067	Group 9
1	0	0	1		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
					Group 16
0					Group 11
1	0	0	2.e-5		Group 15
					Group 16

The final examples 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 or the file entered when using optional keyword file after Group 1. 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.

There are only two options currently supported, option 2: Time-varying solute mass flux input at prescribed nodes, and option 5: Mass flux recirculation and erosion at prescribed zones.

For option 2 the input will be

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.

For option 5 the input will be

Group 2 - EROSION_FACTOR

Group 3 - RECYCLE_FACTOR(J), J = 1, NSPECI

Group 4 - NU_ZONES

Group 5 - PRODZONES(I), I = 1, NU_ZONES

Group 6 - INFLZONES(I),I = 1, NU_ZONES

Input Variable	Format	Description
USROPTION	integer	User subroutine option 2 - Time-varying solute mass flux input at prescribed nodes 5 - Mass flux recirculation and erosion
For usroption = 2		
N_POINTS	integer	Number of flux-time points in the transient input curve
USERFLUX	real	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)
USERTIME	real	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)
For usroption = 5		
EROSION_FACTOR	real	Rate at which solute is removed from the solid in specified nodes (1/yr) These nodes are contained in the designated injection zones.
RECYCLE_FACTOR	real	Fraction of solute that is recycled. NSPECI values are entered (one for each specie in the model)
NU_ZONES	integer	Number of production/injection zone pairs being modeled.
PRODZONES	integer	NU_ZONES zone numbers where solute is being produced.

Input Variable	Format	Description
INFLZONES	integer	NU_ZONES zone numbers where solute is being reinjected. The production/injection zones should be defined

In the fourth example, the userc subroutine is invoked with usroption = 2. 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 there is no mass flux 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       0       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
```

									Group 16
1									Group 11
0	0	0	1	0	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

File "*userc_data.dat*":

2			Group 1
5			Group 2
Mass flux (moles/s)			Group3
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.

In the final example, the userc subroutine is invoked with usroption = 5. The user subroutine input will be read from file *"userc_noerosion.dat"*, in subfolder input_new. A single conservative specie is modeled. Recycling is specified for 65 production/ injection zone pairs (Groups 4, 5, and 6) and all solute produced will be recycled (Group 3) without erosion (Group 2). Note that for this example, even if an erosion factor had been specified, there would be no erosion since a conservative specie is

being modeled ($K_d = 0$). Fluid production/injection is controlled using the **flow** or **boun** input macros.

```
trac
userc    0.0      1.0      1.e-6    1.0                                  Group 1
file
input_new/userc_noerosion_new.dat
1.0      365250.  365250.  1.0                                           Group 2
50       1.2      1.0      1000.0                                       Group 3
tpor                                                                    Group 4
1        0        0        0.18                                         Group 5

1                                                                       Group 6
1                                                                       Group 11
0        0.       0.       1.       1.e-11   2.0      0.2      0.1     Group 12

1        0        0        1                                            Group 13

1        0        0        0.                                           Group 15

-44              0        0        -1.00    1.0      1.e20               Group 16
```

File *"userc_noerosion.dat"*:

```
5                                                                       Group 1
0.0                                                                     Group 2
1.0                                                                     Group3
65                                                                      Group 4
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117     Group 5
118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134
135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181         Group 6
182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213
214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229
230
```

6.2.92 5**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
```

6.2.93 **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 58]. No input is associated with this macro.

6.2.94 **Control statement vcon** (optional)

Variable thermal conductivity information.

(1) Thermal conductivity for intact salt: $\lambda_{IS} = \lambda_{IS,300}\left(300/T\right)^{\gamma_1}$

(2) Thermal conductivity for crushed salt: $\lambda_{CS} = \lambda_{CS,300}\left(300/T\right)^{\gamma_1}$

where $\lambda_{CS,300} = 1.08\left(\alpha_4\phi^4 + \alpha_3\phi^3 + \alpha_2\phi^2 + \alpha_1\phi + \alpha_0\right)$

Parameters related to thermal conductivity are $\lambda_{IS,300}, \gamma_1, \gamma_2, \alpha_4, \alpha_3, \alpha_2, \alpha_1, \alpha_0,$ and ϕ. An additional parameter is the specific heat of salt

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

Group 2 - JA, JB, JC, IVCND (JA, JB, JC - defined on page 33)

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.

Input Variable	Format	Description
VC1F(i)	real	Reference temperature (^oC) 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 are examples of **vcon**. In the first example, a linear conductivity model is defined and applied at each node. The reference temperature is 20^oC, 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

```

For the second example, three models are defined for the entire domain. Model 1 defines the constant thermal conductivity of 16.26 W/m K at 26.85^0C (=300 ^0K) for a stainless steel canister (zone 1). Model 2 defines all parameters for a crushed salt (zone 2). Model 3 defines reference thermal conductivity of 5.4 W/m K at 26.85^0C (=300 ^0K) and exponent 1.14 for the intact salt in the rest of the domain.

```
vcon
1     26.85   16.26   0.
4     26.85   1.08    270.0   370.0   136.0   1.5     5.0     1.14
3     26.85   5.4     1.14

1     0       0       3
-1    0       0       1
-2    0       0       2

```

6.2.95 **Control statement weli** (optional)

Peaceman type well impedance.

Group 1 ITYPEWELL, KEYWEL1,PARMWEL1,PARMWELL2, PARMWEL3

Enter one line per model defined, terminated with a blank line.

Group 2 JA, JB, JC, IZONEWELL (JA, JB, JC - defined on page 33

Input Variable	Format	Description
ITYPEWELL	integer	Well model type
KEYWEL1	integer	
PARAMWEL1	real	Peaceman's rw, wellbore radius (m).
PARAMWEL1	real	Peaceman's h, reservoir thickness (m)
PARAMWEL1	real	Print parameter
IZONEWELL	integer	Peaceman model assignment.

6.2.96 Control statement **wflo** (optional)

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

Group 1 KEYMS1, KEYMS2, KEYMS3, KEYMS3

or if KEYMS4 = 'type'

Group 1 KEYMS1, KEYMS2, KEYMS3, KEYMS4, ITYPSD

Enter one line per model defined, terminated with a blank line.

Group 2JA, JB, JC, ISUBMD (JA, JB, JC - defined on page 33

Input Variable	Format	Description
KEYMS1	character	Macro SKD type. The letters given in ( ) are sufficient to identify the keyword: (p)ressure (h)ead (f)lux
KEYMS2	character	Macro ESKD type. The letters given in ( ) are sufficient to identify the keyword: (s)aturation (t)emperature (e)nthalpy (w)ater (water only source, output saturation as 1.0)
KEYMS3	character	Macro AIPED type: imph - Impedance parameter = 1.0 impl - Impedance parameter = 1.0e-4 impn - Impedance parameter = -1.0 If KEYSM1 = 'flux' the impedance parameter = 0.0, otherwise for any other input, the impedance parameter = 1.0e2.
KEYMS4	character	Flag to indicate submodel type will be input: type

Input Variable	Format	Description
ITYPSD	INTEGER	Submodel type: ITYPSD = 0 , generate 'flow' macro. ITYPSD ≠ 0 , generate 'flo3' macro.
ISUBMD	integer	Submodel assignment.

The following is an example of the **wflo** macro. A flow macro will be written for two output zones. Because, they are specified using two models, the macros will be written to two separate files. The macro file names will be generated using the root file name (as input or determined from the output file name) appended with the model number and suffix '.wflow' (e.g. file.0001.wflow and file.0002.wflow).

```
wflo
flux      wat      impl      na
flux      wat      impl      na

 -308      0        0         1
 -309      0        0         2

```

6.2.97 Control statement **wgtu** (optional)

User defined destributed source boundary weight factors used in conjunction with macro **boun**.

Group 1 - NAREA

Group 2 - IZONE_AREA(I), IAREAF(I), AREA01(I), AREA02(I), I = 1 to NAREA

If (IAREAF(I) = 7) a weight factor is read for each node in the current zone.

Group 3 - II, (JJ, WGT_AREA(JJ), IJ = 1 to II)

If (IAREAF(I) = 8) a weight factor is read for each node in the current zone from the specified file. See Group 3 above for file format.

Group 3 - WFILENAME

Input Variable	Format	Description
NAREA	integer	Number of zones for which a weight factor will be calculated.
IZONE_AREA	integer	Zone on which to calculate weight factors.

Input Variable	Format	Description
IAREAF	integer	Method for calculating weight factor for current zone. 1 : Calculate area of each node in zone 2 : Calculate area of each node based on total area given (area01) and portioned by volume size 3 : Calculate area of each node based on total area given (area01) and portioned by approximate area size 4 : Calculate weighting based on area (used in boun) 5 : Calculate weighting based on area*perm (used in boun) 6 : Calculate weighting based on vol*perm (used in boun) 7 : Read a list of weights for every node in zone 8 : Read a file with weights for every node in zone
AREA01	real*8	Input area.
AREA02	real*8	Base impedance parameter.
II	integer	Number of nodes in the current zone.
JJ	integer	Node number of the ijth node.
WGT_AREA	real*8	Weighting factor for JJ
WFILENAME	character*100	Name of file contining weight factors for the current zone.

6.2.98 Control statement **wtsi** (optional)

Water table, simplified.

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

Input Variable	Format	Description
NFREE	integer	Number of zones to which water table condition are applied.
IZONE_FREE	integer	Zone number.
IFREE_IM_EX	integer	Update parameter (0 - explicit update, 1 - implicit update).
HEAD_TOL	real*8	Tolerance for head (m)
RLPTOL	real*8	Tolerance for saturation

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

```
wtsi
  1        1        50       0.01     0.0001
```

6.2.99 **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.

Input Variable	Format	Description
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 **dpdp** 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

    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
```

6.2.100Control 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 56), **ctrl** (page 60), **nod2** (page 111), **node** (page 110), **mptr** (page 92), **pest** (page 114), **ptrk** (page 117), **sptr** (page 161), **subm** (page 183), **wflo** (page 201), and **time** (page 185): **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**and **wflo** are 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, concentrationand, and for the dua 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. 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 fluxes have been specified in the **rest** macro (or for compatibility with older versions of the code if (ABS (PRNT_RST) = 2 in **ptrk** macro)

Label: 'all fluxes' or 'liquid flux' or 'vapor flux'

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. If both liquid and vapor fluxes as written liquid flux will be output first.

Otherwise

Label: 'no fluxes'

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

Number of particles, final random number seed for transport calculations, final random number seed for particle release position (use by GoldSim)

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 for transport calculations 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 'dpdp' or 'dual' flag is set

The above information includes dual porosity/dual permeability nodes.

```
FEHM V3.1gf 12-02-02 QA:NA       02/14/2012    10:33:49
Unsaturated Diffusion tests
   5000.0000000000000
ngas
trac
nstr
ndpd
ndua
  34.99999999987494        34.99999999987494        29.99740954219060        29.99740954219060
  24.99481908388880        24.99481908388880        19.99222863160355        19.99222863160355
  14.99935303204482        14.99935303204482        10.00000000012507        10.00000000012507
  0.1000000000000000E-98   0.1000000000000000E-98   0.1000000000000000E-98   0.1000000000000000E-98
  0.1000000000000000E-98   0.1000000000000000E-98   0.1727371363921276       0.1727371363921281
  0.4344871249926068       0.4344871249926068       0.7817833455822488       0.7817833455822516
  0.1001154694602094       0.1001154694602094       0.1001154694628803       0.1001154694628803
  0.1001154694707533       0.1001154694707533       0.1001154694901246       0.1001154694901246
  0.1001154722096991       0.1001154722096991       0.1001154822144740       0.1001154822144740
  0.9766094917448133E-01   0.9766094917448133E-01   0.9770095207050181E-01   0.9770095207050181E-01
  0.9774097492577727E-01   0.9774097492577727E-01   0.9778102503811041E-01   0.9778102503811041E-01
  0.9841762151617499E-01   0.9841762151617499E-01   0.9888735221221216E-01   0.9888735221221216E-01
no fluxes
          1
     1.040511468         1.040511468         1.023454397         1.023454397
     1.006402317         1.006402317        0.9893551455        0.9893551455
    0.9701457197        0.9701457197        0.9516070487        0.9516070487
```

Figure 4. Example of FEHM restart (fin) file using original format.

In FEHM Version 3.00 or newer, the format of the file was simplified to allow user selection of output parameters. The modified header contains:

Code version number, date, time

Problem title

Simulation time (days)

Number of nodes (n), type keyword ('dual', 'dpdp' or 'nddp) to designate if dual porosity or double permeability were invoked or not for the simulation.

The remainder of the data is output following a 'keyword' header specifying the type of data to follow. n values will be output for each specified output parameter. Flux data will be output

using the original format following the parameter output and preceding any particle tracking ‘ptrk’ or transport ‘trac’ output. Particle tracking output or transport output will be preceded by a ‘keyword’ header and use the same format as before.

```
FEHM V3.1gf 12-02-09 QA:NA       02/09/2012    11:48:27
Unsaturated Diffusion tests
   5000.0000000000000
       12 nddp
temperature
 34.99999999987494        34.99999999987494        29.99740954219060        29.99740954219060
 24.99481908388880        24.99481908388880        19.99222863160355        19.99222863160355
 14.99935303204482        14.99935303204482        10.00000000012507        10.00000000012507
saturation
 0.10000000000000000E-98  0.10000000000000000E-98  0.10000000000000000E-98  0.10000000000000000E-98
 0.10000000000000000E-98  0.10000000000000000E-98  0.1727371363921276       0.1727371363921281
 0.4344871249926068       0.4344871249926068       0.7817833455822488       0.7817833455822516
pressure
 0.1001154694602094       0.1001154694602094       0.1001154694628803       0.1001154694628803
 0.1001154694707533       0.1001154694707533       0.1001154694901246       0.1001154694901246
 0.1001154722096991       0.1001154722096991       0.1001154822144740       0.1001154822144740
gaspressure
 0.9766094917448133E-01   0.9766094917448133E-01   0.9770095207050181E-01   0.9770095207050181E-01
 0.9774097492577727E-01   0.9774097492577727E-01   0.9778102503811041E-01   0.9778102503811041E-01
 0.9841762151617499E-01   0.9841762151617499E-01   0.9888735221221216E-01   0.9888735221221216E-01
no fluxes
 trac
           1
     1.040511468          1.040511468          1.023454397          1.023454397
     1.006402317          1.006402317         0.9893551455         0.9893551455
    0.9701457197         0.9701457197         0.9516070487         0.9516070487
```

Figure 5. Example of FEHM restart (fin) file using new format.

7.3 History plot file (*filen.his*)

The history plot file contains the following (see Figure 6) :

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

```
FEHM V2.30sun 05-04-19 QA:NA     04/20/2005    08:45:05
***** 2-D Heat Conduction Model *****



 1
     111 0.000000000E+00 0.000000000E+00 0.000000000E+00
headings
node flow enthalpy(Mj/kg) flow(kg/s) temperature(deg C) total pressure(Mpa)
capillary pressure(Mpa) saturation(kg/kg)
 0.0E+0
     111  0.100000000E-19 0.00000000 200.000000 10.0000000 0.00000000 0.00000000
 5.0E-3
     111  0.100000000E-19 0.00000000 199.999999 10.0000000 0.00000000 0.00000000
.
.
.
 4.000049999999937
     111  0.100000000E-19 0.00000000 100.183607 10.0000000 0.00000000 0.00000000
 -4.000049999999937
     111  0.100000000E-19 0.00000000 100.183607 10.0000000 0.00000000 0.00000000
```

Figure 6. Example of history output file, *filen.his*

7.4 Alternate History plot files (*filen.his, filen_param.his*)

The history plot file (*filen.his*) contains the following (see Figure 7):

Code version number, date, time

Problem title

Gas flag (‘ngas’, ‘airw’, or blank)

Tracer flag (‘trac’ or blank)

Stress flag (‘strs’ or blank)

List of parameters written to individual history files (possible paramenters are: pressure, temperature, head, saturation, flow, enthalpy, humidity, zone flux, and global)

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

```
FEHM V2.30sun 05-04-20 QA:NA     04/20/2005    08:50:56
***** 2-D Heat Conduction Model *****



Parameters written to individual history files:
 temperature
for the following nodes:
 1
     111   0.00000000       0.00000000       0.00000000
```

Figure 7. Example of alternate history output file, *filen.his*

If zones for output are specified (see Figure 8)

Number of zones over which output is averaged

And for each zone

Number of nodes in the zone and zone number followed by a list of nodes in the zone.

```
FEHM V2.30sun 05-04-20 QA:NA     04/20/2005    09:10:21
***** 2-D Heat Conduction Model *****



Parameters written to individual history files:
 temperature
for the following nodes and zones:
 1
     111   0.00000000       0.00000000       0.00000000
Number of averaged output zones:    2
    121 Nodes averaged in Zone   -1
0000001 0000002 0000003 0000004 0000005 0000006 0000007 0000008 0000009 0000010
0000011 0000012 0000013 0000014 0000015 0000016 0000017 0000018 0000019 0000020
.
.
.
0000121
     11 Nodes averaged in Zone   -2
0000001 0000012 0000023 0000034 0000045 0000056 0000067 0000078 0000089 0000100
0000111
```

Figure 8. Example of alternate history output file including zones, *filen.his*

The history plot parameter files (*filen_param.his*) contain the following:

Code version number, date, time

Problem title

Blank line

Output parameter title and units

Heading: Time (*units*) Nodes: *Node number 1 . . . Node number n*

-or- if zones are specified

Heading: Time (*units*) Nodes: *Node number 1 . . . Node number n* Zones: *Zone number 1 . . . Zode number n*

And for each time step (time *units* may be seconds, days, or years as specified in **hist** macro)

Time (*units*) followed by parameter value for each specified node and zone. (See Figures 9 and 10)

```
FEHM V2.30sun 05-04-20 QA:NA     04/20/20   09:39:26
***** 2-D Heat Conduction Model *****

 Temperature (C)
 Time (seconds) Nodes: 111
 0.0E+0 200.0
 432.0 199.99999943712558
 864.0 199.99999502157842
.
.
.
 345599.9999999945 100.18362295565089
 345604.31999999453 100.18360733099975
```

Figure 9. Example of alternate history output file, *filen_temp.his*

```
FEHM V2.30sun 05-04-20 QA:NA     04/20/20   09:10:21
***** 2-D Heat Conduction Model *****

 Temperature (C)
 Time (seconds) Nodes: 111 Zones: -1 -2
 0.0E+0 200.0 199.99999999999985 200.0
 432.0 199.99999943712558 187.43603443819043 193.49769265774958
 864.0 199.99999502157842 184.97995686212098 192.16893175137713
.
.
.
 345599.9999999945 100.18362295565089 100.07411373139031 100.11665754330934
 345604.31999999453 100.18360733099975 100.07410742498257 100.11664761680842
```

Figure 10. Example of alternate history output file including zones, *filen_temp.his*

The history plot global parameter file (*filen_param.his*) contains the following:

Code version number, date, time

Problem title

Blank line

Depending on output selected the following possible headings:

mass / energy: Time (days) Total mass in system (kg) Total mass of steam in system (kg) Water discharge (kg) Water input (kg) Total water discharge (kg) Total water input (kg) Net (kg) water discharge Total enthalpy in system (MJ) Enthalpy discharge (MJ) Enthalpy input (MJ) Total enthalpy discharge (MJ) Total enthalpy input (MJ) Net (MJ) enthalpy discharge

water / air: Time (days) Total water in system (kg) Total mass of steam in system (kg) Water discharge (kg) Water input (kg) Net (kg) water discharge Total water discharge (kg) Total water input (kg) Total air in system (kg) Air discharge (kg) Air input (kg) Total air discharge (kg) Total air input kg (kg/s)Net (kg) air discharge

mass / water only (no steam): Time (days) Total water in system (kg) Water discharge (kg) Water input (kg) Total water discharge (kg) Total water input (kg) Net (kg) water discharge

mass / water only (steam): Time (days) Total mass in system (kg) Total mass of steam in system (kg) Water discharge (kg) Water input (kg) Total water discharge (kg) Total water input (kg) Net (kg) water discharge

air only: Time (days) Total air in system (kg) Air discharge (kg) Air input (kg) Total air discharge (kg) Total air input kg (kg/s) Net (kg) air discharge

energy only: Time (days) Total enthalpy in system (MJ) Enthalpy discharge (MJ) Enthalpy input (MJ) Total enthalpy discharge MJ Total enthalpy input (MJ)Net (MJ) enthalpy discharge

And for each time step

Time (days) followed by selected global parameter values.

7.5 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.

```
FEHM V2.30sun 05-04-06 QA:NA     04/06/2005    19:58:25
Check of FEHMN against SORBEQ, All isotherms
 1
     201 0.100000000E+01 0.100000000E+01 0.100000000E+01
 5 5 0 0 0
 1.550709E-4 1   species #001
 4.855185258201169E-29
 1.550709E-4 2   species #002
 3.7325204274237394E-53
 1.550709E-4 3   species #003
 3.7325204274237394E-53
 1.550709E-4 4   species #004
 1.0E-90
 1.550709E-4 5   species #005
 1.0E-90
.
.
.
 1.3E-3 1   species #001
 0.999994799250939
 1.3E-3 2   species #002
 0.9999246650715027
 1.3E-3 3   species #003
 0.9999947753362215
 1.3E-3 4   species #004
 0.9999947668674314
 1.3E-3 5   species #005
 0.9999947643072644
```

Figure 11. Example of solute data history plot file

7.6 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 $\left(\frac{W}{m \cdot K} \right)$ for each node

Porosity, Rock specific heat $\left(\frac{MJ}{kg \cdot K}\right)$, 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 (Š 0 liquid, < 0 vapor) followed by

If injection phase is liquid

Liquid transmissibility / density, Liquid density (kg/m^3), Pressure - Capillary Pressure (MPa), Temperature (°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 (°C)

and if tracer solution is present

Species concentration of vapor phase.

7.7 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.

7.8 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.

7.9 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.

7.10 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 cooordinate

or if boundary type is flux

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

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 183.

7.11 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.

7.12 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.

7.13 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 (m^2) for each specified output node

Heading: "Total Flux (kg/s) Leaving Zone (flxz macro option)"
"Zone NumberSource Sink Net Boundary"

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

The second file (*filen.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

7.14 Particle statistics file (*filen.ptrk)*

The data found in the “.ptrk” file was previously reported in the general output file. From version 2.25 of the code and forward that data will only be reported in the optional, “.ptrk” file unless a coupled GoldSim-FEHM simulation is being run (see the PRNT_RST flag, Section 6.2.59). In addition, the user has the option of selecting which statistics parameters are reported. The default is to report all statistics parameters.

For the default case, the “.ptrk” file contains:

Title line: TITLE="V1=Number Having Entered System, V2=Number Currently In System, V3=Number Having Left System, V4=Number Having Decayed, V5=Number Having Been Filtered, V6=Number That Left This Time”

Header line: VARIABLES="Time (days)" "Sp001 V1" "Sp001 V2" "Sp001 V3" "Sp001 V4" "Sp001 V5" “Sp001 V6” . . . "Sp*nnn* V1" "Sp*nnn* V2" "Sp*nnn* V3" "Sp*nnn* V4" "Sp*nnn* V5" "Sp*nnn* V6"

A heading is output for each variable (V1 to V6) for each species in the model and *nnn* is the total number of species in the simulation. The header line is followed by (for each output time step) the simulation time (days), and the six output variables for each species modeled (Figure 12). When the user selects a subset of the statistics parameters the header line and data will only contain those variables that have been selected for output (Figure 13).

```
TITLE="V1=Number Having Entered System, V2=Number Currently In System, V3=Number Having Left
System, V4=Number Having Decayed, V5=Number Having Been Filtered, V6=Number That Left This Time"
 VARIABLES="Time (days)"  "Sp001 V1" "Sp001 V2" "Sp001 V3" "Sp001 V4" "Sp001 V5" "Sp001 V6"
"Sp002 V1" "Sp002 V2" "Sp002 V3" "Sp002 V4" "Sp002 V5" "Sp002 V6"
365.25000000000   18760   18686      74      0   0    74   18760   18670      90      0   0    90
1278.3750000000   18760   17860     900      0   0   826   18760   17895     865      0   0   775
.
.
.
7270752.1054688   18760     932   16710   1118   0     6   18760     912   16692   1156   0    14
7305000.0000000   18760     929   16712   1119   0     2   18760     905   16695   1160   0     3
```

Figure 12. Example of default “.ptk” particle statistics file.

```
TITLE="V1=Number Having Entered System, V2=Number Currently In System, V3=Number Having Left
System, V4=Number Having Decayed, V5=Number Having Been Filtered, V6=Number That Left This Time"
 VARIABLES="Time (days)"  "Sp001 V1" "Sp001 V2" "Sp001 V3" "Sp001 V4" "Sp001 V6" "Sp002 V1"
"Sp002 V2" "Sp002 V3" "Sp002 V4" "Sp002 V6"
365.25000000000   18760   18686      74      0    74   18760   18670      90      0    90
1278.3750000000   18760   17860     900      0   826   18760   17895     865      0   775
.
.
.
7270752.1054688   18760     932   16710   1118     6   18760     912   16692   1156    14
7305000.0000000   18760     929   16712   1119     2   18760     905   16695   1160     3
```

Figure 13. Example of ".ptk" particle statistics file with five output variables selected.

