November 12, 2019

Release Summary
For Software Release FEHM Version 3.4.0
September 22 2017 to September 12 2019

LANL Software: LA-CC-2012-083 No. C13022 LANL Documents: LA-UR-12-24493

Prepared by:

Shaoping Chu EES-16
Terry Miller EES-16
George A. Zyvoloski

This page is intentionally left blank.

November 12, 2019

FEHM V3.4.0 Release Summary

1 OVERVIEW

The following document is an addendum to the Software User's Manual Version 3.1, LA-UR-12-24493 (UM V3.1.0). This Release Summary describes updates for the FEHM Application Version 3.4.0 release. Information in this document is considered supplemental to UM V3.1. Section 1 of this Summary gives an overview and summary related to the latest FEHM release.

FEHM V3.4.0 (November 2019) adds to the previous releases of FEHM (V3.3.1 September 2017, V3.3.0 December 2015, 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).

The following sections update User Manual 3.1. Section 1 updates the FEHM Capability list and describes release summary, Section 2 summarizes developer's notes on code changes. Section 3 are discussions and examples related to issues using the FEHM software.

1.1 FEHM CAPABILITIES

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. Detailed capabilities are described in FEHM UM V3.1.0 and Appendixes.

1.2 FEHM RELEASE SUMMARY

All changes to software behavior and software document UM V3.1.0 are detailed in UM V3.1 section 6. There are no known effects that change results in models run using the previous version FEHM V3.3.1. The changes did not affect results in the distributed FEHM V&V test suite.

The major changes in the FEHM application from V3.3.1 to this release V3.4.0 are: salt and GDKM development, various modifications to improve the performance of FEHM simulation, fixing several known code bugs, and salt_test in V&V test suite was updated with improved theoretical solution (theoretical_result.txt).

1.2.1 Enhancements include:

- Nonlinear rock heat capacity including rock melting. This capability allows the rock heat capacity with temperature dependence to be used in simulations. The numerical formulation for rock energy was modified to allow internal energy changes that represent rock melting. Several different heat capacity/melting models are available including input in tabular form. This capability will be fully tested and input/output described when the High Temperature version of FEHM is merged in the next release.

- Negative volume. A simple check for negative volumes associated with inconsistent local element node ordering was implemented.

- Humidity boundary conditions. Humidity boundary conditions were added to the BOUN macro.

- Gravity term correction. There are instances in multiphase simulations where a single phase gridblock is next to a multiphase block. In this case, it is possible that the density of one of the phases is not calculated for the single phase block. This affects the gravity term and is now corrected. (Seems to save a few iterations.) First discovered in TOUGH code at the University of Auckland.

- Salt project. Modified several of the salt routines. These changes were primarily for improvements to output. There are some computational efficiency improvements (saltctr).

- SRC_CH2017 (general). Moved vapor pressure lowering calls and improved the free drainage boundary condition for isothermal flow.

- Principal stress output. Changes the output for principal stress. Moved some arrays to use modules for improved efficiency.

- Saved Zone capability. This allows zones to be saved once and for all. This allows sets of zones not to lose nodes because of overlapping nodes defined sequentially. Useful for setting stratigraphy and boundary conditions and for outputting separate contour zones.

- GDKM. This is a major update to the GDKM capability. Corrected known bugs, simplified input, and added models. The models include the generic fracture model (similar to dpdp) and three directional fracture models.

1.2.2 Code bugs fixed:

1. varchk_2.f - fixed a bug in the variable update portion of routine for AWH and SC phase
2. resetv.f - fixed a bug. Now allows a timestep restart when a SC phase is present
3. Residual.f – bug fix so that 135 Xe concentration test passes
4. Sub_gmres1.f - bug fix so that 135 Xe concentration test passes
5. thrmwc.f- continued improvement of ngas physics (AWH) boundary conditions, debugged and modified high temperature table
6. vcon.f - bug fix for earlier implementation of thermal conductivity for porosity above 0.3873 to FEHM. Included temperature dependence calculation and accurate linear slope equation for porosity above 0.3873 situation.

1.2.3 Known Issues:

User-commented known issues:

08/29/2019: *problems compiling FEHM with Intel Visual Fortran*

Checking out the flag "Check Routine Interfaces" in the "Diagnostics" option of the Fortran workspace properties can let compilation of FEHM go through. There are two types of issues: 1) related to the setbit subroutine - the dummy arguments are all integer (4 bytes) while the third argument in the calling is of type real*8. Even if compilation goes through, it is not sure that results could be reliable; 2) there are many cases where the size of dummy arrays are larger than the ones in the calling sequence - but this is not a problem.

1.3 FEHM V3.4.0 DISTRIBUTION

FEHM V3.4.0 LA-CC-2012-083 CODES No. C13022 is available as open source and licensed under the BSD-3 license at https://github.com/lanl/FEHM. FEHM is compiled, tested, and made available for operating systems Ubuntu Linux, Mac OSX, and WINDOWS.

FEHM V3.4.0 Executables:

Linux (64 bit, linux, Ubuntu 16):
xfehm_v3.4.0lnUbuntu16.12Sep19 compiled with gfortran 5.5.0
Windows (64-bit):
FEHM_V3.4.0w64.12Sep19.exe compiled with Intel Parallel Studio XE 2013
MAC:
xfehm_v3.4.0macMojave.12Sep19 compiled with gfortran 8.1.0 static, Darwin Kernel version 18.7.0, OSX Version 10.14.6 Mojave

2 SUMMARY OF DEVELOPER NOTES ON CODE CHANGES

Developer Notes 09/22/2017 to 09/12/2019 ----------

---------- **Code changes 10/07/2017:**
GRAV_TERM:
comji.f - added array grav_wgt(:)
flow_boundary_conditions.f - fixed error on specified air pressure
geneqc.f - set grav wgt = 1 if one of the densities (i.e. rovf(i) le.0)
varchk.f - adjustment to default saturation tolerance

SALT_OLivella:
comai.f - added global parameter 'iaprf' to manage Olivella intrinsic permeability reduction function; added global para meters in support of code modifications
comdi.f - added globalparameters k0f, bkf, por0ffor the Olivella intrinsic permeability reduction function
data.f - zeroed parameter 'iaprf'
porosi.f - Added coding to enable call to Olivella permeability function (including perm initialization); Corrected the number of loop calls and improved cpu.
saltctr.f - Added input coding for Olivella thermal conductivity (and warning in model 4 or 5 are not used with salt). Also added input for Olivella intrinsic perm function. Appended subroutine perm_olivella to the end of saltctr.f. Added conditional to Olivella perm function to enable calculation of a single gridblock to be consistent with loop on porosi.
fehmn_pcx.f - changed comment only near call to porosi()
flow_boundary_conditions.f - added code so that a specified saturation entered in the boun macro works with Richards' equation
vcon.f - Corrected existing model to implement Olivella thermal conductivity function.
(still some question as to Temp units K or C (Olivella, 2011 seems to be C)
Can't find a reference temperature.

