LBNL-7016E

Enhancements to the TOUGH2 Simulator Integrated in iTOUGH2

Stefan Finsterle

Energy Geosciences Division
Lawrence Berkeley National Laboratory
University of California
Berkeley, CA 94720

Finsterle GeoConsulting
Kensington, CA 94708

December 8, 2017
iTOUGH2 V7.1.1

This work was supported, in part, 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. The manuals describing iTOUGH2's capabilities are listed in Table 1.

The core of iTOUGH2 contains slightly modified versions of TOUGH2 modules as documented in *Pruess et al*. [2012] and other reports (see Table 2 below).

Most code modifications are editorial and do not affect the simulation results. Thus, standard TOUGH2 input files can be used in iTOUGH2, and consistent results are obtained if iTOUGH2 is run in forward mode. However, a number of modifications have been made to the version of TOUGH2 that is implemented in iTOUGH2. They enhance the functionality, flexibility, and ease-of-use of the forward simulator. Some of these enhancements are described in this report; more significant developments are documented in separate reports. Table 2 lists manuals describing general aspects of TOUGH2 modules or specific features of the forward simulator. (Note that not all capabilities mentioned in Table 2 are publicly released.)

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 is mainly concerned with issues (i) and (ii).

Table 1. Documentation of iTOUGH2

Module / Feature / Capability	**Reference**	**Source Files**
iTOUGH2 user's guide; theoretical background	*Finsterle*, 2015a	-
iTOUGH2 command reference	*Finsterle*, 2015b	-
iTOUGH2 sample problems	*Finsterle*, 2015c	-
Parallel execution of TOUGH2 runs	*Finsterle*, 1998	it2pvm.f
Geostatistical simulations	*Finsterle and Kowalsky*, 2007	it2gslib.f
Link to external models using PEST protocol	*Finsterle*, 2010	it2pest.f mio.f90
Global sensitivity and data-worth analysis	*Wainwright and Finsterle*, 2015	it2sa.f

Table 2. Documentation of TOUGH2 and Related Modules

Module / Feature / Capability	**Reference**	**Source Files**
Main TOUGH2 reference; general description of code capabilities, mathematical model, numerical scheme, input formats, and sample problems	*Pruess et al.*, 2012	eos1.f eos2.f eos3.f eos4.f eos5.f eos7.f eos7r.f eos8.f eos9.f ewasg.f
Three-phase (gas, aqueous, NAPL), three-component (water, air, VOC) module T2VOC	*Falta et al.,* 1995	eost2voc.f
Three-phase (gas, aqueous, NAPL), multi-component (water, NCGs, VOCs) module TMVOC	*Pruess and Battistelli*, 2002	eostmvoc.f
Three-phase (CO_2-rich phase, aqueous, solid), three-component (water, CO_2, NaCl) module ECO2N	*Pruess*, 2005	eco2n.f
Four-phase (gaseous and liquid CO_2, aqueous, solid), three-component (water, CO_2, NaCl) module ECO2M	*Pruess*, 2011	eco2m.f
Two-phase (gas, aqueous), five-component (water, brine, CO_2 or N_2, tracer, CH_4) module EOS7C	*Oldenburg et al.*, 2004	eos7c.f
Two-phase (steam, liquid), two-component ($water_1$, $water_2$) supercritical module EOS1sc	*Magnúsdóttir and Finsterle*, 2015	eos1sc.f
Two-phase (gas, liquid), two-component (non-Newtonian fluid, air) module EOS3nn	*Wu et al*. 2002	eos3nn.f
Two-phase (gas, liquid), three-component (water, air, miscible, solidifying gel) module EOS11	*Finsterle et al.*, 1994	eos11.f
Non-Darcy flow based on Forchheimer equation	*Finsterle and Witherspoon*, 2001	-
Iterative linear equation solvers	*Moridis and Pruess*, 1997	t2cg22.f t2solv.f
Hysteretic relative permeability and capillary pressure functions	*Doughty*, 2013	it2hyster.f
Wellbore simulator	*Guðmundsdóttir et al.*, 2017	flowell.f
Geostatistical simulations	*Finsterle and Kowalsky*, 2007	it2gslib.f
Two-phase (steam, liquid) flow of water and multiple tracers, module EOS1nT	*Finsterle*, 2017	eos1nt.f

2 Program Options and Control Parameters

2.1 More Options (Block MOMOP)

Table 3 describes additional options invoked by integer flags on a line following keyword MOMOP.

Table 3. More Options

MOP2	Value	Description
1		*Minimum number of Newton-Raphson iterations*
	0, 1	Allow convergence in a single Newton-Raphson iteration
	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 simulations.
	4	Allow convergence in a single Newton-Raphson iteration for positive simulation times, but require at least two for negative times.
2		*Length of element names (default: 5 characters)*
	5–9	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 in block* `GENER`
	0	Generation times ignored
	≥1	Time steps adjusted to match generation times
4		*Vapor pressure lowering*
	0	No vapor pressure lowering
	1	Vapor pressure lowering according to Kelvin's equation, $P_v = P_{sat}$ exp $[P_c M_w/(\rho_l RT)]$
	2	Reduces vapor pressure for $S_l < 0.02$ to zero at $S_l = 0.0$ toprevent liquid disappearance by evaporation
	3	Combines options 1 and 2
5		*Active Fracture Model (see Section 8)*
	0	Active Fracture Model applied to liquid phase only
	≥1	Active Fracture Model applied to all phases
6		*Leverett scaling of capillary pressure if element-specific element-specific permeabilities (see Section 5)*
	0	No Leverett scaling
	1	Rescale capillary pressure: $P_c = P_{c,ref}\sqrt{k_{ref}/k}$
	2	Rescale capillary pressure: $P_c = P_{c,ref}\sqrt{(k_{ref} \cdot \phi_i)/(k \cdot \phi_{ref})}$
7		*Zero nodal distance*
	0	Take absolute permeability from other element
	≥1	Take absolute and relative permeability from other element
8		*Version of sink/source subroutine (default depends on EOS module)*
	0	Take TOUGH2 V2 if available for given EOS module
	1	Take TOUGH2 version of subroutine QU
	2	Take TOUGH2 V2 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
	≥1	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
	≥1	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
	≥1	Potential energy included in enthalpy of all phases [*Stauffer et al*., 2014]

13		*Adjustment of Newton-Raphson increment weighting*
	0	No adjustment
	≥1	Reduce *WNR* by *MOP2*(13) percent if Newton-Raphson iterations oscillate and time step is reduced because *ITER* = *NOITE*
14		*Print input file to the end of output file*
	0	Do not reprint input files
	1	Print TOUGH2 input file to the end of TOUGH2 output file
	2	Print iTOUGH2 input file to the end of iTOUGH2 output file
	3	Print TOUGH2 and iTOUGH2 input files to the end of respective output files
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.
	≥1	Use porosity from block INCON; this assumes that these porosities were not the result from a pore compressibility/expansivity calculation; changes in rock-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		*Primary variable update*
	0	Update all primary variables
	≥1	Do not update primary variable if scaled increment is smaller than $10^{(-MOP2(19)-5)}$
20		*Reading anisotropic permeability modifiers in block* ELEME (*see also block* MAPPI *(Section 5.2) for reading and mapping permeabilities provided on external files)*
	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
21		*Honoring generation times in blocks* TIMBC *(see Section 6.1) and* LOAD *(see Section 11)*
	0	Time stepping ignores times specified in blocks TIMBC and LOAD
	≥1	Time steps adjusted to match times TIMBC and LOAD