7.15 Mass Output from GoldSim Particle Tracking Simulation

To provide a simplified method for tracking solute mass from a FEHM/GoldSim coupled simulation an optional output file may be written that contains cumulative mass output (mg/l) (see the PRNT_RST flag,, Section 6.2.59). The data found in FEHM_GSM_Mass_balance.txt has a format similar to that used for the particle statistics output (see above) and contains:

Title line: TITLE="V1=Mass Having Entered System, V2=Mass Currently In System, V3=Mass Having Left System, V4=Mass Having Decayed, V5=Mass Having Been Filtered, V6=Mass Having Decayed Outside The UZ, V7=Filtered Mass Having Decayed"

Header line: VARIABLES="Time (years)" "Sp001 V1" "Sp001 V2" "Sp001 V3" "Sp001 V4" "Sp001 V5" "Sp001 V6" "Sp001 V7". . . "Sp*nnn* V1" "Sp*nnn* V2" "Sp*nnn* V3" "Sp*nnn* V4" "Sp*nnn* V5" "Sp*nnn* V6" "Sp*nnn* V7"

A heading is output for each variable (V1 to V7) for each species in the model and *nnn* is the total number of species in the simulation. The header line is followed by (for each output time step) the simulation time (days), and the seven output variables for each species modeled.

7.16 Particle Exit Locations and Count Output

To facilitate use of particle tracking simulation statistics an optional output file may be written that contains particle exit locations and count (see the PRNT_RST flag, section, Section 6.2.59). The data found in the ".ptrk_fin" file can also be extracted from the ".fin" files through use of post-processors and/or file editors.

The ".ptrk_fin" file output file contains:

Header line:VARIABLES = "Node" "X" "Y" "Z" "Num_exited" "Zone"

Followed by (for each node where particles have exited the system) the node number, the X, Y, and Z coordinates of the node (m), the number of particles that exited at that node, and the number of the zone (if defined) that contains that node.

7.17 Streamline particle tracking output files (*filen.sptr1, filen.sptr2, filen.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 (*filen.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 (*filen.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 (°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 (Figure 14):

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

or when option 'alt' is specified (Figure 15):

Code version number, date, time

Problem title

Heading: Time (days) Particle# ID Zone Node

Followed by breakthrough time, particle number, particle ID, breakthrough zone, break through node for each particle that reaches the breakthrough zone.

Or when option 'alt xyz' is specified (Figure 16):

Heading: x(m) y(m) z(m) Time (days) Particle # ID Zone Node

Followed by x, y, z coordinate where particle entered breakthrough zone, breakthrough time, particle number, breakthrough zone, break through node for each particle that reaches the breakthrough zone.

```
FEHM V3.1win32 12-02-02 QA:NA    02/02/2012    10:22:34
*** Validation1 Test Problem: 3-D Homogeneous Flow and Transport ***
 Time (days)     Zone1 Particles   . . .
    208000.0          0
    210000.0         40
```

Figure 14. Example of default ".sptr3" file.

```
FEHM V3.1win32 12-02-02 QA:NA    02/02/2012    10:22:41
*** Validation1 Test Problem: 3-D Homogeneous Flow and Transport ***
  Time (days)      Particle#     ID    Zone    Node
   209391.24384033          1    1888       5    1938
   209391.24384033          2    1888       5    1938
.
.
.
   209391.24381900         39   51256       5   51306
   209391.24381900         40   51256       5   51306
```

Figure 15. Example of ".sptr3" file generated using option "alt".

```
FEHM V3.1win32 12-02-02 QA:NA    02/02/2012    10:22:48
*** Validation1 Test Problem: 3-D Homogeneous Flow and Transport ***
x(m)  y(m)  z(m) Time (days)      Particle#     ID    Zone    Node
   19800.0000     0.100000000     -5.50000000      209391.24384033          1    1888       5    1938
   19800.0000     0.200000000     -5.50000000      209391.24384033          2    1888       5    1938
.
.
.
   19800.0000     -2800.90000     -200.000000      209391.24381900         39   51256       5   51306
   19800.0000     -2801.00000     -200.000000      209391.24381900         40   51256       5   51306
```

Figure 16. Example of ".sptr3" file generated using option "alt xyz".

7.18 Contour Output Files for AVS, AVS Express, SURFER or TECPLOT

7.18.1 Log output file (*filen.avs_log*)

The log output file is identical for AVS, AVS Express, Surfer and Tecplot (see Figure 17). It contains:

Code version number, date

AVS log identifier

Problem title

and for each specified time

Output file prefix, Call number, and Time. The output time units will correspond to those selected in the **cont** macro.

```
# FEHM V3.1gf 12-02-09 QA:NA       02/09/2012
#   LOG AVS OUTPUT
# Unsaturated Diffusion tests
# Root filename               Output Time (days)
 output/box.00001       0.00000000
 output/box.00002       1001.68908
 output/box.00003       2002.77258
 output/box.00004       3002.77258
 output/box.00005       4003.12657
 output/box.00006       5000.00000
```

Figure 17. Example of contour log output file.

7.18.2 AVS header output files (*filen.type_head*)

The AVS ASCII (formatted) header files are identical for AVS and AVS Express output (see Figure 18). The data types, "*mat*", "*sca*", "*vec*" or "*con*", are described below. 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)

```
# FEHM V3.1gf 12-02-02 QA:NA       02/14/2012
#  AVS UNSTRUCTURED CELL DATA (UCD) FROM FEHM
# Unsaturated Diffusion tests
# ************************************************
#  To prepare files for input to avs one must
#  concatinate  header/geometry/node_value files.
#  For example, if your FEHM input file was fe.dat,
#  headers are fe10001_sca_head fe10001_vec_head, ...,
#  mesh geometry will be in fe10001_geo,
#  field output will be in fe10001_sca_node,
#            fe10001_vec_node, fe10001_con_dual_node
#
#  A UCD input file can be produced using
#   cat fe10001_sca_head fe10001_geo fe10001_sca_node >
#       fe10001_sca_node.inp
#
#  The UNIX foreach command is useful for processing
#   multiple files. Also use the shell script fehm2avs
#   to perform automatic processing of all output.
# ************************************************
0000000012           5           5           0           0
```

Figure 18. Example of AVS header output file.

7.18.3 Geometry output files (*filen.geo, filen_grid.dat*)

Geometry data will be output when keyword “geom” or “grid” are included in the output variable list. For AVS the geometry data is output in a separate file when keyword ‘geom’ is used. For Tecplot, output of the geometry data depends on the input keyword, ‘geom’ or ‘grid’. If keyword “geom’ is used the geometry data is contained in the first contour file for each type of data requested. If keyword ‘grid’ is used the geometry data is output to a separate Tecplot “grid” file.

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

```
0000000001      0.000000000E+00   0.000000000E+00   0.000000000E+00
0000000002      0.100000000E+01   0.000000000E+00   0.000000000E+00
0000000003      0.000000000E+00   0.200000000E+00   0.000000000E+00
0000000004      0.100000000E+01   0.200000000E+00   0.000000000E+00
0000000005      0.000000000E+00   0.400000000E+00   0.000000000E+00
0000000006      0.100000000E+01   0.400000000E+00   0.000000000E+00
0000000007      0.000000000E+00   0.600000000E+00   0.000000000E+00
0000000008      0.100000000E+01   0.600000000E+00   0.000000000E+00
0000000009      0.000000000E+00   0.800000000E+00   0.000000000E+00
0000000010      0.100000000E+01   0.800000000E+00   0.000000000E+00
0000000011      0.000000000E+00   0.100000000E+01   0.000000000E+00
0000000012      0.100000000E+01   0.100000000E+01   0.000000000E+00
0000000001         1 quad         1       2       4       3
0000000002         1 quad         3       4       6       5
0000000003         1 quad         5       6       8       7
0000000004         1 quad         7       8      10       9
0000000005         1 quad         9      10      12      11
```

Figure 19. Example of AVS geometry output file.

The Tecplot “grid” file contains the following (note that only the grid coordinates used are output):

Header line with problem title

Filetype header

Variable header (coordinates used)

Zone header with grid specification and type (N = number of nodes, E = number of elements)

Followed by

“N” node coordinate sets

Element connectivity for the grid for “E” elements.

```
TITLE = "Unsaturated Diffusion tests"
FILETYPE = "GRID"
VARIABLES = "X coordinate (m)" "Y coordinate (m)"
ZONE T = "GRID", N =       12, E =        5, DATAPACKING = POINT, ZONETYPE
= FEQUADRILATERAL, STRANDID = 0, SOLUTIONTIME = 0.
 0.000000000E+00   0.000000000E+00
 0.100000000E+01   0.000000000E+00
 0.000000000E+00   0.200000000E+00
 0.100000000E+01   0.200000000E+00
 0.000000000E+00   0.400000000E+00
 0.100000000E+01   0.400000000E+00
 0.000000000E+00   0.600000000E+00
 0.100000000E+01   0.600000000E+00
 0.000000000E+00   0.800000000E+00
 0.100000000E+01   0.800000000E+00
 0.000000000E+00   0.100000000E+01
 0.100000000E+01   0.100000000E+01
       1       2       4       3
       3       4       6       5
       5       6       8       7
       7       8      10       9
       9      10      12      11
```

Figure 20. Example of Tecplot grid output file.

Geometry data when contained in the normal tecplot data files uses shared variables. Coordinates and connectivity are output only in the first file. The data files contain the following:

The Tecplot “grid” file contains the following:

Header line with code version number, date, time and problem title

Variable header

Zone header with time, grid specification and type

Followed by

“N” node coordinate, node number and datasets

Element connectivity for the grid for “E” elements

```
TITLE = "FEHM V3.1gf 12-02-09 QA:NA     02/09/2012  11:48:26 Unsaturated
Diffusion tests"
VARIABLES = "X (m)" "Y (m)" "Node" "Vapor_Species_001"
ZONE T = "Simulation time   0.00000000    days", N =       12, E =        5,
DATAPACKING = POINT, ZONETYPE = FEQUADRILATERAL, N =       12, E =        5,
DATAPACKING = POINT, ZONETYPE = FEQUADRILATERAL
   0.00000000       0.00000000     0000000001   1.00000000
   1.00000000       0.00000000     0000000002   1.00000000
   0.00000000      0.200000000     0000000003   1.00000000
   1.00000000      0.200000000     0000000004   1.00000000
   0.00000000      0.400000000     0000000005   1.00000000
   1.00000000      0.400000000     0000000006   1.00000000
   0.00000000      0.600000000     0000000007   1.00000000
   1.00000000      0.600000000     0000000008   1.00000000
   0.00000000      0.800000000     0000000009   1.00000000
   1.00000000      0.800000000     0000000010   1.00000000
   0.00000000       1.00000000     0000000011   1.00000000
   1.00000000       1.00000000     0000000012   1.00000000
       1       2       4       3
       3       4       6       5
       5       6       8       7
       7       8      10       9
       9      10      12      11
```

Figure 21. Example of Tecplot data output file with geometry data included.

7.18.4 Contour data output files (*filen.number_type_node.suffix*)

All the ASCII (formatted) node data files for AVS (suffix ‘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 (suffix ‘avsx’) 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 “ : ”.

All of the node data files for Surfer (suffix ‘csv’) contain a single header line containing:

A label/unit string for each data component separated by “,”

followed by for each node

the associated node data (described by data type below), delimited by “ , ”.

All the node data files for Tecplot (suffix ‘dat’) contain the following headers:

Header line with code version number, date, time and problem title

Filetype header when keyword “grid” is used

Variable header (the variable header is only output to the first data file unless keyword ‘grid’ is used)

Zone header with time (not output for material property files if keyword ‘grid’ is used)

or

Zone header with time and grid specification and type if keyword geom is used

followed by for each node

the associated node data (described by data type below).

The dual or dpdp values for each of these fields will be written to a file with “dual” in the file name .

Contour File Content	Data Type Designation	Output Parameters*
Material properties	mat, mat_dual	Permeability in each active direction (m^2)
		Thermal conductivity in each active direction $\left(\frac{W}{m \cdot K}\right)$
		Porosity
		Rock bulk density (kg/m^3)
		Rock specific heat $\left(\frac{MJ}{kg \cdot K}\right)$
		Capillary pressure (MPa)
		Relative permeability model
		Capillary pressure model.
		Note: Output to the material properities file is dependent on the simulation being performed.
Scalar parameters	sca, sca_dual	Zone number
		Pressure (MPa) - liquid, vapor, capillary
		Temperature (°C)
		Saturation
		CO_2 (Water volume fraction, Liquid CO_2 fraction, Gaseous CO_2 fraction, Dissolved CO_2 mass fraction, Phase state of CO_2
		Head (m)
		Porosity
		Density (kg/m^3) - liquid, vapor
		Permeability in each active direction (m^2)

		Source (kg/s)
		Mass Flux (kg/s)
		Volume weighted Mass Flux (kg/s/m^3)
		Displacement (m) for each specified direction
		Stress (MPa) for each specified direction
		Volume Strain
Vector parameters	vec, vec_dual	Volume Flux ($m^3/m^2/s$) - liquid, vapor
Solute concentrations	con, con_dual	Species concentration (moles/kg)

*Output parameters are dependent upon the simulation being performed and keywords specified in the **cont** macro.

```
11 1 1 1 1 1 1 1 1 1 1
Permeability (m**2) in X, (m**2)
Permeability (m**2) in Y, (m**2)
Thermal Conductivity (W/m*K) in X, (W/m*K)
Thermal Conductivity (W/m*K) in Y, (W/m*K)
Porosity, (non dim)
Rock bulk density (kg/m**3), (kg/m**3)
Rock specific heat (MJ/kg*K), (MJ/kg*K)
Capillary pressure (MPa), (MPa)
Relative permeability model, (flag)
Capillary pressure model, (flag)
0000000001   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000002   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000003   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000004   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000005   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000006   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000007   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000008   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000009   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000010   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000011   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000012   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
```

Figure 22. Example of AVS material properties output file.

```
node  : Permeability (m**2) in X  : Permeability (m**2) in Y  : Thermal Conductivity (W/
m*K) in X  : Thermal Conductivity (W/m*K) in Y  : Porosity  : Rock bulk density (kg/m**3)
: Rock specific heat (MJ/kg*K)  : Capillary pressure (MPa)  : Relative permeability model
: Capillary pressure model
0000000001 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000002 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000003 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000004 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000005 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000006 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000007 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000008 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000009 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000010 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000011 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
0000000012 :  1.000000E-12 :  1.000000E-12 :  1.000000E-05 :  1.000000E-05 :  0.800000
:    1000.00 :    0.00000 :    1 :    1
```

Figure 23. Example of AVS Express material properties output file.

```
node ,  Permeability (m**2) in X ,  Permeability (m**2) in Y ,  Thermal Conductivity (W/
m*K) in X ,  Thermal Conductivity (W/m*K) in Y ,  Porosity ,  Rock bulk density (kg/m**3)
,  Rock specific heat (MJ/kg*K) ,  Capillary pressure (MPa) ,  Relative permeability model
,  Capillary pressure model
0000000001,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000002,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000003,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000004,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000005,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000006,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000007,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000008,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000009,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000010,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000011,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
0000000012,   1.000000E-12,   1.000000E-12,   1.000000E-05,   1.000000E-05,   0.800000
,     1000.00,     0.00000,     1,     1
```

Figure 24. Example of Surfer material properties output file.

```
TITLE = "FEHM V3.1gf 12-02-15 QA:NA     02/15/2012  11:29:32 Unsaturated Diffusion tests"
VARIABLES =  "node" "Permeability (m**2) in X" "Permeability (m**2) in Y" "Thermal
Conductivity (W/m*K) in X" "Thermal Conductivity (W/m*K) in Y" "Porosity" "Rock bulk
density (kg/m**3)" "Rock specific heat (MJ/kg*K)" "Capillary pressure (MPa)" "Relative
permeability model" "Capillary pressure model"
0000000001   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000002   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000003   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000004   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000005   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000006   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000007   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000008   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000009   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000010   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000011   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000012   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
```

Figure 25. Example of Tecplot material properties output file without geometry keyword.

```
TITLE = "FEHM V3.1gf 12-02-15 QA:NA     02/15/2012  07:58:21 Unsaturated Diffusion tests"
FILETYPE = "SOLUTION"
VARIABLES =  "node" "Permeability (m**2) in X" "Permeability (m**2) in Y" "Thermal
Conductivity (W/m*K) in X" "Thermal Conductivity (W/m*K) in Y" "Porosity" "Rock bulk
density (kg/m**3)" "Rock specific heat (MJ/kg*K)" "Capillary pressure (MPa)" "Relative
permeability model" "Capillary pressure model"
0000000001   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000002   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000003   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000004   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000005   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000006   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000007   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000008   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000009   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000010   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000011   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
0000000012   1.000000E-12   1.000000E-12   1.000000E-05   1.000000E-05   0.800000
1000.00       1.000000E-03    0.00000    1    1
```

Figure 26. Example of Tecplot material properties output file with ‘grid’ keyword.

```
VARIABLES =  "X coordinate (m)" "Y coordinate (m)" "node" "Permeability (m**2) in X"
"Permeability (m**2) in Y" "Thermal Conductivity (W/m*K) in X" "Thermal Conductivity (W/
m*K) in Y" "Porosity" "Rock bulk density (kg/m**3)" "Rock specific heat (MJ/kg*K)"
"Capillary pressure (MPa)" "Relative permeability model" "Capillary pressure model"
ZONE T = "Material properties", N =       12, E =        5, DATAPACKING = POINT, ZONETYPE
= FEQUADRILATERAL
  0.00000000        0.00000000      0000000001   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000        0.00000000      0000000002   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  0.00000000       0.200000000      0000000003   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000       0.200000000      0000000004   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  0.00000000       0.400000000      0000000005   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000       0.400000000      0000000006   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  0.00000000       0.600000000      0000000007   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000       0.600000000      0000000008   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  0.00000000       0.800000000      0000000009   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000       0.800000000      0000000010   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  0.00000000        1.00000000      0000000011   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
  1.00000000        1.00000000      0000000012   1.000000E-12   1.000000E-12   1.000000E-
05   1.000000E-05   0.800000        1000.00       1.000000E-03    0.00000    1    1
       1       2       4       3
       3       4       6       5
       5       6       8       7
       7       8      10       9
       9      10      12      11
```

Figure 27. Example of Tecplot material properties output file with ‘geom’ keyword.

7.18.5 SURFER and TECPLOT contour output files with specified ‘zone’

The content of the contour files generated when the ‘zone’ keyword is used in the **cont** macro is the same as that for the regu;ar output with the following exceptions:

Geometry keywords, ‘geom’ and ‘grid’, are ignored;

Data is output only for the nodes in the specified zones;

For “surfer”, a separate file is written for each output zone and the file names generated include the output zone number (using 4 digits, e.g., 0001);

For tecplot, the simulation time is written into a text string, and the zone headers include only the zone number, and output is separated by zone.

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 tyming).

8.2 Compiler Requirements

FEHM Version 2.30 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 Installing a compiled executable

Copy the executable to a location on the current search path. Refer to the Installation Test Plan for the FEHM Application Version 2.30 (10086-ITP-2.30-00).

8.6.2 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_v2.30
```

It should be noted that FEHM uses the GZSOLVE Application (ECD-97) reuse components, solve_new, solve_rdof, and slvesu. The GZSOLVE subroutines are compiled directly into this version of FEHM.

8.6.3 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 2.30 (10086-VTP-2.30-00). See the FEHM VTP for a discussion of the tests performed and their results.

8.6.4 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 2.30 (10086-ITP-2.30-00).

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 86). 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 116) is more convenient. Initial values for noncondensible gas are handled in the **ngas** (page 109) 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 186).

Boundary conditions. Fluid and heat flow boundary conditions can be prescribed through control statements **pres**, **boun** (page 45), **flow** (page 71), and **hflx** (page 81). 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 161), **ngas**, and **air** (page 43).

Rock or Media Properties. These are found in the **rock** (page 141) and **perm** control statements.

Fluid Properties. These are found in control statement **eos** (page 67), 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 132).

Mesh Geometry and Nodal Coordinates. This geometry information is found in control statements **coor** (page 59) and **elem** (page 65). 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 185).

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 60).

Advanced Iteration Control. Reduced degree of freedom methods are invoked with the **iter** (page 87) 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.

Table VI. Required and Optional Macros by Problem Type

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_v2.30

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_v2.30** 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 28 shows the control file that would produce the same results as the terminal responses discussed below and illustrated in Fig. 29. Files that are not needed for

```
/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 28. Input control file for heat conduction example.

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 inzone 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. 29 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. 29. 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_v2.30sun

version  FEHM V2.30sun 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 29. 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 29. 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 29. 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 °C, is modeled after a surface temperature, T_s = 100 °C, is imposed at time, t = 0 (Fig. 30).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. 31. Only a quarter of the square needs to be modeled because of problem symmetry.

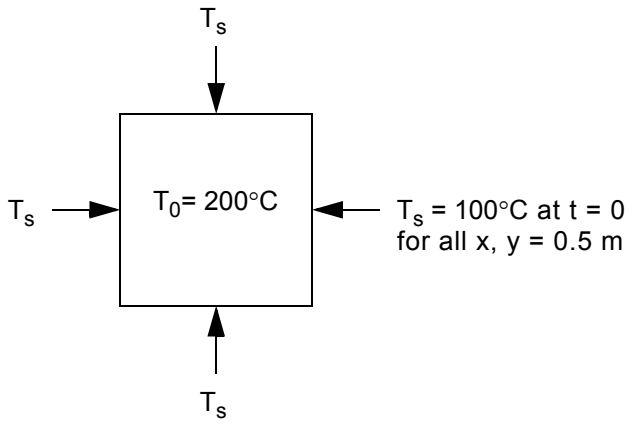

Figure 30. Schematic diagram of 2-D heat conduction problem.

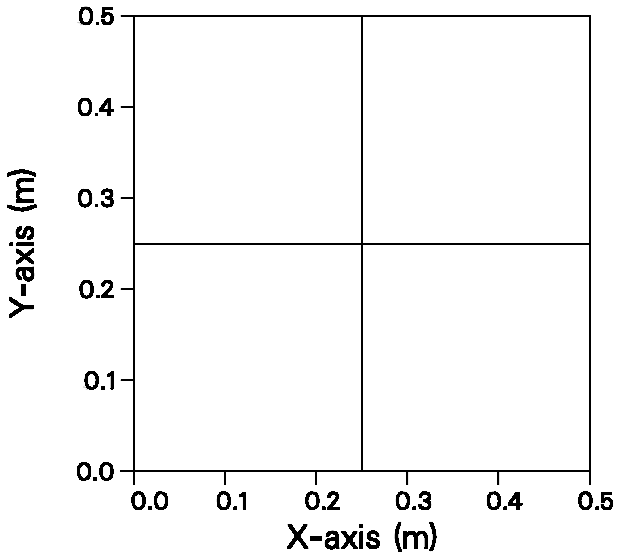

Figure 31. Finite element mesh used for 2-D heat conduction problem.

Table VII. Input Parameters for the 2-D Heat Conduction Problem		
Parameter	Symbol	Value
Rock thermal conductivity	κ_r	2.7 $\frac{W}{m \cdot K}$
Rock density	ρ_r	2700 kg/m^3
Rock specific heat	C_r	1000 $\frac{J}{kg \cdot K}$
Width	a	0.5 m
Length	b	0.5 m
Initial temperature	T_0	200 °C
Surface temperature for all x, y = 0.5 m	T_s	100 °C
Rock thermal diffusivity	$\kappa = \frac{\kappa_r}{\rho_r C_r}$	

The input file (see Fig. 32) 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. 33.