PRINCIPAL_STRESS:
write_avs_head_s.f - changed printout order of header for principal stresses

SRC_CH2017:
psatl.f - modified vapor pressure to use Sparrow for salt repository
vaporl.f - added coding to include sparrows 2003 equation
thrair.f - modified free drainage BC

SV_ZONE:
zone.f - added save zone capability. Can define zone and then use them later in the simulation.
They won't disappear. Also supports special SoilVision output.
write_avs_node_v.f - added coding to support SoilVision output
write_avs_node_s.f - added output for gdkm with blanking and no dual grid
write_avs_node_con.f - added output for gdkm with blanking and no dual grid. Added mods for SoilVision support.

read_avs_io.f - enable SOIL VISION output
initdata2.f - modified to use saved zones.
fehmn_pcx.f - perform restart if equation normalization fails
allocmem.f - added memory allocation for new features and also for SoilVision arrays.
user_ymp.f - moved array elem_temp(:) to use module combi.f

NONLIN_CP:
comdi.f - added variables associated with rock melting
enthp.f - modified for heat conduction only. (might have to set temperature with hflx macro)
input.f - added input call to vrock_ctr.f for rock melting
thermw.f - added coding for rock melting
thrmwc.f - added coding for rock melting. Added coding to implement new humidity related boundary conditions.
vrock_ctr,f - new controller for rock melting
wrtout.f - corrected ethalpy output in *.out file for heat conduction
inrock.f - changed default values to real*8 (i.e. 2500. to 2500.d0)

HUMIDITY:
New humidity BC development:
Note: most of the files were changed to support developments in humidity initialization, output and boundary conditions. 'humidity' is used for the relative humidity fraction.
model_setup.f - added boun sub keywords 'huf' (flowing humidity),'hu' (fixed humidity),
th (humidity temperature), ph (humidity pressure)
co2ctr.f - set relative humidity fraction at input temperature for gas phase, print out relative humidity fraction
flow_humidity_bc.f (new) - calculates the air and water mass fractions of an inflowing gas flow so that the inflow has a specified relative humidity.
wrtcon.f - added coding to output more detail on SIA iterations
flow_boun.f - added coding for new humidity boundary conditions.

GDKM_MODS:
add_gdpm.f - added input for newer gdkm directional models
combi.f - added array for gdkm modifications
coneq1.f - modified diffision/dispersion to have correct weighting with gdkm models
check_rlp.f - made temp array sizes consistent with corresponding variable arrays
dvacalc.f - move dva_save(:) to use module comci.f, Added GDKM weighting
comci.f - added array dva_save(:)
gdkm_connect.f - modified volume fraction calcs
gdkm_volume_fraction_interface.f - used volume fractions to calculate gdkm inteface weights
geneq1.f - modified for new gdkm models, modified grav term weighting for zero phase density
geneq2.f - modified for new gdkm models, no gav weighting mods for isothermal
ingdpm.f - modified input for new gdkm models, major mofification for gdkm input
scanin.f - scan input for new gdkm input. Added coding to identify new humidity related keywords in the boun macro.
startup.f - added and rearranged calls to gdkm calcs
avsio.f - added parameter iogdkmblank for gdkm graphics

setparams.f - added more checks for gdpm and gdkm problems
innode.f - added more checks on input when gdkm not used
solve_dual.f - changes some allocate/deallocate statements to clean up subroutine call
csolve.f - corrected do loops for variable gdkm node number, modified to printout SIA iterations

NEG_VOL:
anonp.f - Added check and correction for negative volumes associated with inconsistent local element node ordering.
gncf3.f - Added check and correction for negative volumes associated with inconsistent local element node ordering.

-------- Code changes 10/29/2017:
varchk.f - adjustment to default saturation tolerance. For pure water, calculated T for 2-phase conditions where s is the variable. Needed for rock melting.
thermw.f - added coding for rock melting. Moved where rock internal energy is calculated.
(so T is available for 2-phase problems)

-------- Code changes 12/13/2017:
bnswer.f - small change for managing "out of bounds restart" parameter (mlz)
cntlin.f - added additional unit numbers for hi temp water data table (like co2 tables)
comai.f - added parameters for tracking number of supercritical water nodes
comsi.f - added variable arrays for tabular mechanical properties
comxi.f - increase size of arrays that identify input tabular data (support for hi temp water data)
dated.f - changed the name of the FEHM version to "FEHM V3.3.J HT 17-12-12" (just a suggestion)
enthp.f - changed to compatibility with High Temp code
psatl.f - added derivative wrt pressure for saturation temperature calculation
fehmn_pcx.f - added another use module (use property_interpolate_1) for hi temp water. Added calls to manage "saved zones".
h2o_properties_new.f - New routine to manage hi temp water properties. Based on Pawar's co2_properties.f
inpres.f - added code to allow super critical water phase state
inrlp.f90 - only a few comments but no active line changes
interpolate_2.f90 - New routine for hi temp water property interpolation. Based on John Doherty's routine for CO2 property interpolation.
iofile.f - added coding to manage opening of hi temp water interpolation table
outbnd.f - corrected multiplier for permeability units. (outputted value only)
psatl.f - added coding to include hi temp water table interpolation
rlperm.f - added coding for supercritical water phase (1 line)
scanin.f - initialized iwater_table = 0, determine if using hi temp table
sther,f - set larger T and P bounds if using water interpolation table
stress_mech_props.f - added coding for tabular input of mechanical properties
stressctr.f - added coding for tabular mechanical properties
thermw.f - added coding for hi temp tabular water properties and temperature dependent
(including rock melting) rock/soil properties
thrmwc.f - added coding for hi temp tabular water properties and temperature dependent

(including rock melting) rock/soil properties
varchk.f - checked for supercritical water phase and phase change
writeio.f – added: use comco2, only: icarb
wrtout.f - added output for supercritical water phase information

Notes:
1. The change to rlperm.f was simple: if (ieosd.eq.4) ieosd =1. Not sure, but this may be needed in routines that are related to the keyword 'rlpm'
2. Parameter "iwater_table" refers to hi temp tabular water properties.

