Title
Enhancements to the TOUGH2 Simulator as Implemented in iTOUGH2.

Author
Finsterle, S.

Publication Date
2015-03-20

Enhancements to the TOUGH2 Simulator as Implemented in iTOUGH2

Stefan Finsterle

Earth Sciences Division
Lawrence Berkeley National Laboratory
University of California
Berkeley, CA 94720

March 2015

This work was supported by the U.S. Dept. of Energy under Contract No. DE-AC02-05CH11231.

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PAGE INTENTIONALLY LEFT BLANK

1 Introduction

iTOUGH2 is a program for parameter estimation, sensitivity analysis, and uncertainty propagation analysis. It is based on the TOUGH2 simulator for non-isothermal multiphase, multicomponent flow and transport in fractured and porous media [*Pruess*, 1987, 1991, 2005, 2011; *Falta et al.*, 1995; *Pruess et al.,* 1999, 2002, 2012; *Doughty*, 2013].

The core of iTOUGH2 contains slightly modified versions of TOUGH2 modules. Most code modifications are editorial and do not affect the simulation results. As a result, standard TOUGH2 input files can be used in iTOUGH2, and identical results are obtained if iTOUGH2 is run in forward mode. However, a number of modifications have been made as described in this report. They enhance the functionality, flexibilitu, and eas-of-use of the forward simulator.

This report complements the reports *iTOUGH2 User's Guide*, *iTOUGH2 Command Referecne*, and the collection of tutorial examples in *iTOUGH2 Sample Problems*.

The key to a successful application of iTOUGH2 is (i) a good understanding of multiphase flow processes, (ii) the ability to conceptualize the given flow and transport problem and to develop a corresponding TOUGH2 model, (iii) detailed knowledge about the data used for calibration, (iv) an understanding of parameter estimation theory and the correct interpretation of inverse modeling results, (v) proficiency in using iTOUGH2 options. This report addresses issue (v) only.

2 More Options (MOMOP)

Table 1 describes additional options invoked by integer flags on a line following keyword MOMOP.

Table 1. More Options

MOP2	**Value**	**Description**
1		Minimum number of Newton-Raphson iterations
	0, 1	Allow convergence in a single Newton-Raphson interation
	2	Perform at least two iterations; primary variables are always updated.
	3	Allow convergence in a single Newton-Raphson iteration for negative simulation times, but require at least two for positive times; useful for steady-state followed by transient siulations.
	4	Allow convergence in a single Newton-Raphson iteration for positive simulation times, but require at least two for negative times.
2	5–9	Length of element names (default: 5 characters) Format of blocks ELEME, CONNE, INCON, and GENER change depending on Element-name length as follows:
		ELEME
	5	(A3,I2,I5,I5,A2,A3,6E10.4)
	6	(A3,I3,I5,I4,A2,A3,6E10.4)
	7	(A3,I4,I4,I4,A2,A3,6E10.4)
	8	(A3,I5,I4,I3,A2,A3,6E10.4)
	9	(A3,I6,I3,I3,A2,A3,6E10.4)
		CONNE
	5	(2(A3,I2),I5,2I5,I5,4E10.4)
	6	(2(A3,I3),I5,2I4,I5,4E10.4)
	7	(2(A3,I4),I5,2I3,I5,4E10.4)
	8	(2(A3,I5),I3,2I3,I5,4E10.4)
	9	(2(A3,I6),I3,2I2,I5,4E10.4)
		INCON
	5	(A3,I2,I5,I5,E15.8,4E12.4)
	6	(A3,I3,I5,I4,E15.8,4E12.4)
	7	(A3,I4,I4,I4,E15.8,4E12.4)
	8	(A3,I5,I4,I3,E15.8,4E12.4)
	9	(A3,I6,I3,I3,E15.8,4E12.4)
		GENER
	5	(A3,I2,A3,I2,I5,2I5,I5,5X,A4,A1,3E10.4)
	6	(A3,I3,A3,I2,I6,2I4,I5,5X,A4,A1,3E10.4)
	7	(A3,I4,A3,I2,I5,2I4,I5,5X,A4,A1,3E10.4)
	8	(A3,I5,A3,I2,I4,2I4,I5,5X,A4,A1,3E10.4)
	9	(A3,I6,A3,I2,I5,2I3,I5,5X,A4,A1,3E10.4)

3		Honoring generation times
	0	Generation times ignored
	>0	Time steps adjusted to match generation times
4		Vapor pressure reduction
	0	No vapor pressure reduction at low liquid saturation
	>0	Reduces vapor pressure for $S_l < 0.02$ to prevent liquid disappearance by evaporation (only certain EOS modules)
5		Active Fracture Model (see Sections A7 and A8)
	0	Active Fracture Model applied to liquid phase only
	>0	Active Fracture Model applied to all phases
6		Leverett scaling of capillary pressure
	0	No Leverett scaling
	>0	Rescale capillary pressure: $P_c = P_{c,ref}\sqrt{k_{ref}/k}$ if element-specific permeabilities are specified (see Section A4)
7		Zero nodal distance
	0	Take absolute permeability from other element
	>0	Take absolute and relative epermeability from other element
8		Version of sink/source subroutine
	0	Take TOUGH2 version of subroutine QU
	>0	Take TOUGH2V2 version of subroutine QU
9		Time stepping after time-step reduction to honor printout time
	0	Continue with time step used before forced time-step reduction
	>0	Continue with time step imposed by forced time-step reduction
10		Writing SAVE file
	0	Write SAVE file only at the end of a forward run
	>0	Write SAVE file after each printout time
11		Water properties
	0	International Formulation Committee (1967)
	1	IAPWS-IF97
	2	EOS1sc only: IAPWS-IF97 for T<800°C IAPWS-95 for T≥800°C
12		Enthalpy of liquid water
	0	Potential energy not included in enthalpy of liquid water
	>0	Potential energy included in enthalpy of liquid water
13		Adjustment of Newton-Raphson increment weighting
	0	No adjustment
	>0	Reduce `WNR` by *`MOP2`*`(13)` percent if Newton-Raphson iterations oscillate and time step is reduced because *`ITER`*=*`NOITE`*
14		Air-entry pressure in Brooks-Corey capillary pressure curve (`ICP`=10)
	0	Spherical model for interpolation between $P_c = P_e$ at $S_l = 0.99$ and $P_c = 0.0$ at $S_l = 1.0$
	1	Step change in P_c from P_e to 0.0 at $S_l = 0.9999$
	2	Linear interpolation between $P_c = P_e$ at $S_l = 0.99$ and $P_c = 0.0$ at $S_l = 1.0$
	3	Spherical interpolation between $P_c = P_e$ at $S_l = 0.99$ and $P_c = 0.0$ at $S_l =$

		1.0
15		Porosity used for calculation of rock energy content
	0	Use porosity of block ROCKS; this assumes that the porosities provided in block INCON were the result of a pore compressibility/expansivity calculation; the "original" porosity from block ROCKS is used to compensate for equivalent rock-grain density changes.
	>0	Use porosity from block INCON; this assumes that these porosities were not the result from a pore compressibility/expansivity calculation; changes inrock-grain density due to pore compressibility/expansivity are not compensated.
16		Porosity-permeability relationships for heterogeneous media
	0	No deterministic correlation
	1	Material-specific empirical correlations (see subroutine `PER2POR`)
17		Invokes wellbore simulator FloWell
	0	Do not run FloWell simulator
	1	Run FloWell for each calibration and printout time
	2	Run FloWell for each time step
18		Porosity update by ROCMECH
	0	Do not update porosity
	1	Apply porosity correction
19		Treatment of residuals
	0	Keep all residuals
	>0	Do not update primary variable if scaled residual is smaller that $10^{(-MOP2(19)-10)}$
20		Reading anisotropic permeability modifiers in block `ELEME`
	0	Read isotropic permeability modifiers from columns 41–50
	1	Read anisotropic permeability modifiers from columns 81–110
	2	Read anisotropic permeability modifiers for `ISOT`=1 from columns 41–50 and for `ISOT`=2 and 3 from columns 91–110

3 Printout Options

3.1 Printout-Control through KDATA

In standard TOUGH2, printout of simulation results is controlled by variables *KDATA* and *MCYPR* in block PARAM.1, as well as by the additional printout times *TIS* given in block TIMES. In iTOUGH2, the calibration times are also stored in array *TIS* which means that at each calibration time the amount of printout specified by variable KDATA is written to the output file. This may make the TOUGH2 output file extremely long, and requires unnecessary CPU time for disk writing, since the TOUGH2 output file is overwritten each time a new TOUGH2 simulation is initiated by iTOUGH2.