The analytical solution for 2-D heat conduction (Carslaw and Jaeger, 1959) is given by

$$T = T_s + \frac{16(T_0 - T_s)}{\pi^2} \sum_{m=0}^{\infty} \sum_{n=o}^{\infty} \frac{(-1)^{m+n}}{(2m+1)(2n+1)} \cos\frac{(2m+1)\pi x}{2a} \cos\frac{(2n+1)\pi y}{2b} e^{-\alpha_{m,n} t}$$

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 34 shows a plot of the simulation results compared to the analytical solution for the selected output nodes at x = y = 0. m and x = y= 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
    0.0054.001000  101994 02

ctrl
    40   1.e-0408
    1    9    1           1

    1.0  0.0  1.0
    10   1.0  0.000050.005
    1    0
coor   Feb 23, 1994 11:39:40
    9
    1         0.0.50  0.
    2         0.250.50 0.
    3         0.500.50 0.
    4         0.0.25  0.
    5         0.250.25 0.
    6         0.500.25 0.
    7         0.0.       0.
    8         0.25 0.  0.
    9         0.50 0.  0.

elem
    4         4
    1         45          2          1
    2         56          3          2
    3         78          5          4
    4         89          6          5

stop
```

Figure 32. 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 33. 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 ncon43 available        43
 storage needed for nop43 available        46
 storage needed for a matrix33 available        33
 storage needed for b matrix33 available        46
 storage needed for gmres81 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 33. FEHM output from the 2-D heat conduction example. (Continued)

```
 Enthalpy discharge this time step:0.297800E+02 MJ (0.689352E-01 MJ/s)
 Enthalpy input this time step:0.000000E+00 MJ (0.000000E+00 MJ/s)
 Total enthalpy discharge:0.297800E+02 MJ (0.689352E-01 MJ/s)
 Total enthalpy input:0.297800E+02 MJ (0.689352E-01 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),  -0.100326E+01 (energy)
**********************************************************************
 Time Step       11

    .

    .

    .
**********************************************************************
 Time Step      801

 Timing Information
         Years            Days        Step Size (Days)
      0.109515E-01      0.400005E+01      0.500000E-04
  Cpu Sec for Time Step =      0.    Current Total =     4.533

                      Equation Performance
  Number of N-R Iterations:         1
  Avg # of Linear Equation Solver Iterations:   2.0
  Number of Active Nodes:        9.
  Total Number of Newton-Raphson Iterations:       801 , Solver:      2402
                      Largest Residuals
EQ1 R=  0.9774E-13 node=     7 x= 0.000     y= 0.000     z= 1.000
   Node   Equation 1 Residual   Equation 2 Residual
             7     0.977369E-13          0.186062E-04
             5     0.621566E-13          0.930309E-05

                      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     100.230    0.              0.
        5  10.000   0.00    0.000     100.115    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:0.675565E+02 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)

 Enthalpy discharge this time step:0.455636E-05 MJ (0.105471E-05 MJ/s)
 Enthalpy input this time step:0.000000E+00 MJ (0.000000E+00 MJ/s)
```

Figure 33. 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 33. FEHM output from the 2-D heat conduction example. (Continued)

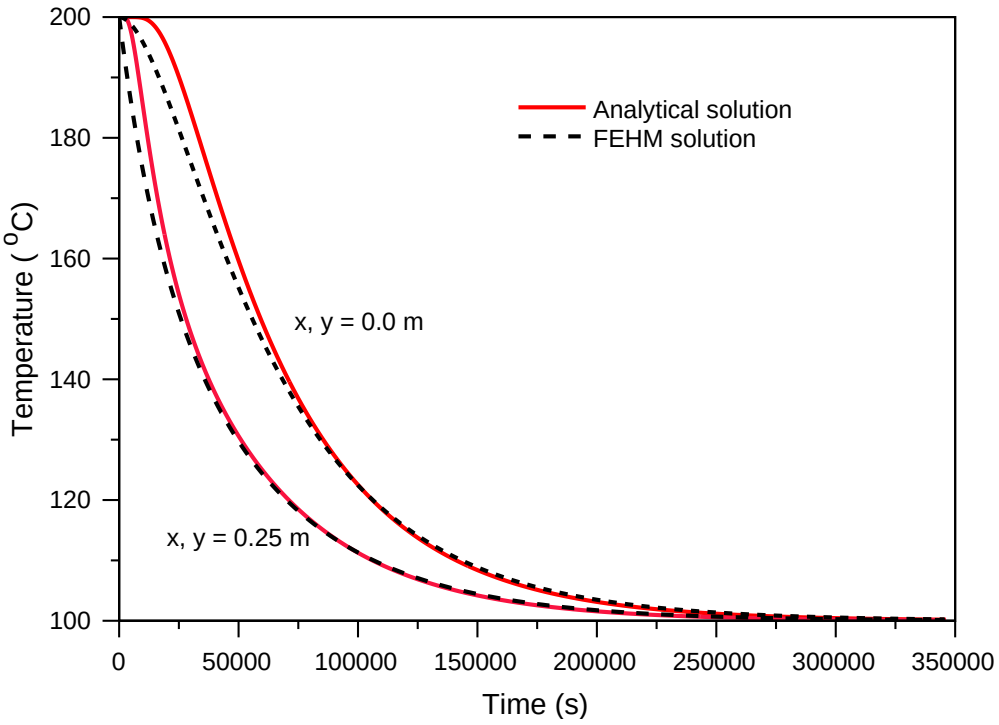

Figure 34. Comparison of analytical and model solution for 2-D heat conduction.

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. 35. Of particular note are the variable initial temperature field, provided to the code through a read file (see Section 5.6 on page 19), and the prescribed pressure and temperature on the right boundary.A partial listing of the input file is provided in Fig. 36. 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.

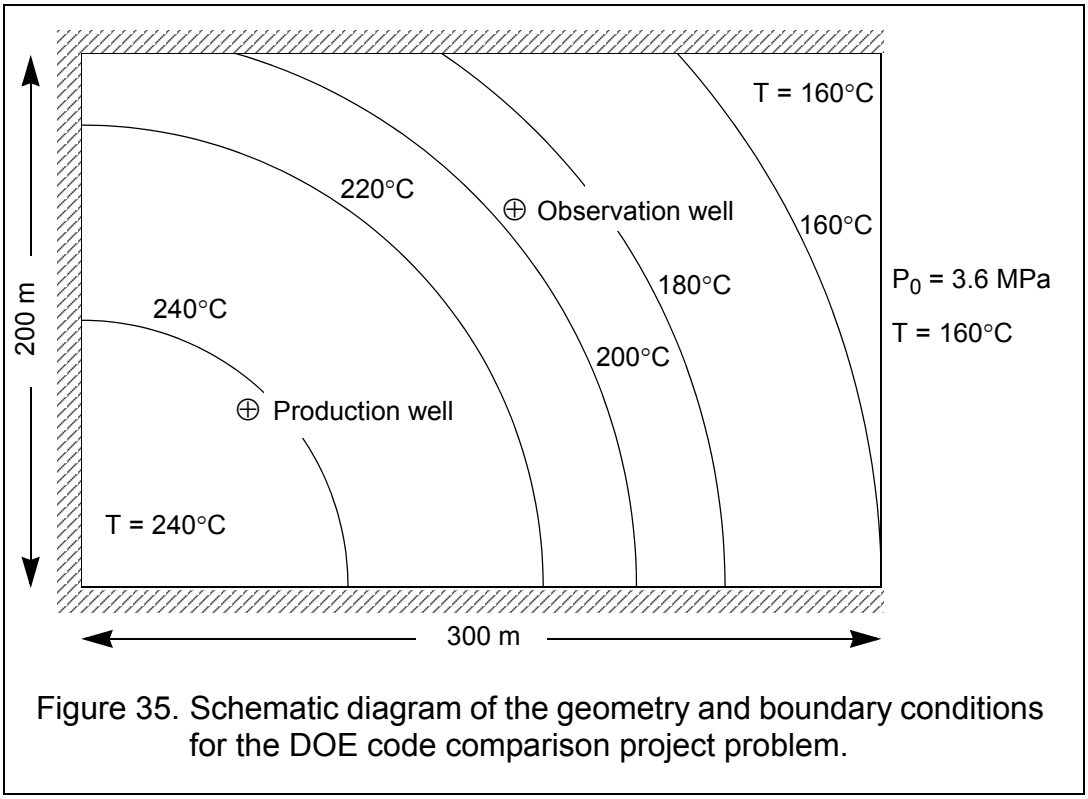

Figure 35. Schematic diagram of the geometry and boundary conditions for the DOE code comparison project problem.

```
*** 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 36. FEHM input file for DOE problem.

Table VIII.Input Parameters for the DOE Code Comparison Project Problem

Parameter	Symbol	Value
Reservoir permeability	k	2.5×10^{-14} m^2
Reservoir porosity	ϕ	0.35
Rock thermal conductivity	κ_r	$1 \frac{W}{m \cdot K}$
Rock density	ρ_r	2563 kg/m^3
Rock specific heat	C_r	$1010 \frac{J}{kg \cdot K}$
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	$0.05 \frac{kg}{m \cdot s}$
Initial Pressure	P_o	3.6 MPa
Production well coordinates:		x = 62.5 m, y = 62.5 m
Observation well coordinates:		x = 162.5 m, y = 137.5 m

Initial temperature distribution (T in °C, r in m):

$$T(x, y, 0) = \begin{cases} 240 & r \le 100 \\ 240 - 160\left(\frac{r-100}{200}\right)^2 + 80\left(\frac{r-100}{200}\right)^4 & 100 < r < 300 \\ 160 & r \ge 300 \end{cases}$$

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. 37, are in excellent agreement with the other codes. The results for the outlet pressure and pressure at an observation well 125 m distant, Fig. 41, 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. 39 and Fig. 40.

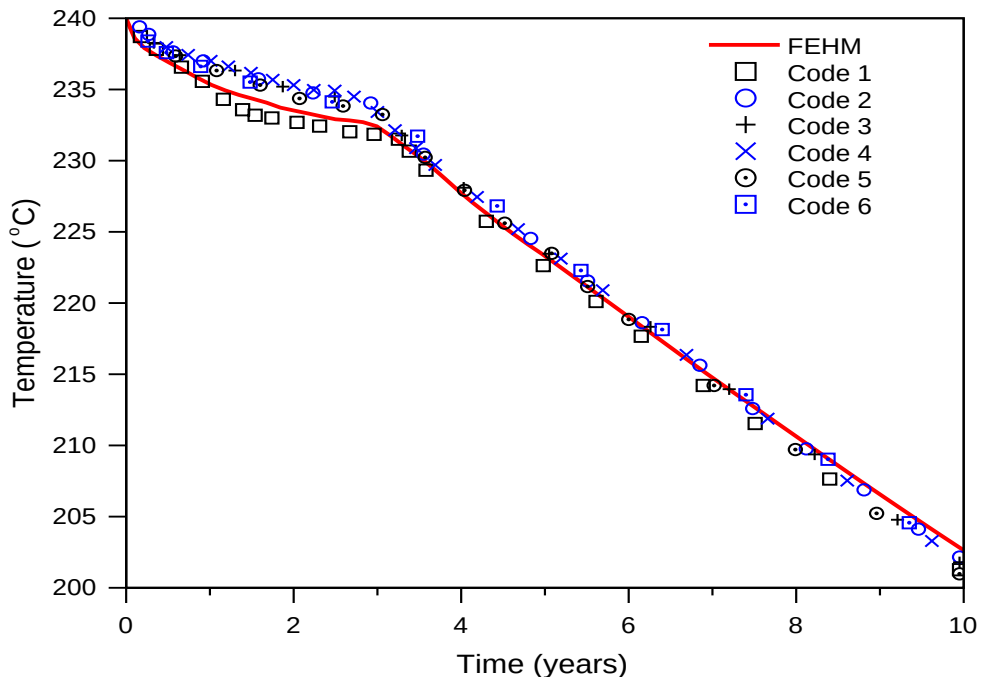

Figure 37. Comparison of FEHM production well temperatures with results from other codes.

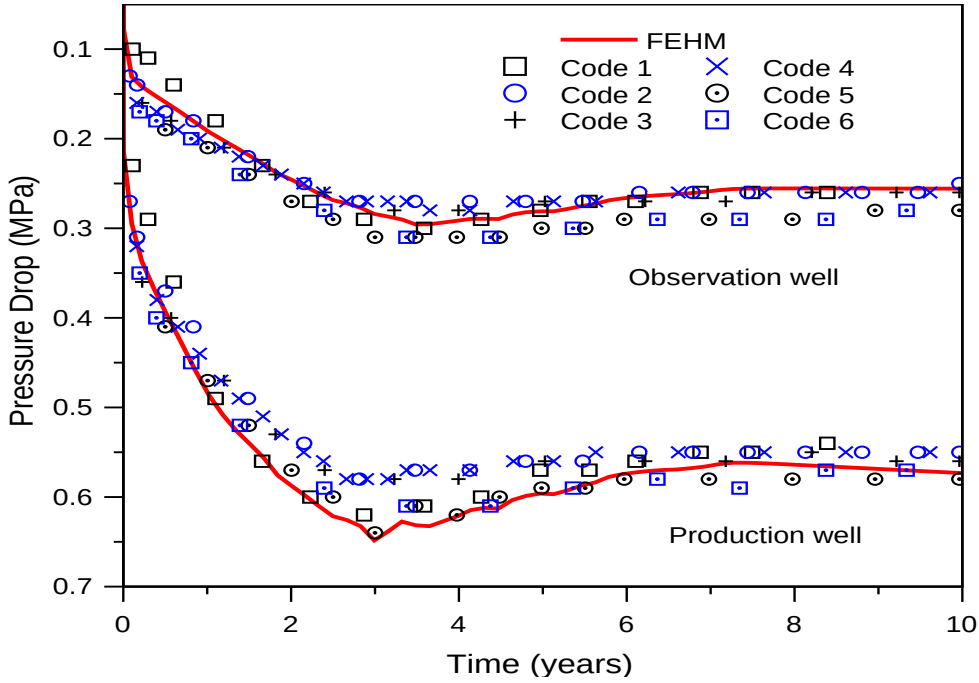

Figure 38. Comparison of FEHM production and observation well pressure drops with results from other codes.

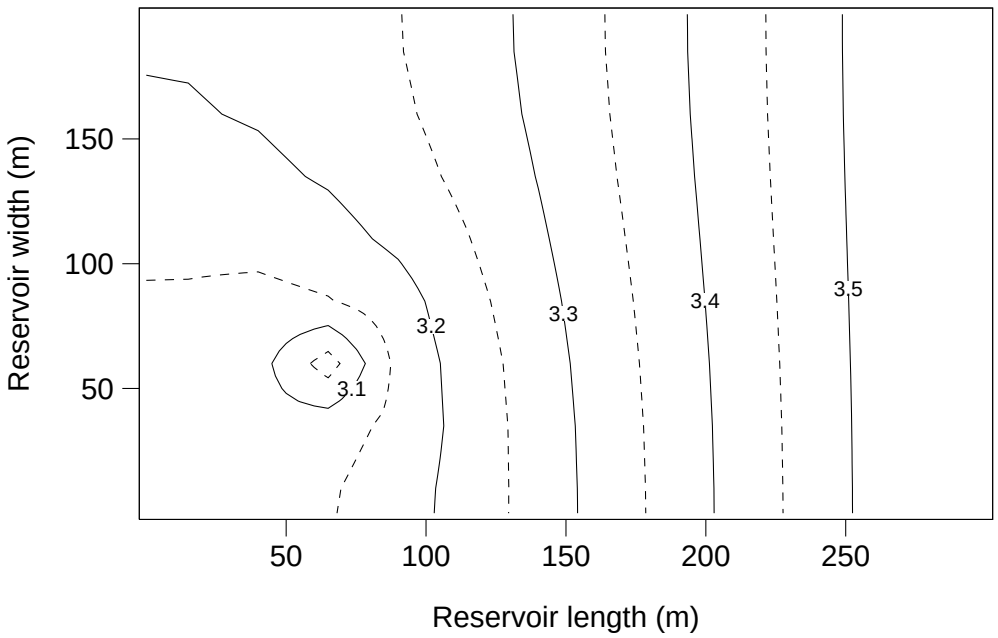

Figure 39. Contour plot of pressure at ten years for the DOE problem.

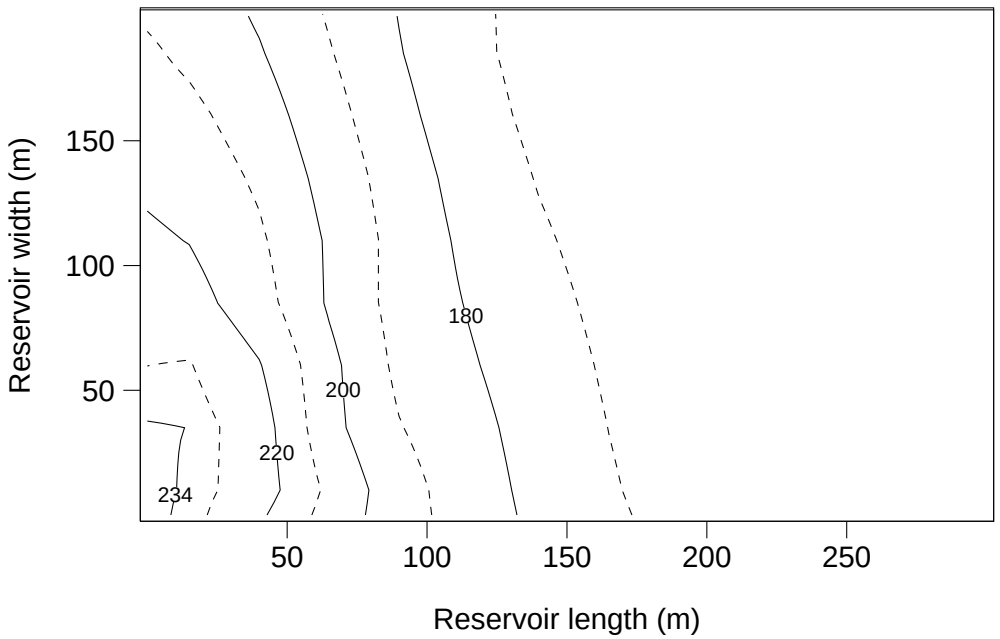

Figure 40. Contour plot of temperature at ten years for the DOE problem.

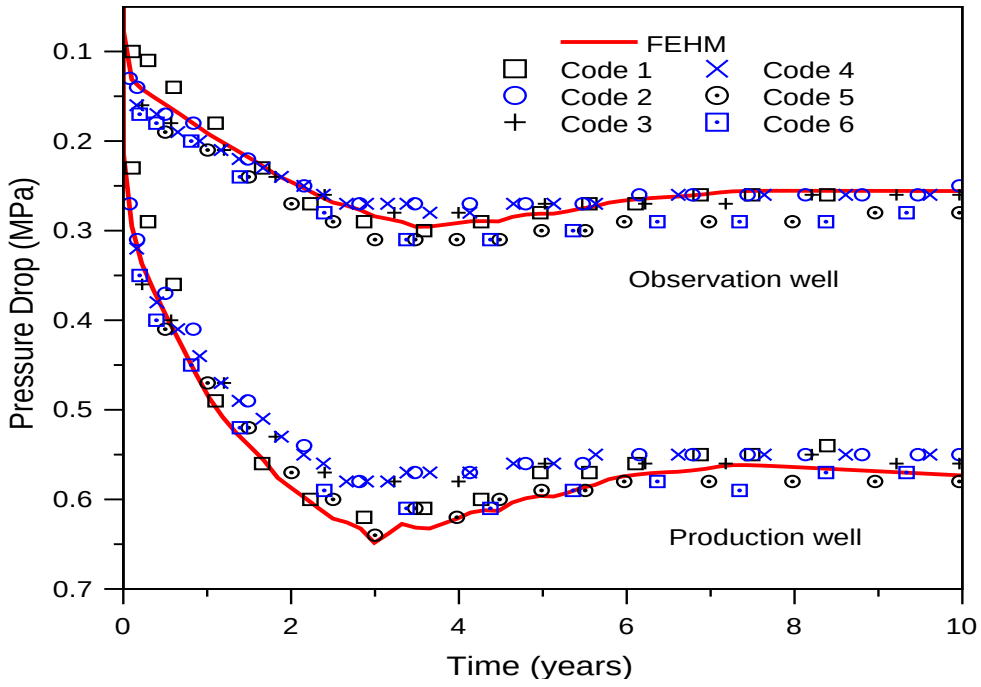

Figure 41. Comparison of FEHM production and observation well pressure drops with results from other codes.

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:

$$CoEDTA_{(aq)} \Leftrightarrow Co_{(aq)} + EDTA_{(aq)}; K_{eq} = 10^{-18}\text{kg/mol} \quad (16)$$

$$Fe_{(aq)} + EDTA_{(aq)} \Leftrightarrow FeEDTA_{(aq)} ; K_{eq} = 6.31\text{x}10^{27}\text{kg/mol} \quad (17)$$

$$Co_{(aq)} \Leftrightarrow Co_{(s)} ; k_f = 2.78\text{x}10^{-4}\ \text{s}^{-1}, k_r = 5.48\text{x}10^{-5}\ \text{s}^{-1} \quad (18)$$

$$CoEDTA_{(aq)} \Leftrightarrow CoEDTA_{(s)} ; k_f = 2.78\text{x}10^{-4}\ \text{s}^{-1},$$

$$k_r = 5.21\text{x}10^{-4}\ \text{s}^{-1} \quad (19)$$

$$FeEDTA_{(aq)} \Leftrightarrow FeEDTA_{(s)} ; k_f = 2.78\text{x}10^{-4}\ \text{s}^{-1},$$

$$k_r = 6.5\text{x}10^{-4}\ \text{s}^{-1} \quad (20)$$

$$CoEDTA_{(s)} + Fe(OH)_{3(s)} \Rightarrow Co_{(s)} + FeEDTA_{(s)};$$

$$k_f = 3.5x10^{-6}\ \text{s}^{-1} \quad (21)$$

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 42 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. 43.

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. 44 and Figure 45.

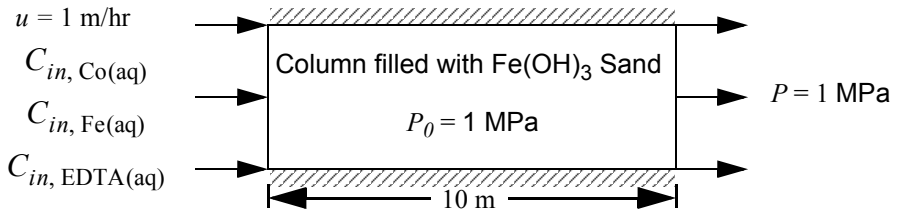

Figure 42. Schematic drawing of the geometry and boundary conditions for the cobalt transport problem.