--------- Code changes 02/10/2018:
add_gdpm.f- modified geomtriv gpkm terms to be closer to dpdp
geneq1.f - Modified connection between primary and secondary nodes to solve the heat conduction GDKM problem
geneq2.f - general cleanup of primary and secondary connection for consistency
geneqc.f - general cleanup of primary and secondary connection for consistency
write_avs_node_s.f - Fixed overflow of gdkm output for infiltration_gdkm verification problem - now passes the tests

-------- Code changes 03/19/2018:
zone.f - changed definition of zone keyword 'all' from setting every node (including secondary porosity nodes) to the designated zone to just the primary nodes. The secondary node are set '+ 100' or '+ 1000' if primary zone number > 100
infiles.f - modified so the zone file identified in the control file correctly sets the secondary porosity nodes.
ingdpm.f - slight modification to insure old style gdkm input always matches the 'dpdp' non directional fracture type
add_gdpm.f - corrected error to insure the correct area term is used in the non-directional fracture model (gdkm_dir = 0) when 'gdkm new' is used

-------- Code changes 06/20/2018:
geneq2_uz_wt.f - added GDKM coding
write_avs_node_s.f - added blanking for more variables when GDKM used
add_gdpm.f - tweaked some geometric factors for directional fracture models
geneq1.f - changed permeability weighting for primary-secondary weighting for GDKM nodes
geneqc.f - changed permeability weighting for primary-secondary weighting for GDKM nodes
coneq1.f - small GDKM changes
setparams.f - initialized some GDKM parameters
geneq2.f - changed permeability weighting for primary-secondary weighting for GDKM nodes
ingdpm.f - changed some input for new gdkm models
gdkm_volume_fraction_interface.f - corrected some minor errors in GDKM interface weighting.

-------- Code changes 09/09/18:
fehmn_pcx.f - moved the following from startup

```
if (ianpe.eq.0) then
  if (irun.eq.1.and.inobr.eq.0) call sx_combine(1)
```

```
      else
        if (irun.eq.1.and.inobr.eq.0) call sx_combine_ani(1)
      endif
```

so that beaking connections between sources or sinks is done consist with boun macro and the flow macro
startup.f - moved the same lines above from startup to fehmn_pcx
model_setup.f - added boun keyword 'fen' for flowing enthalpy. Definition is analogous to flowing temperature
scanin.f - added search for boun keyword 'fen'
zone.f - corrected some coding when creating 'saved' zones; not in common usage yet

---------- Code changes 09/25/18:
thrmwc.f - Added coding so that if, on a gridblock with a specified total pressure, the air partial pressure becomes greater than the specified total pressure is released.
Significant change for ngas problems.
varchk.f - phase_mult_hma : changed from 1.15 to 1.15
comai.f - added a few global variables
rlperm.f - added coding so that macro rlp can be called multiple times
scanin.f - added coding to support multiple rlp calls.
initdata2.f - added coding (padding) to support multiple rlp calls
data.f - add and initialize some variables associated with multiple rlp calls
setparams.f - initialized counter for nultiple rlp calls

-------- Code changes 10/26/18:
zone.f - corrected mismatched header lines in calls to zone_elem (a subroutine in zone.f); separated arrays in deallocate statement to check in allocated; changed zonesavename to character*30
input.f - corrected header line in call to zone_elem; changed zonesavename to character*30
initdata2.f - changed zonesavename to character*30
write_avs_node_con.f - changed zonesavename to character*30
write_avs_node_s.f - changed zonesavename to character*30
write_avs_node_v.f - changed zonesavename to character*30
saltctr.f - changed variable in initdata2 call to include macroread(23) in header; added line 'macroread(23) = .false.' before call to initdata2
comai_1.f - only added comment line to identify macroread(23) as associated with perm_olivella

-------- Code changes 11/01/2018 – 11/27/2018:
thrmwc_2.f - added coding to support new uses of boun keywords and improve numerical performance in the "source term code" section
flow_boun.f - added and used array qaxf(i) to store boun air fraction data
flow_boundary_conditions.f - Implemented array qafx() for applying air fraction boundary conditions. (boun keyword xfa)
model_setup.f - modified array usage for 'fxa' boun keyword
comdi.f - added an allocatable array sk_temp used in thrmwc.f; added an allocatable array qaxf used in flow_boun.f.
comai_1.f - added comment for macroread(24) for macro den
saltctr.f - added macroread(23) = .true. to indicate perm_olive read

hydiss.f - changed macroread(7) to macroread(25)
den_vis_spatial.f - changed macroread(12) to macroread(24)

--------- Code changes 11/30/2018:
flow_boundary_conditions.f - fixed logic that allowed access to an unallocated array when fixed temperatures and fixed enthalpy flows occur in the same simulation.

--------- Code changes 12/10/2018:
thrmwc.f- continued improvement of ngas physics (AWH) boundary conditions
debugged and modified high temperature table
varchk.f - improved phase change code for ngas physics with high temperature table

--------- Code changes 12/30/2018:
thrmwc_5.f - continued improvement of ngas physics (AWH) boundary conditions- fixed some more problems with the supercritical water phase
varchk_2.f - fixed a bug in the varible update portion of routine for AWH and SC phase
resetv.f - fixed a bug, now allows a timestep restart when a SC phase is present.
air_cp.f - attached a subroutine air_sol to air_cp to manage solubility models of air in liquid water. For now limited to the original Henry's law.

--------- Code changes 01/12/2019 (merge with Peter Johnson's development):
allocmem.f - OK to merge, extra variables only
cappr.f - OK to merge, extra code for main PJ stuff fixed problem with ifblock for cap model 1
comdi.f- merged newer code with PJ's now identical in src after 102518
rlperm_1a.f - added PJ code (extra relperm models)
saltctr.f - OK to merge, minor extra PJ code
vcon.f - OK to merge, minor extra PJ code
air_cp.f - added maximum liquid air mass fraction (xnl = 0.1)
varchk_2a.f - small change to phase change criteria

--------- Code changes 01/12/2019:
air_cp.f - corrected henry's law constant
vcon.f - corrected model 4 (PJ merge)
thrmwc_6.f - more tweaks of derivatives for ngas physics

--------- Code changes 01/28/2019:
model_setup.f - changed coding to distinguish between dsa and fxa keywords when multiple boun models exist with sa and fxa respectively
flow_boun.f - added more models that allow the use of ti_linear and corrected an error with distributed source air (dsa). keywords allowed with ti_linear: fd, sw, dsw, dsco2, sa, dsa, se, dse, t