If a negative number is specified for variable *KDATA*, the amount of printout is reduced (see Table 2), saving both disk space and CPU time. Recall that printout at the calibration points is always written to the iTOUGH2 plot file (see command >>> FORMAT). Also, if a TOUGH2 run is terminated due to a convergence failure or using command kit, the full output is automatically generated for the last time step.

Full output is always provided for the times specified in the TOUGH2 block TIMES; the amount of printout is given by the absolute value of *KDATA*. The times provided in TOUGH2 block TIMES can be in arbitrary order; they are sorted internally.

Table 2. Amount of Printout as a Function of Variable *KDATA*

-3	-2	-1	Printout of ...	1	2	3
-	+	+	volume and mass balance	+	+	+
-	-	+	generation rates	+	+	+
-	-	-	most important variables	+	+	+
-	-	-	fluxes and velocities	-	+	+
-	-	-	primary variables	-	-	+

3.2 Excluding Domains from Global Material Balance Printout

Domains with an element volume larger than 10^{20} m^3 or a negative rock grain specific heat *SPHT* or with *SPHT* greater than 10^4 J/kg °C are excluded from global material balance calculations. The absolute value of *SPHT* is used in the heat balance equation. Material balances for individual rock types are available through iTOUGH2 commands >> TOTAL MASS and >>> VOLUME.

4 Secondary Mesh (ELEM2, CONN2)

This enhancement was introduced especially for the estimation of a skin radius (see command >> SKIN) or a MINC parameter (see command >> MINC), both requiring that a new TOUGH2 mesh is generated automatically each iteration without user interference. In order to achieve this, a primary mesh must be generated using the MESHMAKER utility. Elements and connections of this primary mesh can then be overwritten by a secondary mesh provided through blocks ELEM2 and CONN2, which have the same format as blocks ELEME and CONNE, respectively. The names of the elements and connections to be modified by the secondary mesh must be identical with the corresponding ones of the primary mesh. If a secondary mesh is specified, both blocks, ELEM2 and CONN2, must be given; either of the two keywords may be followed by an empty line to indicate that no modification is made.

Figure 1 illustrates an application. A radial mesh is generated using MESHMAKER for simulating a pump test. In block ELEM2, the volume of the first grid block is changed to represent the actual interval volume. In block CONN2, the nodal distance from the first element to the interface is reduced to a very small number as usually done for connections to boundary elements. These last two modifications, which are usually made by editing the mesh file, are now automatically performed whenever a new mesh is generated during an iTOUGH2 run for the estimation of the skin radius.

```
MESHMAKER-----*----2----*----3----*----4----*----5----*----6
RZ2D
RADII
    2    1
 0.000E+00 0.100E+00
LOGAR
   20    2 0.300E+00 0.100E-01
LOGAR
   80    3 1.000E+01
LAYER
    1
 0.100E+01

ELEM2----1----*----2----*----3----*----4----*----5----*----6
A1  1              1 .1250E+00

CONN2----1----*----2----*----3----*----4----*----5----*----6
A1  1A1  2                   1 .1000E-10 .5332E-02 .6283E+00
```

Figure 1. Primary and secondary mesh generation.

5 Permeability Assignments

5.1 Element-by-Element Permeabilities

Heterogeneity is introduced into TOUGH2 models by assigning a certain rock type, for which material properties are defined in block ROCKS, to each grid block. The maximum number of rock types one can specify is given by variable *MAXROC* (see file *maxsize.inc*). For high-resolution simulations of permeability heterogeneity, it is inconvenient to specify hundreds of rock types, one for each grid block. TOUGH2 has been extended so that grid block permeabilities or permeability modifiers can be directly specified in block ELEME, columns 41-50. If a positive value is given, it is interpreted as absolute permeability; if a negative value is provided, it is interpreted as a permeability modifier, i.e., a factor with which the absolute permeability specified in block ROCKS is multiplied. If columns 41–50 are blank for the first element, the element-by-elemnt permabilities are ignored. The anisotropy ratio as given in block ROCKS is preserved in both cases. Alternatively, the same information can be provided through block INCON, columns 31–40. Four additional parameters specific to a grid block can be provided in block INCON, columns 41–80.

The permeability or permeability modifiers are stored in array *USERX(1,N)*, where *N* is the grid block number. If *MOP2(18)* is greater than zero (see Table 1), anisotropic permeability modifiers are provided and storied in *USERX(i,N)*, *i*=1…3. The fourth parameter, *USERX(4,N)*, is reserved for a factor scaling capillary strength. To write element-by-element parameters to file *SAVE*, set *MOP(13)* to 1.

Using a permeability modifier instead of permeability itself has the advantage that the mean of the permeability field can easily be changed (or estimated) by adjusting parameter *PER(ISOT)* in block ROCKS.

5.2 Permeability and Porosity Regions

Instead of providing permeabilities and porosities for individual elements or material domains through blocks ELEME or INCON, the option described in this section allows one to specify permeabilities over a certain geometric region. The location and size of this region is parameterized and can thus be subjected to parameter estimation by inverse modeling (e.g., for sensitivity analysis on fault location). For the system response to be a smooth function of the location and size of the permeability region, the region must comprise multiple elements, with a smooth function describing the adjustment of the permeability to its background value as a function of distance from the center of the region. This option can only be used if element coordinates are provided in columns 51–80 in block ELEME.

The permeabilities and porosities will be changed over all elements within a user-specified region. This region is defined by the location of its center and the extent. The shape can be

either an ellipsoid, a rectangular box or cube, or a cylinder. The ellipsoid and box are aligned with the coordinate axes; the cylinder may be arbitrarily oriented.

The box region includes all elements i with coordinates X_i, Y_i, and Z_i that lie within a reactangular box, i.e.,

$$\begin{aligned} d_X &= |X_i - X_c| \le L_X \\ d_Y &= |Y_i - Y_c| \le L_Y \\ d_Z &= |Z_i - Z_c| \le L_Z \end{aligned} \tag{1a}$$

The ellipsoidal region includes all elements i with coordinates X_i, Y_i, and Z_i that satisfy the equation

$$d^2 = \left(\frac{X_i - X_c}{L_X}\right)^2 + \left(\frac{Y_i - Y_c}{L_Y}\right)^2 + \left(\frac{Z_i - Z_c}{L_Z}\right)^2 < 1 \tag{1b}$$

Here, X_i, Y_i, and Z_i are the center coordinates of the box or ellipsoid, and L_X, L_Y, and L_Z are the three half-lengths of the box or the three semi-axes of the ellipsoid. The region is aligned with the coordinate axes.

A cylindrical region is defined by the starting and end coordinates of its axis, and a radius. All elements within this arbitrarily oriented cylinder are selected.

The permeabilities and porosities of the elements within the region will be calculated as a simple weighted average of the respective background values k_0 and ϕ_0 (i.e., the values in the `ROCKS` block assigned to the element in block `ELEME`) and those specified for the particular region, i.e., k_{reg} and ϕ_{reg}:

$$k_i = \omega \cdot k_{reg} + (1-\omega) \cdot k_0 \tag{2a}$$

$$\phi_i = \omega \cdot \phi_{reg} + (1-\omega) \cdot \phi_0 \tag{2b}$$

where ω_i is one of the following "influence functions":

$$\omega = 1 \tag{3a}$$

$$\omega(d) = 1 - d \tag{3b}$$

$$\omega(d) = \left(1 - \left(\frac{3}{2}d - \frac{1}{2}d^3\right)\right) \tag{3c}$$

$$\omega(d) = a^{-d} \tag{3d}$$

$$\omega(d) = \min(1, (1-d)/a) \tag{3e}$$

Here, d is a normalized distance from the center of the region (for boxes and ellipsoids) or from the axis of the cylinder. In Eqs. (3d) and (3e), a is a user-specified parameter,

provided at the end of the region-definition parameters (see Table 3). The effect of the incluence function is that the farther away the element is from the region's center, the lower is the weight assigned to the region-specific permeability and porosity, i.e., the properties near the edges of the region approach those of the background material. Note that if Eq. (3a) is chosen, each element within the region will have the same property value. However, changing the geometry of the region will not lead to a smooth, differentiable change of properties; consequently, this parameter value cannot be used identifying the permeability structure using a gradient based algorithm.