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	$C_{in,\,\mathrm{Co}}$	3.1623x10^{-5} M
Fe Inlet Concentration	$C_{in,\,\mathrm{Fe}}$	0 M
EDTA Inlet Concentration	$C_{in,\,\mathrm{EDTA}}$	3.1623x10^{-5} M
Boundary conditions:	At l = 0	u = 1 m/hr
	At l = 1	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 43. 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
** IDCPNT(IC),CPNTNAM(IC),IFXCONC(IC),CPNTPRT(IC) (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),IMMPRT(IM)(ID # and name of immoblie 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 43. 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 43. 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
coor   n/a
     202
     1    0.000001.000000.00000
     2    0.100001.000000.00000
     3    0.200001.000000.00000
     .
     .
     .
     200 9.800000.000000.00000
     201 9.900000.000000.00000
     202 10.000000.000000.00000

elem
     4    100
     1    10210321
     2    10310432
     3    10410543
     .
     .
     .
     98   1992009998
     99   20020110099
     100 201202101100

stop
```

Figure 43. FEHM input file for reactive transport problem (Continued).

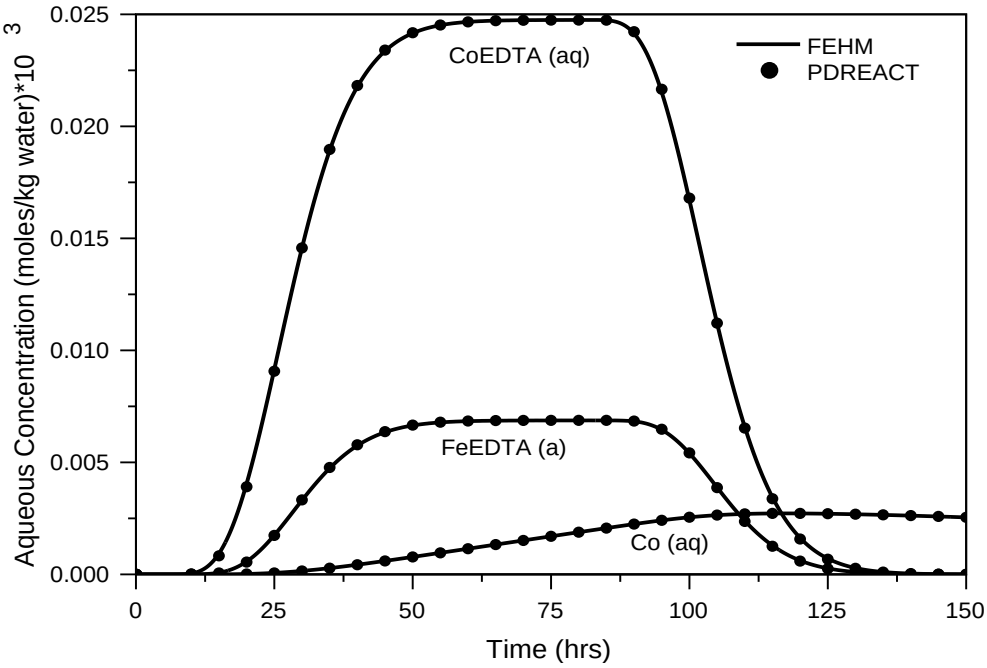

Figure 44. Comparison of FEHM and PDREACT for the breakthrough curves of aqueous species.

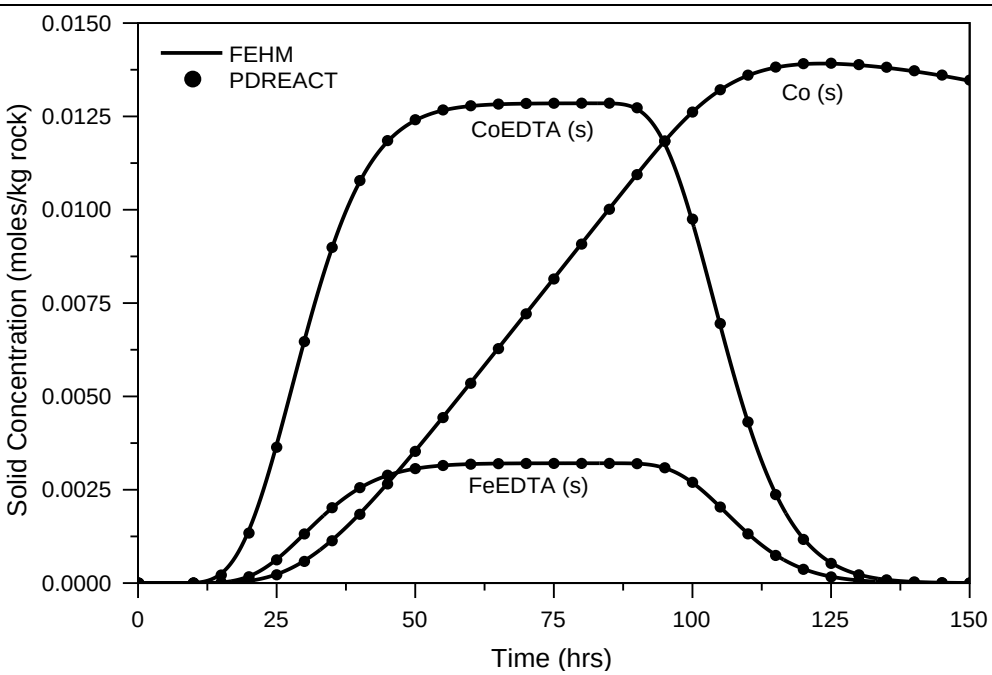

Figure 45. Comparison of FEHM and PDREACT for the exit concentration versus time for solid species.

10.0 USER SUPPORT

For information and latest updates visit https://fehm.lanl.gov
For technical support email fehm-help@lanl.gov

Update August 8, 2013

FEHM UM V3 Appendix
For Software Release FEHM Version 3.2.0
July 11, 2013

This is an addendum to

Software Users Manual (UM) for FEHM Application Version 3.1.0

LA-UR-12-24493

Prepared by:

Terry Miller EES-16
Shaoping Chu EES-16

FEHM UM V3 Appendix
For Software Release FEHM Version 3.2.0

1 PURPOSE

The following document is an addendum to the Software Users Manual Version 3.1.0, LA-UR-12-24493 (UM V3.1) and This Appendix describes updates to the FEHM Application Version 3.2.0. The use of FEHM V3.2.0 has not changed from FEHM V3.1.0 except to add or improve functionality. Information in this document is considered supplemental to UM V3.1 and also applies to earlier version 3 releases of the User Manual (UM V3). For detail on how to use the FEHM software refer to the UM V3.1 or refer to the on-line Wiki pages found at https://ancho.lanl.gov/fehm/trac/wiki (currently restricted to Los Alamos National Laboratory (LANL) employees and collaborators). Publicly available Documentation and Publications can be found on-line at https://fehm.lanl.gov/ (unrestricted).

FEHM V3.2.0 (July 2013) adds to the previous releases of FEHM (V3.1.0 April 2012, and V2.3 January 2008). This FEHM software is a continuation of QA work performed for the Yucca Mountain Project (YMP) under Software Configuration Control Request (SCCR) (Software Tracking Numbers STN: 10086-2.21-00 August 2003, V2.22, STN 10086-2.22-01, V2.23, STN 10086-2.23-00, V2.24-01, STN 10086-2.24-01, and V2.25, STN 10086-2.25-00).

2 FEHM V3.2.0 APPLICATION OVERVIEW

The FEHM Application is used to simulate groundwater and contaminant flow and transport in deep and shallow, fractured and un-fractured porous media throughout the US DOE complex. FEHM has proved to be a valuable asset on a variety of projects of national interest including Environmental Remediation of the Nevada Test Site, the LANL Groundwater Protection Program, geologic CO2 sequestration, Enhanced Geothermal Energy (EGS) programs, Oil and Gas production, Nuclear Waste Isolation, and Arctic Permafrost.

The following capabilities describe the FEHM V3.2.0 Application.

- 3-dimensional complex geometries with unstructured grids
- saturated and unsaturated media
- simulation of production from gas hydrate reservoirs
- simulation of geothermal reservoirs
- non-isothermal, multi-phase flow of gas, water, oil
- non-isothermal, multi-phase flow of air, water
- non-isothermal, multi-phase flow of CO2, water
- multiple chemically reactive and sorbing tracers
- preconditioned conjugate gradient solution of coupled linear equations
- fully implicit, fully coupled Newton Raphson solution of nonlinear equations
- double porosity and double porosity/double permeability capabilities
- control volume (CV) and finite element method (FE) methods
- coupled geomechanics (THM) problems (fluid flow and heat transfer coupled with stress/deformation) including non-linear elastic and plastic deformation, nonlinear functional dependence of rock properties (e.g. permeability, porosity, Young's modulus) on pressure, temperature and damage/stress

3 FEHM V3.2.0 RELEASE SUMMARY

All changes to software behavior and software document UM V3.1 are detailed in the following section 6. There are no known effects that change results in models run using the previous version FEHM V3.1. The changes did not affect results in the distributed FEHM V&V test suite.

The major changes in the FEHM application from V3.1 to this release V3.2, are improvements to geothermal, CO2, and stress capabilities. All changes are described in detail in release notes and Appendix documents posted on the fehm.lanl.gov web page. Enhancements to this FEHM release include:

- Additional stress models have been added, including stress permeability for shear-stimulated fractures and settings for incremental shear for porosity damage.
- Enhanced functionality for C02 problems have been added. The User Manual documentation has been updated for both "carb" and "grad". Output for macro "rlpm" has been generalized to include both co2 gas/water and co2 liquid/water.
- For mass, energy balances in porous media, "anpe" (anisotropic permeability) has been modified to work with dual or dpdp or gdpm or gdkm for the isothermal case. The macro "anpe" works with "trac" for transport and "gdkm" works with isothermal and thermal.
- The macro "trxn" is a new macro, which does not add functionality but is a user-friendly interface to "trac" and "rxn".

4 FEHM V3.2.0 V&V TEST SUITE

Before distribution of FEHM software, tests are executed and verified as acceptable on LANL computers with operating systems Linux, Mac OSX, and WINDOWS. FEHM V3.2.0 test results do not differ from FEHM V3.1 results. The overall validation effort for the FEHM software consists of a suite of directories and scripts that test the model whenever possible, against known analytical solutions of the same problem. The test suite was developed under YMP QA for FEHM RD.10086-RD-2.21-00 and is provided to modelers along with the FEHM application.

The following is a summary of new tests that have been incorporated into the FEHM_VVSECT script test suite V3.2.0 and is based on a comparison to those tests documented for FEHM V2.21.

The following is a partial list of the added tests (and associated test name).

- macro co2 with co2pres, co2flow, and co2frac (co2test)
- macro evap (evaporation)
- macros ngas and trac (wvtest)
- infiltration for ecm and dpm models with Richard’s equation (infiltration*)
- new concentration dependent density formulation (cden_test)
- concentration flux output option (cflxz_test)
- new stress permeability model (stress_prem)
- using the rlpm macro instead of rlp macro, using equation and table input.
- using trxn format input in place of the trac and rxn macros.

5 FEHM V3.2.0 DISTRIBUTION

FEHM V3.2.0 is compiled, tested, and made available for operating systems Linux, Mac OSX, and WINDOWS. Compressed archive files are created from each of the V&V Test suites for distribution with the FEHM executable.

The operating system and compiler for this FEHM V3.2.0 release are:
Linux (64-bit): compiled with gfortran 4.5.1
MAC OSX: compiled with gfortran 4.8.1
PC Windows (32-bit): compiled with Intel Parallel Studio XE 2011
PC Windows (64-bit): compiled with Intel Parallel Studio XE 2011

6 MACRO CONTROL STATEMENTS (SUPPLEMENT TO UM V3 SECTION 6)

The input specifications for the following macros (control statements) in FEHM V3.2.0 have been modified to enhance or clarify use during the modeling process. Each macro description is considered supplemental to documentation in UM V3, Section 6.2, and each description is complete.

The following updated macros are listed in alphabetic order.

6.1 ANPE

This update supplements UM V3.1 Section 6.2.8

Anisotropic permeability input, adds cross terms to the **perm** macro.

The ANPE keyword implements a flux-continuous anisotropic permeability tensor with cross terms. The cross terms can either be input directly or grid rotation angles inputted and the cross terms calculated by FEHM. FEHM implements the method presented by Lee et al 2002.

Lee et al., 2002, Implementation of a Flux-Continuous Finite-Difference Method for Stratigraphic Hexahedron Grids, SPE Journal, Volume 7, Number 3, DOI 10.2118/80117-PA.

The ANPE is incompatible with keywords GDKM, GDPM, DUAL, and DPDP.

Group 1 - JA, JB, JC, ANXY, ANXZ, ANYZ (JA, JB, JC - defined on page 33 of UM V3.1)

Input Variable	Type	Def	Description
ANXY	real	1.e-30	Anisotropic permeability in the xy-direction (m2).
ANXZ	real	1.e-30	Anisotropic permeability in the xz-direction (m2).
ANYZ	real	1.e-30	Anisotropic permeability in the yz-direction (m2).

6.2 CARB (OPTIONAL)

This update supplements UM V3.1 Section 6.2.11

Macro carb is used to set up a CO_2 problem. Input following the problem type is grouped using sub keywords.

Group 1- IPRTYPE

Group 1 - KEYWORD

KEYWORD “co2pres”

JA, JB, JC, PHICO2, TCO2, ICES

KEYWORD “co2flow”

JA, JB, JC, SKTMP, ESKTMP, AIPED, IFLG_FLOWMAC

KEYWORD “co2diff”

JA, JB, JC, DIFF, TORTCO2

KEYWORD “co2frac”

JA, JB, JC, FW, FL, YC, CSALT, INICO2FLG

KEYWORD “userprop”

DENC, DENCP, DENCT, ENC, ENCP, ENCT, VISC, VISCP, VISCT

DENW, DENWP, DENWT, ENW, ENWP, ENWT, VISW, VISWP, VISWT

KEYWORD “brine”

Input is terminated with KEYWORD “*end carb*” or “*endcarb*”

Input Variable	Type	Def	Description
IPRTYPE	int		The settings for IPRTYPE are: 1 = Water only problem (2 DOFs) 2 = CO_2 only problem (2 DOFs) 3 = CO_2-water problem, no solubility (3 DOFs) 4 = CO_2-water problem, with solubility (4 DOFs) 5 = CO_2-water-air w/ solubility (5 DOFs)
KEYWORD	char		Remaining input is grouped using sub-macro keywords.
KEYWORD “end carb” or “endcarb”	End of **carb** input.		
KEYWORD “co2pres”	Set up the initial pressure (uses the same format as the pres macro)		
PHICO2	real	0.	Initial CO_2 pressure (MPa).
TCO2	real	0.	Initial CO_2 temperature (°C)
ICES	int	0	Initial guess for phase state of CO_2 (actual phase will be calculated internally), ICES settings are: 1 = liquid 2 = two-phase liquid and vapor 3 = vapor 4 = super-critical CO_2.

KEYWORD "co2flow"	Set up co2 flow boundary conditions (similar to the flow macro used to set up water boundary conditions)		
SKTMP	real	0.	Values for SKTMP are conditional, the following apply: When IFLG_FLOWMAC = 1, 2, 3, 4, 5 or 9 CO_2 flowing pressure (MPa). If SKTMP = 0 the initial value of pressure will be used for the flowing pressure. When IFLG_FLOWMAC = 6 or 7 water mass flow rate (kg/s). When IFLG_FLOWMAC = 8 CO_2 flowing saturation.
ESKTMP	real	0.	Values for ESKTMP are conditional, the following apply: When IFLG_FLOWMAC = 1, 2, 3, 6, 7 or 9 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 (°C) and the enthalpy calculated accordingly. When IFLG_FLOWMAC = 4 or 5 CO_2 flowing saturation. When IFLG_FLOWMAC = 8 mass fraction of CO2
AIPED	real	0.	Values for AIPED are conditional, the following apply: When IFLG_FLOWMAC = 1, 2, 4 or 9 CO_2 impedance parameter. When IFLG_FLOWMAC = 5 or 6 value is ignored. When IFLG_FLOWMAC = 7 CO_2 mass fraction in water. When IFLG_FLOWMAC = 8 Water mass flow rate (kg/s).
IFLG_FLOWMAC	int	0	Flag specifying boundary condition type, IFLG_FLOWMAC values: 1 = Constant pressure boundary condition with inflow or outflow allowed. AIPED is user specified 2 = Constant pressure boundary condition with only outflow allowed. AIPED is user specified 3 = Constant pressure boundary condition. AIPED is calculated in the code based on block geometric parameters. 4 = Constant pressure and constant saturation boundary condition. AIPED is user specified 5 = Constant pressure and constant saturation boundary condition. AIPED is calculated in the code based on block geometric parameters. 6 = Constant free phase CO_2 mass flow rate boundary condition. 7 = Constant source of water with specified mass fraction of CO2 (kg/s) 9 = Partial explicit update of nonlinear part of CO_2 constant pressure
KEYWORD "co2diff"	Read CO_2 diffusivity in water		
DIFF	real	0.	Diffusion
TORTCO2	real	0.	Tortuosity for CO_2-water vapor diffusion.
KEYWORD "co2frac"	Read initial CO_2, air, water/brine saturation. FG, CO2/air-rich gas saturation (volume fraction), FG = 1 - FW - FL.		
FW	real	0.	Water-rich liquid saturation (volume fraction).
FL	real	0.	CO_2-rich super-critical/liquid phase saturation (volume fraction).
YC	real	0.	Mass fraction of CO_2 in the CO_2-rich phase.
CSALT	real	0.	Initial salt concentration in water for brine (ppm) (only used if "brine" keyword is invoked.
INICO2FLG	integer	0	Flag to override CO_2 fractions read from restart file. If set to 1 the input values are used instead of those read from the restart file.
KEYWORD "userprop"	Read user defined properties for CO_2 and brine		
DENC	real		CO_2 density (kg/m^3)
DENCP	real		Derivative of density with respect to pressure.
DENCT	real		Derivative of density with respect to temperature.
ENC	real		CO_2 enthalpy (MJ/kg).

ENCP	real		Derivative of enthalpy with respect to pressure.
ENCT	real		Derivative of enthalpy with respect to temperature.
VISC	real		CO_2 viscosity (Pa s)
VISCP	real		Derivative of viscosity with respect to pressure.
VISCT	real		Derivative of viscosity with respect to temperature.
DENW	real		Brine density (kg/m^3)
DENWP	real		Derivative of density with respect to pressure.
DENWT	real		Derivative of density with respect to temperature.
ENW	real		Brine enthalpy (MJ/kg).
ENWP	real		Derivative of enthalpy with respect to pressure.
ENWT	real		Derivative of enthalpy with respect to temperature.
VISW	real		Brine viscosity (Pa s)
VISWP	real		Derivative of viscosity with respect to pressure.
VISWT	real		Derivative of viscosity with respect to temperature.
KEYWORD "brine"		Invoke option for brine in the simulation. (salt-concentration dependent CO2 solubility)	

In the following example, zone 1 is injecting CO_2 dissolved water at 0.001 kg/s. The temperature is 20°C. The water has a dissolved CO_2 mass fraction of 0.3. The code will check internally whether the user specified mass fraction exceeds the equilibrium mass fraction calculated using the pressure and temperature values of the injection node. In case it does exceed that value, it is fixed at the equilibrium mass fraction. The user can specify a value of "zero" and the code will automatically fix the dissolved CO_2 mass fraction at the equilibrium value. Zone 2 is maintained at initial pressure using "aiped" calculated internally.

```
carb
    4

co2pres
     1      0      0      3.      20.      4
    -1      0      0      13      20.      4
    -2      0      0      .6      20.      4

co2frac
     1   0   0    1.0         0.0       0   100000      0
    -1   0   0    0.9465    .0535       0   0.          0.

co2flow
    -2 0 0    0          -20.    -1.e-1     3
    -1 0 0   -0.0001     -20.      0.        6