--------- Code changes 02/17/2019:
flow_boun.f - added more models that allow the use of ti_linear and corrected an error with distributed source air (dsa). keywords allowed with ti_linear: fd, sw, dsw, dsco2, sa, dsa, se, dse, t, ft, hd, hdo, s

--------- Code changes 03/01/2019 – 05/22/2019:
startup_1.f-no substantive changes (a debug line)
write_avs_node_s.f - added threshhold for saturation
thrair_1.f - modified seepage face boundary condition (flo3 input) for use with Richards equation solution
steady.f - modified tolerance changing algorithm when using flowrate to determine steady state
comsteady.f - added global variables to support changes made to steady.f
user_ymp1.f - added some operations for manipulating zones. (these do not affect any tests)
fehmn_pcx_1.f - comment lines only
comai_1.f - added some variables
incoord.f - small change to indicate bad grid
wrtout_1.f - added output (additional columns) for zone numbers and phase state for water. Improved accuracy of phase change count.
flxz_2.f - added output (additional column) for average zone saturation in output for water
gradctr_1.f - corrected error in visualization when grad macro is used. Does not affect simulation but only zones in some contour output files when grad macro is used

--------- Code changes 05/22/2019 – 06/11/2019:
user_ymp.f - Changed the general 'a' format, allowed in Intel Fortran, to a format that specifies the number of characters (ie a8) so that the routine compiles in linux.

--------- Code changes 06/13/2019:
residual.f - line 187, anorm_tol change to 1.d-90
sub_gmres1.f - line 370 change tols = 1.d-30, line 427 change 1.d-20 to 1.d-60

--------- Code changes 06/27/2019:
wrtout_1.f - added output (additional columns) for zone numbers and phase state for water. Improved accuracy of phase change count. Added output for zone and phase state information for printout nodes
flxz_2.f - added output (additional column) for average zone saturation in output for water. Corrected memory error when idof = 13 (s always=1)
setconnarray_1.f - added code that writes a file (red_fac_itfc.txt) for visualization of the interface nodes.

--------- Code changes 07/08/2019:
combi.f - added global variables for the number of interface connections affected by itfc macro.
scanin.f - saved the actual number of classes (lines) of interface connections values entered on input for itfc macro.
setconnarray.f - fixed error in colloid part of itfc. The code was previously going through the colloid loop even if there was no colloid input. Added vtk output along with text file for visualizing the interface positions for the reduction factors.

---------- Code changes 08/06/2019:
flxz_2.f - added output (additional column) for average zone saturation in output for water. Corrected memory error when idof = 13 (s always=1)
airctr_2.f - modified phase change criteria 2-phase to gas

combi_1.f - added variables associated with itfc macro (visualization)
gradctr_1.f - correction for case of reading gradient info from aux file.
(As happens when gradctr used multiple times.)
pest_1.f - added for information for flowrate columns
scanin_1.f - added logical variables associated with itfc macro
setconnarray_3.f - added option to find all connections to a particular zone for setting interface multipliers
write_avs_node_con_2.f - significant changes for transport species to get heading correct for tec files. (This could cause some problems.)
wrtout_1.f - write out additional columns with zone number and phase state (minor tweaks over last version)
reset_1.f - reset saturation = 0 ti ieos=2 for Richard's Eq only (could be significant)
steady_1.f - minor improvements to steady macro. Forced variable tolerance to be monotonic.
comsteady_1.f - added variable which saves last timestep tolerance
comdi_1.f - added last timestep source array
thrair_2.f - added capability to manage maximum source changes for constant pressure nodes and seepage faces (currently disabled).

---------- **Code changes 08/28/2019:**
vcon.f - bug fix for pjjohnson's implementation of thermal conductivity for porosity above 0.3873 to FEHM. Included temperature dependence calculation and accurate linear slope equation for porosity above 0.3873 situation.

---------- **Code changes 09/03/2019:**
airctr.f - only comments
comai.f - changed sat_ich from integer to real*8 (sat value below which head is set to zero for output)
comdi.f - added allocatable arrays for renubering algorithms
comsteady.f - added parameters related for improved algorithm for changing eq tolerance
contr.f - corrected errors when using macro 'chea' for output in head for non-head problems
convctr.f - fixed conflict with with crl(1,1) usage for density in isothermal problems and output using subroutine water_density.
diagnostics.f - added zone id to timestep restart information
fehmn_pcx.f - eliminated negative porosity node (eliminated nodes) from phase change count
headctr.f - implimented a saturation id for head output when "chea" macro used
input.f - corrected "chea" default saturation for zero head designation
pest.f - added output for zone id, average sat of a zone, and velocity
renum.f - added infrastructure for some renumbering algorithms
reset.f - if richard's eq enabled, only allow reset to liquid or two-phase condition (gas phase included in two phase)
startup.f - added some comments
steady.f - improved the algorithm for reducing eq tolerance as flow balance converges. Added input parameters:
"tstr" : specify time(days) to start new tolerance reduction algorithm
"sacc" : set tolerance for nodal mass change
thrair.f - added some comments

End Developer Notes 09/22/2017 to 09/12/2019 ----------

3 DISCUSSION AND EXAMPLES

This section summarizes and adds further discussion for topics not covered in the User Manual V3.1 documentation, but will be added to the GitHub version of the Manual.

3.1 BOUNDARY CONDITIONS FOR THE AIR-WATER-HEAT MODULE IN FEHM AND A DESCRIPTION OF VERIFICATION PROBLEMS

The ability to impose a variety of boundary conditions is crucial in Air-Water-Heat (AWH) module in FEHM because of the variety of applications ranging from the simulation of nuclear waste disposal in salt formations and related laboratory experiment, the simulation of near surface geothermal systems such as geysers where the mixture air and water vapor is significant, and the burial of nuclear waste in the unsaturated zone (Yucca mountain).

The numerical formulation of the three conservation equations (AWH) is tightly coupled. The reader is referred to the Zyvoloski (2007) for numerical details. The set of discrete conservation equations for each process requires separate boundary conditions. Boundary conditions are related to variables.