The option is invoked by selecting "XYZ" as the first three characters of the material name (see variable MAT in block ROCKS.1). If a region is requested, the following parameters are read in free format, starting on the next line:

IPERMGEOM: Defines geometry of the region
1 = Box
2 = Ellipsoid
3 = Cylinder
4 = Cube

IPERMINFF: Defines influence function. If negative, the complementary region is selected (i.e., all elements outside the defined geometry).
0 = Constant (Eq. 3a)
1 = Linear (Eq. 3b)
2 = Spherical (Eq. 3c)
3 = Exponential (Eq. 3d)
4 = Constant-linear (Eq. 3e)

XREGION(*i*) (see Table 3)

XREGINFF Parameter *a* (only if |*IPERMINFF*| = 3 or 4)

Table 3. Geometrical Parameters Defining Region

IREGGEOM	*XREGION(i)*								
	1	2	3	4	5	6	7	8	9
1 (box)	X_{min}	Y_{min}	Z_{min}	X_{max}	Y_{max}	Z_{max}	aximuth	dip	plunge
2 (ellipsoid)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	aximuth	dip	plunge
3 (cylinder)	X_S	Y_S	Z_S	Z_E	Y_E	Z_E	radius	-	-
4 (cube)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	aximuth	dip	plunge

If the region is not aligned with the coordinate axes, set *IREGGEOM* negative and provide three correction angles (azimuth, dip, and plunge). Figure 2 shows an input file that uses a rotated ellipsoidal region to represent an inclined fault. The parameters *XREGION*(*i*) can be varied through the iTOUGH2 command `>> REGION` and thus be estimated using inverse modeling. For example, the location and extent of the fault can be varied.

```
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7
MATRX                       0.1 1.000E-17 1.000E-17 1.000E-17
XYZFT                       0.5 1.000E-11 1.000E-11 1.000E-11
-2  2
50.0        60.0       -50.0
 6.0        20.0       100.0
30.0         0.0        80.0
```

Figure 2. TOUGH2 input file for assigning ellipsoidal permeability region.

5.3 Geostatistics

Spatially correlated permeability fields can be generated internally using methods of the geostatistical library GSLIB [*Deutsch and Journel*, 1992]. A special user's guide [*Finsterle and Kowalsky*, 2007] describes the approach and options.

6 Boundary Conditions and Sink/Source Terms

6.1 Time-Dependent Dirichlet Boundary Conditions

Time-dependent Dirichlet boundary conditions can be read from the input file following keyword `TIMBC`, or from the file given on the line following keyword "`TIMBC F`".

FILENAME	:	Filename with boundary condition data (only if "`TIMBC F`"); make sure the file is available in the local directory
NTPTAB	:	Number of elements with time-dependent boundary conditions Repeat the following entries *NTPTAB* times
NBCP, *NBCPV*	:	Number of times and identification number of primary variable
BCELM	:	Name of boundary element (start in Column 1)
TIMEBCV, *PGBCEL*	:	Time and value of primary variable *NBCPV* at boundary element *BCELM*; repeat this entry *NBCP* times

All values are read in free format. Boundary values will be linearly interpolated between table entries. An example is given in Figure 3.

```
TIMBC----1----*----2----*----3----*----4----*----5----*----6----*----7
2
3  1
TOP99
    0.0    1.0E5
86400.0    1.1E5
 1.0E50    1.1E5
2  2
  T 1
    0.0    10.1
  1.0E7    10.9
```

Figure 3. TOUGH2 input file for assigning time-dependent Dirichlet boundary conditions.

6.2 Free-Drainage Boundary Condition

A free drainage boundary condition for liquid flow can be implemented, in which gravity is the only driving force, i.e., (capillary) pressure pressure gradients are ignored across an interface to the boundary gridblock. This type of boundary condition comes into effect at each connection in which one of the gridblocks belongs to rock type `DRAIN`.

6.3 User-Specified Boundary Conditions

In TOUGH2, Neumann boundary conditions are specified by introducing sinks and sources in block GENER. Dirichlet boundary conditions can be implemented by assigning very large volumes to grid blocks adjacent to the boundary so that the thermodynamic conditions in those elements do not change from fluid and heat exchange with finite-size grid blocks in the model domain.

Prescribed, but time-varying boundary conditions can be implemented by specifying appropriate (large) sinks and sources in grid blocks having a very large volume or by using keyword `TIMBC` as described in Section 6.1. Moreover, for simple step changes, iTOUGH2 offers an alternative option (see command `>> RESTART TIME`). Another possibility is described in this section. The user can provide values of the primary variables for selected elements as a function of time. The function has to be programmed into subroutine USERBC, which can be found in file *it2user.f*. In subroutine USERBC, the user has the possibility to provide the value of one or more primary variables for selected elements as a function of time. For example, these values can be calculated internally or read from a file. The header of subroutine USERBC is shown in Figure 4. The element number *N* or element name *CELEM* can be used to identify the boundary grid block. The user is supposed to return a value for one or several of the primary variables through array *X*. In the example given below, a table of time versus pressure data is read from file *atm_pres.dat* and assigned to element `'ATM 0'` using the linear interpolation function INTERP1. Note that either the full path to file *atm_pres.dat* must be given, or the file must be copied to the temporary directory using option `-fi filename`.

Subroutine USERBC is called only if *MOP(22)* is either 1 or 2. If *MOP(22)* is 2, the EOS module is called after completion of a time step to ensure that the user-specified thermodynamic conditions are updated. If *MOP(22)* is 1, subroutine USERBC is called, but no additional call to subroutine EOS is made. This allows one to make time-dependent changes to TOUGH2 variables that are not primary variables, and that do not require a recalculation of the thermodynamic state.

```
************************************************************************
      SUBROUTINE USERBC(N,CELEM,VOLUME,TIME,X)
************************************************************************
*  User specified boundary condition                                   *
*  Set MOP(22).GE.1                                                    *
*      MOP(22)=1 don't call EOS                                        *
*      MOP(22)=2 call EOS                                              *
*  Return user specified boundary condition (vector X)                 *
*  for element CELEM at time TIME and/or change the VOLUME of CELEM    *
************************************************************************

      CHARACTER CELEM*5

      PARAMETER (MDATA=5000)

      DIMENSION X(*),DTIME(MDATA),DVALUE(MDATA)

      SAVE DTIME,DVALUE,IREAD

      IF (CELEM.EQ.'ATM 0') THEN
C --- Read table from a file
         IF (IREAD.EQ.0) THEN
            IREAD=IREAD+1
            OPEN(UNIT=39,FILE='atm_pres.dat',STATUS='OLD')
            I=0
 1001       CONTINUE
            I=I+1
            READ(39,*,END=1002) DTIME(I),DVALUE(I)
            GOTO 1001
 1002       CONTINUE
            NDATA=I-1
            CLOSE(39)
         ENDIF
         CALL INTERP1(TIME,X(1),DTIME,DVALUE,NDATA)
      ENDIF
```

Figure 4. Subroutine USERBC for specifying time-dependent boundary conditions.

6.4 Tabular Input of Time-Dependent Rates

In addition to the standard input format, time-dependent generation rates (i.e., if *LTAB* > 1 in block GENER.1) can be provided as a free-format table with time in the first column, injection or production rate in the second column, and (if *ITAB* is not left blank) specific enthalpy in the third column. The number of table rows is given by *LTAB*. The tabular format is chosen by providing the character "T" or "D" in Column 7 after keyword GENER. Moreover, time and rate conversion factors can be given in Columns 11–20 and 21–30. If character "D" is specified in Column 7, time can be given in (any) date forma; it will be converted to seconds (relative to the first date given). These conversion factors only apply to sinks/source with time-dependent generation rates (i.e., constant rates given in Columns 41–50 of block GENER.1 are not affected). Figure 5 shows an example, in which time-dependent water injection rates (in m^3/hour) are given as a four-entry table, with time given in hours. The options discussed in this section are only available if sinks/sources are given directly in the TOUGH2 input deck. The external file *GENER* has to be provided in the standard format.

```
MOMOP----1----*----2----*----3----*----4----*----5----*----6----*----7
  1
----*-7--1----GTFACT----GRFACT----*----4----*----5----*----6----*----7
GENER T       3600.0 2.7778E-7 ---*----4----*----5----*----6----*----7
A11 1INJ 1    0    0    0    4     WATE
  0.00  1.0
  6.00  1.0
  6.01  0.0
1.0e50  0.0
```

Figure 5. TOUGH2 input format for specifying time-dependent generation rates in tabular format, and using conversion factors for time and rate.

6.5 Honoring Generation Times

By setting *MOP2*(3) = 1 in block MOMOP (see Figure 5), each time specified for any sink/source with time-dependent generation rates will be honored in the simulation, i.e., time stepping will automatically be adjusted to coincide with the time when the generation rate changes.

6.6 Material-Related Sinks/Sources

Instead of providing sinks/sources for individual elements, the option described in this section allows one to specify sinks/sources for all elements that belong to a certain material domain. The injection or production rates specified this way arevolume-specific, i.e., they will have units of kg/s per m^3. A mass rate in kg/s will then be internally calculated based on the volume of the element that belongs to the identified material type. The option is invoked by the sink/source code name "MAT*ii*" (see variables *SL* and *NS* in block GENER.1), where the integer *ii* is the sequence number of the material domain as entered in block ROCKS. An example is shown in Figure 6.