end carb
```

6.3 GRAD (OPTIONAL)

This update supplements UM V3.1 Section 6.2.42

Gradient model input.

Group 1 – NGRAD

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

Group 2 is repeated (NGRAD times) for each gradient model being defined.

Input Variable	Type	Description
NGRAD	int	Number of gradient models
IZONE_GRAD	int	Zone associated with model
CORDG	real	Reference coordinate (m)
IDIRG	int	Direction of gradient (1,2, or 3).
IGRADF	int	Variable to which gradient is applied. The settings for IGRADF are: 1 = initial pressure 2 = initial temperature 3 = initial saturation 4 = Fixed pressure 5 = Fixed enthalpy -5 = Fixed inflowing temperature 6 = Initial methane pressure 7 = Fixed Methane pressure 8 = depending on how hflx macro is configured this is either a temperature or a heat flux 9 = initial CO_2 pressure 10 = Fixed CO_2 cell pressure 11 = Pressure for secondary material in gdkm or gdpm model 12 = initial Temperature for matrix in gdkm or gdpm model NOTE: IGRADF = 4,5,-5,7 requires that the node is previously defined as a boundary node in a flow macro (or equivalent) IGRAFD = 11 or 12 requires gdkm or gdpm macro IGRADF = 8 requires a hflx macro
VAR0	real	Value of variable at reference point (m).
GRAD1	real	Gradient. Units: Pressure MPa/m, T degrees C/m, enthalpy KJ/kg/m, heat flux MW/m

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

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

6.4 STRS (OPTIONAL)

This update supplements UM V3.1 Section 6.2.85

The solid mechanical deformation module is invoked with this control statement.

Group 1- ISTRS, IHMS

Group 2- KEYWORDS

The remaining input is entered in subgroups defined by additional keywords. These keywords are all optional unless otherwise noted, but the user should be careful to ensure the problem is completely defined with the keywords selected.

Input associated with KEYWORDS is shown below and described in more detail in the following table. Unless otherwise specified, blank lines are not permitted.

KEYWORD “initcalc” (optional, no other input)

KEYWORD “bodyforce” (optional, no other input)

KEYWORD “fem” (optional, no other input)

KEYWORD *“reldisp”* (optional)

KEYWORD *“stresspor”* (optional)

KEYWORD “excess_she” (optional)

FRICTION_OUT, STRENGTH_OUT, PP_FAC_OUT

KEYWORD “permmodel” (optional)

IPERM, SPMF1, SPMF2, . . . SPMF13

(as many models as needed,one per line, terminated by a blank line)

JA, JB, JC, MODEL_NUMBER (JA, JB, JC – defined page 33 of UMV3)

KEYWORD “elastic” (required)

JA, JB, JC, ELASTIC_MOD, POISSON

KEYWORD “nonlinear” (optional)

NONLIN_MODEL_FLAG

If the value of NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young’s modulus and Poisson’s ratio:

E_INI, DEDT, POISSON_INI, DNUEDT

Else, if the value of NONLIN_MODEL_FLAG = 91then a table lookup is used:

YOUNG_TEMP_FILE

KEYWORD “plastic”

NUMPLASTICMODELS

The following are repeated NUMPLASTICMODELS times

PLASTICMODEL, MODULUS, NU, [PLASTICPARAM1, PLASTICPARAM2]

JA, JB, JC, MODELNUMBER

KEYWORD “biot” (required)

JA, JB, JC, ALPHA, PP_FAC

KEYWORD “stressboun” (required)

SUB-KEYWORD ‘distributed’ or ‘lithostatic’ (optional)

or

SUB-KEYWORD ‘lithograd’ SDEPTH GDEPTH (optional)

JA, JB, JC, BOUNVAL, KQ

KEYWORD “tolerance (required)

STRESS_TOL

KEYWORD “end strs” (required)

The input is terminated with keyword “*end strs*” or “*endstrs*”.

Input Variable	Type	Def	Description
ISTRS	int	0	ISTRS values indicate the program’s state of stress: 0 = skip stress solution 1 = 3-D solution
IHMS	int		IHMS is used to identify the amount and frequency of coupling between TH and M parts of the code. The following values apply: -1 = stress solved only at the end of the TH (flow) simulation -2 = stress solved at the beginning and end of the TH (flow) simulation (useful for establishing a lithostatic load -3 = stress solved iteratively within the NR_loop for each timestep of the TH (flow) simulation
KEYWORD “end strs” or “endstrs”		End of **strs** input.	

KEYWORD "initcalc"	Initiate an initial stress calculation that is useful for establishing lithostatic stress.		
KEYWORD "bodyforce"	Sets a body force if gravity is non-zero. Force is calculated using the rock density information provided in the rock macro.		
KEYWORD "reldisp"	Use relative displacement in the calculation of volume strains, permeability models, and output.		
KEYWORD "stresspor"	Explicitly update the porosity after each time step.		
KEYWORD "fem"	Use the Finite Element modules for forming displacement equations, and calculating stresses. Although optional, it is strongly recommended that this keyword be included.		
KEYWORD "principal"	For stress output to the files generated by the cont macro, output the principal stress values and the orientation of the axis.		
KEYWORD "strainout"	Create a file, strain.out, containing x, y, z, node number, εxx, εyy, εzz, εxy, εxz, εyz		
KEYWORD "excess_shear"	This controls stress output to files generated by the 'cont' macro, output the excess shear stress and the direction of the failure plane given in the equation below, as well as the Young's modulus $\tau_{excess}(\max) = \frac{1}{2}(\sigma_1 - \sigma_3)(\mu^2 + 1)^{1/2} - \frac{1}{2}\mu(\sigma_1 + \sigma_3) - S_0$ Where τ_{excess} is the excess shear, σ_1 and σ_3 are the maximum and minimum principal stresses, μ is the coefficient of friction, and S_0 is the shear strength. The angle β between this plane and the orientation of the maximum principal stress is given by $\tan(2\beta) = -\frac{1}{\mu}$. NOTE: The values specified here are used only for output, and are superseded by values specified in 'permmpdel' macro.		
FRICTION_OUT	real		Coefficient of friction
STRENGTH_OUT	real		Cohesion
PP_FAC_OUT	real		Pore pressure factor similar to Biot's coefficient in the 'biot' macro.
KEYWORD "zone"	The format and inputs for this are described in the zone macro. Inclusion of zone macros within the strs macro are allowed to facilitate input associated with the following keywords.		
KEYWORD "permmodel"	This keyword identifies the stress or displacement dependent permeability model. The permeability model can be invoked in a fully coupled, sequentially coupled, or explicitly coupled manner.		
IPERM	int	1	Specifies the type of permeability model used, input parameters specified on this line change depending on the model selected.
IPERM = 1	Equivalent to no stress permeability model, no further input parameters		
IPERM = 2	Stress permeability model dependent on tensile stress in the coordinate directions. Changes are linear in stress up to the prescribed maximum change. Tensile stress in a given coordinate direction affects the permeability in the other two directions. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f		
SPM1F	real		Minimum tensile stress (x direction) for damage to occur.
SPM2F	real		Minimum tensile stress (y direction) for damage to occur
SPM3F	real		Minimum tensile stress (z direction) for damage to occur
SPM4F	real		Damage factor for elastic modulus in x direction.
SPM5F	real		Damage factor for elastic modulus in y direction.
SPM6F	real		Damage factor for elastic modulus in z direction.
SPM7F	real		Maximum factor for x-permeability.

SPM8F	real		Maximum factor for y-permeability.
SPM9F	real		Maximum factor for z-permeability.
IPERM =21	Mohr-coulomb failure criteria on a user specified plane. Here z-prime is along the normal to the plane of failure, and y-prime is along the plane of median principal stress. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f, spm10f, spm11f, spm12f, spm13f		
SPM1F	First direction cosin of the normal to the user specified fault plane		
SPM2F	Second direction cosin of the normal to the user specified fault plane		
SPM3F	Third direction cosin of the normal to the user specified fault plane		
SPM4F	Friction coefficient of shear in the fault plane.		
SPM5F	Shear strength of the fault plane.		
SPM6F	Factor in effective stress calculation where effective_stress = sigma - (pp_fac*pore pressure)		
SPM7F	Range of excess shear stress over which damage is ramped		
SPM8F	Maximum multiplier for young's modulus in x-prime direction.		
SPM9F	Maximum multiplier for young's modulus in y-prime direction		
SPM10F	Maximum multiplier for young's modulus in z-prime direction		
SPM11F	Maximum multiplier for permeability x-prime direction.		
SPM12F	Maximum multiplier for permeability y-prime direction.		
SPM13F	Maximum multiplier for permeability z-prime direction.		
IPERM = 22	Mohr-coulomb failure criteria on the plane that maximizes the excess shear. Here z-prime is along the normal to the plane of failure, and y-prime is along the plane of median principal stress. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f, spmf10, spmf11f, spm12f		
SPM1F	real		Friction coefficient of shear in the fault plane.
SPM2F	real		Shear strength of the fault plane.
SPM3F	real		Factor in effective stress calculation where effective_stress = sigma - (pp_fac*pore pressure)
SPM4F	real		Range of excess shear stress over which damage is ramped
SPM5F	real		Maximum multiplier for young's modulus in x-prime direction.
SPM6F	real		Maximum multiplier for young's modulus in y-prime direction
SPM7F	real		Maximum multiplier for young's modulus in z-prime direction
SPM8F	real		Maximum multiplier for permeability x-prime direction.
SPM9F	real		Maximum multiplier for permeability y-prime direction.
SPM10F	real		Maximum multiplier for permeability z-prime direction.
SPM11F	real	1.	Optional, Maximum multiplier for porosity
SPM12F	real	0	Optional, permeability is made a function of current stresses minus this factor time the initial stresses. NOTE: if this option is to be used without the porosity modification, a value of 1 must be entered for SPM11F
IPERM = 91	Table input from a file, name given on the next line		
FILENAME	char		Name of the file with permeability model factors. The file has the following format:Line 1: # of rows in the tableLines 2 through (# of rows)+1: stress, x-factor, y-factor, z-factor
IPERM = 24	Stress permeability calculated for ensemble of shear-stimulated fractures. An isotropic fracture distribution is assigned for control volume. The Mohr-Coulomb failure criterion is evaluated for each fracture. If it fails, it's in-plane permeability is enhanced according to a calculated stress drop and shear displacement. The total change in nodal permeability is the average of all fractures. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f		
SPM1F	real		Shear fracture toughness. Converts calculated stress drop to shear displacement according to disp = stress_drop/toughness.
SPM2F	real		Static friction coefficient, for calculating onset of Mohr-Coulomb failure.
SPM3F	real		Dynamic friction coefficient, for calculating stress drop, where stress_drop = (static-

			dynamic)*(norm_strs-pressure)-cohesion
SPM4F	int		Number of fractures per control volume.
SPM5F	real		Shear displacement for onset of permeability enhancement.
SPM6F	real		Shear displacement interval over which permeability enhancement completes.
SPM7F	real		Logarithm of maximum per-fracture permeability multiplier.
SPM8F	real		Fracture cohesion, used in calculation of Mohr-Coulomb failure.
SPM9F	real		Fracture density in control volume (default = 1), can be used as a secondary, isotropic permeability multiplier.
KEYWORD "elastic"		For linear elastic material.	
ELASTIC_MOD	real		Young's modulus. (MPa)
POISSON	real		Poisson's ratio.
KEYWORD "nonlinear"			
NONLIN_MODEL _FLAG	int		If NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young's modulus and Poisson's ratio. Input: E_INI, DEDT, POISSON_INI, DNUEDT
			If NONLIN_MODEL_FLAG= 91 then a table lookup is used. Input: YOUNG_TEMP_FILE
E_INI	real		Value of Young's modulus at the reference temperature (MPa).
DEDT	real		Derivative of Young's modulus with respect to temperature (MPa/ °C)
POISSON_INI	real		Value of Poisson's ratio at the reference temperature.
DNUEDT	real		Derivative of Poisson's ratio with respect to temperature (per °C)
YOUNG_TEMP_F ILE	char		Name of the file with nonlinear model values. The file has the following format: Line 1: # of rows in the table (nentries_young) Line 2 through (# of rows)+1: temperature, young's modulus, poisson's ratio
KEYWORD "plastic"			
NUMPLASTICMO DELS	int		Number of plastic models. NOTE: Current version allows only ONE plastic model for the entire domain.
PLASTICMODEL	real		The Plastic model number PLASTICMODEL has the following values: If PLASTICMODEL = 1 Isotropic, linear elastic solid. If PLASTICMODEL = 2 von Mises model. If PLASTICMODEL = 3 Drucker-Prager model. PLASTICPARAM1 and PLASTICPARAM2 are entered only for the von Mises model. PLASTICPARAM1, PLASTICPARAM2 and PLASTICPARAM3 are entered only for the Drucker-Prager model.
MODULUS	real		Young's modulus in the elastic region (MPa).
NU	real		Poisson's ratio in the elastic region.
PLASTICPARAM 1	real		Yield stress for von Mises model (MPa).
PLASTICPARAM 2	real	0.	Currently not used.
KEYWORD "biot"			
ALPHA	real	0	Volumetric coefficient of thermal expansion (per °C)
PP_FAC	real		Factor multiplying the pore pressure coupling term in the stress-strain relations.
KEYWORD "stressboun"		Enter boundary conditions for the mechanical deformation equations. These can be a combination of specified values of displacements, stresses, or forces.	
SUB-KEYWORD 'distributed'		Distribute the applied force in proportion to areas of the members of the zone to which the force is applied.	
SUB-KEYWORD		BOUNVAL and KQ are interpreted as multipliers of the lithostatic stress and the stress	

'lithostatic'			direction. The lithostatic stress is always in the vertical (downward) direction. The z-axis is taken to be positive upwards. In the ctrl macro the direction of gravity must be set to 3.
SUB-KEYWORD 'lithograd' SDEPTH GDEPTH			BOUNVAL and KQ are interpreted as the stress gradient and stress direction. The parameters sdepth and gdepth are read on the same line as lithograd, and the KQth diagonal component of the stress at any node is calculated as follows, where z is the vertical coordinate of the node (see Figure 3 below). Stress(kq) = (sdepth+gdepth-z)*bounval
SDEPTH	real		Depth (m) of the reference level from the free surface of the earth corresponding to the level specified by GDEPTH, i.e., model elevation of GDEPTH meters is equivalent to SDEPTH meters depth.
GDEPTH	real		In the coordinate system of the model, the z coordinate of the reference level.
BOUNVAL	real	0	This is a fixed displacement, specified stress, or specified force depending on the value of KQ and optional keywords. No keyword, and kq > 0 : prescribed displacement (m) in the kq direction No keyword and kq < 0 : applied stress (MPa) in the kq direction Keyword = 'lithograd' and kq > 0 : the stress gradient (MPa/m) in the kq direction Keyword = 'distributed' and kq < 0 : prescribed force (MN) in the kq direction.
KQ	int	0	Parameter that determines the type of boundary condition kq = 1 or -1: prescribed value in the x direction kq = 2 or -2: prescribed value in the y direction kq = 3 or -3: prescribed value in the z direction
KEYWORD "tolerance" (required)			
STRESS_TOL	real	0	The tolerance for solution of the stress equations as follows: STRESS_TOL > 0 STRESS_TOL is the reduction of initial residual of the stress equations STRESS_TOL < 0 STRESS_TOL is the required absolute value of the residual of the normalized equations

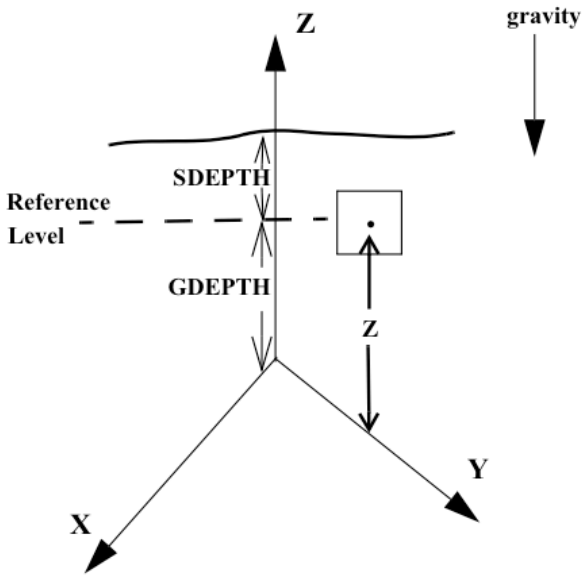

Figure 3. Schematic illustrating variables for 'lithograd' SUB-KEYWORD.

In the 3D example below, the option to explicitly couple stress with heat-mass equations is invoked. Initial stresses and displacements are calculated, a body force due to gravity is applied, optional strain output is activated, computations are performed using the finite element module, material is specified to be elastic, with temperature dependence of Young's modulus and Poisson's ratio specified in a file called "EvsT.txt", linear coefficient of thermal expansion 1.4e-5/0C, Biot's coefficient equal to 0. Zone 3 is pinned in all 3 directions, zones 4 And 5 are constrained in the X direction, and zones 6 and 7 are constrained in the Y direction. Tolerance for the stress solution is set to 1.e-3.

```
strs
1 -3
initcalc
bodyforce
strainout
fem
elastic
1 0 0 1.59e4 0.25

nonlinear
91
EvsT.txt
biot
1 0 0 5.4e-5 0.

zone
2 ! top,Z=300
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 299.99 299.99 299.99 299.99

3 ! bottom, Z=0
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
0.1 0.1 0.1 0.1 -0.1 -0.1 -0.1 -0.1

stressboun
-3 0 0 0. 3

stressboun
-3 0 0 0. 2

stressboun
-3 0 0 0. 1

zone
4 ! back X=20
19.99 20.01 20.01 19.99 19.99 20.01 20.01 19.99
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 -1. -1. -1. -1.
5 ! front X=0
-0.01 +0.01 +0.01 -0.01 -0.01 +0.01 +0.01 -0.01
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-4 0 0 0. 1

stressboun
```

```
-5 0 0 0. 1

zone
6 ! right, Y=0
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
0.01 0.01 -0.01 -0.01 0.01 0.01 -0.01 -0.01
300.01 300.01 300.01 300.01 -1. -1. -1. -1.
7 ! left, Y=60.
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
60.01 60.01 59.99 59.99 60.01 60.01 59.99 59.99
300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-6 0 0 0. 2

stressboun
-7 0 0 0. 2

tolerance
-1.e-3

end stress
```

6.5 TRXN (OPTIONAL INTERFACE FOR TRAC AND RXN)

This is a new macro added since UM V3.1. It does not add functionality but is a user-friendly interface to **trac** and **rxn**.

Overview ¶

The trxn macro is designed to replace the trac (and optionally rxn) macros with a more convenient, user-friendly input format. Rather than reading input as sets of nameless numbers, the trxn macro reads several "blocks" of data in which parameters are defined. The blocks may be specified in any order (with the exception of the ctrl and lookup blocks), and all blocks are optional, except for the header and comp blocks. Any parameters that are not specified will be given default values (usually zero).

The trxn macro relies heavily on zones, and uses zones for applying all variables that can vary by node. For this reason, a zone macro must be supplied in the input file before trxn is read. A time macro must also be given before trxn.

If the pound character ("#") appears on a line, everything after it on that line will be treated as a comment. Lines in which the first character is a pound sign are treated as blank lines. Blocks must not contain any blank or commented lines (although they can contain comments that do not start at the beginning of the line), and blocks must be separated by at least one blank or commented line. In the comp, water, rock, gas, disp, sorp, and assign blocks, entire columns can be commented out. If an asterisk ("*") (separated from surrounding tokens by whitespace) appears in the column header line of one of these blocks, the contents of every column to the right of the asterisk will be ignored. An entire block (until the next blank line) can be skipped by placing the keyword null before the block name.

The blocks are as follows:

TRXN block: ctrl ¶

This block contains control parameters, all on the same line. If it is supplied, it must be the first block in trxn. Its options are as follows:

Option	Description
rxnon	Enables reactions, which are by default disabled. Reactions will not occur in a simulation if rxnon is not given, even if reaction-related blocks are specified.
co2_couple	Enables CO_2 coupling for rxn.
debug	Enables the output of debugging information.

Example of the ctrl block; In this example, CO_2 coupling and reactions are enabled.

```
ctrl co2_couple rxnon
```

TRXN block: include ¶

The include block is designed to facilitate the construction of "libraries" of rock/water/gas types, component properties, dispersion models, etc. that can be shared between simulations. It allows blocks to be included from external files as though these files' contents were placed at the location of the include statement. External include libraries have the exact same syntax as the standard trxn macro. The first non-blank (and non-commented) line of the include file should be trxn library. Following this, as many blocks as desired may be supplied. The end trxn line should be given at the end of the library file; otherwise, reading will terminate early. Multiple include statements referencing several different files may be used in the same trxn macro. Libraries may include other libraries.

The syntax is as follows:

include /scratch/nts/ms/trxn/test-problems/standard.trxl
This will include the library located at /scratch/nts/ms/trxn/test-problems/standard.trxl at the current location in the input file.

As an example, a simple library might look something like this:

```
# Standard library for use with all test problems

trxn library

header
0  1  1e-6  1
0  100000   100000   100000
5  2  1  1  0
iskip=0, rsdmax=1e-9

disp  lx ly lz vx vy vz
std_x 0.1   1e-9  1e-9  0.1   1e-9  1e-9
std_y 1e-9  0.1   1e-9  1e-9  0.1   1e-9
std_z 1e-9  1e-9  0.1   1e-9  1e-9  0.1

sorp  ltype a1l   a2l   bl vtype a1v   a2v   bv
```

```
.std
*  con   0  0  1  con   0  0  1

diff  l=1e-9, v=1e-9

end trxn
```

All problems that include this library will be given the standard values for header and liquid and vapor diffusion coefficients of 1×10^{-9}, and will have one sorption model ("std") and three dispersion models ("std_x", "std_y", and "std_z") available.

Caution should be used when including libraries to prevent multiple definitions of any blocks, as this may cause unexpected behavior. trxn will print a warning message to the error file if a block is supplied more than once, and these warnings should be considered. Also, care should be taken to avoid placing any problem-specific parameters in the libraries.

TRXN block: header ¶

This block contains basic constant values, and is the first three lines of the trac macro copied verbatim.

Line	Variables
1	AN0 AWC EPC UPWGTA
2	DAYCS DAYCF DAYHF DAYHS
3	IACCMX DAYCM DAYCMM DAYCMX NPRTTRC
4	Optional parameters

Please refer to the **trac** section of the FEHM User's Manual for the meanings of the parameters on the first three lines. On the fourth line, optional parameters may be defined, using the form variable=value (no spaces), with commas between these pairs. The variables that may be set here are ISKIP, RSDMAX, and STRAC_MAX. Please refer to the rxn section of the FEHM User's Manual for the meanings of these variables. (STRAC_MAX is the maximum allowable saturation for vapor and Henry's Law species. When using trac, it is read if provided in the input file; however, it is not documented in the trac portion of the FEHM User's Manual.) ISKIP and RSDMAX are used only if reactions are enabled (see rxn below). If these optional variables are omitted, they are given the values shown in the example below. Below is an example of the header block.

```
header
0.0   1.0   1.0e-6   1.0          # ANO, AWC, EPC, UPWGTA
0.0   1e20  1e20    1e20          # DAYCS, DAYCF, DAYHF, DAYHS
10    2.0   1.0    150000.0   1  # IACCMX, DAYCM, DAYCMM, DAYCMX, NPRTTRC
iskip=0, rsdmax=1e-9, strac_max=0.99
```

Please note that the keyword userc is not supported in the header block. For userc support, please refer to the userc block below. Also, unlike in trac, NPRTTRC may not be omitted from the header.

TRXN block: userc ¶

This block invokes the solute transport user subroutine as specified in the trac section of the user's manual. On the same line as the macro name should be given the path to a file containing userc parameters. See the trac section of the user's manual for more information on the userc input format.

Below is an example of userc:

```
userc input/userc.dat
```
In this example, the userc subroutine is called with an input file located at input/userc.dat.

TRXN block: comp ¶

The comp block is used to define each component present in the simulation. (A component is any group of compounds, ions, etc., all of the same phase, that the user wishes to be treated as a single entity by the tracer solver.) It contains one line for each component, and each line consists of a phase designation for that component and the name of the component. The phase designation is one of "aqueous", "solid", "gas", and "henry"; these may be shortened to the first character to save time.

If kinetic reactions (rxn macro) are being simulated, two additional columns may be included. The master column indicates the "master species" for each aqueous or Henry's Law component. These master species are arbitrarily chosen forms of the components, by convention the form that is expected to dominate in reactions. The third column, guess, allows the user to specify a guess for the initial uncomplexed concentration of each aqueous and Henry's Law component. This is not necessary unless the chemical speciation solver has difficulty converging with the default value of 1×10^{-9}, in which case specifying a more representative value may help. If this is not necessary, leave the column out entirely, or place asterisks in the rows of components that do not need help converging.

Below is an example of the comp block:

```
comp     master   guess
aqueous H   H+ *
aq C_a      HCO3- 8.6e-6
a Na_a      Na+   *
a Ca_a      Ca++  *
a C_a2      CO3-- *
a Cl_a      Cl-   2.0e-9
a U238      U02   *
a Th234     ThO2  *
solid AlO3  *  *
s NaCl      *  *
s NaHCO3 *  *
s CaCl2     *  *
gas O2      *  *
g N2     *  *
henry C_h2  C6H6  1.2e-8
h Cl_h      Cl2   *
h C_h    CO2   *
```

In this example, there are 17 components. "H", "C_a", "Na_a", "Ca_a", "C_a2", "Cl_a", "U238", and "Th234" are aqueous; "AlO3", "NaCl?", "NaHCO3", and "CaCl2" are solid; "O2" and "N2" are gaseous, and "C_h", "Cl_h", and "C_h2" may be liquid or gas according to Henry's Law.

TRXN block: water, rock, gas ¶

These blocks are identical in form, and they are used to assign concentrations in the simulation for components of different states. These blocks allow the user to specify different "water types", "rock types", and "gas types", which may consist of different combinations of components specified in comp in different concentrations. See the moles block below for an alternative input format.

On the same line as the block name, the names of components specified in comp are placed, separated by tabs or spaces. These are column headers. Below this line, one line is given to each "type" desired. Each line consists of the name of the type, followed by numbers representing the concentrations of each of the components given in the column headers in that type. If the columns are separated by tabs, this layout forms a neat table with types down the left side, components across the top, and the concentration of each component in each type at the intersections of the rows and columns.

Only aqueous and Henry's Law components may be included in the water block, only solid components in the rock block, and only gaseous and Henry's Law components in the gas block. In the water block, a special column header, pH, may be included. This is the same as heading the column with "H" (and may only be done if "H" is specified in comp and is aqueous), but allows the user to enter H+ concentration in terms of pH rather than molarity. If a concentration in the grid is negative, it is assumed that the value entered is the base-ten logarithm of the actual concentration, and the value is adjusted accordingly.

An asterisk ("*") in a space where a number is expected is the same as a 0. If a component is omitted entirely from the table, it is assumed that that component is not present in the simulation. The unit for all numbers in these tables is molal, with the exception of the pH column, if present.

A negative value in the grid for a water or gas inflow type indicates that the concentration of that solute will be held constant at inflow nodes of that water type. Negative values in initial condition types should be avoided.

An example of each of the four block types, consistent with the sample comp block above, is given below:

```
water pH C_a   Na_a  C_a2  C_h2
wt1   7.43  1e-6  1.34e-5  1e-10 0.3
wt2   5.4   1e-3  0.002 2.4e-6   2.3
wt3   2.3   0  10 0  0

rock  AlO3  NaCl  NaHCO3
tuff  3.4e-2   1.24  1.6e-6
granite  3e-5  1.5e-2   6e-5
clay  0.3   3.9e-3   0.03

gas   O2 N2 Cl_h  C_h
vt1   0.23  0.10  1.24  2
vt2   1.02  0.012 0.2   0
```

In this example, there are three water types ("wt1", "wt2", and "wt3"), three rock types ("tuff", "granite", and "clay"), and two vapor types ("vt1" and "vt2"). Water type "wt1" has a pH of 7.3, an HCO3- concentration of 1×10^{-6} *m*, an Na+ concentration of 1.34×10^{-5} *m*, a CO3-- concentration of 1×10^{-10} *m*, and a C6H6 concentration of 0.3 *m*.

TRXN block: print ¶

The print block allows the user to specify for which components and complexes information is to be printed to tracer output(.trc) files. This block occupies only one line. After the keyword print is given a list of aqueous component and complex names that are to be printed, delimited by spaces, tabs, or commas. The keywords all and none may be given instead of the list, specifying that information is to be printed for all or none of the aqueous components and complexes, respectively. The default action, if no print block is specified, is the same as specifying print all. This information is only used if reactions are enabled.
Below is an example of the print block:

```
print UO2 HCO3- Na_a Ca_a
```

In this example, information will be printed only about UO2, HCO3-, Na_a, and Ca_a.

TRXN block: moles ¶

This block allows the user to specify initial solute conditions as the total number of moles contained in each zone. FEHM will distribute the specified number of moles evenly throughout the volume of the zone. The moles block conflicts with the water, rock, and gas columns in the assign block, and they cannot both appear in the same trxn macro.

The moles block allows the use of custom zone definitions. Unlike standard zones, these zones are permitted to overlap, and the concentrations at nodes in overlapping zones become cumulative. If these custom zones are to be used, a separate file must be created, containing a valid zone macro that defines the desired zones. Due to limited functionality in the trac macro, all zones in this macro must be specified using the nnum method and zones must be numbered rather than named. (See the manual section on the zone macro for more information.) Alternatively, the zones defined in previous zone macros may be used. On the same line as the moles keyword in this block, the path to the alternate zone file should be provided. If the standard zone definitions are to be used, leave the path blank or explicitly turn the alternate zone processing off by providing an asterisk in the place of the path. The next lines form a table with component names from comp across the top and zone numbers down the left-hand side. At the intersections of rows and columns is placed the number of moles of that component initially in that zone. Any omitted zones or components are assigned a zero concentration.

Below is an example of the moles block:

```
moles input/moles.zone
   C_a   Na_a  C_a2   C_h2
2  0     1.6   0.02   1
3  0     1.7   0.6    0.89
5  0     2     0.045  2.2
4  1.2   0     0.102  1.61
```

In this example, zone definitions are loaded from the file input/moles.zone, which remain in effect for the rest of the moles macro. Zone 2 contains 1.6 moles of Na_a, 0.02 moles of C_a2, and 1 mole of C_h2 distributed evenly throughout it.

TRXN block: hparam ¶

The hparam block is used to assign Henry's Law parameters to different Henry's Law components. Each Henry's Law component appearing in comp must be included in this block. Below the block title hparam, each Henry's Law component is given one line. Each of these lines consists of the name of the Henry's Law component, the model that will be used to simulate that component, and then several parameter=value pairs (separated by commas), specifying parameters for each model. The models and their parameters are as follows:

Option	Description
1 or hoff	van't Hoff model. Parameters: ah - A_H; dhh - ΔH_H.
2 or multi	Multi-parameter fit. Parameters: ahn - $A_{H,}$ *n*, for each *n* from 1 to 5.
3 or wvp	Use water vapor pressure. Parameters: ah - A_H; hh - Henry's constant modifier $H = P_{wv} \cdot \Delta H_H$.

Below is an example of an hparam block that uses all three of the above models:

```
hparam
C6H6  hoff  ah=0.34, dhh=3.2
Cl2   multi ah1=4.12, ah2=2.4, ah3=4, ah4=0.18, ah5=1.1
CO2   wvp   hh=2.8, ah=0.04
```

In this example, Henry's Law component C6H6 is simulated using the van't Hoff model. Its A_H value is 0.34, and its ΔH_H value is 3.2.

TRXN block: diff ¶

This block sets the molecular diffusion coefficients and diffusion models for liquid and vapor components. Due to technical restrictions imposed by FEHM's solver, there can only be one liquid and one vapor diffusion coefficient for the entire model. The diff block consists of one line. After the keyword are up to four name/value pairs, where the name is l to set the liquid diffusion coefficient, v to set the vapor diffusion coefficient, lm to set the liquid diffusion model, or vm to set the liquid diffusion model.

The possible values for the diffusion models are as follows:

Option	Description
0 or con	Constant molecular diffusion coefficient
1 or mq	Millington-Quirk diffusion model
2 or cw	Conca-Wright diffusion model for liquid, or alternate Millington-Quirk model for vapor
3 or adif	Diffusion model calculated as a function of pressure and temperature using tortuosity from adif macro

Below is an example of the diff block:

```
diff  l=1e-9, v=1.6e-8  lm=0, vm=con
```

In this example, the liquid diffusion coefficient is 1×10^{-9}, and the vapor diffusion coefficient is 1.6×10^{-8}. Both liquid and vapor use a constant molecular diffusion coefficient.

TRXN block: disp ¶

This block is used to set the dispersivity constants of regions of the simulation. Dispersivity parameters are applied to dispersivity models, which are applied to zones in the assign block. Model names run down the left side of the block, and parameters run across the top. Dispersivity can be supplied either for the X, Y, and Z directions, or for the longitudinal and transverse directions. Parameter names consist of two characters: the first l or v for liquid or vapor, and the second x, y, z, l, or t for X, Y, Z, longitudinal, or transverse, respectively. Parameter names from the two modes of setting dispersion cannot be mixed.

Below is an example of the disp block:

```
disp  lx ly lz vx vy vz
model1   0.2   3  1.5   0.28  3.6   1.92
model2   0.18  0.9   2.361 1.22  0.56  0.58
```

In this example, X/Y/Z dispersivity is set. There are two models, named model1 and model2. The liquid dispersivity for model1 is 0.2 in the X direction, 3 in the Y direction, and 1.5 in the Z direction. The vapor dispersivity for model1 is 0.28 in the X direction, 3.6 in the Y direction, and 1.92 in the Z direction.
sorp ¶
This block sets adsorption parameters for selected components. On the same line as the block title should appear any or all of the following column headers:

Header	Description
ltype	The type of adsorption model to be used to simulate liquid adsorption. See below for the possible adsorption models from which to choose.
a1l	α_1 parameter in liquid adsorption model.
a2l	α_2 parameter in liquid adsorption model.
bl	β parameter in liquid adsorption model.
vtype	The type of adsorption model to be used to simulate vapor adsorption. See below for the possible adsorption models from which to choose.
a1v	α_1 parameter in vapor adsorption model.
a2v	α_2 parameter in vapor adsorption model.
bv	β parameter in vapor adsorption model.

If a column header is not specified, it is assumed to be zero for every component.
The following are the available adsorption models to use for ltype and vtype:

Option	Description
0 or con	Conservative solute
1 or lin	Linear sorption isotherm
2 or freu	Freundlich sorption isotherm
3 or mfreu	Modified Freundlich sorption isotherm
4 or lang	Langmuir sorption isotherm

The α_1, α_2, and β parameters are used differently according to the adsorption model chosen:

Model	Expression	α_1	α_2	β
Linear	$C_r = K_d \cdot C_l$	K_d	0	1
Freundlich	$C_r = \Lambda \cdot C_l^{\beta}$	Λ	0	$0 < \beta < 1$
Modified Freundlich	$C_r / (C_{r,max} - C_r) = \Lambda \cdot C_l^{\beta}$	$\Lambda \cdot C_{r,max}$	Λ	$0 < \beta < 1$
Langmuir	$C_r = (r_b \cdot C_l) / (1 + r \cdot C_l)$	r_b	r	1

For a more in-depth description of the models used for sorption, please refer to the FEHM Models and Methods Summary.

Models are designated by a line starting with a period ("."), immediately followed by the name of the model and nothing else. On the next several lines, components specified in comp may be given, with their corresponding parameters for the current model in the correct column. The same components, in the same order, must be given in each model. If a component is liquid- or vapor-only, then asterisks should be placed in the columns that do not apply to that component. An asterisk can also be placed in the column containing component names to indicate that all components that have not yet been explicitly given sorption parameters should use the model on the line with the asterisk. Thus, to assign the same sorption parameters to all components, only one line per model would be supplied, containing an asterisk in the first column.
Below is an example sorp block:

```
sorp  ltype a1l   a2l   bl vtype a1v   a2v   bv
.model1
CO3-- freu  3.02  0.061 0.89  *  *  *  *
C6H6  lin   0.89  3.16  0.2   freu  0.35  4.519 0.688
CO2   mfreu 1.20  3.31  0.4   con   0.01  2.01  0.61
.model2
CO3-- con   0.001 2.02  0.88  *  *  *  *
C6H6  con   1.20  0.58  1.12  lin   2.2   2.043 2.7
CO2   mfreu 3.006 1.0   9.8   mfreu 0.229 3.434 2.33
```

In this example, C6H6 is modeled using the linear sorption isotherm model with a liquid α_l parameter of 0.89 and a vapor α_l parameter of 0.35 in adsorption model model1, and modeled using the conservative solute model with a liquid α_l parameter of 1.20 and a vapor α_l parameter of 2.2 in model model2.
If sorp is omitted, it is assumed to contain zeros for all values except the β parameters, which are assumed to be 1.

TRXN block: cden ¶

The cden block allows the user to input the molecular weights of aqueous and aqueous Henry's Law components and have the code adjust the density of the water according to the concentrations of these components. It should not be used if trxn is preceded by a cden macro, as some values may be modified. cden accepts several lines, each consisting of the name of a master species and its molecular weight, separated by spaces, tabs, or commas.
Below is an example of the cden block:

```
cden
HCO3- 61
CO3-- 60
C6H6  78
```

In this example, HCO3- is defined to have a molecular weight of 61, CO3-- to have a molecular mass of 60, and C6H6 to have a molecular mass of 78.
If database lookup is enabled (see the lookup block below), one of the lines in cden may consist of only an asterisk. If such a line is provided, all components that lookup dynamically imports will be inserted into the cden block with the appropriate molecular weights. These imported molecular weights can be overridden by explicitly listing the component and its desired molecular weight on a separate line.

TRXN block: group ¶

The information in this block is only used if reactions are enabled (see ctrl above). It is used to group aqueous components that take part in rapid kinetic reactions. On the line below the keyword group, place one line for each group. Each line should contain the names of all aqueous components present in that group, separated by spaces, tabs, or commas.
Below is an example of the group block:

```
group
H
C_a Cl_a
Na_a Ca_a Ca_a2
U238 Th234
```

In this example, H is in its own group, C_a and Cl_a are in a group together, Na_a, Ca_a, and Ca_a2 are grouped together, and U238 and Th234 are grouped together.

TRXN block: cplx ¶
The cplx block allows the user to specify instantaneous reactions that form aqueous complexes from the aqueous master species and non-aqueous components specified in comp. One equation is specified on each line, using a slightly modified version of the standard reaction format detailed in the rxn block below. The left side of the reaction should contain only the name of the aqueous complex, without a number denoting its stoichiometry (the stoichiometry of the aqueous complex must be 1). The right side should contain stoichiometry/compound pairs as specified by the standard format. If a compound needs to be removed to make the aqueous complex, negate its stoichiometry. After the equation, two comma-separated values in the format variable=value (not padded by spaces) are required: ckeq for the equilibrium constant of that complex, and heq for the enthalpy of the formation of that complex. On the same line as the keyword cplx, the keyword log10 may be provided to denote that the values for all constants in the block will be given as the base-ten logarithm of the actual values. Asterisks supplied in place of the equilibrium constant or enthalpy value always signify a zero, even if the log10 keyword is specified.
If the keyword equi is supplied on the same line as the block name, the equi block below will be consulted to calculate equilibrium constants as functions of temperature. In this case, the log10 keyword and any values given to the right of the equations will be ignored.
If the database lookup option is enabled (see the lookup block below), complexes from the % CPLX section of the database file may be imported using a line that omits the right-hand side of the equation, and possibly the equilibrium information as well. If such a line is encountered, the code will search for the named complex in the lookup database, and import the complex's constituents, stoichiometry, and equilibrium-related constants. The comp block will be automatically updated to include all components required for the imported complexes. Note that the equi option cannot be used if database lookup is enabled, as this conflicts with the five-parameter fit used by PHREEQC databases for calculating the equilibrium constant as a function of temperature. These equilibrium values can be overridden by placing explicit values for ckeq and heq on the same line.
Below is an example of cplx:

```
cplx log10
CaHCO3+ = 1 Ca++ + 1 HCO3-       ckeq=-13.456, heq=0
OH- = -1 H+                      ckeq=-14,     heq=0
Ca2UO2(CO3)3 = 2 Ca++ + 2 UO2 + 3 HCO3- + -3 H+ ckeq=-20.26, heq=*
MgHPO4
```

In this example, there are four complexes. The complex Ca2UO2(CO3)3 is made by combining 2 Ca++, 2 UO2, and 3 HCO3- and removing 3 H+. The information for the complex MgHPO4 is automatically

imported from the database given in the lookup block. All six equilibrium and enthalpy values are given as the base-ten logarithms of their actual values. The equilibrium constant for CaHCO3+ is $1.0\times10^{-13.456}$. The enthalpies of CaHCO3+ and OH- are $1.0\times10^{0} = 1$, but the enthalpy of Ca2UO2(CO3)3 is zero.

TRXN block: equi ¶

This block allows equilibrium constants for aqueous complexes from the cplx block above to vary with temperature. If the keyword equi is provided in the cplx block above, trxn will not read equilibrium constants in cplx; instead, the equi block will be used to determine the constants as a function of temperature.
Every aqueous complex appearing in cplx must be included in equi. For each aqueous complex, a line should be provided that contains only the name of the complex, prefixed by a single period. On the lines that follow, the equilibrium constants should be specified by placing on each line a temperature in ° C, a tab or space, and the equilibrium constant at that temperature. The number of temperatures need not be the same for each complex.

Below is an example of the equi block:

```
equi
.CaHCO3+
0  1e-10
10 2e-10
40 4e-9
.OH-
0  1e-14
.CaUO2(CO3)3
20 1.3e-8
40 2.64e-6
60 8.7e-4
80 2.1e-1
```

In this example, CaHCO3+ has an equilbirium constant of 1×10^{-10} at 0° C, 2×10^{-10} at 10° C, and 4×10^{-9} at 40° C. OH- has an equilibrium constant of 1×10^{-14} at all temperatures.

TRXN block: dist ¶

This block specifies distribution models that can be used for reaction types 1 and 2 to describe the distribution coefficient as a function of temperature. For each distribution model, a line beginning with a period and then the name of the model (without a separating space) is given. This is followed by a set of temperature/distribution coefficient pairs, one per line, for as many lines as desired. FEHM will perform a piecewise linear interpolation between the values given to determine the value of the distribution coefficient for intervening temperatures.

Below is an example of the dist block:

```
dist
.model1
0  1
10 2
20 4
30 8
```

```
40 16
.model2
0  20
50 40
100   70
```

There are two models in this example. The first one, model1, has five data points at 10° C intervals from 0 to 40° C. The distribution coefficient at 0° C is 1, the coefficient at 10° C is 2, and the coefficient at 20° C is 4. Intermediate temperatures are linearly interpolated, so the distribution coefficient for model1 at 25° C is 6.

TRXN block: sol ¶

The sol block specifies solubility models that can be used for reaction types 7 and 8. For each solubility model, a line beginning with a period, immediately followed by the model name, is given. This is followed by temperature/solubility coefficient pairs, one per line, for as many lines as desired. FEHM will perform a piecewise linear interpolation between the values given to determine the solubility coefficient for intervening temperatures.
Below is an example of the sol block:

```
sol
.model1
0  0
10 1e-16
20 1e-4
30 1e-2
50 1
```

This example contains one model, model1. In this model, the solubility coefficient changes from 0 to 1 over a range from 0° C to 50° C. The solubility coefficient at 10° C is 1×10^{-16}, and the coefficient at 20° C is 1×10^{-4}. Because FEHM linearly interpolates between successive values of the solubility coefficient, the coefficient at 40° C is 0.505.

TRXN block: lookup ¶

This block enables the dynamic lookup process for mineral dissolution and aqueous complexation. lookup uses a lookup database (generated by trxndb from a USGS PHREEQC geochemical database) to determine which reactions occur among a specific set of minerals and aqueous complexes. The lookup block adds data to the comp, cden, group, cplx, and rxn blocks based on the information provided it.
In order to use lookup, a database in trxn's standard format must be supplied.

The trxn lookup format is as follows:

Keyword % MASTER Blank line(s) Component 1 parameters: component 1 name, component 1 master species name, component 1 molar mass Component 2 parameters... Blank line(s) Keyword % CPLX

Blank line(s)
Complex 1 name
Complex 1 products: product 1 stoichiometry, product 1 name; product 2 stoichiometry, product 2 name...
Equilibrium constant for this complex
Enthalpy of reaction for this complex
Up to five values defining temperature dependence of equilibrium constant: A_1, A_2, A_3, A_4, A_5, where $\log_{10} K = A_1 + A_2 \cdot T + A_3 / T + A_4 \cdot \log_{10} T + A_5 / T^2$
Blank line(s)
Complex 2 parameters...
Blank line(s) Keyword % MIN
Blank line(s)
Mineral 1 name, mineral 1 formula
Mineral 1 products
Equilibrium constant for this mineral
Enthalpy of reaction for this mineral
Up to five values defining temperature dependence of equilibrium constant
Blank line(s)
Mineral 2 parameters...
Blank line(s)
Keyword % END

Comments can be included anywhere in the input file by using a pound sign ("#").
Below is a brief example of a database in this form.

```
% MASTER

E  e- 0
H  H+ 1.007942
Mn(+2)   Mn++  54.938045
Mn(+3)   Mn+++ 54.938045
F  F- 18.9984032
Al AlOH++   26.9815386
Si H4SiO4   28.08554
Mg Mg++  24.305
Ca Ca++  40.0782
S  HS-   32.0652
Fe Fe++  55.845

% CPLX

Al+++
1 AlOH++ 1 H+
-5.00
11.49
 -38.253 0.0 -656.27 14.327

Mn++
1 e- 1 Mn+++
-25.510
25.800


MnF+
```

```
1 Mn++ 1 F-
0.840

% IMM

Sepiolite Mg2Si3O7.5OH:3H2O
3 H4SiO4 -4 H+ 2 Mg++
15.760
-10.700

Fluorite CaF2
1 Ca++ 2 F-
-10.600
4.690
66.348 0.0 -4298.2 -25.271


Pyrite FeS2
2 HS- -2 e- 1 Fe++ -2 H+
-18.479
11.300

% END
```

In this example, there are three complexes and three minerals. The first complex is Al+++, which is formed by the master species AlOH++ and H+. Its equilibrium constant is -5.00, its enthalpy of formation is 11.49, and four of five possible temperature-dependence parameters are supplied. For the complex MnF+, no enthalpy of formation or temperature-dependence parameters are supplied.

A conversion script, trxndb, is available to automate conversion of certain USGS PHREEQC input files to the appropriate format. The converter script understands a limited subset of the complete syntax used in PHREEQC input files; if it gives improper results or errors, ensure that the input file is in a consistent format and that the keywords log_k, delta_h, and -analytic are used rather than their shortened alternatives. An example file that can be converted flawlessly by trxndb can be found at /scratch/nts/ms/trxn/geochem/phreeqc.dat. The converter script is written in Perl and located at /scratch/nts/ms/trxn/geochem/trxndb. The script should be called with the name of the PHREEQC input file as the only argument. It will create a file in the same directory with the same name and extension trxd containing the trxn-compatible database.

For example:

```
$ ls
phreeqc.dat
$ /scratch/nts/ms/trxn/geochem/trxndb phreeqc.dat
45 master species, 187 solution equations read
58 mineral equations read
45 master species, 180 solution equations written
57 mineral equations written
$ ls
phreeqc.dat  phreeqc.trxd
$
```

The lookup block must be the first block in the trxn macro (excepting the ctrl block if it is used) and consists of one line, which contains the block name lookup and the full path to the database file. The lookup block should be followed by a blank or commented line.
Below is an example of the lookup block.

```
lookup /scratch/nts/ms/trxn/geochem/phreeqc.trxd
```

In this example, the database is located at /scratch/nts/ms/trxn/geochem/phreeqc.trxd.

Once lookup has been enabled, the cden, cplx, and rxn blocks may be modified by the user to utilize the information from the lookup database. Please see the sections for these blocks for the appropriate syntax to take advantage of this information. If the debug option is provided in the ctrl block, the final version of each block will be printed to the output (.out) file, which may aid in debugging if the results of the run are not as expected.

TRXN block: rxn ¶

The rxn blocks are used to model kinetic reactions between simulated compounds. Seven types of reactions my be used, each with its own input parameters and input format. rxn is intended to be specified multiple times, once for each reaction that is taking place. For each reaction, the first line of the block contains the keyword rxn and the number representing the type of the reaction (see below). The next several lines are used for the parameters unique to that reaction type, which are detailed below. The end of the rxn block is signaled by a blank line.
Most of these reactions take one line of input in the standard reaction format, which is used to specify reactants, products, and stoichiometries simultaneously. In this format, each reactant or product is specified by a number denoting the stoichiometry of that compound, a space, and then the name of the compound as given in comp. Compounds are separated from each other by a plus sign padded on either side with spaces (" + "). The products and reactants are separated from each other by a token containing an equals sign ("="). Optionally, directional indicators may be added to the equals sign to indicate the direction of the reaction (e.g., "=>"). The reactants must be placed on the left side of the equals sign, and the products on the right side. For example:
6 HCl + 2 Al_s => 2 AlCl3 + 3 H2
The reaction types and their parameters are as follows. A more detailed description of the mechanics of each type of reaction can be found at the end of the rxn section of the FEHM User's Manual.

Number	Type	Description
1	Linear kinetic reaction	This reaction accepts one aqueous reactant and one solid product with 1:1 stoichiometry. The first parameter line for this reaction is a reaction in standard format, without the stoichiometric coefficients. The second parameter line in a list of comma-separated name/value pairs, where the acceptable names are rate to specify the rate of the reaction and distcoef to specify the distribution coefficient. The value for distcoef my be a real number of the coefficient is constant, or the name of a model specified in dist (without the beginning period).
2	Langmuir kinetic reaction	This reaction's parameter input is identical to the input for reaction type 1 except for an extra available parameter for the name/value pairs. This parameter is maxconc, used to set the maximum concentration that can sorb.
3	General kinetic reaction	This reaction accepts its first line of of input as a generic reaction in the standard format described above. This is followed by a line of name/value pairs, for which the name can be forward to set the forward rate constant for the reaction or reverse to set the reverse rate constant.
4	Dual Monod	The first three to six parameter lines of this reaction consist of a name, a colon, and

	biodegradation reaction	one or more component/complex names. The name substrate accepts a single immobile component name that is to be the substrate that is degraded. The name electronacceptor accepts the stoichiometry and name (separated by a space) of the aqueous complex that is the electron acceptor for the reaction. The name biomass accepts the stoichiometry and name of the solid component that is the biomass produced by the reaction. The name reactants is optional and accepts the stoichiometries and names of any extra reactants that are participating in the reaction. Likewise, products is optional and accepts the names of any extra products of the reaction. Only a total of three additional reactants and products can be specified. If only certain forms of the substrate are biodegradable, those can be listed with the biodegradable name. The last parameter line contains a list of name/value pairs as follows: ks - the half-maximum-rate concentration of the substrate; ka - the half-maximum-rate concentration of the electron acceptor; decay - the microbial decay rate (1/hr); phthreshold - the pH threshold above which biodegradation will not occur; qm - the maximum rate of substrate utilization (mol/kg biomass/hr); yield - the microbial yield coefficient (kg biomass/mol substrate); xminit - the minimum concentration of biomass (mol/kg rock).
5	Radioactive decay reaction	This reaction accepts one aqueous component reactant and one aqueous component product with 1:1 stoichiometry. This first parameter line is a reaction in standard format without stoichiometric coefficients. The reactant is the component that is decaying, and the product is the decay product. If the decay product is not being modeled in the simulation, an asterisk may be given in place of the product name. The second parameter line contains a single name/value pair. The name is halflife, and the value is the half-life of the reaction in years.
7 and 8	Precipitation/ dissolution reaction	For reaction type 7, the rates are based on the total aqueous concentration of the components; whereas for reaction type 8, the rates are based on the free-ion concentrations alone. This reaction accepts one solid component and any number of aqueous master species. The first line of input for this reaction should be a reaction in standard format, containing only a solid component on one side, and at least one master species on the other. The second line should contain the following name/value pairs: solubility - the solubility product (either a real number or the name of a solubility model specified in sol); rate - the reaction rate (mol/m^2/s); sarea - the surface area of the mineral (m^2/m^3 rock); molecularweight - the molecular weight of the mineral; density - the density of the mineral. molecularweight and density should only be provided for reaction type 8. For reaction types 7 and 8 only, if the database lookup option is enabled (see the lookup block above), an alternate form of the reaction can be input. On the first line of the reaction, provide only the name of a mineral defined in the % IMM section of the lookup database. The information on the reactants and products, as well as the solubility constant, are imported from the database, and any components that are necessary for the reaction are dynamically added to the comp block. However, the rate and sarea parameters (and molecularweight and density parameters for reaction type 8) are not contained in the database, and must still be set by the user on the second line of the reaction.

Please note that in all name/value pairs, the name and value must be separated by only an equals sign that is not padded by spaces.

Here is an example of each reaction type. Two versions of reaction type 5 are given to demonstrate the optional unsimulated daughter species, and two versions of reaction type 7 are provided to demonstrate the database lookup option.

```
rxn 1
Ca_a <=> Ca_s
```

```
rate=10, distcoef=2

rxn 2
Ca_a <=> Ca_s
rate=10, maxconc=2, distcoef=model2

rxn 3
3 H + 2 Ca + 5 U238 <=> Cl + 2 Cl2 + 3 C_a
forward=2, reverse=1.5

rxn 4
substrate:  U238
electronacceptor:  3 H
biomass:  Ca
reactants:  2 Cl
products:  1 Na, 5 Th234
biodegradable:  UO2
ks=1.2,   ka=1.35,   decay=0.69,   phthreshold=8.2,   qm=0.20,   yield=1.2,
xminit=0.067

rxn 5
U238 => Th234
halflife=20

rxn 5
U234 => *
halflife=10

rxn 7
NaCl <=> Na + Cl
solubility=0.0231, rate=1.02, sarea=2.2

rxn 7
Quartz
rate=0.2, sarea=1

rxn 8
CaCl2 <=> Ca + 2 Cl
solubility=model1, rate=0.2, sarea=5, molecularweight=60, density=5.25
```

TRXN block: assign ¶

This block allows the user to assign the parameters stored in models in the above blocks to the zones defined in the zone macro. The assign block also allows assignment of some other parameters that are specific to zones.
Zone numbers run down the side of the assign block, and parameters run across the top. The following parameters may be supplied; all are optional:

Option	Description	Default Value
water	Sets the initial water type filling the zone. This parameter must be the name of a water type defined in the water block.	Pure water
boun	Sets the concentrations of species coming from inflow nodes in this zone. This parameter may consist of a water type defined in water and/or a gas type defined	Pure water

	in gas. If both a water type and a gas type are flowing in, they should be separated by a period (not padded by spaces). If there is no inflow in this zone, give an asterisk for this parameter. An asterisk can also be used to specify inflow of pure water.	
time	Sets the time range during which the inflow nodes in this zone are injecting, in days. A lone zero ("0") gives no inflow. An asterisk provides no inflow if the entry in the boun column is "*" or if the boun column is missing, but provides inflow over the entire simulation otherwise. A single number other than zero will give inflow for one day starting at the specified day. A range separated by a greater-than sign (">") not padded by spaces will run injection from the first number specified to the second number. If the number before or after the greater-than is ommitted, it will default to the beginning or end of the simulation, respectively. Thus, ">30" will run injection from time 0 through time 30, "30>" will run injection from time 30 to the end of the simulation, and ">" will run injection for the entire simulation.	Inflow for the entire simulation if valid water/gas types are specified in the boun column, no inflow otherwise
rock	Sets the composition of the rock in this zone. This parameter must be the name of a rock type defined in rock.	Rock contains no relevant species
gas	Sets the initial composition of the gas in this zone. This parameter must be the name of a gas type defined in gas. If there is no gas in this zone, give an asterisk for this parameter.	No gas
disp	Sets the dispersivity constants for this zone. This parameter must be the name of a dispersivity model defined in disp.	No dispersivity
sorp	Sets adsorption parameters for this zone. This parameter must be the name of an adsorption model defined in sorp. While the models in sorp are defined by starting the line with a period, do not include the period in this parameter.	No adsorption for any components
tpor	Sets the optional tracer porosity for this zone. This parameter must be a real number from 0 to 1.	0.32
opt	Sets miscellaneous options per-zone. Options should be separated by periods with no spaces. The available options are: const causes the concentrations of solutes in the inflow water to be held constant at nodes in the current zone; accum enables solute accumulation in the current zone. Note that const and accum are mutually exclusive. An asterisk can be used to specify no options.	No options

An asterisk may be given for any of the parameters above for a given zone, in which case the default values given above are used. If an entire column is omitted, that parameter will be given the default values shown above for every zone. If a zone is omitted, it will receive default values for every column. Nodes that are not in any zone at the time when trxn is called will also receive default values.

Below is an example of the assign block.