Before describing the specifics of the boundary conditions, it is useful to describe the phase states and phase transition criteria that are possible with the AWH module:
There are three phase states possible in the AWH:

1. Liquid water. The liquid water phase consists of water and dissolved air. It is assumed that the density of water depends only on temperature and pressure but not on dissolved air. The active variables used in FEHM for the liquid phase are total pressure, air partial pressure, and temperature. The dissolved air mass fraction, used in numerical formulation, is easily calculated from the active variable
2. Two-phase. Both a gas and liquid phase exist. The liquid phase is as described above. The gas phase consists of air and water vapor. Properties (density, enthalpy, viscosity) are calculated as a mixture. The partial water pressure is a (known) function of temperature and may be influenced by capillary pressure and chemical species (salt). Humidity, the ratio of the partial water vapor pressure to the maximum water vapor pressure, is assumed to be 1.0. The active variables used in FEHM for two-phase conditions are total pressure, liquid saturation, and temperature.
3. Gas phase. A single fluid phase comprised of air and water vapor. Properties are calculated as a mixture. Humidity, the ratio of the partial water vapor pressure to the maximum water vapor pressure, is calculated as the ratio of water partial pressure to the maximum (saturation) water partial pressure at the current gridblock temperature. The active variables used in FEHM for gas phase conditions are total pressure, air partial pressure, and temperature.

Boundary conditions for the AWH coupled processes equations range from obvious to subtle. A specified pressure condition for the active variable pressure is obvious. A fixed energy flow does not, at least to me, seem enough to be a stable boundary condition and produce a stable solution, when used alone, for a set of equations that results in a temperature field. After describing boundary conditions related to the conservation equations, example simulations are presented. To facilitate use in FEHM, the example simulations will use the *boun* macro and the keywords available with that macro. Active and derived variable have the following acronyms:

1. P - total pressure (MPa)
2. T - temperature (C)
3. S - saturation
4. H - humidity
5. P_a - air partial pressure (MPa)
6. Q_w - flow rate water (kg/sec)
7. Q_a - flow rate air (kg/sec)
8. Q_e - enthalpy rate air (Mw)
9. X_a - air mass fraction

Boundary conditions related to the water balance equation:

1. P (specified)
2. H (specified) H will generate a set of Q_w and Q_a that will hold the gridblock at H.
3. H_f (specified) H_f divide an inflowing air flowrate into air and water flowrates will that will produce an inflow with H_f.
4. S(specified)
5. Q_w (specified)
6. FX_a(specified) FX_a will divide an inflowing water flowrate into air and water flowrates.

Boundary conditions related to the air balance equation:

1. P_a (specified)
2. H (specified) H will generate a set of Q_w and Q_a that will hold the gridblock at H.
3. H_f (specified) H_f divide an inflowing air flowrate into air and water flowrates will that will produce an inflow with H_f.
4. Q_a (specified)
5. FX_a(specified) FX_a will divide an inflowing water flowrate into air and water flowrates.

Boundary conditions related to the energy balance equation:

1. T (specified) T will generate an energy flowrate sufficient to hold the gridblock to T
2. T_f (specified) T_f will change the enthalpy of the incoming water flowrate so that the temperature of the water flowrate is T. Given the mixture of air and water flowrates in AWH simulations, T_f is not recommended unless only (pure) water inflow is occurring at a gridblock.
3. Q_e (specified)

It is worth remarking on inflow and outflow conditions. First, because of pressure changes that can occur in a simulation, notably heating or cooling, the gridblocks with specified pressures can change during a simulation from inflow to outflow and vice versa. Second, outflow gridblocks have a more limited set of conditions that can be applied to them than inflow gridblocks. This is because most outflow properties are dependent on the most recent fluid and rock properties such as temperature and relative permeability within the gridblock. In contrast, the inflow streams occur from outside source where the properties are known. The boundary conditions T, H, P, and P_a are responsible for the direction of mass and energy flows.
The following combinations have been tested in FEHM and run well. They are presented in *boun* format. Results will be presented later.

1. Inflow (pw,t,fxa) = (0.1001,20.,0.99). Note that pw = P in *boun* terminology. The *boun* keyword 'if' is the impedance factor.

```
model 1
ti
2
0. 1.e20
pw
0.1001 0.1001
t
20. 20.
if
1.e-2 1.e-2
fxa
0.99 0.99
```

2. Inflow (pw,t,sa) = (0.1001,20.,-8.e-3). Note that pw = P in *boun* terminology. The *boun* keyword 'if' is the impedance factor. Instead of splitting the water flow, an air source is supplied. Note sa = Q_a .

```
model 2
ti
2
0. 1.e20
pw
0.1001 0.1001
t
20. 20.
if
1.e-2 1.e-2
sa
-8.e-3 -8.e-3
```

3. Inflow (sw,T,sa) = (-1.e-7, 20.,-1.e-2). Note that pw = P in *boun* terminology. The *boun* keyword 'if' is the impedance factor. Replaced P with sw =Q_w

```
model 3
ti
2
0. 1.e20
```

```
t
20. 20.
if
1.e-2 1.e-2
sw
-1.e-7 -1.e-7
sa
-1.e-2 -1.e-2
```

4. Inflow (pw,T,pa) = (0.1001, 20.,0.1001). Note that pw = P in *boun* terminology. The *boun* keyword ‘if’ is the impedance factor. Have both P and P_a; since P (=pw) is the total pressure and P_a = pa is the partial air pressure, this simulation produces an almost dry (humidity = 0) inflow from the prescribed pressures.

```
model 4
ti
2
0. 1.e20
t
20. 20.
if
1.e-2 1.e-2
pw
0.1001 0.1001
pa
0.1001 0.1001
```

5. Inflow (pw,T,fxa) = (0.1001, 80.,0.99999). Note that pw = P in *boun* terminology. The *boun* keyword ‘if’ is the impedance factor. Similar to Model 1 except raised the T and made the stream drier.

```
model 5
ti
2
0. 1.e20
t
80. 80.
if
1.e-2 1.e-2
pw
0.1001 0.1001
fxa
0.99999 0.99999
```

6. Inflow (sw,T,fxa) = (-1.e-3, 80.,0.98). Similar to Model 5 except replaced pw with sw(= Q_w).

```
model 6
ti
2
0. 100.
t
```

80. 80.
if
1.e-2 1.e-2
sw
-1.e-3 -1.e-3
fxa
0.98 0.98

7. Inflow (sa,T,huf) = (-1.e-3, 20.,0.02). Similar to Model 6 except replaced sw with sa (= Q_a) and set the inflow humidity (H_f = huf) to 0.02. This will split the air flow into air and water flows give the specified gas flow the H_f.

model 7
ti
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
sa
-1.e-3 -1.e-3
huf
0.02 .02

8. Inflow (pa,T,huf) = (0.1003, 20.,0.02). Similar to Model 7 except replaced sa with pa.

model 8
ti
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
pa
0.1003 0.1003
huf
0.02 .02

9. Inflow (pa,T,huf) = (0.1003, 20.,0.02). Similar to Model 8 except changed the conditions (ph,th) with which the inflowing gas humidity is calculated.

model 9
ti
2
0. 100.
t
80. 80.
if
1.e-2 1.e-2
pa

0.1003 0.1003
huf
0.02 0.02
ph
0.1 0.1
th
80. 80.