For example, infiltration can be conveniently specified if a one-meter thick land surface layer assigned to a unique material type is provided. Using material-related volume-specific mass flow rates, the infiltration in kg/s (≈ mm/s) can be provided in a single GENER-block entry; it will be internally converted to mass flow rates that are proportional to the surface areas of all infiltration elements.

The model-related sinks/sources option is also convenient to specify volumetric generation rates, e.g., for hydrogen generation due to corrosion, gas from biodegradation in landfills, or radionuclide and gas generation rates in rock masses.

```
GENER----1----*----2----*----3----*----4----*----5----*----6----*----7
 AA 1MAT 3                  1     COM4 3.1688E-08
```

Figure 6. TOUGH2 input format for specifying material-related, volume-specific generation rates.

6.7 Sink/Source Regions

Instead of providing sinks/sources for individual elements, the option described in this section allows one to specify a sink/source over a certain geometric region. The location and size of this region is parameterized and can thus be subjected to parameter estimation by inverse modeling (e.g., for contaminant source identification or optimization of well locations). For the sink/source response to be a smooth function of its location and size, the sink/source region must comprise multiple elements, with a smooth function describing the decline of the rate with distance from the center of the region. This option can only be used if element coordinates are provided in columns 51–80 in block `ELEME`.

The injection or production rate will be distributed over all elements within a user-specified region. This region is defined by the location of its center and the extent. The shape can be either an ellipsoid, a rectangular box or cube, or a cylinder. The ellipsoid and rectangle are aligned with the coordinate axes; the cylinder may be arbitrarily oriented.

The box region includes all elements i with coordinates X_i, Y_i, and Z_i that lie within a reactangular box, i.e.,

$$\begin{aligned} d_X &= |X_i - X_c| \le L_X \\ d_Y &= |Y_i - Y_c| \le L_Y \\ d_Z &= |Z_i - Z_c| \le L_Z \end{aligned} \tag{4a}$$

The ellipsoidal region includes all elements i with coordinates X_i, Y_i, and Z_i that satisfy the equation

$$d^2 = \left(\frac{X_i - X_c}{L_X}\right)^2 + \left(\frac{Y_i - Y_c}{L_Y}\right)^2 + \left(\frac{Z_i - Z_c}{L_Z}\right)^2 < 1 \tag{4b}$$

Here, X_i, Y_i, and Z_i are the center coordinates of the box or ellipsoid, and L_X, L_Y, and L_Z are the three half-lengths of the box or the three semi-axes of the ellipsoid. The region is aligned with the coordinate axes.

A cylindrical region is defined by the starting and end coordinates of its axis, and a radius. All elements within this arbitrarily oriented cylinder are selected.

The total generation rate Q of the sink/source is distributed among these elements as follows:

$$q_i = Q\frac{\omega_i}{\sum \omega_i} \tag{5}$$

where ω_i is one of the following "influence functions":

$$\omega = 1 \tag{6a}$$
$$\omega(d) = V \cdot (1-d) \tag{6b}$$
$$\omega(d) = V \cdot \left(1 - \left(\frac{3}{2}d - \frac{1}{2}d^3\right)\right) \tag{6c}$$
$$\omega(d) = V \cdot a^{-d} \tag{6d}$$
$$\omega(d) = \min(1, (1-d)/a) \tag{6e}$$

Here, V is the element volume, and d is a normalized distance from the center of the region (for boxes and ellipsoids) or from the axis of the cylinder. In Eqs. (6d) and (6e), a is a user-specified parameter, to be provided after the region-definition parameters (see Table 4). The effect of the incluence function is that the farther away the element is from the region's center, the smaller is the rate assigned to this element. The rate is also weighted by the element's volume. Note that if Eq. (6a) is chosen, each element within the region will have the same volume-weighted sink/source strength. However, changing the geometry of the region will not lead to a smooth, differentiable reallocation of the generation rates; consequently, this parameter value cannot be used for contaminant source identification or well location optimization using a gradient-based algorithm.

The option is invoked by the sink/source code name "`XYZ..`" (see variable `SL` in block `GENER.1`; variable *`NS`* is an arbitrary two-digit integer). If a region is requested, the following parameters are read in free format, starting on the line following `GENER.1`:

`IPERMGEOM`: Defines geometry of the region
1 = Box
2 = Ellipsoid
3 = Cylinder
4 = Cube

`IREGINFF`: Defines influence function. If negative, the complementary region is selected (i.e., all sinks/sources outside the defined geometry).
0 = Constant (Eq. 6a)
1 = Linear (Eq. 6b)
2 = Spherical (Eq. 6c)
3 = Exponential (Eq. 6d)
4 = Constant-linear (Eq. 6d)

`XREGION`(`i`) (see Table 4)

`XREGINFF` Parameter a (only if |*`IREGINFF`*| = 3 or 4)

Table 4. Geometrical Parameters Defining Region

IREGGEOM	*XREGION(i)*								
	1	2	3	4	5	6	7	8	9
1 (box)	X_{min}	Y_{min}	Z_{min}	X_{max}	Y_{max}	Z_{max}	aximuth	dip	plunge
2 (ellipsoid)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	aximuth	dip	plunge
3 (cylinder)	X_S	Y_S	Z_S	Z_E	Y_E	Z_E	radius	-	-
4 (cube)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	aximuth	dip	plunge

If the region is not aligned with the coordinate axes, set *IREGGEOM* negative and provide 3 correction angles (azimuth, dip, and plunge). Figure 7 shows an input file that uses an elliptical source region. Note that the vertical semi-axis (variable *XREGION*(6)) of the ellipsoid is very small, so only elements at an elevation of -4.5 m (variable *XREGION*(1)) will be selected. The parameters *XREGION*(*i*), *i*=1,…,7 can be varied through the iTOUGH2 command >> REGION and thus be estimated using inverse modeling.

Since the flow generation rates are calculated internally, set *MOP*(4)= 2 to see the allocation of the total flow rate of 1 kg/s to the eight elements within the ellipsoid (see Figure 8).

```
Test ellipsoidel source region option
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7
SAND1                     0.5 1.000E-12 1.000E-12 1.000E-12

MESHM----1----*----2----*----3----*----4----*----5----*----6----*----7
XYZ

NX      10       1.0
NY      10       1.0
NZ      10       1.0


PARAM----1-MOP: 123456789012345678901234----*----5----*----6----*----7
   2   1       1   2
                      1.0E-5

                 0.5
START----1----*----2---------3----*----4---------5---------6----*----7
INCON----1----*----2---------3----*----4---------5---------6----*----7

GENER----1----*----2---------3----*----4---------5---------6----*----7
    1XYZ 1                    1     WATE        1.0
2  1   5.0       4.7      -4.5       2.5       1.1       0.1

ENDCY----1----*----2----*----3----*----4----*----5----*----6----*----7
```

Figure 7. TOUGH2 input file for testing ellipsoidal sink/source regions.

```
QQQQQQQQQQQ SUBROUTINE QU QQQQQQQQQQ  --- [KCYC,ITER] = [   1,  1]
ELEMENT A55 4   SOURCE XYZ 1 --- FLOW RATE = 0.132698E+00
ELEMENT A56 4   SOURCE XYZ 1 --- FLOW RATE = 0.203361E-01
ELEMENT A55 5   SOURCE XYZ 1 --- FLOW RATE = 0.259560E+00
ELEMENT A56 5   SOURCE XYZ 1 --- FLOW RATE = 0.874065E-01
ELEMENT A55 6   SOURCE XYZ 1 --- FLOW RATE = 0.259560E+00
ELEMENT A56 6   SOURCE XYZ 1 --- FLOW RATE = 0.874065E-01
ELEMENT A55 7   SOURCE XYZ 1 --- FLOW RATE = 0.132698E+00
ELEMENT A56 7   SOURCE XYZ 1 --- FLOW RATE = 0.203361E-01
```

Figure 8. Excerpt of TOUGH2 output file showing allocation of generation rates within ellipsoidal source region.

7 Relative Permeability and Capillary Pressure Functions

Subroutines RELP and PCAP provide the relative permeability and capillary pressure functions, respectively (see *Pruess* [1987]). They are frequently modified to accommodate particular needs. The user should therefore carefully check the functional form and required parameters before selecting a certain curve through variable `IRP` and `ICP`, respectively (see also command `>>> CHARACTERISTIC`).

The functions provided in this version include the three-phase curves used by the T2VOC module [*Falta et al.*, 1995]. Additional functions are provided as described in the following subsections.

