

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

Update August 8, 2013

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 fehmlanl.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 CO₂ 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 gdp 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](https://doi.org/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 CO₂ problem. Input following the problem type is grouped using sub keywords.

Update August 8, 2013

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, DENC, 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 CO ₂
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 CO ₂ (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, CO ₂ /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 CO₂ dissolved water at 0.001 kg/s. The temperature is 20°C. The water has a dissolved CO₂ 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 CO₂ 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.

Update August 8, 2013

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) IGRADF = 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.
```

Update August 8, 2013

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)

FRICITION_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:

Update August 8, 2013

E_INI, DEDT, POISSON_INI, DNUEDT

Else, if the value of NONLIN_MODEL_FLAG = 91 then 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, ϵ_{xx} , ϵ_{yy} , ϵ_{zz} , ϵ_{xy} , ϵ_{xz} , ϵ_{yz}
KEYWORD “excess_shear”			<p>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</p> $\tau_{excess}(\max) = \frac{1}{2}(\sigma_1 - \sigma_3)(\mu^2 + 1)^{1/2} - \frac{1}{2}\mu(\sigma_1 + \sigma_3) - S_0$ <p>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</p> $\tan(2\beta) = -\frac{1}{\mu}.$ <p>NOTE: The values specified here are used only for output, and are superseded by values specified in ‘permmpdel’ macro.</p>
FRICION_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 str 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			<p>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.</p> <p>Input Variables: iperm, spm1f, spm2f, spm3f, spm4f, spm5f, spm6f, spm7f, spm8f, spm9f</p>
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_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.
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 °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.			

Update August 8, 2013

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 `txn`. Its options are as follows:

Option	Description
<code>rxnon</code>	Enables reactions, which are by default disabled. Reactions will not occur in a simulation if <code>rxnon</code> is not given, even if reaction-related blocks are specified.
<code>co2_couple</code>	Enables CO ₂ coupling for <code>rxn</code> .
<code>debug</code>	Enables the output of debugging information.

Example of the `ctrl` block; In this example, CO₂ 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 `txn` macro. The first non-blank (and non-commented) line of the include file should be `txn library`. Following this, as many blocks as desired may be supplied. The end `txn` 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 `txn` macro. Libraries may include other libraries.

The syntax is as follows:

```
include /scratch/nts/ms/txn/test-problems/standard.txl
```

This will include the library located at `/scratch/nts/ms/txn/test-problems/standard.txl` 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

txn 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 a11 a21 b1 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	ANO AWC EPC UPWGTA
2	DAYCS DAYCF DAYHF DAYHS
3	IACCMX DAYCM DAYCMM DAYCMX NPRTRC
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, NPRTRC
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, NPRTRC 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    UO2   *
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 $1 \times 10^{-6} m$, an Na⁺ concentration of $1.34 \times 10^{-5} m$, a CO₃⁻⁻ concentration of $1 \times 10^{-10} m$, 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 txn 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
C12   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 b1 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_1 parameter of 0.89 and a vapor α_1 parameter of 0.35 in adsorption model model1, and modeled using the conservative solute model with a liquid α_1 parameter of 1.20 and a vapor α_1 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 ¶

Update August 8, 2013

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 CaHCO_3^+ is $1.0 \times 10^{-13.456}$. The enthalpies of CaHCO_3^+ and OH^- are $1.0 \times 10^0 = 1$, but the enthalpy of $\text{Ca}_2\text{UO}_2(\text{CO}_3)_3$ 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, CaHCO_3^+ has an equilibrium 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
```

Update August 8, 2013

```
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 trxn timer 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: A1, A2, A3, A4, A5, where
log10 K = A1 + A2 · T + A3 / T + A4 · log10 T + A5 / T2
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, `trxn`, 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 `trxn` 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/trxn`. 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/trxn 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 `txn` 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/txn/geochem/phreeqc.trxd
```

In this example, the database is located at `/scratch/nts/ms/txn/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 may 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 is a list of comma-separated name/value pairs, where the acceptable names are <code>rate</code> to specify the rate of the reaction and <code>distcoef</code> to specify the distribution coefficient. The value for <code>distcoef</code> may be a real number if the coefficient is constant, or the name of a model specified in <code>dist</code> (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 <code>maxconc</code> , used to set the maximum concentration that can sorb.
3	General kinetic reaction	This reaction accepts its first line 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 <code>forward</code> to set the forward rate constant for the reaction or <code>reverse</code> 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	<p>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.</p> <p>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).</p>
5	Radioactive decay reaction	<p>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.</p>
7 and 8	Precipitation/dissolution reaction	<p>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/m²/s); sarea - the surface area of the mineral (m²/m³ 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.</p> <p>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.</p>

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, pthreshold=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 omitted, 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 `txn` 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 `txn` keyword to end the reading of the `txn` 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 all  a2l  b1
.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 1=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

```

Update August 8, 2013

```
# 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 txn
```

Below are the trac and rxn macros that were replaced by the above txn, 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
** IDCNT(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), IMPRT(IM) (ID # and name of immobile spec, NIMM
rows)
1 Co-EDTA[s] 0
2 Fe-EDTA[s] 0
3 Cobalt[s] 0
** IDVAP(IV), VAPNAM(IM), VAPPRT(IV) (ID # and name of vapor spec, NVAP rows)
** Skip nodes
0
** 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 Enthalpy, 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 coefficient (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 coefficient (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 coefficient (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 rxn interface¶

Please note that the following features present in trac and rxn have been removed in rxn:

- 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 rxn format. These can be found in /scratch/nts/ms/rxn/test-problems. The input files contain the original trac and/or rxn macros (turned off), along with an equivalent rxn macro. The output for the rxn 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 rxn:

- 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: $\lambda_{IS} = \lambda_{IS,300} (300/T)^{\gamma_1}$

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

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

Parameters related to thermal conductivity are $\lambda_{IS,300}$, γ_1 , γ_2 , α_4 , α_3 , α_2 , α_1 , α_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°C (=300 °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°C (=300 °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.

Government report disclaimer

This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither Los Alamos National Security, LLC, the U.S. Government nor any agency thereof, nor any of their employees make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by Los Alamos National Security, LLC, the U.S. Government, or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of Los Alamos National Security, LLC, the U.S. Government, or any agency thereof.