10. Inflow (pa,T,hu) = (0.1003, 20.,0.90). Similar to Model 9 except changed to (hu = H) which generates air and water flow by fixing the humidity of the gridblock.

model 10
ti
2
0. 100.
t
80. 80.
if
1.e-2 1.e-2
hu
0.90 0.90
ph
0.1 0.1
th
80. 80.

11. Inflow (sw,T,sa) = (-1.e-14, 20.,-8.e-3). Similar to Model 2 except changed sw)to a very small value as a stability check

model 11
ti
2
0. 1.e20
sw
-1.e-14 -1.e-14
t
20. 20.
if
1.e-2 1.e-2
sa
-8.e-3 -8.e-3

12. Inflow (s,T,a) = (0.6, 20.,0.097). Changed pw to s (=S) to check flow induced by specified saturation. Also tests air flow when S is used. The boundary conditions s and pw cannot be used in the same model.

model 12
ti
2
0. 1.e20
t
20. 20.

if
1.e-2 1.e-2
s
0.6 0.6
pa
0.097 0.097

13. Outflow (pw,T,sa) = (0.1, 20.,NA). Outflow model where total pressure (pw) and temperature (T) are prescribed. Outflow condition depends on value of pw relatively lower than other pressures in the model. Works as inflow if pw is higher than the pressure in the model

model 13
ti
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
pw
0.1 0.1

Development of test problems. The 12 inflow boundary conditions and the common outflow boundary condition (model 13) will be tested on a one dimensional column with uniform rock properties. As this input file has all 13 models listed in the boun macro, the different boundary condition sets are obtained by selecting the appropriate model number in the input file. For the test problem, zone 5 is the inflow node and zone 6 is the outflow node. Thus test 2 is obtained by the following lines at the end of the boun macro:

```
-5 0 0 2
-6 0 0 13
```

All the tests will use model 13 as the outflow at zone 6. Below is the complete boun macro for test 2.

```
#----------Boun----------
# gaz 112018
# models 1-12 are largely inflow (applied at node 20)
# model 13 is a fixed atmospheric pressure (applied at node 1)
boun
model 1
ti
2
0. 1.e20
pw
0.1001 0.1001
```

t
20. 20.
if
1.e-2 1.e-2
fxa
0.99 0.99
model 2
ti
2
0. 1.e20
pw
0.1001 0.1001
t
20. 20.
if
1.e-2 1.e-2
sa
-8.e-3 -8.e-3
model 3
ti
2
0. 1.e20
t
20. 20.
if
1.e-2 1.e-2
sw
-1.e-7 -1.e-7
sa
-1.e-2 -1.e-2
model 4
ti
2
0. 1.e20
t
20. 20.
if
1.e-2 1.e-2
pw
0.1001 0.1001

pa
0.1001 0.1001
model 5
ti
2
0. 1.e20
t
80. 80.
if
1.e-2 1.e-2
pw
0.1001 0.1001
fxa
0.99999 0.99999
model 6
ti
2
0. 100.
t
80. 80.
if
1.e-2 1.e-2
sw
-1.e-3 -1.e-3
fxa
0.98 0.98
model 7
ti
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
sa
-1.e-3 -1.e-3
huf
0.02 0.02
model 8
ti

```
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
pa
0.1003 0.1003
huf
0.02 0.02
model 9
ti
2
0. 100.
t
80. 80.
if
1.e-2 1.e-2
pa
0.1003 0.1003
huf
0.02 0.02
ph
0.1 0.1
th
80. 80.
model 10
ti
2
0. 100.
t
80. 80.
if
1.e-2 1.e-2
hu
0.90 0.90
ph
0.1 0.1
th
80. 80.
```

```
model 11    # like model 2 except pw changed to sw
ti
2
0. 1.e20
sw
-1.e-14 -1.e-14
t
20. 20.
if
1.e-2 1.e-2
sa
-8.e-3 -8.e-3
model 12    # will initially produce inflow
ti
2
0. 1.e20
t
20. 20.
if
1.e-0 1.e-0
s
0.6 0.6
pa                  # (also test pa 0.08 0.08)
0.09 0.09           # (also test sa -1.e-2 -1.e-2)
model 13
ti
2
0. 100.
t
20. 20.
if
1.e-2 1.e-2
pw
0.1 0.1

-5 0 0 2
-6 0 0 13
```

To use this data file for a V and V test, I suggest comparing future runs with the data generated from my runs and on the contour scalar output (2nd instance). For example, for model 1.

rh_case_m1.00002_sca_node.dat

ZONE T ="Simulation time 0.300000000 days"

0000000001	-1.88201387	0.100000001	20.0000000	0.899306480E-02
0000000002	-1.88200888	0.100005263	19.9998500	0.899292959E-02
0000000003	-1.88200416	0.100010524	19.9997587	0.899265920E-02
0000000004	-1.88199971	0.100015785	19.9997055	0.899225366E-02
0000000005	-1.88199553	0.100021046	19.9996759	0.899171296E-02
0000000006	-1.88199162	0.100026307	19.9996601	0.899103713E-02
0000000007	-1.88198798	0.100031568	19.9996521	0.899022617E-02
0000000008	-1.88198461	0.100036828	19.9996483	0.898928008E-02
0000000009	-1.88198151	0.100042088	19.9996469	0.898819885E-02
0000000010	-1.88197869	0.100047348	19.9996469	0.898698248E-02
0000000011	-1.88197613	0.100052608	19.9996480	0.898563095E-02
0000000012	-1.88197384	0.100057867	19.9996504	0.898414424E-02
0000000013	-1.88197183	0.100063126	19.9996548	0.898252233E-02
0000000014	-1.88197008	0.100068385	19.9996625	0.898076518E-02
0000000015	-1.88196861	0.100073644	19.9996758	0.897887276E-02
0000000016	-1.88196741	0.100078902	19.9996981	0.897684503E-02
0000000017	-1.88196648	0.100084160	19.9997343	0.897468193E-02
0000000018	-1.88196582	0.100089418	19.9997911	0.897238343E-02
0000000019	-1.88196543	0.100094675	19.9998767	0.896994945E-02
0000000020	-1.88196531	0.100099933	20.0000000	0.896737993E-02

The comparison should be for each column.