```
assign   water rock  gas   boun     time   disp    sorp     tpor
1  wt3   granite     *     vt1.wt1  20>30  model1  model1   0.28
2  wt2   clay       vt2   *        *      model2  model2   0.69
3  wt2   granite    vt2   *        *      model1  model2   0.32
```

In this example, there are three zones. The first zone is initially filled with water of type wt3 and no gases, and water of type wt1 and gas of type vt1 are flowing into it starting at time 20 days and ending at time 30 days. Its rock is of type granite, it uses the dispersivity parameters defined in model1 and the adsorption parameters defined in model1, and has a tracer porosity of 0.28. Note also that zones 2 and 3 have no inflow.

After specifying all applicable blocks, use the end trxn keyword to end the reading of the trxn macro.

TRXN Example ¶

Below is a complete, commented working example trxn macro, with its accompanying zone macro for reference. This example was taken from the multi_solute test case in FEHM's standard verification suite.

```
zone
default
all
bound
nnum
2  1 102

trxn

# In the zone macro above, all nodes are placed into the zone named
"default",
# except for nodes 1 and 202, which are placed in the zone named "bound".
# Zone "bound" will be used for inflow.

ctrl rxnon # Enable reactions.

# Include header information from trac.
header
1.d-80   1.0   1.e-6 0.5
1. 2000  1.0   2000
5  5.0   1.e-6 2.8333e-3   0
iskip=0, rsdmax=1e-10

# There are six components in this simulation:  aqueous cobalt, iron, and
EDTA,
# and solid Co-EDTA, Fe-EDTA, and cobalt.
comp     master
a Cobalt    Cobalt_a
a Iron      Iron_a
a EDTA      EDTA_a
s Co-EDTA_s *
s Fe-EDTA_s *
s Cobalt_s  *

# There is only one type of water, called "inflow", which contains 3.16e-5 M
# aqueous cobalt, 1e-13 M aqueous iron, and 3.16e-5 M aqueous EDTA.
water Cobalt   Iron  EDTA
inflow   3.16e-5  1e-13 3.16e-5

# There is one sorption model.   It models all components with a linear
sorption
#  isotherm, using alpha-1 and alpha-2 parameters of zero, and a beta
parameter
# of 1.
sorp     ltype a1l   a2l   bl
.smod
Cobalt   lin    0    0     1
Iron     lin    0    0     1
EDTA     lin    0    0     1

# We assign the liquid diffusion coefficient for the simulation to be 1e-9.
diff  l=1e-9

# There is one model for dispersivity, "dmod".  It sets the dispersivity to
```

```
# 0.05 in the X direction, and 1e-34 in the Y and Z directions.
disp  lx    ly    lz
dmod  0.05  1e-34 1e-34

# There are two groups for the coupled solver.  One contains cobalt and EDTA,
# and the other contains iron.
group
Cobalt EDTA
Iron

# Here, two aqueous complexes are defined:  Co-EDTA and Fe-EDTA.  Co-EDTA is
# composed of one aqueous cobalt and one aqueous EDTA; Fe-EDTA is composed of
# one aqueous iron and one aqueous EDTA.  Both complexes have enthalpy zero;
# Co-EDTA has equilibrium constant 1e18 and Fe-EDTA has equilibrium constant
# 6.31e27.
cplx
Co-EDTA_a = 1 Cobalt_a + 1 EDTA_a   ckeq=1e18, heq=0
Fe-EDTA_a = 1 Iron_a + 1 EDTA_a     ckeq=6.31e27, heq=0

# There are four reactions taking place in this simulation.

# Reaction 1 is a linear kinetic reaction describing the dissolution of
# Co-EDTA.  The distribution coefficient is 0.533, and the rate of reaction
is
# 1.
rxn 1
Co-EDTA_a <=> Co-EDTA_s
distcoef=0.533, rate=1

# Reaction 2, also linear kinetic, describes the dissolution of cobalt.  The
# distribution coefficient is 5.07, and the rate of reaction is again 1.
rxn 1
Cobalt_a <=> Cobalt_s
distcoef=5.07, rate=1

# Reaction 3 is also linear kinetic and describes the dissolution of Fe-EDTA.
# The distribution coefficient here is 0.427, and the rate of reaction is 1.
rxn 1
Fe-EDTA_a <=> Fe-EDTA_s
distcoef=0.427, rate=1

# Reaction 4 is a general kinetic reaction that describes the complexation of
# solid cobalt and Fe-EDTA to form Co-EDTA.  The forward rate constant is
#  1.26e-2,  and  the  reaction  never  occurs  in  reverse  (the  reverse  rate
constant
# is zero).
rxn 3
Co-EDTA_s = Fe-EDTA_s + Cobalt_s
for=1.26e-2, rev=0

# Finally, attributes are assigned to zones.  The first zone, "default", con-
# tains most of the nodes; the zone "bound" contains only the inflow nodes.
# The initial water filling both zones is pure water, as signified by the
# asterisks in the water column, and the rock does not contain any species
that
# participate in the reactions, as signified by the lack of a rock column.
# (The water column could also have been left out entirely, but is included
# here for clarity.)  No inflow occurs in the default zone, as shown by the
```

```
# asterisk in the boun column, but water of the  "inflow" type defined in the
# water block is flowing in through the nodes in the bound zone from time 1
# through time 4.167 of the simulation.  The dispersivity model "dmod" and
the
# sorption model "smod" are applied to both zones.
assign   water boun   time       disp  sorp
default  *     *        0        dmod  smod
bound    *     inflow   1>4.167  dmod  smod

end trxn
```

Below are the trac and rxn macros that were replaced by the above trxn, for reference and comparison.

```
trac
1.d-80 1.0 1.e-6 0.5
1. 2000 1.0 2000
5 5.0 1.e-6 2.8333e-3
6
1
1 0. 0. 1. 1.e-9 .05 1.e-34 1.e-34

1 202 1 1


1 202 101 3.1623e-5 1.0 4.16667

1
1 0. 0. 1. 1.e-9 .05 1.e-34 1.e-34

1 202 1 1


1 202 101 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


1 202 101 3.1623e-5 1.0 4.16667

0


0


0


end trac
rxn
** NCPLX, NUMRXN
```

```
2,4
** Coupling of the aqueous components (dRi/dUj)
2
1 0 1
0 1 0
**   IDCPNT(IC),CPNTNAM(IC),IFXCONC(IC),CPNTPRT(IC)   (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),IMMPRT(IM)(ID  #  and  name  of  immoblie  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
** 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
** 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
end rxn
```

Differences using trac or rxn vs trxn interface¶

Please note that the following features present in trac and rxn have been removed in trxn:

- trac

Inflow concentrations for solids
Varying molecular diffusion coefficient by either species or location
Setting different dispersivities for different species

- rxn

Using IFXCONC to specify that concentrations are free-ion only (all concentrations in water, rock, and gas must be total aqueous concentrations)
Enabling reactions at specific nodes only
Reaction type 6

- Both

JA JB JC format for entering specific nodes for tracer injection, etc. (These must be defined by zones.)
If these features are desired, old-style input using trac and rxn must be used instead.

Further resources and verification ¶

Several test problems from the standard FEHM test suite that have trac and/or rxn macros have been converted to the trxn format. These can be found in /scratch/nts/ms/trxn/test-problems. The input files contain the original trac and/or rxn macros (turned off), along with an equivalent trxn macro. The output for the trxn macro has been verified against the output for the original macros in each of these cases; the results of the comparison can be found in the plot/plot.png and plot/plot-orig.png files in the test directory. The location of the input file in the test directory is included in parentheses after the test problem name. The following test problems have been verified to work successfully with trxn:

- baro_trans (input/baro_trans.in)
- cden_test (input/static-multi1.dat)
- dissolution (input/dissolution.in)
- fracture_transport (input/tangtestN.in)
- henrys_law (input/henrytest.in)

- multi_solute (input/multi_solute.in)
- sorption (input/sorption.in)
- transport3D (input/3d_trac.dat)

In addition, several new test problems have been developed in order to better test the full range of trxn's functionality. These are also found in /scratch/nts/ms/trxn/test-problems, and documentation of the problem setup and expected results can be found in /scratch/nts/ms/trxn/test-problems/meta/doc. The following test problems fall into this category:

- multirock
- inflow
- decay

Bugs, error handling, and limitations of trxn ¶

- trxn will try to print out a nice error message if something goes wrong. However, this is not guaranteed.
- A zone macro with at least one valid zone is required before trxn. This zone macro may contain no more than 100 zones.
- If a block is specified multiple times, the values from the last block should be used; however, do not rely on this feature.
- The maximum permitted length of input lines is 200 characters.
- The maximum number of characters in a name (of a component, zone, model, etc.) is 40.
- The maximum number of specifications (models, components, etc.) that any given block may contain is 100.
- The zone macro preceding trxn may not contain more than 100 zones.
- Model and species names should be strictly alphanumeric plus the five characters "(", ")", "+", "_", and "-". Use of other characters may cause incorrect behavior.
- If only the trac blocks of the macro are given, trxn should be compatible with an old rxn macro, as long as the trxn macro is read before the rxn macro. However, this has not been tested and may not work reliably. Furthermore, trxn is not compatible with the old trac macro.

Debug tools for trxn ¶

- The keyword debug in the ctrl block will enable some informational output that may be useful for debugging problems. The stop keyword in the ctrl block will halt FEHM immediately after reading and processing trxn.
- null blocks are ignored by default, but are printed if debugging output is enabled.

6.6 VCON (OPTIONAL)

This update supplements UM V3.1 Section 6.2.94

Thermal conductivity:

(1) Thermal conductivity for intact salt: $\lambda_{IS} = \lambda_{IS,300}\left(300/T\right)^{\gamma_1}$

(2) Thermal conductivity for crushed salt: $\lambda_{CS} = \lambda_{CS,300}\left(300/T\right)^{\gamma_1}$

where $\lambda_{CS,300} = 1.08\left(\alpha_4\phi^4 + \alpha_3\phi^3 + \alpha_2\phi^2 + \alpha_1\phi + \alpha_0\right)$

Parameters related to thermal conductivity are $\lambda_{IS,300}, \gamma_1, \gamma_2, \alpha_4, \alpha_3, \alpha_2, \alpha_1, \alpha_0$, and ϕ.

Variable thermal conductivity information.

Group 1 - IVCON(I), VC1F(I), VC2F(I), VC3F(I), VC4F(I), VC5F(I), VC6F(I), VC7F(I), VC8F(I), VC9F(I)

Group 2 - JA, JB, JC, IVCND (JA, JB, JC - defined on page 33 of UM V3.1)

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	Type	Description
IVCON(i)	int	Model type for ith conductivity model. IVCON(i) values are: 1 = linear variation of thermal conductivity with temperature 2 = square root variation of thermal conductivity with liquid saturation. 3 = intact salt 4 = crushed salt
VC1F(i)	real	Reference temperature (°C) for IVCON(i): If IVCON(i) = 1, 3, 4 Conductivity (W/m K) at liquid saturation = 1 for IVCON(i) = 2.
VC2F(i)	real	Reference conductivity (W/m K) for IVCON(i): If IVCON(i) = 1 Conductivity (W/m K) 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. An exponent for IVCON(i) = 3. Coefficient of ϕ^4 for IVCON(i) = 4.
VC4F(i)	real	Coefficient of ϕ^3 for IVCON(i) = 4. Not used for other models
VC5F(i)	real	Coefficient of ϕ^2 for IVCON(i) = 4. Not used for other models
VC6F(i)	real	Coefficient of ϕ^1 for IVCON(i) = 4. Not used for other models
VC7F(i)	real	Coefficient of ϕ^0 for IVCON(i) = 4. Not used for other models
VC8F(i)	real	An exponent for IVCON(i) = 4. Not used for other models
IVCND	int	Number referring to the default is 1

The following is an example of VCON. Three models are defined for the entire domain.
Model 1 defines the constant thermal conductivity of 16.26 W/m K at 26.85^{0}C (=300 ^{0}K) for the stainless steel canister (zone 1). Model 2 defines all parameters for the crushed salt (zone 2). Model 3 defines reference thermal conductivity of 5.4 W/m K at 26.85^{0}C (=300 ^{0}K) and exponent 1.14 for the intact salt in the rest of the domain.

```
vcon
1   26.85     16.26     0.
4   26.85     1.08   -270.0    370.0     -136.0    1.5    5.0    1.14
3   26.85     5.4    1.14

 1  0  0  3
-1  0  0  1
-2  0  0  2
```

6.7 ZNEG (PROBLEM SOLVING)

The **zneg** keyword is a problem-solving tool that allows the user to zero out geometric coefficients that represent negative areas. It is useful for investigating causes of instability in highly nonlinear simulations. A message will be written to the output file that states how many coefficients were set to zero.

Use with caution as the numerical grid will no longer be consistent in the mathematical sense.

December 03, 2015

FEHM UM V3 Appendix
For Software Release FEHM Version 3.3.0
December 03, 2015

This is an addendum to

Software Users Manual (UM) for FEHM Application Version 3.2.0

LA-UR-12-24493

Prepared by:

Shaoping Chu EES-16
Terry Miller EES-16

December 03, 2015

FEHM UM V3 Appendix
For Software Release FEHM Version 3.3.0

1 PURPOSE

The following document is an addendum to the Software Users Manual Version 3.1.0, LA-UR-12-24493 (UM V3.1.0) and FEHM UM V3 Appendix for Software Release FEHM Version 3.2.0. This Appendix describes updates for the FEHM Application Version 3.3.0 release. The use of FEHM V3.3.0 has not changed from FEHM V3.2.0 except to add or improve functionality. Information in this document is considered supplemental to UM V3.1.0 and Appendix (V3.2.0), and also applies to earlier version 3 releases of the User Manual (UM V3). For detail on how to use the FEHM software refer to the UM V3.1.0 or refer to the on-line Wiki pages found at https://ancho.lanl.gov/fehm/trac/wiki (currently restricted to Los Alamos National Laboratory (LANL) employees and collaborators). Publicly available Documentation and Publications can be found on-line at https://fehm.lanl.gov/ (unrestricted).

FEHM V3.3.0 (December 2015) adds to the previous releases of FEHM (V3.2.0 July 2013, V3.1.0 April 2012, and V2.3 January 2008). This FEHM software is a continuation of QA work performed for the Yucca Mountain Project (YMP) under Software Configuration Control Request (SCCR) (Software Tracking Numbers STN: 10086-2.21-00 August 2003, V2.22, STN 10086-2.22-01, V2.23, STN 10086-2.23-00, V2.24-01, STN 10086-2.24-01, and V2.25, STN 10086-2.25-00).

2 FEHM V3.3.0 APPLICATION OVERVIEW

The FEHM Application is used to simulate groundwater and contaminant flow and transport in deep and shallow, fractured and un-fractured porous media throughout the US DOE complex. FEHM has proved to be a valuable asset on a variety of projects of national interest including Environmental Remediation of the Nevada Test Site, the LANL Groundwater Protection Program, geologic CO_2 sequestration, Enhanced Geothermal Energy (EGS) programs, Oil and Gas production, Nuclear Waste Isolation, and Arctic Permafrost.

The following capabilities describe the FEHM V3.3.0 Application.

- 3-dimensional complex geometries with unstructured grids
- saturated and unsaturated media
- simulation of production from gas hydrate reservoirs
- simulation of geothermal reservoirs
- non-isothermal, multi-phase flow of gas, water, oil
- non-isothermal, multi-phase flow of air, water
- non-isothermal, multi-phase flow of CO2, water
- multiple chemically reactive and sorbing tracers
- preconditioned conjugate gradient solution of coupled linear equations
- fully implicit, fully coupled Newton Raphson solution of nonlinear equations
- double porosity and double porosity/double permeability capabilities
- control volume (CV) and finite element method (FE) methods
- coupled geomechanics (THM) problems (fluid flow and heat transfer coupled with stress/deformation) including non-linear elastic and plastic deformation, nonlinear functional

dependence of rock properties (e.g. permeability, porosity, Young's modulus) on pressure, temperature and damage/stress

3 FEHM V3.3.0 RELEASE SUMMARY

All changes to software behavior and software document UM V3.1.0 are detailed in section 6. There are no known effects that change results in models run using the previous version FEHM V3.2.0. The changes did not affect results in the distributed FEHM V&V test suite.

The major changes in the FEHM application from V3.2.0 to this release V3.3.0 are: salt development, updates to stress capabilities, various modifications to improve the performance of FEHM simulation such as enhancements to air-water-heat physics module, and fixing several known code bugs. All changes are described in detail in release notes and Appendix documents posted on the fehm.lanl.gov web page. Enhancements to this FEHM release include:

- Work performed on improving the performance of simulations with salt

  A new controller module (saltctr.f) was created to manage simulations with salt. This was implemented for three primary reasons: First, to isolate the added salt-related hardwired software modifications into one location so that those modifications can be merged with the primary FEHM software suite and allow a platform for future software development; Second, to easily allow (via user input) different salt capabilities to be tested separately to assess individual modification importance and sensitivity; Third, to allow the testing of numerical algorithms associated with the averaging and temporal updating of the highly nonlinear salt-related physics.

  Key additions to capability include:
  1. The ability to run salt simulations with a variety of combinations of water vapor formulations with vapor pressure lowering with different salt concentrations and capillary pressure vapor pressure lowering.
  2. The option to initialize grid blocks to saturation temperature at total fluid pressure or total fluid pressure at saturation temperature. The partial pressure of air in these conditions was set to a prescribed small number.
  3. Improved numerical performance with temporal averaging of porosities permeabilities. This simple change allowed significantly smaller porosities to be reached in a simulation while maintaining large timesteps. It also was the difference between simulations finishing and not finishing for some parameter combinations.

- Enhancements to air-water-heat physics module

  Several modifications were made to the air-water-heat physics module of FEHM that is invoked with the *ngas* keyword. These were made to improve the performance of FEHM when simulating the dry out of rock. Initial testing shows a very good improvement in performance.

  Key additions to capability include:
  1. The ability to set a fixed total pressure with a flowing mass fraction of air. This is useful when applying and high pressure source of low humidity air to dry reservoir rock.
  2. Improved the ability to simulate a constant saturation node by allowing both fixed saturation and fixed pressure at the node. The simulated physics allows the saturation to be maintained at a fixed value by addition/removal of water. The pressure is maintained by the addition/removal of air.
  3. The outflow fluid mix with constant pressure was corrected to include the correct mixture of air and water

4. The boundary conditions for air-water-heat physics are now available with the boun keyword. New keyword 'fxa' for flowing ngas mass fraction added. Several errors in other ngas boundary condition now work.

- Additional stress models have been added
  Mohr-coulomb failure criteria applied to an ensemble of fractures with orientations distributed according to fracture orientation data. Pre-failure increase in permeability occurs according to the Bai et al. (1999) model. Post-failure increase in permeability occurs according to an empirical model based on Lee and Cho (2002).

- The macro “bodyforce” add greater flexibility for specifying body forces, allow user to apply body force on each node.

- The macro "cont” has been modified to add a new keyword “(heat)flux” which adds the option for printing out the x,y,z heat flux vector (W/m2) analogous to the velocity vector.

- The macro "boun" has been modified to add a new keyword “fxa” for handling the boundary conditions for air-water-heat physics.

- Known code bugs fixed
  1. Modification to dvacalc.f to fix a bug in setting the limits of dva.
  2. diagnostics.f: Fixed array accessing error that stopped code on some platforms.
  3. Code change to exponential.f to avoid NaN problem.
  4. write_avs_node_con.f: Fixed contour output problems for vapor species, and array out of bound problems.

  No known bugs in this V3.3.0 release

4 FEHM V3.3.0 V&V TEST SUITE

Before distribution of FEHM software, tests are executed and verified as acceptable on LANL computers with operating systems Linux, Mac OSX, and WINDOWS. FEHM V3.3.0 test results do not differ from FEHM V3.2.0 results for most existing test cases. A few existing test cases have been modified and several new test cases have been added for the new model capability development. The overall validation effort for the FEHM software consists of a suite of directories and scripts that test the model whenever possible, against known analytical solutions of the same problem. The test suite was developed under YMP QA for FEHM RD.10086-RD-2.21-00 and is provided to modelers along with the FEHM application.

The following is a summary of new tests that have been incorporated into the FEHM_VVSECT script test suite V3.3.0 and is based on a comparison to those tests documented for FEHM V3.2.0.

The following is a list of the modification (*) and added tests:

- MULTI_SOLUTE*:

Multi-Solute Transport with Chemical Reaction - trac macro
Multi-Solute Transport with Chemical Reaction - trxn macro

- SORPTION*:

One Dimensional Reactive Solute Transport - trac macro
One Dimensional Reactive Solute Transport - trxn macro

- TRANSPORT3D*:

Three-Dimensional Radionuclide Transport Problem - trac_rlp

Three-Dimensional Radionuclide Transport Problem - trxn_rlp
Three-Dimensional Radionuclide Transport Problem - trac_rlpm

- HEAT_FLUX:

Heat Flux Contour Output Test
Comparison of heat flux contour output test results

- SALT_TEST:

Salt Variable Conductivity Test
Comparison of Salt Conductivity Model to Theoretical Results

- STRESS_BODYFORCE:

Stress Bodyforce Test
Comparison of Stress Bodyforce Model test results

5 FEHM V3.3.0 DISTRIBUTION

FEHM V3.3.0 is compiled, tested, and made available for operating systems Linux (Ubuntu, Red Hat), Mac OSX, and WINDOWS. Compressed archive files are created from each of the V&V Test suites for distribution with the FEHM executable.

The operating system and compiler for this FEHM V3.3.0 release are:
Linux (64-bit): Ubuntu - compiled with gfortran 4.8.2
Red Hat – compiled with gfortran 4.5.1
MAC OSX 10.9.5 (mavericks): compiled with gfortran 4.9.0
PC Windows (32-bit): compiled with Intel Parallel Studio XE 2011
PC Windows (64-bit): compiled with Intel Parallel Studio XE 2011

6 MACRO CONTROL STATEMENTS (SUPPLEMENT TO UM V3 SECTION 6)

The input specifications for the following macros (control statements) in FEHM V3.3.0 have been modified to enhance or clarify use during the modeling process. Each macro description is considered supplemental to documentation in UM V3.1.0, Section 6.2, and each description is complete.

The following updated macros are listed in alphabetic order.

6.1 BODYFORCE (OPTIONAL)

This update supplements UM V3.1.0 Section 6.2

Macro BODYFORCE is used to allow user to apply body force on each node. Input following the problem type is grouped using sub keywords.

Optional keyword on the macro line:

“force” – user specify body force, body force (3 component vector, i.e. (x, y, z) vector in Mega-Newtons of force, unit in kg*m/s^2) given on node by node basis on subsequent lines (zone syntax supported)

“acceleration” – user specify acceleration, body force calculated using acceleration (3 component vector in acceleration, unit in kg/s^2) given on node by node basis on subsequent lines (zone syntax supported)

Group 1 - JA, JB, JC, bodyforce_X, bodyforce_Y, bodyforce_Z (JA, JB, JC - defined on page 33 of UM V3.1.0)

Input Variable	Type	Def	Description
bodyforce_X	real	1.e-30	X component of the force (kg*m/s^2), if using keyword “force” X component of the acceleration (m/s^2), if using keyword “acceleration”
bodyforce_Y	real	1.e-30	Y component of the force (kg*m/s^2), if using keyword “force” Y component of the acceleration (m/s^2), if using keyword “acceleration”
bodyforce_Z	real	1.e-30	Z component of the force (kg*m/s^2), if using keyword “force” Z component of the acceleration (m/s^2), if using keyword “acceleration”

In the following example, the three cases implement the same problem (gravity = 9.8 in -z direction) but using the three available syntax inputs.

```
Case 1: FEHM determines body force based on gravitational load
bodyforce

Case 2: user specifies body force (density x gravity)
bodyforce force
1  0  0  0  0  -0.0245

Case 3: user specifies acceleration, FEHM determines body force (density x
acceleration)
bodyforce acceleration
1  0  0  0  0  -9.8
```

6.2 BOUN

This update supplements UM V3.1.0 Section 6.2.9

Add a new keyword “fxa” for flowing ngas mass fraction, used for the boundary conditions for air-water-heat physics. Example see updates to NGAS macro (section 6.4).

6.3 CONT (OPTIONAL)

This update supplements UM V3.1.0 Section 6.2.18

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’, ‘avsx’, ‘surf’, or ‘tec’)
If CHDUM = ‘ zone’ that line is followed by
    NSURF
    IZONE_ISURF(I), I=1, NSURF

The new change is as follow:

Add a new keyword to input variable CHDUM – (heat)flux.

The new keyword "(heat)flux" adds the option for printing out the x, y, z heat flux vector (W/m2) analogous to the velocity vector.

The following is an example of using keyword (heat)flux.