7.1 Modified Brooks-Corey Model

A modified version of the Brooks-Corey model [*Luckner et al.,* 1989] has been implemented. In order to prevent the capillary pressure from decreasing towards negative infinity as the effective saturation approaches zero, a linear function is used for saturations S_l below a certain value $(S_{lr}+\varepsilon)$, where ε is a small number. The slope of the linear extrapolation is identical with the slope of the capillary pressure curve at $S_l = S_{lr}+\varepsilon$. Alternatively, the capillary pressure is prevented from becoming more negative than $-p_{c,\max}$.

The modified Brooks-Corey model is invoked by setting both `IRP` and `ICP` to 10. The model is described by the following set of equations (the input parameters are listed in Table 5):

$$S_{ec} = \frac{S_l - S_{lrc}}{1 - S_{lrc}} \tag{8a}$$

$$S_{ek} = \frac{S_l - S_{lrk}}{1 - S_{lrk} - S_{gr}} \tag{8b}$$

$$p_c = -p_e \left(S_{ec}\right)^{-1/\lambda} \qquad \text{for} \quad S_l \geq (S_{lrc}+\varepsilon) \tag{9a}$$

$$p_c = -p_e \left(\frac{\varepsilon}{1-S_{lrc}}\right)^{-1/\lambda} + \frac{p_e}{\lambda}\frac{1}{1-S_{lrc}}\left(\frac{\varepsilon}{1-S_{lrc}}\right)^{-\frac{1+\lambda}{\lambda}} \left(S_l - S_{lrc} - \varepsilon\right) \qquad \text{for} \quad S_l < (S_{lrc}+\varepsilon) \tag{9b}$$

$$p_c \geq -p_{c,\max} \tag{10}$$

$$k_{rl} = S_{ek}^{\frac{2+3\lambda}{\lambda}} \tag{11a}$$

$$k_{rg} = (1 - S_{ek})^2 \left(1 - S_{ek}^{\frac{2+\lambda}{\lambda}}\right) \tag{11b}$$

$$k_{rg} = 1 - k_{rl} \tag{11c}$$

Table 5. Input Parameters for Modified Brooks-Corey Model

Parameter	Variable	Description
`IRP`	10	select Brooks-Corey relative permeability model
`RP(1)`	S_{lrk}	residual liquid saturation for relative permeability functions
`RP(2)`	S_{gr}	residual gas saturation
`RP(3)`	(flag)	if zero, use (11b), otherwise (11c)
`ICP`	10	select Brooks-Corey capillary pressure model
`CP(1)`	λ	pore size distribution index
`CP(2)`	p_e	gas entry pressure [Pa]
		if `CP(2)` negative and `USERX(1,N)` non-zero, apply Leverett's rule: p_e = -CP(2)$\sqrt{\text{USERX(1, N)/PER(NMAT)}}$
		if `USERX(4,N)` positive then p_e = `USERX(4,N)`
		if `USERX(4,N)` negative then p_e = -`USERX(4,N)`·`CP(2)`
`CP(3)`	ε or	if `CP(3)` = 0 then $p_{c,\max} = 10^{50}$, $\varepsilon = -1$
	$p_{c,\max}$	if 0 < `CP(3)` < 1 use linear model (9b) for $S_l < S_{lr} + \varepsilon$
		if `CP(3)` ≥ 1, then $p_{c,\max}$ = `CP(3)`, $\varepsilon = -1$
`CP(6)`	S_{lrc}	if zero, then $S_{lrc} = S_{lrk}$

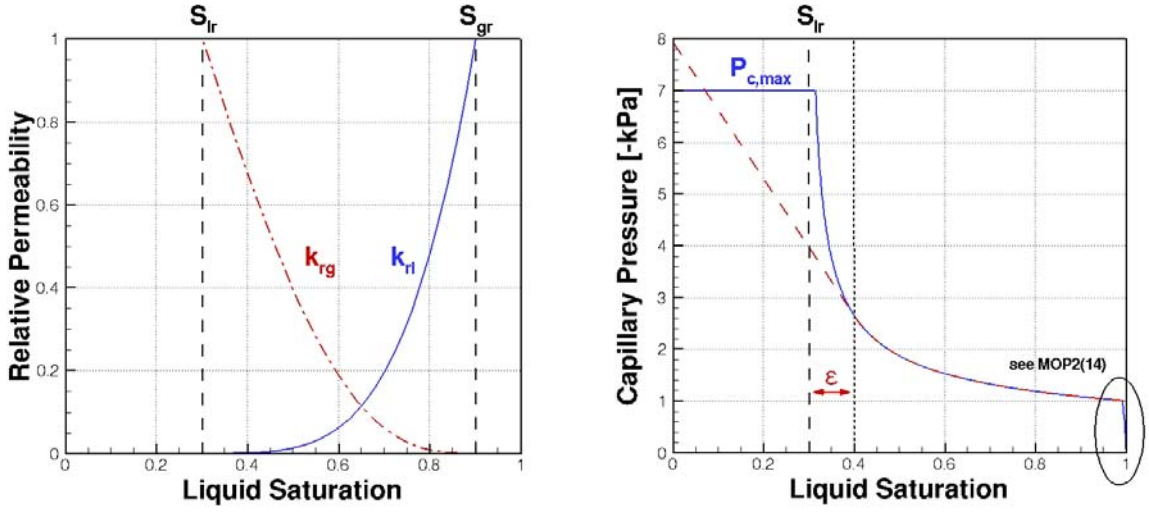

Figure 9. Modified Brooks-Corey relative permeability and capillary pressure curves.

7.2 Modified van Genuchten Model

A modified version of the van Genuchten model [*Luckner et al.,* 1989] has been implemented. In order to prevent the capillary pressure from decreasing towards negative infinity as the effective saturation approaches zero, a linear function is used for saturations S_l below a certain value $(S_{lr}+\varepsilon)$, where ε is a small number. The slope of the linear extrapolation is identical with the slope of the capillary pressure curve at $S_l = S_{lr}+\varepsilon$. Alternatively, the capillary pressure is prevented from becoming more negative than $-p_{c,\max}$.

The modified van Genuchten model is invoked by setting both `IRP` and `ICP` to 11. The model is described by the following set of equations (the input parameters are listed in Table 6**Error! Reference source not found.**):

$$S_{ec} = \frac{S_l - S_{lrc}}{1 - S_{lrc}} \tag{12a}$$

$$S_{ekl} = \frac{S_l - S_{lrk}}{1 - S_{lrk}} \tag{12b}$$

$$S_{ekg} = \frac{S_l}{1 - S_{gr}} \tag{12c}$$

$$S_{ec^*} = \frac{\varepsilon}{1 - S_{lrc}} \tag{12d}$$

$$p_c = -\frac{1}{\alpha}\left[\left(S_{ec}\right)^{(\gamma-1)/m} - 1\right]^{1/n} \quad \text{for} \quad S_l \geq (S_{lrc}+\varepsilon) \tag{13a}$$

$$p_c = -\frac{1}{\alpha}\left[S_{ec^*}^{(\gamma-1)/m} - 1\right]^{1/n} - \beta\cdot\left(S_l - S_{lrc} - \varepsilon\right) \quad \text{for} \quad S_l < (S_{lrc}+\varepsilon) \tag{13b}$$

with

$$\text{linear extension:} \quad \beta = -\frac{(1-\gamma)}{\alpha n m}\cdot\frac{1}{(1-S_{lrc})}\cdot\left(S_{ec^*}^{(\gamma-1)/m} - 1\right)^{\frac{1}{n}-1} S_{ec^*}^{\left(\frac{\gamma-1-m}{m}\right)}$$

$$p_c = -\frac{1}{\alpha}\left[S_{ec^*}^{(\gamma-1)/m} - 1\right]^{1/n}\cdot 10^{\beta\cdot(S_l - S_{lrc} - \varepsilon)} \quad \text{for} \quad S_l < (S_{lrc}+\varepsilon) \tag{13c}$$

with

$$\text{log-linear extension:} \quad \beta = -\log_{10}(e)\cdot\left(\frac{1-m}{m}\cdot\frac{\gamma-1}{\varepsilon}\cdot\frac{1}{S_{ec^*}^{(1-\gamma)/m} - 1}\right)$$

$$p_c \geq -p_{c,\max} \tag{13d}$$

$$k_{rl} = S_{ekl}^{\gamma} \cdot S_{ekl}^{(1-\gamma)\eta} \cdot \left[1-\left(1-S_{ekl}^{(1-\gamma)/m}\right)^{m}\right]^{2} \tag{14a}$$

$$k_{rg} = \left(1-S_{ekg}\right)^{\zeta}\left[1-S_{ekg}^{1/m}\right]^{2m} \tag{14b}$$