22		*Format for reading coordinates in block* `ELEME`
	0	Read coordinates in format 3E10.4 from columns 51–80
	≥1	Read coordinates in format 3E20.14 from columns 51–110
23		*Makes choice of 5- or 9-point mesh when using* `MESHM`
	0	5-point mesh
	1	9-point mesh; use only with rotation angle `DEG` = 0 or 90
	2	9-point mesh; symmetry at I=1
24		*Format (text or binary) of* MESH *input and output files*
	0	Input: text Output: text
	1	Input: binary Output: binary
	2	Input: text Output: binary
	3	Input: binary Output: text
25		*Check mesh for duplicate element names*
	0	Do not check the mesh
	1	Check the mesh
26		*Printout format (see block* `OUTPU`*, Section 3.1)*
	0	Traditional output format (add 3 to the following options to print in both traditional and CSV formats)
	1	CSV/TECPLOT/GNU format (types separated; times combined)
	2	CSV/TECPLOT/GNU format (types separated; times separated)
	3	CSV/TECPLOT/GNU format (types combined; times combined)
27		*Element naming in* `MESHM`
	0	Traditional element naming scheme
	1	Element name is equal to consecutive element number
28		*Scaling of capillary pressure by temperature-dependent surface tension*
	0	No scaling
	1	$P_c(T) = P_c(20) \cdot [\sigma(T)/\sigma(20)]$
29		*Phase transition window (for certain EOSs only)*
	0	Hairtrigger phase transition
	≥1	Finite-window phase transition; window size: $10^{-MOP2(29)}$
30		*Switch from hairtrigger to finite-window phase transition (only if* `MOP2(29)` *> 0)*
	0	Always use finite-window phase transition
	≥1	Switch from hairtrigger to finite-window phase transition if *ITER* ≥ `MOP2(30)`
31		*Reduce convergence criterion* `RE1` *for positive simulation times*
	0	Keep `RE1` constant
	≥1	Reduce `RE1` by `MOP2(31)` orders of magnitude when simulation time transitions from negative to positive (useful when running a steady-state simulation for negative times)
32		(reserved)
33		(reserved)

34		*Convergence criterion based on maximum primary variable increment*
	0	Convergence criterion *RE3* not engaged
	≥1	Newton-Raphson iterations converge if maximum scaled primary variable increment is less than *RE3* $= 10^{(-MOP2(34)-5)}$
35		*Apply pathway dilation, i.e., permeability as a function of depth- and saturation-dependent threshold pressure; set IRP to negative value to read additional line of parameters, see subroutine PATHWAYDILATION*
	0	Do not apply pathway dilation
	1	Do apply pathway dilation
	2	Do apply pathway dilation and permeability adjustment due to swelling/shrinking of clays
36		*Strategies to recover from NaNs in linear solve*
	0	Do not apply NaN strategy
	1	Write and read SAVE file when a NaN appears during linear solve
	≥2	Write and read SAVE file when a NaN appears during linear solve, and perturb primary variables with random Gaussian noise with a standard deviation of $10^{(-MOP2(36)-5)}$
37		*Control of full printout (saves CPU time and disk space if running in parallel)*
	0	Print out according to option in block PARAM
	1	Do not print out full output if evaluating Jacobian
	2	Do not print out full output if running on a child process
38		*Initial temperature of confining unit for semi-analytical heat exchange (see Section 10)*
	0	Initial temperature of last element (uniform)
	≥1	Initial temperatures of adjacent elements (non-uniform)
39	1	(reserved)
40		*Set variable ZERO used to initialize primary variables after phase change*
	0	*ZERO* $= 10^{-6}$
	≥1	*ZERO* $= 10^{-MOP2(40)}$

2.2 Maximum Number of Time Steps (Variable *MCYC*)

Variable *MCYC* in block PARAM.1 sets the maximum number of time steps. If *MCYC* = 9999, the maximum number of time steps is set to 10^6. If *MCYC* is negative, the maximum number of time steps is set to 2^{-MCYC}.

2.3 Array Dimensions (Block DIMEN)

Memory for (almost) all arrays is dynamically allocated, with the required array dimensions automatically determined by iTOUGH2. If special options are invoked (e.g., region definitions in blocks ROCKS and GENER, see Sections 5.3 and 6.10, respectively), the required array sizes cannot be automatically determined. In these cases, the user has to provide the maximum number of materials and/or sinks/sources in a data block DIMEN. User-provided array dimensions may also be required for arrays related to the number of printout times specified in block TIMES. An unambiguous error message will be issued, instructing the user to provide array dimensions as needed. Only dimensions specifically requested by iTOUGH2 need to be specified; others may be left blank, in which case iTOUGH2 determines the maximum dimension internally. Figure 1 shows an example of block DIMEN. Block DIMEN has to come after the title and before any other TOUGH2 input block. (Dimenions for iTOUGH2-related arrays are provided in the iTOUGH2 input file using first-level command > DIMENSIONS.)

```
DIMEN----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
        20     10000       500       100      1000      1000      1000     10000
        10       100        50        20     10000
         ^         ^         ^         ^         ^         ^         ^         ^
    MAXMAT   MAXTIME     MAXSS MAXGENTAB     MAXDX     MAXDY     MAXDZ    MAXRAD
MAXPROPVST MAXTIMVST MAXREGION MAXREGPAR   MAXLTAB

MAXMAT        : Maximum number of materials in block ROCKS
MAXTIME       : Maximum number of times in TOUGH2 block TIMES and iTOUGH2 block TIME)
MAXSS         : Maximum number of sinks/sources in GENER
MAXGENTAB     : Maximum number of time-dependent rates in GENER
MAXDX         : Maximum number of grid increments in X direction (for XYZ MESHM)
MAXDY         : Maximum number of grid increments in Y direction (for XYZ MESHM)
MAXDZ         : Maximum number of grid increments in Z direction (for XYZ MESHM)
MAXRAD        : Maximum number of grid increments in radial direction (for RZ2D MESHM)
MAXPROPVST    : Maximum number of time-dependent properties
MAXTIMPROPVST : Maximum number of times when properties change
MAXREGION     : Maximum number of regions
MAXREGPAR     : Maximum number of parameters defining region
MAXLTAB       : Maximum number of time-dependent generation rates (if LTAB="COUNT")
```

Figure 1. Block DIMEN for specifiying maximum array dimensions.

3 Printout Options

3.1 Printout Format (Block OUTPU)

The standard TOUGH2 output consists of blocks of predefined element-, connection-, and generation-related output variables. The amount and frequency of printout depends on variables *KDATA* (see Section 3.2) and *MCYPR* as well as the times specified in block TIMES (in iTOUGH2, printout frequence also depends on the calibration times specified in the iTOUGH2 input file).

Visualization of simulation results generally requires a reformatting program (such as EXT), which parses the standard output file along with spatial information provided in the MESH file and then generates a plot file in a format suitable for visualization (e.g., specifically TECPLOT and PetraSim).

The purpose of the enw input block OUTPU is (1) to allow the user to select the output variables to be printed, and (2) to provide output in formats that can either directly be used by standard visualization software, or that is more ease to parse by post-processing programs. It should be noted that while EXT calculates the components flow vectors at the centers of grid-blocks, in the current version of the OUTPU feature, flow rates and velocities are simply printed for each connection.

The line following the data block identifier OUTPU contains a keyword indicating the desired output format, currently either CSV, TECPLOT (or VISIT), PETRASIM or GNU. The number of output variables to be printed, MOUTVAR, is given on the next line, followed by MOUTVAR lines, each containing a keyword and either none, one, or two interger values (read in free format anywhere after Column 20) for further specification of the output variable. Table 4 summarizes the available keywords and options. The output variables will be written to separate files (one for each type and/or time). Alternatively, they can be combined according to the option selected by variable MOP2(26) (see Table 3). An example OUTPU block is shown in Figure 2; excerpts of resulting output files for element-, connection- and generation-related output variables are shown in Figure 3, Figure 4, and Figure 5, respectively.

The variables identified in block OUTPU are also used for printing out time series data (at each time step) at elements, connectsions, and sinks/sources according to TOUGH blocks FOFT, COFT, and GOFT. If no block OUTPU is specified, the default variable sets ISET=1, ISET=2, and ISET=3 (see Table 4) will be printed for element-, connection-, and generation-related time series, respectively.

Block OUTPU must be specified *after* block MULTI.