3.2 DEVELOPMENT OF A HIGH TEMPERATURE TABULAR WATER PROPERTIES CAPABILITY IN FEHM

The High Temperature Water Properties Capability was developed by George Zyvoloski with funding from Olympic Research Corporation under the Department of Energy Project "Development of Thermally Formed Plugs for Deep Borehole Waste Disposal Applications". See Lowry (2017) for details. The project seeks to evaluate the storing nuclear waste in deep boreholes in a granitic environment. The conditions surrounding and in the waste emplacement area are severe. Temperatures reaching 1600 C and above are anticipated in some scenarios. The temperature range includes fluid the critical point temperature of water and the melting

temperatures granitic rock and steel. More importantly, temperature differences between the emplacement area and the surrounding rock produced very large stresses with attendant rock failure.

1. *Development of the High Temperature Property Module (Fluid properties to 1600C)*

At the heart of the high temperature fluid property module is an efficient variable resolution property table (Doherty, 2006) that was originally developed for CO2 property evaluation. High resolution data is necessary near the phase change boundaries and near the critical point of water (374 C, 22 MPa). See Figure 1. This table is populated with fluid properties from NIST Database 23.The property look-up modules incorporate water boiling curves that correspond exactly with table points and results in accurate determination of phase changes and function derivatives near phase and super critical boundaries. Transition to and from the super critical region are easily incorporated.

Verification simulation

A 1D slab 50 m long (100 gridblocks) was chosen to highlight the differences between the rational polynomial based functions to represent water properties. The input properties are shown in Table 1.

	Test a: low permeability	Test b: high permeability
Injection rate(kg/s)	4.e-4	4.e-4
Injection energy (Mj/kg)	4.0	3.5
Outlet pressure (MPa)	10.0	10.0
Outlet temperature (deg C)	100.	100.
Permeability(m^2)	2.e-16	1.e-12
porosity	0.2	0.2
Thermal conductivity (W/(m.K)	2.7	2.7
Gridblock flow area (m^2)	1.0	1.0
Gridblock thickness (m)	0.5	0.5
Heat capacity(Kj/kg.K)	1250.	1250.
Rock density (kg/m^3)	2563.	2563.

Table 1 Input parameters for supercritical water properties table test

Two test cases were run. Case “a” uses a relatively low permeability of 2.e-16 m^2 to produce pressures and temperatures that start in the liquid water phase and later enter the supercritical region. Case “b” utilizes a high permeability to keep the pressure low enough to start in the liquid region and transition to the vapor region as hot fluid enters the system. The low permeability case produced pressures at the source node of about 45 MPa. During the simulation, the bulk of the gridblocks moved from a phase state in the liquid region to the supercritical region. In the simulation, the source gridblock quickly rises to 45 MPa and is 350 C at 6 days- still in the liquid region as seen from Figure 1. Two things should be noted. The curves for rational polynomial functional and tabular data representation overlay at 6 days for both the pressure and temperature.

This is shown in Figures 2 and 3 respectively. The upper temperature limit for the rational polynomial representation is 360 C.

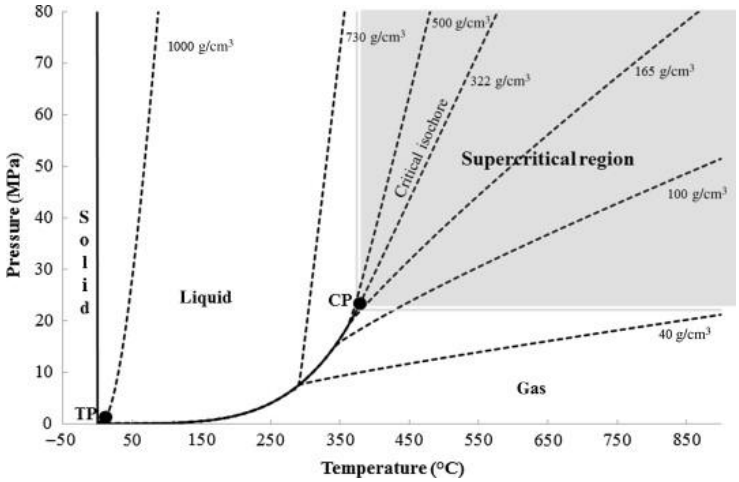

Figure 1. Water phase diagram showing the super critical region.

The tabular water properties simulations were continued to 365 days. Most of the gridblocks transitioned to the supercritical region. The pressure results show an interesting inflection near the critical point-see Figure 2. The temperature of the source node rose to 800C.

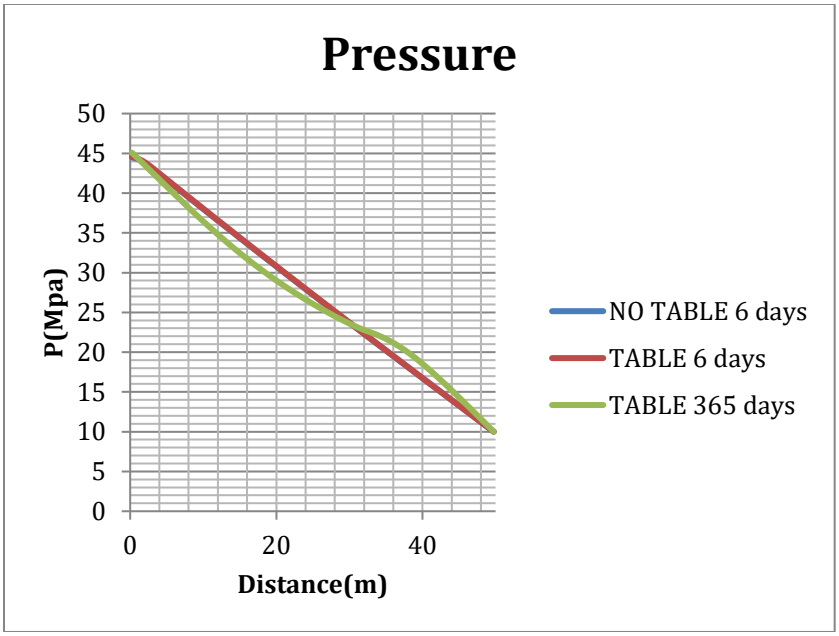

Figure 2. Pressures for case “a”, the low permeability test.