or

$$k_{rg} = 1-k_{rl} \tag{14c}$$

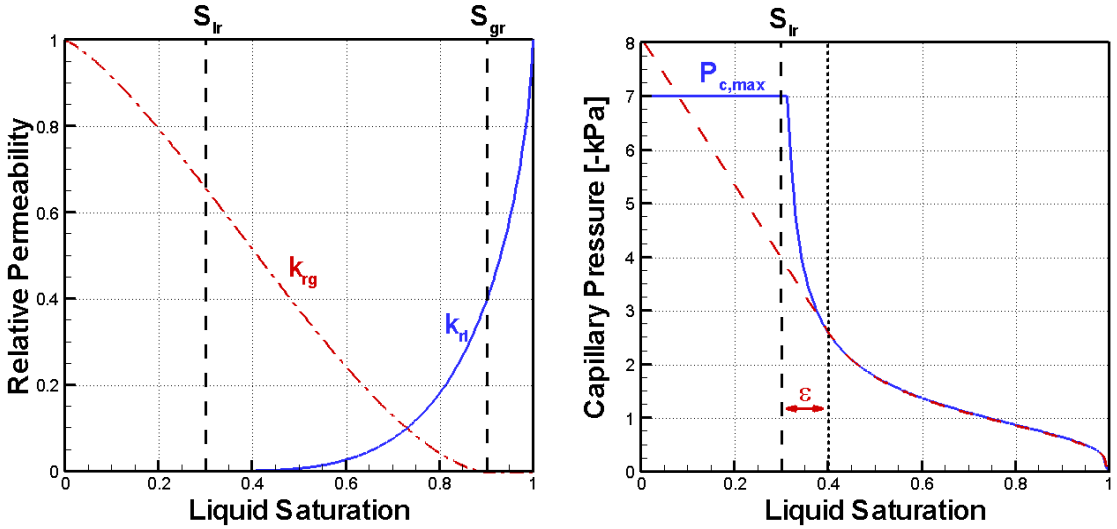

Figure 10. Modified van Genuchten relative permeability and capillary pressure curves.

Table 6. Input Parameters for Modified van Genuchten Model

Parameter	Variable	Description
		Relative Permeability Function
IRP	11	select van Genuchten relative permeability model
RP(1)	S_{lrk}	residual liquid saturation for relative permeability functions, if negative, $S_{lrk} = 0$ for calculating gas relative permeability, absolute value is used for calculating liquid relative permeability
RP(2)	S_{gr}	residual gas saturation if negative, $S_{gr} = 0$ for calculating liquid relative permeability, absolute value is used for calculating gas relative permeability
RP(3)	(flag)	if zero, use (14b), if non-zero, use (14c)
RP(4)	η	exponent in (14a), default = ½
RP(5)	ε_k	use linear function between $k_{rl}(S_e = 1-\varepsilon_k)$ and 1.0.
RP(6)	a_{fm}	Constant fracture-matrix interaction reduction factor, in combination with Active Fracture Model (see Section 8)
RP(7)	ζ	exponent in (10b), default = 1/3
		Capillary Pressure Function
ICP	11	select van Genuchten capillary pressure model
CP(1)	n	parameter related to pore size distribution index (see also *CP(4)*)
CP(2)	$1/\alpha$	parameter related to gas entry pressure [Pa] USERX(4,N)>0 : $1/\alpha_i$ = USERX(4,N) USERX(4,N)<0 : $1/\alpha_i$ = USERX(4,N) · CP(2) CP(2)<0: apply Leverett scaling rule: $1/\alpha_i = 1/\alpha_{ref} \cdot \sqrt{k_i / k_{ref}}$ where: $1/\alpha_{ref} = \lvert$ CP(2) $\rvert$ k_{ref} = PER(NMAT) USERX(1,N)>0 : k_i = USERX(1,N) USERX(1,N)<0 : k_i = USERX(1,N) · PER(NMAT)
CP(3)	ε or $p_{c,\max}$	CP(3)=0 : $p_{c,\max} = 10^{50}$, $\varepsilon = -1$ 0<CP(3)<1 : ε = CP(3); use linear extension (13b) CP(3) ≥ 1 : $p_{c,\max}$ = CP(3), $\varepsilon = -1$ –1<CP(3)<0 : ε = |CP(3)|; use log-linear extension (13c)
CP(4)	m	if zero then m=1-1/CP(1), else m=CP(4) and n=1/(1-m)
CP(5)	T_{ref}	if negative, |CP(5)| is reference temperature to account for temperature dependence of capillary pressure due to changes in surface tension
CP(6)	γ	parameter of Active Fracture Model (see Section 8)
CP(7)	S_{lrc}	if zero, then $S_{lrc} = S_{lrk}$

8 Active Fracture Model

8.1 Active Fracture Concept

There is evidence that only a portion of the connected fracture network conducts water under unsaturated conditions. The fractures contributing to liquid flow are referred to as "active fractures". The Active Fracture Concept (AFC) was developed by *Liu et al.* [1998] to describe gravity-dominated, non-equilibrium, preferential liquid flow in fractures, which is expected to be similar to fingering in unsaturated porous media. AFC is based on the hypothesis that (1) the number of active fractures is small compared with the total number of connected fractures, (2) the number of active fractures within a grid block is large so that the continuum approach is valid, and (3) the fraction of active fractures, f_a, is related to water flux and equals one for a fully saturated system, and zero if the system is at residual saturation. The following power function of effective liquid saturation, S_e, fulfills these conditions:

$$f_a = S_e^{\gamma} \tag{15}$$

Here, γ is a positive constant depending on properties of the fracture network, and S_e is the effective liquid saturation given by

$$S_e = \frac{S_l - S_{lr}}{1 - S_{lr}} \tag{16}$$

Capillary pressure and relative permeability functions are modified to account for the fact that the effective saturation in the active fractures, S_{ea}, is larger than the effective saturation of the total fracture continuum:

$$S_{ea} = \frac{S_e}{f_a} = S_e^{1-\gamma} \tag{17}$$

Using the van Genuchten model, capillary pressure and liquid relative permeability are given, respectively, by

$$p_c = -\frac{1}{\alpha}\left[S_e^{(\gamma-1)/m} - 1\right]^{1/n} \tag{18}$$

and

$$k_{rl} = S_e^{(1+\gamma)/2}\left\{1 - \left[1 - S_e^{(1-\gamma)/m}\right]^m\right\}^2 \tag{19}$$

The fracture-matrix interface area reduction factor (see Section A8) is given by

$$a_{fm} = S_e^{1+\gamma} \tag{20}$$

The AFC is invoked by selecting $\gamma > 0$, which is provided as an additional parameter of the standard van Genuchten model (`ICP=7`) through variable `CP(6,NMAT)`. Fracture-matrix interface area reduction is invoked by selecting `ISOT` between -10 and -12 (see Table 7).

8.2 Reduction of Fracture-Matrix Interface Area

There is evidence that fracture-matrix interaction in the unsaturated zone is reduced as a result of fracture coatings as well as preferential flow in the fractures as invoked by flow instabilities (fingering) and small-scale heterogeneities. A number of options for reducing fracture-matrix interface area have been implemented for use in a dual-permeability flow simulation. Interface area reduction is applied to connections with a negative value for variable *ISOT*, which is provided in the CONNE block. Different modifiers are used depending on the value of *ISOT* and *MOP*(8) as summarized in Table 7.

Table 7. Option for Reducing Fracture-Matrix Interface Area

ISOT	*MOP*(8)	Interface area reduction factor a_{fm}
positive	any	No interface area reduction, i.e., $a_{fm} = 1$
negative	1	$a_{fm} = RP(6, NMAT)$
-1, -2, -3	0	$a_{fm} = S_\beta$
	2	$a_{fm} = S_\beta \cdot RP(6, NMAT)$
-4, -5, -6	0	$a_{fm} = k_{r\beta}$
	2	$a_{fm} = k_{r\beta} \cdot RP(6, NMAT)$
-7, -8, -9	0	$a_{fm} = RP(6, NMAT)$
-10, -11, -12	0	$a_{fm} = S_e^{1+\gamma}$ (see Section 8.1)

a_{fm}	:	Fracture-matrix interface area reduction factor.
S_β	:	For flow of phase β, upstream saturation of phase β.
$k_{r\beta}$	:	For flow of phase β, upstream relative permeability of phase β.
$RP(6, NMAT)$	:	6th parameter of rel. perm. function of upstream element; if zero (i.e., not specified), reset to one.