Table 4. Keywords of Block OUTPU and Related Output Variables

Keyword	*ID1*	*ID2*	Output Variable
SET	ISET	–	Prints predefined sets of output variables ISET = 1 : Set of main element-related output variables ISET = 2 : Set of main connection-related output variables ISET = 3 : Set of main generation-related output variables
NO COMMA	–	–	Omit commas between values
NO QUOTES	–	–	Omit quotes around values
NO NAME	–	–	Omit element names
COORDINATE COORD	NXYZ	–	Grid-block or connection coordinates; mesh dimension and orientation are automatically determined, or can be specified through variable NXYZ: NXYZ = 1 : Mesh is 1D "X " NXYZ = 2 : Mesh is 1D " Y " NXYZ = 3 : Mesh is 1D " Z" NXYZ = 4 : Mesh is 2D "XY " NXYZ = 5 : Mesh is 2D "X Z" NXYZ = 6 : Mesh is 2D " YZ" NXYZ = 7 : Mesh is 3D "XYZ"
INDEX	–	–	Index (running number) of elements, connections, or sinks/sources
TIME	–	–	Simulation time
KCYC, ITERATION	–	–	Time step index
DOMAIN	X_{min} X_{max} Y_{min} Y_{max} Z_{min} Z_{max}	–	Only print out if element is within bounding box (provide vertices of box on one line, starting after Column 18)
Element-related output variables			
MATERIAL ROCK TYPE	–	–	Material number
MATERIAL NAME ROCK NAME ROCK TYPE NAME	–	–	Material name
ABSOLUTE PERMEABILITY	ISOT	–	Absolute permeability in directon ISOT; if ISOT = 0, permeabilities related to all three directions are printed
POROSITY	–	–	Porosity
TEMPERATURE	–	–	Temperature
PRESSURE	IPH#	–	Pressure of phase IPH
SATURATION	IPH#	–	Saturation of phase IPH

RELATIVE	IPH#	–	Relative permeability of phase IPH
VISCOSITY	IPH#	–	Viscosity of phase IPH
DENSITY	IPH#	–	Density of phase IPH
ENTHALPY	IPH#	–	Enthalpy of phase IPH
MASS FRACTION	IPH	IC	Mass fraction of component IC in phase IPH
DIFFUSION1	IPH#	–	Diffusion parameter group 1 ($\phi\tau_0\tau_\beta\rho_\beta$) of phase β = IPH
DIFFUSION2	IPH#	–	Diffusion parameter group 2 ($\frac{P_0}{P}\left[\frac{T+273.15}{273.15}\right]^\theta$) of phase β = IPH
PSAT	–	–	Saturated vapor pressure
PRIMARY	IPV	–	Primary variable No. IPV; if IPV = 0, all NK+1 primary variables are printed
SECONDARY	IPH#	ISP	Secondary parameter No. ISP related to phase IPH, where ISP = 0 : All secondary parameters ISP = 1 : Phase saturation ISP = 2 : Relative permeability ISP = 3 : Viscosity ISP = 4 : Density ISP = 5 : Enthalpy ISP = 6 : Capillary pressure ISP = 7 : Diffusion parameter group 1 ISP = 8 : Diffusion parameter group 2
SUBSIDANCE	–	–	One-dimensional deformation, see Section 11
Connection-related output variables			
TOTAL FLOW TOTAL FLOW RATE	–	–	Total flow rate
FLOW RATE FLOW RATE	IPH	IC	Advective flow rate. IPH=0; IC=0: Total flow IPH>0; IC=0: Flow of phase IPH IPH<0; IC=0: Flow of each phase IPH>0; IC>0: Flow of component IC in phase IPV IPH>0; IC<0: The flow of each component in phase IPH
DIFFUSIVE FLOW	IPH	IC	Diffusive flow rate. IPH=0; IC=0: Total flow IPH>0; IC=0: Flow of phase IPH IPH<0; IC=0: Flow of each phase IPH>0; IC>0: Flow of component IC

			in phase IPV IPH>0; IC<0: The flow of each component in phase IPH
HEAT FLOW	–	–	Heat flow rate
VELOCITY	IPH#	–	Flow velocity of phase IPH
Generation-related output variables			
GENERATION GENEARTION RATE	–	–	Production or injection rate
FLOWING ENTHALPY	–	–	Flowing enthalpy
FRACTIONAL FLOW	IPH#	–	Fractional flow of phase IPH (production only)
WELLBORE PRESSURE	–	–	Wellbore pressure (wells on deliverability only)
# If IPH = 0, the output variables of all phases are printed			

```
OUTPU----1----*----2----*----3----*----4----*----5----*----6
CSV
9
PRESSURE
SECONDARY           1 3
SECONDARY           1 4
MASS FRACTION       2 2
FLOW                1
VELOCITY            1
GENERATION
FRACTIONAL FLOW     1
FLOWING ENTHALPY
```

Figure 2. Example data block OUTPU.

```
"              ELEM","          PRES_GAS","          VISC_GAS","          DENS_GAS","          X_AIR_LIQ"
"                 ","               (PA)","            (PA*S)","         (KG/M**3)","                (-)"
"TIME [sec]  0.10000000E+01"
"              A1101",  0.401236819721E+06,  0.184570692607E-04,  0.465919882449E+01,  0.644956432615E-04
"              A2101",  0.147507792677E+06,  0.184570734485E-04,  0.171287344952E+01,  0.237110757024E-04
"              A3101",  0.116720218365E+06,  0.184570720594E-04,  0.135536558761E+01,  0.187621756067E-04
"              A4101",  0.113034689131E+06,  0.184570711567E-04,  0.131256899168E+01,  0.181697497798E-04
"              A5101",  0.112593498390E+06,  0.184570707619E-04,  0.130744588788E+01,  0.180988311033E-04
"              A6101",  0.112540683073E+06,  0.184570706029E-04,  0.130683260888E+01,  0.180903413682E-04
"              A7101",  0.112534360176E+06,  0.184570705403E-04,  0.130675919361E+01,  0.180893250018E-04
"              A8101",  0.112533603089E+06,  0.184570705159E-04,  0.130675040489E+01,  0.180892033046E-04
...
"TIME [sec]  0.30000000E+01"
"              A1101",  0.400995615118E+06,  0.184570681555E-04,  0.465639836140E+01,  0.644568724749E-04
"              A2101",  0.193338269214E+06,  0.184570735777E-04,  0.224506094596E+01,  0.310779891137E-04
"              A3101",  0.131359741970E+06,  0.184570735571E-04,  0.152536085210E+01,  0.211153850575E-04
"              A4101",  0.116601086798E+06,  0.184570724951E-04,  0.135398217228E+01,  0.187430259548E-04
"              A5101",  0.113380367818E+06,  0.184570716453E-04,  0.131658299037E+01,  0.182253154829E-04
"              A6101",  0.112706320690E+06,  0.184570711235E-04,  0.130875595137E+01,  0.181169665889E-04
...
```

Figure 3. Element-related output variables in CSV format based on `OUTPU` block shown in **Figure 2**.

```
"              ELEM1","              ELEM2","          FLOW_GAS","           VEL_GAS"
"                 ","                 ","            (KG/S)","             (M/S)"
"TIME [sec]  0.10000000E+01"
"              A1101","              A2101", -0.352311719350E-08, -0.150029808352E-05
"              A2101","              A3101", -0.422441958852E-09, -0.322959016651E-04
"              A3101","              A4101", -0.504964190656E-10, -0.487876354994E-05
"              A4101","              A5101", -0.604382074745E-11, -0.602969130852E-06
"              A5101","              A6101", -0.723495860628E-12, -0.724632771562E-07
"              A6101","              A7101", -0.866146117386E-13, -0.867914300296E-08
"              A7101","              A8101", -0.103710092602E-13, -0.103927648693E-08
"              A8101","              A9101", -0.124248446811E-14, -0.124509924289E-09
...
"TIME [sec]  0.30000000E+01"
"              A1101","              A2101", -0.288336365213E-08, -0.122860138353E-05
"              A2101","              A3101", -0.852256531812E-09, -0.497104674654E-04
"              A3101","              A4101", -0.202352019394E-09, -0.173716307947E-04
"              A4101","              A5101", -0.441276789154E-10, -0.426779727997E-05
...
```

Figure 4. Connection-related output variables in CSV format based on `OUTPU` block shown in **Figure 2**.

```
"              ELEM","            SOURCE","               GEN","            FF_GAS","              ENTG"
"                 ","                 ","             (KG/S)","               (-)","            (J/KG)"
"TIME [sec]  0.10000000E+01"
"              AQ102","              TOP 2", -0.162800000000E-10,  0.100000000000E+01,  0.105761536312E+06
"              AQ103","              TOP 3", -0.193900000000E-10,  0.100000000000E+01,  0.105761536231E+06
"TIME [sec]  0.30000000E+01"
"              AQ102","              TOP 2", -0.162800000000E-10,  0.100000000000E+01,  0.105761535951E+06
"              AQ103","              TOP 3", -0.193900000000E-10,  0.100000000000E+01,  0.105761535806E+06
...
```

Figure 5. Generation-related output variables in CSV format based on `OUTPU` block shown in **Figure 2**.