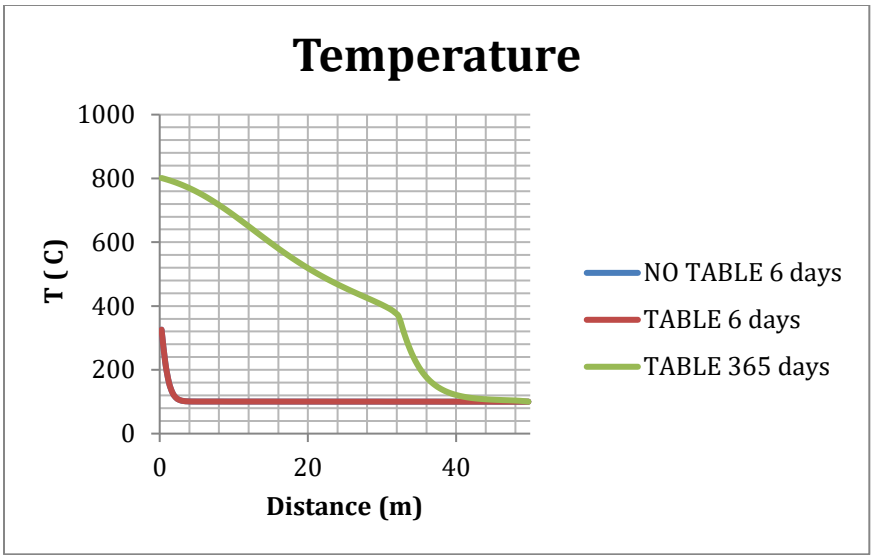

Figure 3. Temperatures for case “a”, the low permeability test.

The simulations for case “b”, the high permeability case showed temperatures that were virtually indistinguishable at 15 days (Figure 3) for the rational polynomial and tabular representations of water properties. The rational polynomial representation reached its upper limit at 15 days. Results were similar for pressure; the pressure changes (because of the high permeability) were very small and not shown. All the pressures were close to 10 MPa. At 365 days, the tabular data run reached about 540 C at the injection gridblock. Referring to Figure 1, all the gridblocks, excepting near the constant pressure outlet gridblock, for this longer run were in the gas phase region below the supercritical region. The saturations for these blocks was 0.

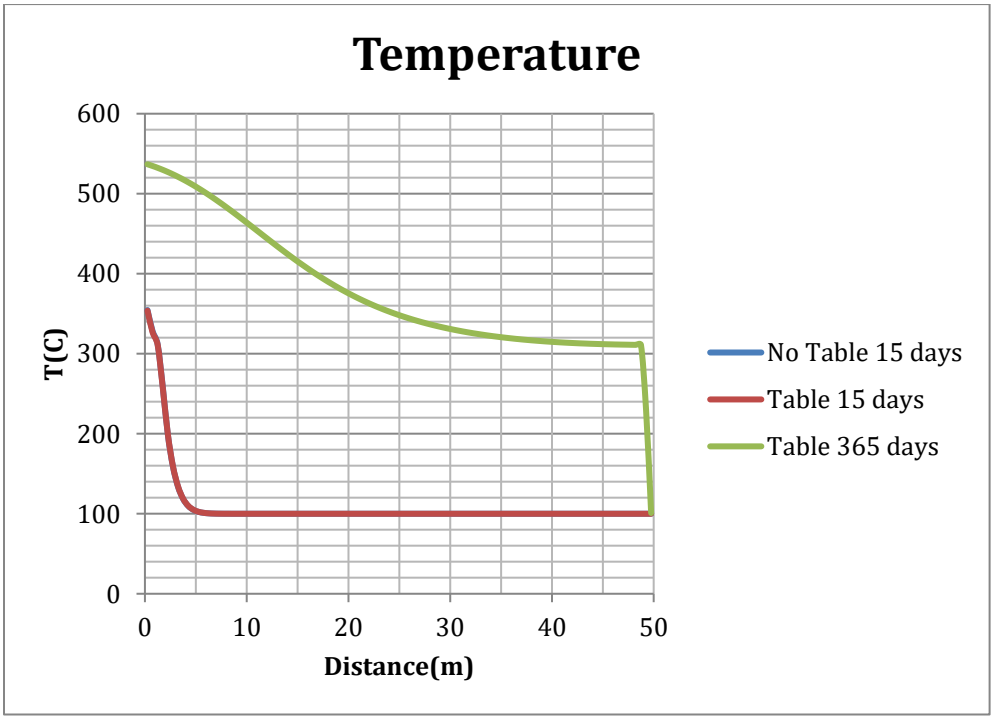

Figure 4. Temperature results for the high permeability test.

The numerical performance of the simulations with the tabular water properties in terms of stability of the very nonlinear equations, the iteration count, and CPU time were excellent. One noted difference was it was useful when using the tables that when the simulation required a time step restart, sometimes it took the timestep to a small very small value and quickly returned the timestep to an appropriately large value if a reasonable timestep multiplier was provided. The piecewise continuous nature of the table with discontinuous derivatives is likely the reason for this behavior. Thus it is recommended that a small minimum timestep be used, say 1.e-6 days, and a timestep multiplier of 1.5 to 2.

The water properties table was constructed in a manner completely analogous to the co2 properties table that has been used for a decade in FEHM (Doherty, 2006). The details of construction will be presented in a future report. The use of the water properties table is easy to implement in an existing FEHM simulation. Two changes are necessary:

1. The control file (e.g. fehmn.files) should identify the tabular data file:

   input: input/heat1d_fluid_FDM_0a.in
   outp: output/heat1d_fluid_FDM_0a.out
   hist: output/heat1d_fluid_FDM_0a.his
   root: output/heat1d_fluid_FDM_0a
   h2oin: input/h2o_props_gaz_6a.txt

   all
   0

   Where h2o_props_gaz_6a.txt is the water properties table.

2. In the input file, the tabular water table is invoked in the *eos* macro:

   #**
   eos
   table
   #**

Notes:
The two verification tests and the tabular water properties file are included in the HITEMP_VERIFICATION verification package. The folders include results with and without tabular data. Each of these folders have results for low (case a) and high permeability (case b). Also included is “doe” verification problem that has been modified to use tabular water properties. It can be compared with the “standard” doe problem in the test suite.

References:

1. Zyvoloski G., 2007. FEHM: A control volume finite element code for simulating subsurface multi-phase multi-fluid head and mass transfer, Los Alamos National Laboratory Report LA-UR-07-3359, Los Alamos, New Mexico, USA.
2. FEHM website: https://fehm.lanl.gov/
3. Doherty J., 2006. Fast Lookup of CO2 Properties, LANL informal report.
4. Lowry, B., 2017, *Development of Thermally Formed Plugs for Deep Borehole Waste Disposal Applications*, DOE SBIR award no. DE-SC0010135, Office of Science, DOE program manager: Mark Tynan.

Government report disclaimer