10 Coupled Overland – Subsurface Flow

iTOUGH2 provides the capability to fully couple overland flow (solving the non-inertial, diffusion wave form of the Saint-Venant equations) with subsurface flow using an approach similar to that proposed by *Weill et al.* [2009]. The momentum and continuity equations are given by:

$$S_{f,i} = -\nabla(z_l + h_s) \tag{21}$$

$$\frac{\partial h_s}{\partial t} + \nabla \cdot (h_s \vec{U}) = q_s \tag{22}$$

where $S_{f,i}$ is the friction slope [-] in the direction i, z_l is land surface elevation [L], h_s is the water depth on the surface, $\vec{U}$ is the depth averaged flow velocity [LT^{-1}], and q_s is a source/sink term [LT^{-1}]. The Manning-Strickler formula is used for relating velocity to friction slopes:

$$U_i = \frac{h_s^{2/3}}{n_{man}} \sqrt{S_{f,i}} \tag{23}$$

where n_{man} is the Manning roughness coefficient [$L^{-1/3}T$]. The diffusion-wave form of the Saint-Venant equations assumes slowly varying flow.

In order to couple the surface and subsurface flow equations, the approach developed by *Weill et al.* [2009] is followed. A surface layer of thickness e is expected to be present at the top of the numerical model. For liquid flow within the surface layer, Eqs. (21)–(23) are combined into a form that is similar to that describing flow in a porous medium:

$$\frac{\partial h_s}{\partial t} - \nabla \cdot \left(K_s \nabla (z_l + h_s) \right) = q_s \tag{24}$$

Here, the non-diagonal terms of the hydraulic conductivity tensor K_s are zero and the diagonal components are

$$K_{s,xx} = \frac{h_s^{5/3}}{n_{man} \sqrt{\nabla_x (z_l + h_s)}} \tag{25}$$

$$K_{s,yy} = \frac{h_s^{5/3}}{n_{man} \sqrt{\nabla_y (z_l + h_s)}} \tag{26}$$

$$K_{s,zz} = k_{zz} \frac{k_{rl}}{\mu_l} \tag{27}$$

The horizontal hydraulic conductivities describe surface water flow, while the vertical hydraulic conductivity describes resistance to liquid flow between the surface and subsurface layer, with k_{zz} equal to the vertical permeability of the subsurface layer. The liquid pressure in the surface layer is assumed hydrostatic. Because liquid and gas pressures are continuous across the surface/subsurface boundary, negative water depths occur when there is no runoff. The volumetric liquid content in the surface layer is defined as

$$\theta_l = \begin{cases} 0 & \text{for } h_s < 0 \\ h_s / e & \text{for } h_s \geq 0 \end{cases} \tag{28}$$

For vertical liquid flow, the liquid relative permeability is set to one, unless $h_s/e < 10^{-5}$, when it is specified as zero. To capture the pressure head due to ponding in the surface layer, a positive capillary pressure is calculated as a function of h_s.

For gas flow within the surface layer and between the surface and subsurface layers, the regular subsurface flow equationsa re used. If runoff occurs in the surface layer, i.e., $\theta_l > 0$ then $k_{rg} = 0$ for pressure gradients from the surface to the subsurface layers such that no gas flows between the surface and subsurface layers (note, however, that it is possible for pressurized gas to escape the subsurface and flow to the surface layer), and $k_{rg} = 1$ within the surface layer such that gas flows freely in the surface layer. If there is no runoff, $\theta_l = 0$ $k_{rg} = 1$ and the intrinsic permeability of the surface layer is assumed isotropic and equal to the vertical intrinsic permeability of the subsurface layer.

To implement coupled surface water – groundwater flow, the user has to set up a TOUGH2 model—similar to that shown in Figure 11—as follows:

(1) Create a material in block `ROCKS` named `SURWA`. All elements representing surface water must be related to this material type.

(2) For material `SURWA`, set porosity close to 1.0, set *`NAD`*=1, and provide two Manning's coefficients referring to the first and second direction (i.e., for *`ISOT`*=1 and 2) and the surface layer thickness (consistent with its definition in block `ELEME`) in columns 51–60, 61–70, and 71–80, respectively. Do *not* specify material-dependent relative permeability and capillary pressure function using *`NAD`*=2; these functions are provided internally.

(3) To specify a zero-water-depth-gradient boundary condition, create boundary elements (either inactive or of large volume), assign them to a material named SURZG, and connect them to the surface water elements at the desired locations.

(4) Create a material in block `ROCKS` named `ATMOS` and assign properties suitable for representing atmospheric conditions.

(5) Note that the absolute permeabilities in materials SURWA, SURGZ, and ATMOS will be used to calculate gas flow only; liquid flow will be determined by the Saint-Venant equation.

(6) Generate a mesh with an atmospheric element (or layer) and a surface-water layer (typically of thickness 1 m). Assign materials ATMOS and SURWA to these two layers.

(7) Connect the atmosphere to the surface-water layer, and the surface-water layer to the subsurface system. Connections between surface-water elements must be assigned to directions *ISOT*=1 and 2; the slope of the surface is provides through variable *BETAX*. Connections between surface-water and subsurface elements must be assigned to direction *ISOT*=3; the nodal distance from the surface-water element to the interface with the subsurface element is internally set to zero.

Surface-water flow is solved simultaneously and fully coupled with subsurface flow using the standard TOUGH2 implicit scheme. Note that time-step size may be governed by the relatively fast flow occurring in the surface-water layer.

```
Surface water - subsurface water flow test problem
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
SURWA    1     2650.     0.999   1.0e-12   1.0e-12    1.e-12         1.     1000.
                                                        0.1       0.1       1.0
SANDY    2     2650.     0.300   1.0e-12   1.0e-12    1.e-12         1.     1000.

    7           .818      0.25       1.0
    7           .818      0.25 .00023488      1.e7       1.0
ATMOS    2     2650.     0.999   1.0e-12   1.0e-12    1.e-12         1.     1000.

    3            0.9
    1            0.0       0.0       1.0

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
----*----1 MOP: 123456789*123456789*1234----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
  -4 100       1100000900000000400006000
    0.0E00               0.001    1.0e20                10.0
     1.E-5
               1.0E5               10.750               20.00
MOMOP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
2
ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A11 0          ATMOS0.1000E+520.2240E-02                0.75     0.500+.0000E+00
SWA 1          SURWA0.1000E+010.1000E+01               0.500     0.500    -0.500
A21 1          SANDY0.1000E+010.1000E+01               0.500     0.500    -1.005
A21 2          SANDY0.1000E+010.1000E+01               1.500     0.500    -1.005
SWA 2          SURWA0.1000E+010.1000E+01               1.500     0.500    -0.500

CONNE
SWA 1SWA 2                  10.5000E+000.5000E+000.1000E+01       0.1
SWA 1A21 1                  30.0000E-000.5000E-000.1000E+010.1000E+01
A21 1A21 2                  10.5000E+000.5000E+000.1000E-01
SWA 2A21 2                  30.0000E-000.5000E-000.1000E+010.1000E+01
A11 0SWA 1                  30.0000E-000.5000E-000.1000E-100.1000E+01
A11 0SWA 2                  30.0000E-000.5000E-000.1000E-100.1000E+01

INCON----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A11 0           0.99900000E+00
 0.1000000000000E+06 0.1000000000000E+01 0.2000000000000E+02

ENDCY----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
```

Figure 11. TOUGH2 input file for testing coupled surface water – subsurface flow.

11 Semi-Analytical, Radial Heat Exchange

Radial, conductive heat exchange between fluids in a discretized wellbore and the formation is calculated using a semi-analytical, time-convolution method. The time-dependent temperature evolution in the wellbore is calculated numerically using TOUGH2. At each time step, radial heat transfer with the formation is calculated by superposition of analytical solutions of heat flow that are dependent on the temperature differences between subsequent time steps.

Carslaw and Jaeger [1959, pp. 334–339] provided an approximate solution for heat conduction between a cylinder and surrounding media where the temperature of the cylinder is maintained constant. If the initial temperature difference between the two domains is $\Delta T = T_w - T_f$ (where T_w and T_f are the temperatures in the well and the formation, respectively), the heat flux q from the wellbore to the formation can be calculated using as the product of a heat transfer function and the temperature using Equ. (29) for small values of the dimensionless time $t_d = \alpha t / r_0^2$, where α is the thermal diffusivity, and Equ. (30) for large values of t_d:

$$q = f_1(t_d) \cdot \Delta T = \frac{k\Delta T}{r_0}\left\{(\pi t_d)^{-0.5} + \frac{1}{2} - \frac{1}{4}\left(\frac{t_d}{\pi}\right)^{0.5} + \frac{1}{8}t_d - \ldots\right\} \tag{29}$$

$$q = f_2(t_d) \cdot \Delta T = \frac{2k\Delta T}{r_0}\left\{\frac{1}{\ln(4t_d) - 2\gamma} - \frac{\gamma}{\left[\ln(4t_d) - 2\gamma\right]^2} - \ldots\right\} \tag{30}$$

Here, k is thermal conductivity (W m^{-1} K^{-1}), r_0 is the wellbore radius (m), and γ is the Euler constant (0.57722).