3.2 Printout Control (Variable *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 5), 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.
For *KDATA* = 4 or 5, select variables for, respectively, elements and connections, along with coordinates are printed in Tecplot format to files named **_E.tec* and **.C.tec*.

Table 5. Amount of Printout as a Function of Variable *KDATA*

-5	-4	-3	-2	-1	Printout of …	1	2	3	4	5
-	-	-	+	+	volume and mass balance	+	+	+	+	+
-	-	-	-	+	generation rates	+	+	+	+	+
-	-	-	-	-	element variables	+	+	+	+	+
-	-	-	-	-	connection variables	-	+	+	+	+
-	-	-	-	-	primary variables	-	-	+	+	+
-	-	-	-	-	elements Tecplot	-	-	-	+	+
-	-	-	-	-	connections Tecplot	-	-	-	-	+

3.3 Excluding Domains from Global Material Balance Printout (Variable *SPHT*)

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 Mesh Generation

4.1 Meshmaker Options

The following options have been added for more convenient generation of Cartesian and radial meshes using the built-in mesh generator MESHM:

- Meshmaker can generate meshes with element names consisting of 5 (default), 6, 7, 8, and 9 characters. The length of the element name (variable IELL) can be indicated in Column 6, immediately after keyword MESHM, or by setting MOP2(2) = IELL (see Table 3). The first three characters of the element name are alphanumeric; the remaining (IELL-3) positions of the element name must be integer numbers.
- If an asterisks ("*") is specified in Column 7 on the line that contains keyword MESHM, lists of grid increments (for Cartesian meshes), radial distances and layer thicknesses (for radial meshes) can be specified in free format (rather than format E10.4).
- If MOP2(27) is set to 1, or if a plus sign ("+") or "C" is specified in Column 8 on the line that contains keyword MESHM, the generated element names are indentical to the element index, i.e., integers from 1 for the first element to NEL for the last element.
- For Cartesian meshes, the coordinates can be shifted by ΔX, ΔY, and ΔZ provided in Columns 11–20, 21–30, and 31–40 of record XYZ.1 (i.e., after variable DEG).
- For radial meshes (keywords RZ2D or RZ2DL), a material number can be specified in Columns 6–10 on the line following keywors RADII, EQUID, LOGAR. The corresponding material name will be added to the corresponding elements, allowing, for example, to directly indicate whether the elements represent the well, skin zone, or formation.
- If mesh information (i.e., blocks ELEME and CONNE) are specified in the input file, they will be written to file MESH in the same format, i.e., without a potential loss of accuracy due to the limitation imposed by the E10.4 format.
- The input and/or output MESH files can be either ASCII text fiels (default) or binary files (see Table 3, MOP2(25)).

4.2 Secondary Mesh (Blocks 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 6 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 6. Primary and secondary mesh generation.

5 Material Property Assignment

5.1 Element-by-Element Permeabilities (Block ELEME)

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-element permabilities are ignored. 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 anisotropy ratio as given by ratios of *PER(1)*, *PER(2)*, and *PER(3)* in block ROCKS is preserved, i.e., the single permeability modifier is applied to all three permeabilities. To read three permeabilities or permeability modifiers, i.e., one for each of the three directions, *MOP2*(20) has to be set according to Table 3. (Permeabilities and permeability modifiers can also be read from an external file and mapped onto the TOUGH2 mesh; see block MAPPI described in Section 5.2). Note that the permeability or permeability modifiers are stored in array *USERX(i,N)*, *i*=1…3. 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 Element-by-Element Permeabilities from External Files (Block MAPPI)

Instead of specifying element-by-element permeabilities or permeability modifiers in block ELEME as described in Section 5.1, heterogeneous and anisotropic, potentially transformed permeabilities or permeability modifiers may be generated externally and provided on a file, which is read in and then mapped onto the TOUGH2 mesh. Various file formats as well as mapping procedures are supported; the related information has to be provided in block MAPPI, which contains the following records (read in free format):

Record MAPPI.1

FILENAME : Filename containing (potentially transformed) permeabilities in one of the supported formats (see *MAPFMT*); make sure the file is available in the local directory

Record MAPPI.2

MAFMT : File format

0: GSLIB format as internally generated by iTOUGH2 [*Finsterle and Kowalsky*, 2007]

1: *VALUE*(i), $i = 1, \ldots,$ *MAPNVAL*
Values need to be provided in same order as elements in block ELEME

2: *X Y Z VALUE*(i), $i = 1, \ldots,$ *MAPNVAL*
Coordinates followed by values; the coordinates may or may not be identical to those in block ELEME

3: *I J K VALUE*(i), $i = 1, \ldots,$ *MAPNVAL*
Grid indices followed by values; assumes values are provided on a regular grid (see record MAPPI.6); the TOUGH2 grid does not need to be regular)

4: *I J K NFRACT COND*(i), $i = 1, \ldots,$ *MAPNVAL*
Format used by discrete fracture network to continuum grid mapper of *Parashar and Reeves* [2011]; assumes values are provided on a regular grid with uniform grid spacing (see record MAPPi.6); the TOUGH2 grid does not need to be regular

MAPHEADER : Number of header lines to be skipped before data reading

MAPNVAL : Number of values to be read per grid block (1 for single permeability or modifier applied to all three permeabilities; 3 for permeabilities or modifiers applied to different directions)

MAPPOINTS : Number of data lines (points in space where permeability information is provided)

Record MAPPI.3

IFIELD1 : Identifies variable

0: porosity, *PHI(N)*

1–3: permeabilities or permeability modifiers, *USERX(i,N)*

4: capillary-strength parameter (see Section 7) *USERX(4,N)*

5: user specified, *USERX(5,N)*

IFIELD2 : Interpret as value or modifier

0: value

1: modifier

IFIELD3 : Variable transformation

0: no transformation

1: value represents decadic logarithm

2: value represents natural logarithm

Record MAPPI.4

NMAPROCK : Number of material types to be given (see *IMAPROCK*).

Mapping can be restricted to certain zones within the TOUGH2 model, where a zone is defined by all elements that belong to one or more material domains. This option may be useful to exclude the properties of special elements (e.g., those representing boundary elements, wells, engineered structures, etc.) If *NMAPROCK* is zero, the heterogeneous values will be mapped onto the entire TOUGH2 grid.

IMAPROCK : Material type numbers (sequence number of material as entered in block ROCKS) of domains to be included in mapping, *IMAPROCK*(i), $i = 1, \ldots,$ *NMAPROCK*

Record MAPPI.5

MAPPING : Type of mapping and averaging procedure

1: nearest neighbor; the property value from the point nearest to the TOUGH2 element center point will be taken

2: arithmetic mean; the arithmetic mean of all property values within a user-specified radius from the TOUGH2 element center point will be taken

3: geometric mean; the geometric mean of all property values within a certain radius from the TOUGH2 element center point will be taken

4: harmonic mean: the harmonic mean of all property values within a certain radius from the TOUGH2 element center point will be taken

SEARCHR : Search radius; specifying a non-zero search radius for *MAPPING* = 1 may speed up the mapping, as the nearest point is considered identified as soon as the distance is less than the search radius. For *MAPPING* = 2, 3, or 4, a non-zero search radius has to be specified. Three options exist to specify the search radius r, where SR is an input parameter to be provided by the user, V is

the volume of the given TOUGH2 element, and D = 1, 2, or 3 is the dimension of the model:

$$r = \begin{cases} -SR \cdot V^{1/D} & SR < 0 \\ V^{1/D} & \text{for} \quad SR = 0 \\ SR & SR > 0 \end{cases}$$

Record `MAPPI.6` (only if ***MAPFMT*** = 3 or 4)

X0, Y0, Z0 : Origin of coordinate system used in the external file (the TOUGH2 mesh may be shifted by specifying ***SHIFTX***, ***SHIFTY***, and ***SHIFTZ*** in columns 51–60, 61–70, and 71–80, respectively, on the line containing keyword `ELEME`).