```
cont
tec     1000    11574.074
geom
temperature
heatflux
liquid
velocity
end
```

6.4 NGAS (OPTIONAL)

This update supplements UM V3.1.0 Section 6.2.61

The new changes to keyword *ngas* input:

Group 1 - ICO2D
Group 2 - JA, JB, JC, PCO2
Group 3 - JA, JB, JC, HUM_SAT, PFLOWA

Group 4 - JA, JB, JC, QCD, AIPED

Input Variable	Type	Def	Description
ICO2D	Int	3	solution descriptor for ngas
PCO2	real	0.	initial partial pressure of non condensable gas
HUM_SAT	real	0.	Specified humidity (If > 0), Specified saturation (if <0, abs value used for specified saturation)
PFLOWA	real	0.1	Specified total pressure used with specified
QCD	real	0.	Specified ngas flowrate (kg/s) when AIPED = 0. , ngas mass fraction when AIPED ne 0.
AIPED	real	1.	IMPEDANCE factor

Example 1. Changes in ngas input showing new ngas mass fraction input. Flow macro shown for completeness.

```
flow
1 1 1 0.105 -30. 1.e0
6 6 1 0.10  -30. 1.e0

ngas
3
1  6  1 1.e-8   <initial ngas partial pressure

                < no humidity or specified saturation
1 1 1 0.99 1.   < if inflow, 0.99 ngas mass fraction applied, aiped = 1.
6 6 1 0.99 1.   < node 6, if outflow, will ignore these values
                < and use in place values (if inflow, see above)
```

Example 2. Boundary conditions for ngas inputted in the boun macro. Flow macro was not needed.

```
ngas
3
1  6  1 -80  < "-80" means the initial ngas partial pressure = P_tot-Pv(80)
             where P_tot is the total pressure and Pv(80) is the water vapor
             pressure at 80 C.
             < no humidity input
             < no ngas mass fraction or ngas flow rate


boun
model 1
ti
2 0. 5.
pw
0.105 0.10 (note specified pressure changed at 5 days)
ft
30. 30.
if
1.e0 1.e0
fxa
0.99 0.99  < flowing mass fraction (fxa) of ngas denoted by 'xa',only for inflow
ts
0.0001 0.0001
model 2
ti
2 0. 5.
pw
0.10 0.105 (note specified pressure changed at 5 days, produces a flow reversal)
ft
30. 30.
if
1.e0 1.e0
fxa
0.001 0.001

1 1 1 1
6 6 1 2
```

6.5 SALT input (OPTIONAL)

This update supplements UM V3.1.0 Section 6.2

User input description for the salt controller:
The input structure for the SALT module is similar to that of CO_2 and Stress Modules. That is, it makes use of sub-keywords within the SALT input section. This module allows input that also available from other keywords (*ppor*, *adif*, *vapl*) but is logically included here as well.

KEYWORD "saltppor" Keyword specifying type of porosity compressibility model
Group 1 - IPOROS (only IPOROS = 6 or 7 allowed)

Group 2 - JA, JB, JC, POR1, POR2 ,POR3, POR4 (1 parameter entered for IPOROS = 6,
4 parameters entered for IPOROS = 7)

A warning message is written to the output file and the ".err" file if a salt porosity model is not entered.

KEYWORD "saltvcon"

Only one thermal conductivity model (4) is implemented for salt in the saltctr.f. It is based on the thermal conductivity for crushed salt (Bechthold et al, 2004).

Group 1 - IVCON(I), VC1F(I), VC2F(I) , VC3F(I), VC4F(I), VC5F(I), VC6F(I), VC7F(I), VC8F(I)

Group 2 - JA, JB, JC, IVCND

Input Variable	Format	Description
IVCON(I)	integer	model type IVCON(I) = 4 (only salt model available)
VC1F(I)	real	reference temperature (C)
VC2F(I)	real	porosity-related used in Bechthold equation
VC3F(I)	real	Coefficient of 4th order term in Bechthold equation
VC4F(I)	real	Coefficient of 3rd order term in Bechthold equation
VC5F(I)	real	Coefficient of 2nd order term in Bechthold equation
VC6F(I)	real	Coefficient of 1st order term in Bechthold equation
VC7F(I)	real	Constant term in Bechthold equation
VC8F(I)	real	Power law term in Bechthold equation
IVCND	integer	model number to apply to nodes

A warning message is written to the output file and the ".err" file if a salt thermal conductivity model is not entered.

KEYWORD "saltden"

KEYWORD "saltadif"

Group 1 - TORT

The appropriate diffusion models are TORT = 333 and TORT = 666 and are based on the Millington Quirk model as programmed originally by Pruess. If other models are used, a warning message is written to the output file and the ".err" file.

KEYWORD "saltvapr"

Group 1 - IVAPRSALT

IVAPRSALT - identifies the vapor pressure lowering model

Reference: Sparrow (2003) Desalination.

Note: The Sparrow formulation has no capillary vapor pressure lowering.

Input Variable	Format	Description
IVAPRSALT	integer	IVAPRSALT = 0 - traditional FEHM h2o vapor pressure fit with vapor pressure no lowering
IVAPRSALT	integer	IVAPRSALT = 1 - Sparrow vapor pressure model with no salt
IVAPRSALT	integer	IVAPRSALT =2 - Sparrow vapor pressure model with salt

IVAPRSALT	integer	IVAPRSALT =3 - Sparrow vapor pressure model with salt and capillary vapor pressure lowering (not yet implemented)
IVAPRSALT	integer	IVAPRSALT =4 - traditional FEHM vapor pressure model with capillary vapor pressure lowering
IVAPRSALT	integer	IVAPRSALT =5 - traditional FEHM vapor pressure model with salt and no capillary vapor pressure lowering
IVAPRSALT	integer	IVAPRSALT =6 - traditional FEHM vapor pressure model with salt and capillary vapor pressure lowering

KEYWORD "saltnum"

This keyword manages the updating of the nonlinear salt related variables

Group 1 - ACTION, VALUE

Input Variable	Format	Description
ACTION	character	Salt process to be modified ACTION = " permavg" - average permeability after every *tracer* timestep ACTION = " poravg" - average porosity after every *tracer* timestep ACTION = " pormin" - set minimum porosity
VALUE	real	parameter value related to process ACTION = " permavg", VALUE = 1-use new time step permeability VALUE = 0-use old time step permeability VALUE = 0.5-use average permeability ACTION = " poravg", VALUE = 1-use new time step porosity VALUE = 0-use old time step porosity VALUE = 0.5-use average porosity ACTION = " pormin", VALUE = minimum porosity for ppor model 7

KEYWORD "saltend"

6.6 STRS (OPTIONAL)

This update supplements UM V3.1.0 Section 6.2.85

The solid mechanical deformation module is invoked with this control statement.

Group 1- ISTRS, IHMS

Group 2- KEYWORDS

The remaining input is entered in subgroups defined by additional keywords. These keywords are all optional unless otherwise noted, but the user should be careful to ensure the problem is completely defined with the keywords selected.

Input associated with KEYWORDS is shown below and described in more detail in the following table. Unless otherwise specified, blank lines are not permitted.

KEYWORD “initcalc” (optional, no other input)

KEYWORD “bodyforce” (optional, no other input)

KEYWORD “fem” (optional, no other input)

KEYWORD *“reldisp”* (optional)

KEYWORD *“stresspor”* (optional)

KEYWORD “excess_she” (optional)

FRICTION_OUT, STRENGTH_OUT, PP_FAC_OUT

KEYWORD “permmodel” (optional)

IPERM, SPMF1, SPMF2, . . . SPMF13

(as many models as needed,one per line, terminated by a blank line)

JA, JB, JC, MODEL_NUMBER (JA, JB, JC – defined page 33 of UMV3)

KEYWORD “elastic” (required)

JA, JB, JC, ELASTIC_MOD, POISSON

KEYWORD “nonlinear” (optional)

NONLIN_MODEL_FLAG

If the value of NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young’s modulus and Poisson’s ratio:

E_INI, DEDT, POISSON_INI, DNUEDT

Else, if the value of NONLIN_MODEL_FLAG = 91then a table lookup is used:

YOUNG_TEMP_FILE

KEYWORD “plastic”

NUMPLASTICMODELS

The following are repeated NUMPLASTICMODELS times

PLASTICMODEL, MODULUS, NU, [PLASTICPARAM1, PLASTICPARAM2]

JA, JB, JC, MODELNUMBER

KEYWORD “biot” (required)

JA, JB, JC, ALPHA, PP_FAC

KEYWORD “stressboun” (required)

SUB-KEYWORD ‘distributed’ or ‘lithostatic’ (optional)

or

SUB-KEYWORD ‘lithograd’ SDEPTH GDEPTH (optional)

JA, JB, JC, BOUNVAL, KQ

KEYWORD “tolerance (required)

STRESS_TOL

KEYWORD “end strs” (required)

The input is terminated with keyword “*end strs*” or “*endstrs*”.

Input Variable	Type	Def	Description
ISTRS	int	0	ISTRS values indicate the program’s state of stress: 0 = skip stress solution 1 = 3-D solution
IHMS	int		IHMS is used to identify the amount and frequency of coupling between TH and M parts of the code. The following values apply: -1 = stress solved only at the end of the TH (flow) simulation -2 = stress solved at the beginning and end of the TH (flow) simulation (useful for establishing a lithostatic load -3 = stress solved iteratively within the NR_loop for each timestep of the TH (flow) simulation
KEYWORD “end strs” or “endstrs”			End of **strs** input.
KEYWORD “initcalc”			Initiate an initial stress calculation that is useful for establishing lithostatic stress.
KEYWORD “bodyforce”			Sets a body force if gravity is non-zero. Force is calculated using the rock density information provided in the rock macro.
KEYWORD “reldisp”			Use relative displacement in the calculation of volume strains, permeability models, and output.
KEYWORD “stresspor”			Explicitly update the porosity after each time step.
KEYWORD “fem”			Use the Finite Element modules for forming displacement equations, and calculating stresses. Although optional, it is strongly recommended that this keyword be included.
KEYWORD “principal”			For stress output to the files generated by the cont macro, output the principal stress values

	and the orientation of the axis.		
KEYWORD "strainout"	Create a file, strain.out, containing x, y, z, node number, εxx, εyy, εzz, εxy, εxz, εyz		
KEYWORD "excess_shear"	This controls stress output to files generated by the 'cont' macro, output the excess shear stress and the direction of the failure plane given in the equation below, as well as the Young's modulus $\tau_{excess}(\max) = \frac{1}{2}(\sigma_1 - \sigma_3)(\mu^2+1)^{1/2} - \frac{1}{2}\mu(\sigma_1+\sigma_3) - S_0$ Where τ_{excess} is the excess shear, σ_1 and σ_3 are the maximum and minimum principal stresses, μ is the coefficient of friction, and S_0 is the shear strength. The angle β between this plane and the orientation of the maximum principal stress is given by $\tan(2\beta) = -\frac{1}{m}$. NOTE: The values specified here are used only for output, and are superseded by values specified in 'permmpdel' macro.		
FRICTION_OUT	real		Coefficient of friction
STRENGTH_OUT	real		Cohesion
PP_FAC_OUT	real		Pore pressure factor similar to Biot's coefficient in the 'biot' macro.
KEYWORD "zone"	The format and inputs for this are described in the zone macro. Inclusion of zone macros within the strs macro are allowed to facilitate input associated with the following keywords.		
KEYWORD "permmodel"	This keyword identifies the stress or displacement dependent permeability model. The permeability model can be invoked in a fully coupled, sequentially coupled, or explicitly coupled manner.		
IPERM	int	1	Specifies the type of permeability model used, input parameters specified on this line change depending on the model selected.
IPERM = 1	Equivalent to no stress permeability model, no further input parameters		
IPERM = 2	Stress permeability model dependent on tensile stress in the coordinate directions. Changes are linear in stress up to the prescribed maximum change. Tensile stress in a given coordinate direction affects the permeability in the other two directions. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f		
SPM1F	real		Minimum tensile stress (x direction) for damage to occur.
SPM2F	real		Minimum tensile stress (y direction) for damage to occur
SPM3F	real		Minimum tensile stress (z direction) for damage to occur
SPM4F	real		Damage factor for elastic modulus in x direction.
SPM5F	real		Damage factor for elastic modulus in y direction.
SPM6F	real		Damage factor for elastic modulus in z direction.
SPM7F	real		Maximum factor for x-permeability.
SPM8F	real		Maximum factor for y-permeability.
SPM9F	real		Maximum factor for z-permeability.
IPERM =21	Mohr-coulomb failure criteria on a user specified plane. Here z-prime is along the normal to the plane of failure, and y-prime is along the plane of median principal stress. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f, spm10f, spm11f, spm12f, spm13f		
SPM1F	First direction cosin of the normal to the user specified fault plane		
SPM2F	Second direction cosin of the normal to the user specified fault plane		
SPM3F	Third direction cosin of the normal to the user specified fault plane		
SPM4F	Friction coefficient of shear in the fault plane.		
SPM5F	Shear strength of the fault plane.		

SPM6F	Factor in effective stress calculation where effective_stress = sigma - (pp_fac*pore pressure)		
SPM7F	Range of excess shear stress over which damage is ramped		
SPM8F	Maximum multiplier for young's modulus in x-prime direction.		
SPM9F	Maximum multiplier for young's modulus in y-prime direction		
SPM10F	Maximum multiplier for young's modulus in z-prime direction		
SPM11F	Maximum multiplier for permeability x-prime direction.		
SPM12F	Maximum multiplier for permeability y-prime direction.		
SPM13F	Maximum multiplier for permeability z-prime direction.		
IPERM = 22	Mohr-coulomb failure criteria on the plane that maximizes the excess shear. Here z-prime is along the normal to the plane of failure, and y-prime is along the plane of median principal stress. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f, spmf10, spmf11f, spm12f		
SPM1F	real		Friction coefficient of shear in the fault plane.
SPM2F	real		Shear strength of the fault plane.
SPM3F	real		Factor in effective stress calculation where effective_stress = sigma - (pp_fac*pore pressure)
SPM4F	real		Range of excess shear stress over which damage is ramped
SPM5F	real		Maximum multiplier for young's modulus in x-prime direction.
SPM6F	real		Maximum multiplier for young's modulus in y-prime direction
SPM7F	real		Maximum multiplier for young's modulus in z-prime direction
SPM8F	real		Maximum multiplier for permeability x-prime direction.
SPM9F	real		Maximum multiplier for permeability y-prime direction.
SPM10F	real		Maximum multiplier for permeability z-prime direction.
SPM11F	real	1.	Optional, Maximum multiplier for porosity
SPM12F	real	0	Optional, permeability is made a function of current stresses minus this factor time the initial stresses. NOTE: if this option is to be used without the porosity modification, a value of 1 must be entered for SPM11F
IPERM = 24	Stress permeability calculated for ensemble of shear-stimulated fractures. An isotropic fracture distribution is assigned for control volume. The Mohr-Coulomb failure criterion is evaluated for each fracture. If it fails, it's in-plane permeability is enhanced according to a calculated stress drop and shear displacement. The total change in nodal permeability is the average of all fractures. Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f		
SPM1F	real		Shear fracture toughness. Converts calculated stress drop to shear displacement according to disp = stress_drop/toughness.
SPM2F	real		Static friction coefficient, for calculating onset of Mohr-Coulomb failure.
SPM3F	real		Dynamic friction coefficient, for calculating stress drop, where stress_drop = (static-dynamic)*(norm_strs-pressure)-cohesion
SPM4F	int		Number of fractures per control volume.
SPM5F	real		Shear displacement for onset of permeability enhancement.
SPM6F	real		Shear displacement interval over which permeability enhancement completes.
SPM7F	real		Logarithm of maximum per-fracture permeability multiplier.
SPM8F	real		Fracture cohesion, used in calculation of Mohr-Coulomb failure.
SPM9F	real		Fracture density in control volume (default = 1), can be used as a secondary, isotropic permeability multiplier.
IPERM = 25	Mohr-coulomb failure criteria applied to an ensemble of fractures with orientations distributed according to fracture orienation data. Pre-failure increase in permeability occurs according to the Bai et al. (1999) model. Post-failure increase in permeability occurs according to an empirical model based on Lee and Cho (2002). Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f		
SPM1F	real		Shear fracture toughness. Converts calculated stress drop to shear displacement according to disp = stress_drop/toughness.

SPM2F	real		Static friction coefficient, for calculating onset of Mohr-Coulomb failure.
SPM3F	real		Dynamic friction coefficient, for calculating stress drop, where stress_drop = (static-dynamic)*(norm_strs-pressure)-cohesion
SPM4F	int		Number of fractures per control volume.
SPM5F	real		Shear displacement for onset of permeability enhancement.
SPM6F	real		Shear displacement interval over which permeability enhancement completes.
SPM7F	real		Logarithm of maximum per-fracture permeability multiplier.
SPM8F	real		Fracture cohesion, used in calculation of Mohr-Coulomb failure.
IPERM = 91		Table input from a file, name given on the next line	
FILENAME	char		Name of the file with permeability model factors. The file has the following format:Line 1: # of rows in the tableLines 2 through (# of rows)+1: stress, x-factor, y-factor, z-factor
KEYWORD “elastic”		For linear elastic material.	
ELASTIC_MOD	real		Young’s modulus. (MPa)
POISSON	real		Poisson’s ratio.
KEYWORD “nonlinear”			
NONLIN_MODEL_FLAG	int		If NONLIN_MODEL_FLAG = 1 then this model is for linear dependence on temperature of Young’s modulus and Poisson’s ratio. Input: E_INI, DEDT, POISSON_INI, DNUEDT
			If NONLIN_MODEL_FLAG= 91 then a table lookup is used. Input: YOUNG_TEMP_FILE
E_INI	real		Value of Young’s modulus at the reference temperature (MPa).
DEDT	real		Derivative of Young’s modulus with respect to temperature (MPa/ °C)
POISSON_INI	real		Value of Poisson’s ratio at the reference temperature.
DNUEDT	real		Derivative of Poisson’s ratio with respect to temperature (per °C)
YOUNG_TEMP_FILE	char		Name of the file with nonlinear model values. The file has the following format: Line 1: # of rows in the table (nentries_young) Line 2 through (# of rows)+1: temperature, young’s modulus, poisson’s ratio
KEYWORD “plastic”			
NUMPLASTICMODELS	int		Number of plastic models. NOTE: Current version allows only ONE plastic model for the entire domain.
PLASTICMODEL	real		The Plastic model number PLASTICMODEL has the following values: If PLASTICMODEL = 1 Isotropic, linear elastic solid. If PLASTICMODEL = 2 von Mises model. If PLASTICMODEL = 3 Drucker-Prager model. PLASTICPARAM1 and PLASTICPARAM2 are entered only for the von Mises model. PLASTICPARAM1, PLASTICPARAM2 and PLASTICPARAM3 are entered only for the Drucker-Prager model.
MODULUS	real		Young’s modulus in the elastic region (MPa).
NU	real		Poisson’s ratio in the elastic region.
PLASTICPARAM 1	real		Yield stress for von Mises model (MPa).
PLASTICPARAM 2	real	0.	Currently not used.
KEYWORD “biot”			
ALPHA	real	0	Volumetric coefficient of thermal expansion (per °C)
PP_FAC	real		Factor multiplying the pore pressure coupling term in the stress-strain relations.
KEYWORD “stressboun”		Enter boundary conditions for the mechanical deformation equations. These can be a combination of specified values of displacements, stresses, or forces.	
SUB-KEYWORD		Distribute the applied force in proportion to areas of the members of the zone to which the	

'distributed'		force is applied.	
SUB-KEYWORD 'lithostatic'		BOUNVAL and KQ are interpreted as multipliers of the lithostatic stress and the stress direction. The lithostatic stress is always in the vertical (downward) direction. The z-axis is taken to be positive upwards. In the ctrl macro the direction of gravity must be set to 3.	
SUB-KEYWORD 'lithograd' SDEPTH GDEPTH		BOUNVAL and KQ are interpreted as the stress gradient and stress direction. The parameters sdepth and gdepth are read on the same line as lithograd, and the KQth diagonal component of the stress at any node is calculated as follows, where z is the vertical coordinate of the node (see Figure 3 below). Stress(kq) = (sdepth+gdepth-z)*bounval	
SDEPTH	real		Depth (m) of the reference level from the free surface of the earth corresponding to the level specified by GDEPTH, i.e., model elevation of GDEPTH meters is equivalent to SDEPTH meters depth.
GDEPTH	real		In the coordinate system of the model, the z coordinate of the reference level.
BOUNVAL	real	0	This is a fixed displacement, specified stress, or specified force depending on the value of KQ and optional keywords. No keyword, and kq > 0 : prescribed displacement (m) in the kq direction No keyword and kq < 0 : applied stress (MPa) in the kq direction Keyword = 'lithograd' and kq > 0 : the stress gradient (MPa/m) in the kq direction Keyword = 'distributed' and kq < 0 : prescribed force (MN) in the kq direction.
KQ	int	0	Parameter that determines the type of boundary condition kq = 1 or -1: prescribed value in the x direction kq = 2 or -2: prescribed value in the y direction kq = 3 or -3: prescribed value in the z direction
KEYWORD "tolerance" (required)			
STRESS_TOL	real	0	The tolerance for solution of the stress equations as follows: STRESS_TOL > 0 STRESS_TOL is the reduction of initial residual of the stress equations STRESS_TOL < 0 STRESS_TOL is the required absolute value of the residual of the normalized equations

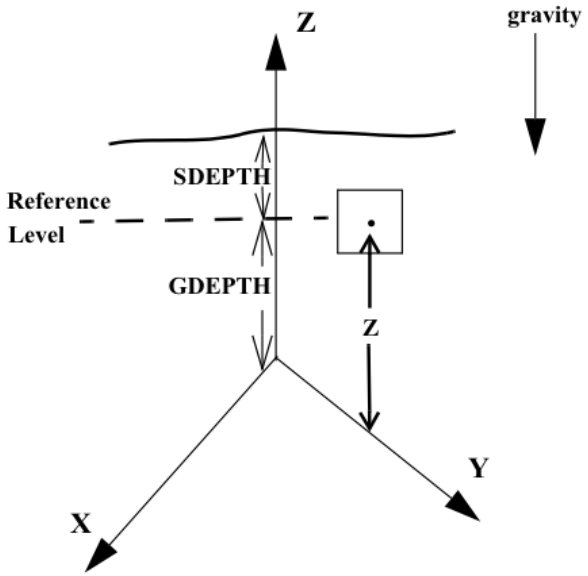

Figure 3. Schematic illustrating variables for 'lithograd' SUB-KEYWORD.

In the 3D example below, the option to explicitly couple stress with heat-mass equations is invoked. Initial stresses and displacements are calculated, a body force due to gravity is applied, optional strain output is activated, computations are performed using the finite element module, material is specified to be elastic, with temperature dependence of Young's modulus and Poisson's ratio specified in a file called "EvsT.txt", linear coefficient of thermal expansion 1.4e-5/0C, Biot's coefficient equal to 0. Zone 3 is pinned in all 3 directions, zones 4 And 5 are constrained in the X direction, and zones 6 and 7 are constrained in the Y direction. Tolerance for the stress solution is set to 1.e-3.

```
strs
1 -3
initcalc
bodyforce
strainout
fem
elastic
1 0 0 1.59e4 0.25

nonlinear
91
EvsT.txt
biot
1 0 0 5.4e-5 0.

zone
2 ! top,Z=300
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 299.99 299.99 299.99 299.99

3 ! bottom, Z=0
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
0.1 0.1 0.1 0.1 -0.1 -0.1 -0.1 -0.1

stressboun
-3 0 0 0. 3

stressboun
-3 0 0 0. 2

stressboun
-3 0 0 0. 1

zone
4 ! back X=20
19.99 20.01 20.01 19.99 19.99 20.01 20.01 19.99
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 -1. -1. -1. -1.
5 ! front X=0
-0.01 +0.01 +0.01 -0.01 -0.01 +0.01 +0.01 -0.01
+1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15
300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-4 0 0 0. 1

stressboun
-5 0 0 0. 1
```

```
zone
6 ! right, Y=0
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
0.01 0.01 -0.01 -0.01 0.01 0.01 -0.01 -0.01
300.01 300.01 300.01 300.01 -1. -1. -1. -1.
7 ! left, Y=60.
-1.e+15 +1.e+15 +1.e+15 -1.e+15 -1.e+15 +1.e+15 +1.e+15 -1.e+15
60.01 60.01 59.99 59.99 60.01 60.01 59.99 59.99
300.01 300.01 300.01 300.01 -1. -1. -1. -1.

stressboun
-6 0 0 0. 2

stressboun
-7 0 0 0. 2

tolerance
-1.e-3

end stress
```