The heat transfer functions f_1 and f_2 express the amount of heat flux with time due a unit temperature difference. As shown in *Zhang et al.* [2011], the heat transfer functions f_1 and f_2 are approximately the same at the dimensionless time t_d = 2.8. Therefore, $t_d = 2.8$ is considered the critical dimensionless time to switch from f_1 to f_2.

During fluid injection and production, and as a result of the heat exchange processes, temperature changes continuously over time at any point within the wellbore and at the wellbore-formation interface. Based on superposition, the radial heat flux across each wellbore element to the surrounding formation is a time-convolution result of varying temperature. The discretized form at each time step can be expressed by the following:

$$q_{total} = \sum_{i=1}^{d-1} f(t_d - t_i) \cdot \Delta T(t_i) \tag{31}$$

Here, t_d represents the current time after d time steps, and t_i represents the time after i time steps; the function f is f_1 if $t_d - t_i \le 2.8$, and f_2 if $t_d - t_i > 2.8$. The temperature difference

$\Delta T(t_i)$ is the temperature in the well at time step *i*, minus the formation temperature at the interface at the previous time step, i.e., $\Delta T(t_i) = T_w(t_i) - T_f(t_{i-1})$.

To implement the solution into the TOUGH2 simulator, we need to calculate q_{total} and its derivative at each time step, and incorporate these two terms into the heat balance equation and the corresponding linearized form, which is needed for the implicit solution of the fully coupled system of mass and heat flow equations in the well. This requires the algorithm to store the temperature history for each wellbore element. This may be problematic if the time history becomes very long, which increases the computational demand of the time-convolution approach, and potentially reaches the limit of the computer's storage capacity. To mitigate this problem, a maximum number of time steps can be defined, beyond which the contributions from earlier temperature changes are lumped into a single term, which is calculated based on the time-weighted average of the individual temperature changes $\Delta T(t_i)$ and associated times t_i.

To make the algorithm flexible for handling various wellbore configurations and thermal conditions in the rock formation, the code gives the user an option to choose between uniform or depth-dependent formation properties, wellbore radii, and geothermal gradients.

The radial semi-analytical heat exchange model is currently incompatible with the option to perform semi-analytical linear heat exchange with confining beds (see `MOP(15)`=1 in *Pruess et al.* [1999]). There are two options (selected by `MOP(15)`=5 or 6, respectively) to invoke radial heat exchange:

Option 1 (`MOP(15)`=5): Constant Well and Formation Properties

Provide a material named `QLOSS` with the following parameters:

`DROK`: Rock grain density [kg/m^3] of formation near well
`POR`: Well radius [m]
`PER(1)`: Reference elevation [m]; specify Z coordinate in block `ELEME`, Columns 71–80
`PER(2)`: Reference temperature [°C]
`PER(3)`: Geothermal gradient [°C/m]
`CWET`: Heat conductivity near well [W/kg °C] of formation near well
`SPHT`: Rock grain specific heat [J/kg °C] of formation near well

Option 2 (`MOP(15)`=6): Variable Well and Formation Properties

Provide an external file named *radqloss.dat* with information in the following format:

First line: `NMATQLOSS`: number of elevations with geometric and thermal data
Provide `NMATQLOSS` lines with the following data in free format:

Elevation [m], well radius [m], initial temperature [°C], `CWET`, `DROK`, `SPHT`

Between elevations, properties are calculated using linear interpolation. Figure 12 shows and example input file using Option 1, for the simple heat injection into a single gridblock [*Zhang et al.*, 2011].

```
Semi-analytical radial heat exchange
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
WELL0          2650.      0.01    1.E-09    1.E-09    1.E-09      2.10     1000.
QLOSS          2650.      0.05    -.5000       20.      0.00      2.10     1000.

ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A1  1              10.7854E-020.3142E+00          0.2500E-01          -.5000E+00

CONNE----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8

MULTI----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    2    3    2    6
PARAM----1 MOP: 123456789*123456789*1234 ---*----5----*----6----*----7----*----8
   29999    999910000010000020054  1
             5.0e+10   1.0e+03
    1.e-05      1.e0                                    1.e-7
           1.013E+05             0.20e+2              0.0e+0

GENER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A1  1INJ 1                   12     HEATA
         0.0E7         1.0E7         2.0E7         3.0E7
         4.0E7         5.0E7         6.0E7         7.0E7
         8.0E7         9.0E7         1.0E8         5.e10
         8.e+1         2.e+1         8.e+1         2.e+1
         8.e+1         2.e+1         8.e+1         2.e+1
         8.e+1         2.e+1         8.e+1         2.e+1
         8.e+5         2.e+5         8.e+5         2.e+5
         8.e+5         2.e+5         8.e+5         2.e+5
         8.e+5         2.e+5         8.e+5         2.e+5

ENDCY----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
```

Figure 12. TOUGH2 input file for testing radial heat exchange option.

ACKNOWLEDGMENT

This work was supported, in part, by the U.S. Dept. of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

Carslaw, H.S. and J.C. Jaeger, 1959. *Conduction of Heat in Solids, 2nd edition*. Oxford University Press, Oxford, pp. 336

Doughty, C., *User's Guide for Hysteretic Capillary Pressure and Relative Permeability Functions in TOUGH2*, Report LBNL-6533E, Lawrence Berkeley National Laboratory, Berkeley, Calif., March 2013.

Deutsch, C.V., and A.G. Journel, *GSLIB, Geostatistical Software Library and User's Guide*, Oxford University Press, New York, New York, 1992.

Falta, R. W., K. Pruess, S. Finsterle, and A. Battistelli, *T2VOC User's Guide*, Report LBL-36400, Lawrence Berkeley Laboratory, Berkeley, Calif., 1995.

Finsterle, S., *Parallelization of iTOUGH2 Using PVM*, Report LBNL-42261, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1998.

Finsterle, S., *iTOUGH2 User's Guide,* Report LBNL-40401, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015a.

Finsterle, S., *iTOUGH2 Command Reference,* Report LBNL-40401 (Updated reprint), Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015b.

Finsterle, S., *iTOUGH2 Sample Problems*, Report LBNL-40402 (Updated reprint), Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015c.

Finsterle, S., and M.B. Kowalsky, *iTOUGH2-GSLIB User's Guide*, Report LBNL/PUB-3191, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2007.

Liu, H.-H., C. Doughty, and G. S. Bodvarsson, An active fracture model for unsaturated flow and transport in fractured rocks, *Water Resour. Res.*, *34*(10), 2633–2646, 1998.

Luckner, L., M. Th. van Genuchten, and D. Nielsen, A consistent set of parametric models for the two-phase flow of immiscible fluids in the subsurface, *Water Resour. Res.*, *25*(10), 2187–2193, 1989.

Moridis, G.J., and K. Pruess, *T2SOLV: An Enhanced Package of Solvers for the TOUGH2 Family of Reservoir Simulation Codes*, Report LBNL-40933, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1997.

Pruess, K., *TOUGH User's Guide*, Report NUREG/CR-4645, Nuclear Regulatory Commission (also Report LBL-20700, Lawrence Berkeley Laboratory, Berkeley, Calif.), 1987.

Pruess, K., *TOUGH2—A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow*, Report LBL-29400, Lawrence Berkeley Laboratory, Berkeley, Calif., 1991.

Pruess, K., C. Oldenburg, and G. Moridis, *TOUGH2 User's Guide, Version 2.0*, Report LBNL-43134, Lawrence Berkeley Laboratory, Berkeley, Calif., 1999.

Pruess, K. and A. Battistelli, *TMVOC, A Numerical Simulator for Three-Phase Non-isothermal Flows of Multicomponent Hydrocarbon Mixtures in Saturated-Unsaturated Heterogeneous Media*, Report LBNL-49375, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2002.

Pruess, K., ECO2N: A TOUGH2 Fluid Property Module for Mixtures of Water, NaCl, and CO_2, Report LBNL-57952, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2005.

Pruess, K., *ECO2M: A TOUGH2 Fluid Property Module for Mixtures of Water, NaCl, and CO_2, Including Super- and Sub-Critical Conditions, and Phase Change Between Liquid and Gaseous CO_2*, Report LBNL-4590E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2011.

Pruess, K., C. Oldenburg, and G. Moridis, *TOUGH2 User's Guide, Version 2.1*, Report LBNL-43134, Lawrence Berkeley Laboratory, Berkeley, Calif., 1999.

Weill S., E. Mouche, and J. Patin, A generalized Richards equation for surface/subsurface flow modeling, *J. Hydrol., 366*, 9–20, 2009.

Zhang, Y., L. Pan, K. Pruess, and S. Finsterle, A time-convolution approach for modeling heat exchange between a wellbore and surrounding formation, *Geothermics*, *40*(4), 251–266, doi:10.1016/j.geothermics.2011.08.003, 2011.