DX, DY, DZ : Grid spacings

During the mapping process, each element of the TOUGH2 grid is visited. First, it is checked whether the element belongs to a material type that defines the mapping zone. If so, the coordinates of the element are compared to the coordinates of grid points specified in the external file, and if the distance between the two points is less than the search radius, the property values are retained, averaged, and eventually assigned to the element.

Figure 7 shows an example of block `MAPPI` in a TOUGH2 input file; Figure 8 shows an excerpt of the data file *Log10PermModifier.dat*.

```
MAPPIng--1----*----2----*----3----*----4----*----5----*----6----*----7
Log10PermModifier.dat
    3    1    3       9600
    1    1    1
    3               1    4    5
    1   1.0E-03
   5.0  5.0 -28.75
  10.0 10.0   2.50
```

Figure 7. TOUGH2 input file for mapping permeability modifiers from external file.

```
X-index  Y-index  Z-index  log10(pmx1)  log10(pmx2)  log10(pmx3)
      1        1        1        -1.03        -1.14        -1.98
      1        1        2         0.65         0.83        -0.76
      1        1        3         2.98         2.22         1.56
    ...
     40       20       12        -1.71        -0.98        -2.83
```

Figure 8. Data file containing log-10 permeability modifiers on indexed regular mesh.

5.3 Permeability and Porosity Regions (Block ROCKS)

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 an ellipsoid, a rectangular box or cube, a cylinder or truncated cone, or a prism.

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

A cylindrical region is defined by the starting and end coordinates of its axis, and a radius.

The irregular base of the prism is defined by a polygon.

The ellipsoid, box and cube can be rotated. However, the axis of the irregular prism must be aligned with one of the coordinate axes.

The permeabilities and porosities of the elements within (or, aleternatively, outside) 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}:

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

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

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

$$\omega = 1 \quad (3a)$$

$$\omega(d) = 1 - d \quad (3b)$$

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

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

$$\omega(d) = \min(1, (1-d)/a) \quad (3e)$$

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 6). The effect of the influence 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:

`IREGGEOM`: Defines geometry of the region; set negative to also provide rotation angles (see Table 6) (not available for prism).
1 = Box
2 = Ellipsoid
3 = Cylinder
4 = Cube
5 = Polygonial prism (aligned with one of the coordinate axes)
6 = Right, circular, truncated cone

`IREGINFF`: 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)

NPREGPOLY: (only if *IPREGGEOM* = 5): Number of polygon points, *NP*

IJKPRISMAXIS: (only if *IREGGEOM* = 5): Alignment of polygonial prism axis
1 = prism axis aligned with *X*-axis
2 = prism axis aligned with *Y*-axis
3 = prism axis aligned with *Z*-axis

XREGION(*i*) (see Table 6)

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

Table 6. 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}	azimuth	dip	plunge
2 (ellipsoid)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	azimuth	dip	plunge
3 (cylinder)	X_B	Y_B	Z_B	X_E	Y_E	Z_E	R	-	-
4 (cube)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	azimuth	dip	plunge
5 (prism)	$Axis_{min}$ [@]	$Axis_{max}$ [@]	$C_{1,1}$ [%]	$C_{1,2}$ [%]	… [%]	… [%]	$C_{NP,1}$ [%]	$C_{NP,2}$ [%]	
6 (cone)	X_B	Y_B	Z_B	X_E	Y_E	Z_E	R_B	R_E	-

[@]: Minimum and maximum coordinate of prism axis, i.e., location of prism bases; coordinate variable given by *IJKPRISMAXIS*

[%]: Coordinates for *NP* polygon points (clockwise or anti-clockwise); two coordinates per point: (*Y*,*Z*) for *IJKPRISMAXIS* = 1; (X,*Z*) for *IJKPRISMAXIS* = 2; (*X*,Y) for *IJKPRISMAXIS* = 3

If the region is not aligned with the coordinate axes, set *IREGGEOM* negative and provide three correction angles (azimuth, dip, and plunge). A polygonial prism must be aligned with one of the prism axes. Figure 9 shows an input file that uses a rotated ellipsoidal region to represent an inclined fault. In addition, a prismatic engineered structure is generated by indicating the start and end of the prism axis in X-diection, and by defining the prism base using an 11-point polygon. Figure 10 shows how region definitions are reported in the TOUGH2 output file. The resulting permeability distribution is visualized in Figure 11. Note that the order in which regions are defined affects the final property distribution if regions ioverlap or intersect each other (later regions modifying the previously defined regions).

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.

```
Ellipsoidal, prismatic, and cylindrical property regions
DIMEN----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
         5
MESHM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
XYZ

NX     100       0.2
NY     100       0.2
NZ     100       0.2



ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
GRANI    0    2670.0      0.01  1.00E-16                                  800.0
XYZFR    0    2670.0      0.40  1.00E-10                                  800.0
   -2    2
      10.0      11.9      -8.2
       1.0      10.0      22.0
     130.0     -30.0      65.0
XYZBE    0    2670.0      0.40  1.00E-20                                  800.0
    5    4   11    1
      -1.0      17.2
       5.0     -15.0
       5.0     -10.0
       7.0      -8.0
      10.0      -7.0
      13.0      -8.0
      15.0     -10.0
      15.0     -15.0
      11.0     -15.0
      11.0     -12.0
       9.0     -12.0
       9.0     -15.0
     0.98
XYZHE    0    4000.0      0.01  1.00E-09                                  800.0
    3    0
      -1.0      10.0     -10.5
       2.5      10.0     -10.5
       1.5

OUTPU----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
TECPLOT
3
COORD
PERMEABILITY          1
POROSITY

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
 8 3   1       11000009000010004000050000
        0.                 1.0
     1.E-5
 1.1000000000000E+05
MOMOP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
                    1
ENDCY----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
```

Figure 9. TOUGH2 input file for assigning ellipsoidal, prismatic, and cylindrical regions.

```
REGIONS: PERMEABILITY (K), SINK/SOURCE (S), OBSERVATION (O)
============================================================
REGION SKO  SOURCE    GEOMETRY     INFLUENCE  X-(S|C)   Y-(S|C)   Z-(S|C) X-END|DX/2 Y-END|DY/2 Z-END|DZ/2 RADIUS
           MATERIAL               FUNCTION
-------------------------------------------------------------------------------------------------------------------

     1  K   XYZFR   ELLIPSOID     SPHERICAL   10.000    11.900    -8.200      1.000     10.000     22.000
         AZIMUTH          DIP         PLUNGE  130.000   -30.000    65.000

     2  K   XYZBE        PRISM  CONST.-LIN.
           MIN AND MAX AXIS COORDINATES      -1.000    17.200
           POLYGON POINT                      5.000   -15.000
           POLYGON POINT                      5.000   -10.000
           POLYGON POINT                      7.000    -8.000
           POLYGON POINT                     10.000    -7.000
           POLYGON POINT                     13.000    -8.000
           POLYGON POINT                     15.000   -10.000
           POLYGON POINT                     15.000   -15.000
           POLYGON POINT                     11.000   -15.000
           POLYGON POINT                     11.000   -12.000
           POLYGON POINT                      9.000   -12.000
           POLYGON POINT                      9.000   -15.000
           INFLUENCE FUNCTION PARAMETER       0.980

     3  K   XYZHE     CYLINDER      CONSTANT   -1.000    10.000   -10.500      2.500     10.000    -10.500  1.500
-------------------------------------------------------------------------------------------------------------------
```

Figure 10. Excerpt from TOUGH2 output file, documenting region definitions.

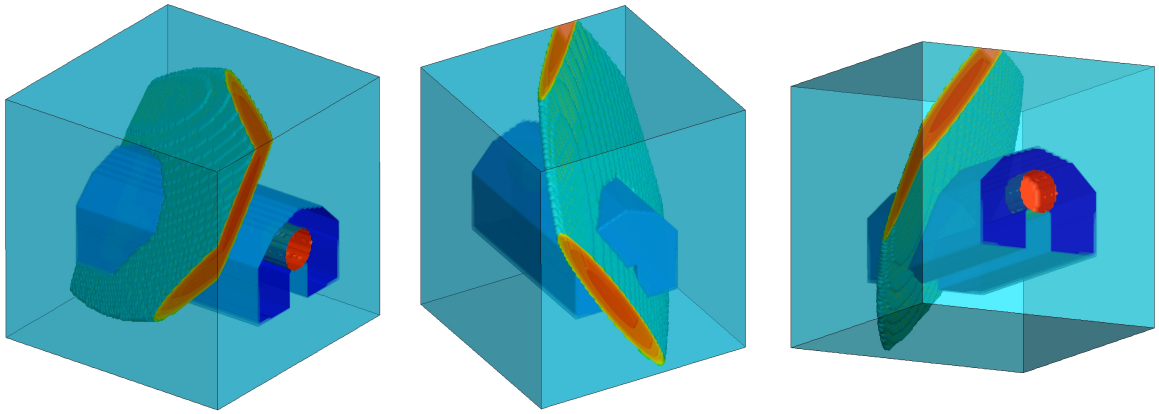

Figure 11. Permeability distribution generated by region assignment using the input file shown in **Figure 9**.

5.4 Time-Dependent Material Properties (Block ROCKS)

By default, the material properties specified in block ROCKS remain constant during a simulation. A noteable exception is porosity, which may change as a function of pressure and/or temperature provided that a non-zero pore compressibility and/or expansivity is specified.

All properties (except porosity) specified in block ROCKS can be made time-dependent. This option assumes that the material properties are not dynamically calculated as a function of the solution vector (i.e., the primary variables), but instead are known beforehand and thus can be prescribed as a function of time. Time-dependent property values must be provided in tabular form on separate files, one file for each material type for which at least one propery is time-dependent. The name of the file needs to be given at the end of record ROCKS.1, at any postion with a column number greater than 80. An example is shown in Figure 12, where the names of the files containing time-dependent material property values are given in Columns 81–91. Note that the properties of material LAY02 do not change with time.

Figure 13 shows the content of the user-supplied file with time-dependent material properties. The file contains a single header line identifying which input parameters are considered time-dependent. The heading must be identical to the variable names shown for the parameters in block ROCKS in Appendix E of the TOUGH2 manual [*Pruess et al*., 1999], i.e., DROK, PER(i) (or PERM, if all three permeabilities are updated to the same value), CWET, SPHT, COM, EXPAN, CDRY, TORTX, GK, FOC, XKD(i), RP(i), and CP(i). In addition, parameters PFT(i) and CFT(i) (see *Magnúsdóttir and Finsterle* [2015]) can be made time-dependent. Index i in the variable names has to be replaced by the appropriate integer to identify the desired array element. The header line must also contain the keyword TIME to identify the column holding time (in seconds). The first time must be equal or earlier than the simulation starting time TSTART, and the last time must be equal or later than the end time of the simulation.

Following the header line, an arbitrary number of lines with time-dependent property values can be given. In the example shown in Figure 13, all properties remain constant for the first day, after which permeability is decreasing (variable PERM is used to first change the permeabilities in all three directions, and PER(3) is used to then overwrite the permeability in the third direction), pore compressibility is increasing (e.g., due to some chemically or thermally induced degradation of rock strength), and associated changes in the pore size distribution and capillary-strength parameters of the van Genuchten curves.

As a simulation proceeds through time, property values as assigned after each time step to values that are linearly interpolated between the points provided in the table. Note that different time-dependent variables may be specified for different materials.

If a property is declared time-dependent, it is a user-specified quantity and thus cannot be estimated or updated in iTOUGH2 (potential approaches suitable for estimating time-dependent properties include the use iTOUGH2-PEST [*Finsterle*, 2010] the RESTART option of iTOUGH2, or defining and estimating appropriate user-specified parameters through subroutine USERPAR).

```
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8----*----9--
LAY01     2    2680.0       0.39  1.00E-12  1.00E-12  1.00E-13       2.0    1000.0 MAT1fT.dat
     1.E-9
   7            0.80       0.20      1.00      0.00
   7            0.80       0.15  6.40E-04  1.00E+08      1.00
LAY02     2    2680.0       0.22  2.62E-13  2.62E-13  2.62E-13       2.0    1000.0
     1.E-9
   7            0.90       0.25      1.00      0.00
   7            0.90       0.20  6.40E-04  1.00E+08      1.00
LAY03     2    2680.0       0.25  1.00E-13  1.00E-13  1.00E-14       2.0    1000.0 MAT3fT.dat
     1.E-9
   7            0.66       0.10      1.00      0.00
   7            0.66       0.08  7.38E-04  1.00E+08      1.00
```

Figure 12. ROCKS block with names of files containing time-dependent material properties.

```
      TIME      PERM    PER(3)       COM     RP(1)        CP(3)
         0  1.00E-12  1.00E-13  1.00e-09      0.80      6.40E-04
    864000  1.00E-12  1.00E-13  1.00e-09      0.80      6.40E-04
   8640000  1.00E-14  1.00E-15  5.00e-09      0.60      6.40E-05
```

Figure 13. Example file with time-dependent material properties.

5.5 Geostatistics (Block GSLIB)

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 (Block `TIMBC`)

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. Block `TIMBC` must be specified *before* the optional block `LOAD` (see Section 11). An example is given in Figure 14.

```
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 14. TOUGH2 input file for assigning time-dependent Dirichlet boundary conditions.

6.2 Free-Drainage Boundary Condition (Material *`DRAIN`*)

A free drainage boundary condition for liquid flow can be implemented, in which gravity is the only driving force, i.e., (capillary) 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 (Subroutine *`USERBC`*)

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 15. 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 15. Subroutine USERBC for specifying time-dependent boundary conditions.

6.4 Tabular Input of Time-Dependent Rates (Block GENER)

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*. If *LTAB* = 'COUNT', the number of table entries is automatically counted. 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 format; it will be converted to seconds (relative to the first date given; the first and last dates in a time series can be set as "-infinity" and "infinity"; negative times in regular time units may also be specified without affecting the reference date). 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 16 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 16. TOUGH2 input format for specifying time-dependent generation rates in tabular format, and using conversion factors for time and rate.

6.5 Cycling Generation Rate Sequences (Block GENER)

In standard TOUGH2, the first time in any GENER block that defines time-dependent rates must be smaller than the starting time of the simulation, and the last time must be greater than the end time of the simualation. In iTOUGH2, starting and end times in the GENER block do not need to fulfill this requirement; instead, the following rates will be taken:

- For times smaller than the first time in GENER, the rate at the first time is taken.
- If the simulation time exceeds the last tabulated time (i.e., that with index *LTAB*), the sequence of rates specified in $F2(i)$ (and $F3(i)$), $i = 1, \ldots, LTAB$, is repeated, allowing to simulate repeated cycles by specifying a single cycle.

Figure 20 shows an example, in which, starting after 100 days, a 20-day injection-production cycle (with three-day waiting periods in between) is repeated until the end of the simulation.

```
----*----1----*----2----*----3----*----4----*----5----*----6----*----7
GENER T      86400.0
A5612INJ 1                    5     WATEe       5.0
  100.0     0.0  1E6
  100.05    5.0  1E6
  107.00    5.0  1E6
  107.05    0.0  1E6
  120.00    0.0  1E6
A7612PRO 1                    6     MASS       -4.0
   100.0    0.0
   110.00   0.0
   110.05  -5.0
   117.00  -5.0
   117.05   0.0
   120.00   0.0
```

Figure 17. TOUGH2 input format for specifying a cycling injection-production schedule.

6.6 Sinusoidal Generation (Block GENER)

Sinusoidal generation rates of the form

$$Q(t) = Q_0 + Q_A \cdot \sin\left(2\pi \cdot \frac{t_0 + t}{t_p}\right)$$

where t is simulation time, t_0 is the time lag, t_p is the time period (1/frequency), Q_0 is the reference rate, and Q_A is the rate amplitude. Sinusoidal generation rates can be modeled by specifying the following source term in block GENER:

(1) Define course code name "SIN**". If a material-related source is specified (see Section 6.9), define the source element name as "SIN**" and the source code name as "MAT*ii*", where *ii* is the sequence number of the material domain as entered in block ROCKS.

(2) Set LTAB = 2

(3) Provide time information in GENER.1.1, i.e., F1(1) = t_p, and F1(2) = t_0.

(4) Provide rate information in GENER.1.2, i.e., F2(1) = Q_A, and F2(2) = Q_0.

(5) If ITAB is not blank, provide enthalpy information in GENER.1.3, i.e., F3(1) = Q_A, and F3(2) = Q_0.

An example GENER block for the reinjection of produced wate, NaCl, and CO_2 is shown in Figure 19.

```
GENER----1----*----2----*----3----*----4----*----5----*----6----*----7
A11 1SIN 1                    2     WATE
       86400.0       43200.0
           2.0           5.0
SIN 2MAT 3                    2     WATE
         3600.0          0.0
           1.0           0.0
```

Figure 18. TOUGH2 input format for specifying sinusoidal generation rates.

6.7 Reinjection of Produced Fluid Components (Block GENER)

Circulation by reinjection of produced fluids is common practice in some environmental remediation technologies as well as geothermal and other applications. As the composition and enthalpy of the produced fluid mixture is a simulation result that depends on changing reservoir conditions, the reinjection rate cannot be specified a priori, but must be calculated internally. Fluid circulation can be modeled by specifying the following sink and source terms in block GENER:

(1) Define production wells, using type MASS followed by a mass production rate. All production wells need to be defined first, before injection wells are specified.

(2) Define reinjection wells using the course code name "CIR*ii*", where "*ii*" is the order number (i.e., sink/source entry in block GENER) of the production well to which the injection well is linked.

(3) For the reinjection well, specify (i) the component that shall be reinjected (Columns 36–39), (ii) the fraction of the produced component rate that shall be reinjected (Columns 41–50; e.g., specify a value less than 1.0 to represent fluid losses in the surface system), and (iii) an enthalpy if different from that of the produced fluid (Columns 51–60).

(4) Repeat Step (3) for all components that shall be reinjected, using the same element and source code name.

An example GENER block for the reinjection of produced water, NaCl, and CO_2 is shown in Figure 19.

```
GENER----1----*----2----*----3----*----4----*----5----*----6----*----7
A1110PRO 1                    1     MASS       -2.0
A11 1CIR 1                    1     COM3        1.0
A11 1CIR 1                    1     COM2        1.0
A11 1CIR 1                    1     WATE        1.0
```

Figure 19. TOUGH2 input format for specifying fluid reinjection.

6.8 Honoring Generation Times (Variable *MOP2(3)*)

By setting *MOP2*(3) = 1 in block MOMOP (see Figure 16), 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.

6.9 Material-Related Sinks/Sources (Block GENER)

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 are volume-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 20.

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

6.10 Sink/Source Regions (Block GENER)

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 an ellipsoid, a rectangular box or cube, a cylinder or truncated cone, or a prism.

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

A cylindrical region is defined by the starting and end coordinates of its axis, and a radius.

The irregular base of the prism is defined by a polygon.

The ellipsoid, box and cube can be rotated. However, the axis of the irregular prism must be aligned with one of the coordinate axes.

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 7). The effect of the influence 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`:

`IREGGEOM`: Defines geometry of the region set negative to also provide rotation angles (see Table 6) (not available for prism).
1 = Box
2 = Ellipsoid
3 = Cylinder
4 = Cube
5 = Polygonial prism (aligned with one of the coordinate axes)
6 = Right, circular, truncated cone

`IREGINFF`: 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)

`NPREGPOLY`: (only if *`IREGGEOM`* = 5): Number of polygon points, *`NP`*

`IJKPRISMAXIS`: (only if *`IREGGEOM`* = 5): Alignment of polygonial prism axis
1 = prism axis aligned with *X*-axis
2 = prism axis aligned with *Y*-axis
3 = prism axis aligned with *Z*-axis

`XREGION`(*`i`*) (see Table 7)

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

Table 7. 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}	azimuth	dip	plunge
2 (ellipsoid)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	azimuth	dip	plunge
3 (cylinder)	X_S	Y_S	Z_S	X_E	Y_E	Z_E	R	-	-
4 (cube)	X_c	Y_c	Z_c	L_X	L_Y	L_Z	azimuth	dip	plunge
5 (prism)	$Axis_{min}$ [@]	$Axis_{max}$ [@]	$C_{1,1}$ [%]	$C_{1,2}$ [%]	…[%]	…[%]	$C_{NP,1}$ [%]	$C_{NP,2}$ [%]	
6 (cone)	X_B	Y_B	Z_B	X_E	Y_E	Z_E	R_B	R_E	-

[@]: Minimum and maximum coordinate of prism axis, i.e., location of prism bases; coordinate variable given by `IJKPRISMAXIS`

[%]: Coordinates for `NP` polygon points (clockwise or anti-clockwise); two coordinates per point: (Y,Z) for `IJKPRISMAXIS` = 1; (X,Z) for `IJKPRISMAXIS` = 2; (X,Y) for `IJKPRISMAXIS` = 3

If the region is not aligned with the coordinate axes, set `IREGGEOM` negative and provide 3 correction angles (azimuth, dip, and plunge). Figure 21 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 22). See Section 5.3 for additional region definition examples.

```
Test ellipsoidal 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 21. 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 22. 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 8):

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

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

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

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

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

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

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

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

Table 8. 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: $p_e = -CP(2) \cdot \sqrt{PER(NMAT)/USERX(1,N)}$
		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 $S_{lrc} = S_{lrk}$

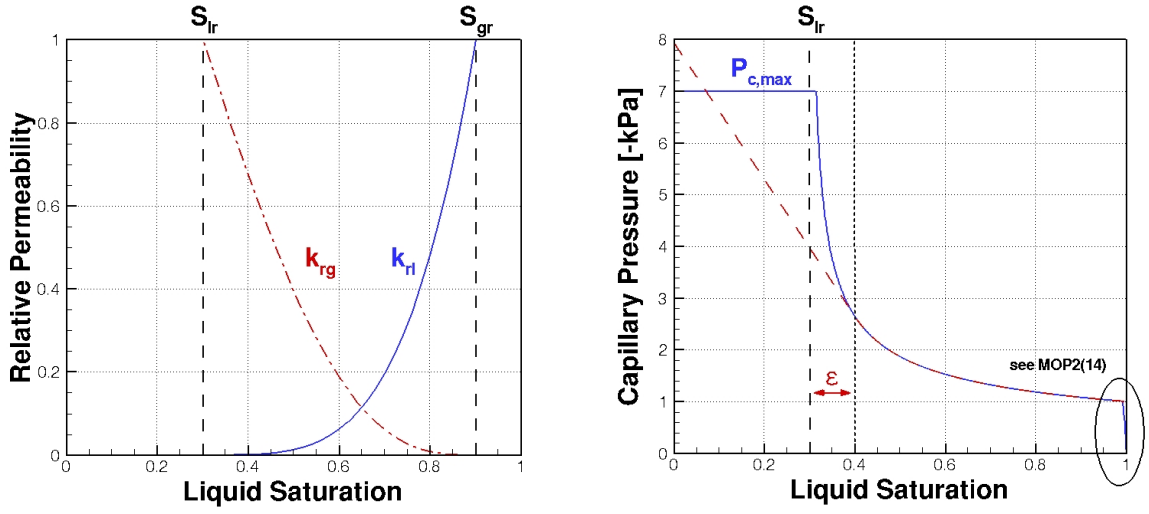

Figure 23. Modified Brooks-Corey relative permeability and capillary pressure curves.

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 9):

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

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

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

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

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

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

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

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

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

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

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

or

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

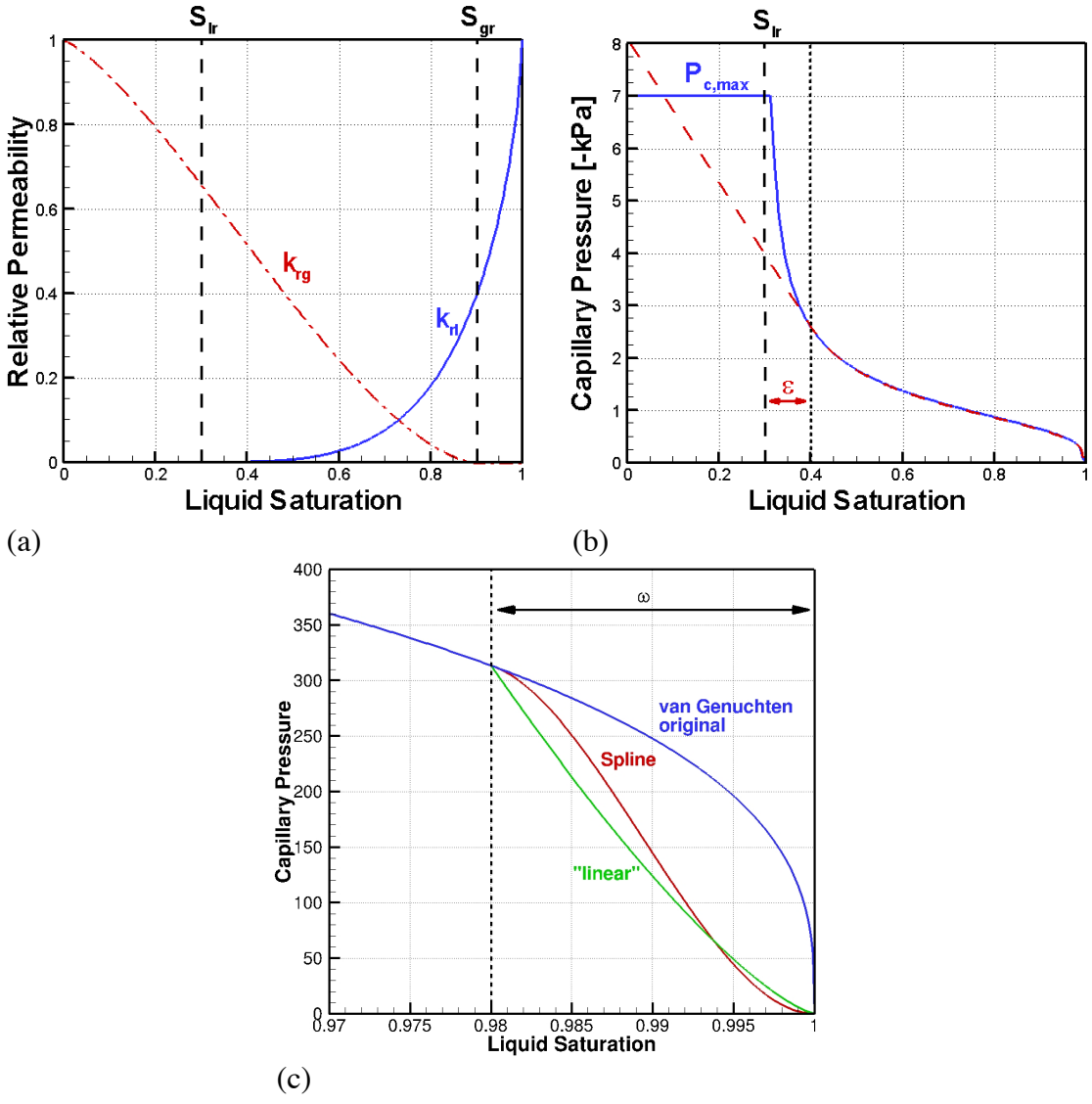

Figure 24. Modified van Genuchten (a) relative permeability and (b) capillary pressure curves with (c) options for capillarity near full saturation (for |CP (5)| = 0.02)

Table 9. 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: $1/\alpha_i = 1/\alpha_{ref} \cdot \sqrt{k_{ref}/k_i}$ 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 : $\varepsilon = \lvert$CP(3)$\rvert$; 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)	ω	Zero: Use original van Genuchten curve near full saturation Positive: Use cubic spline for $S_l > 1 - \omega$ Negative: Use "linear" interpolation for $S_l > 1 - \lvert\omega\rvert$
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 10).

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 10.

Table 10. 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.

9 Coupled Overland – Subsurface Flow (Material **SURWA**)

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 equations are 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 25—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`, `SURZG`, 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       11000009000000004000060 00
    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 25. TOUGH2 input file for testing coupled surface water – subsurface flow.

10 Semi-Analytical Radial Heat Exchange (Variable `MOP(15)`)

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 26 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    999910000001000020054  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 26. TOUGH2 input file for testing radial heat exchange option.

11 One-Dimensional Deformation Calculation (Block SUBSI)

Approximate, one-dimensional deformations are calculated based on porosity changes induced by changes in fluid pressure and/or temperature. No stress-strain calculation is performed. Relative porosity changes calculated by TOUGH2 expand or contract the distance between elements belonging to the same column. These changes are then integrated along the column to calculate new element coordinates. Volumes and nodal distances are not changed, i.e., the TOUGH2 mesh is not dynamically adapted.

Columns and their orientations are defined by a new parameter *ISOSUBSID*. A column consists of all elements that are linked by connections with *ISO* = *ISOSUBSID*. It is assumed that the TOUGH variable *ISO* with values of 1, 2, and 3 refer to connections that are (approximately) aligned with the X- Y-, and Z-axis, respectively. The default for *ISOSUBSID* is 3, creating vertical columns. The resulting deformation at the top of each column thus refers to subsidence (negative) or uplift (positive).

Each element in the mesh is assigned to a column and ordered according to its distance from a reference coordinate (variable *SUBSREFLEVEL*). Deformations at this reference point are defined as being zero. By default, *SUBSREFLEVEL* is set to a large neagive value. For vertical columns, this means that a no-deformation boundary condition is defined at the bottom of the model.

No stress-strain calculation is performed, and there is no mechanical interaction between neighboring columns, i.e., the entire volumetric strain is assumed to lead to deformation in only one direction (defined by *ISOSUBSID*).

To invoke this calculation, non-zero values must be given for pore compressibility (TOUGH2 variable *COM*) or pore expansivity (TOUGH2 parameter *EXPAN*) along with a new block labeled by keyword `SUBSI`. On the line following keyword `SUBSI`, the (optional) variables *ISOSUBSID* and *SUBSREFLEVEL* can be given in format (I5,5X,E10.4).

The results of this simplified, one-dimensional deformation calculation are reported in a separate output block in the standard TOUGH2 output file, or in CSV format using label `SUBSIDANCE` in block `OUTPU` (see Section 3.1).

Element-related one-dimensional deformations can be extracted for iTOUGH2 analyses using iTOUGH2 command `>> SUBSIDANCE`.

12 One-Dimensional Hydrogeomechanics (Block `LOAD`)

Following *Walsh et al*. [2012], the effects of an external, one-dimensional, vertical stress change on porosity is added to TOUGH2. Such enforced stress and porosity changes result in pore-pressure changes and related fluid redistributions. The model requires specification of a material-dependent loading efficiency and a one-dimensional specific storage coefficient. The underlying assumptions of horizontally bedded formations and uniaxial strain limit the applicability of this model to very simple systems in which changes in mechanical loading occur over a large area (e.g., due to continental glaciation, laterally extensive erosion and deposition events, Earth tides, and atmospheric pressure fluctuations).

Hydrogeomechanical coupling under a homogeneous and laterally extensive load is implemented within the mass accumulation term. Porosity (ϕ) in TOUGH2 is not constant, but is updated at the end of each iteration to account for changes in pressure. The expression for the updated porosity for the current time step (ϕ_t), including hydro-geomechanical effects, is

$$\phi_t = \phi_{t-1} + \phi_{t-1} \cdot c_\phi \cdot dp - \phi_{t-1} \cdot S_{S-1D} \cdot \zeta \cdot d\sigma_{zz} \tag{32}$$

where

ϕ_{t-1} : porosity (-) at previous timestep
c_ϕ : pore compressibility (Pa^{-1}), variable *COM* in block `ROCKS.2`
dp : pressure change (Pa) during the time step
S_{S-1D} : one-dimensional specific storage coefficient (Pa^{-1})
ζ : one-dimensional loading efficiency (-)
$d\sigma_{zz}$: change in vertical loading (Pa) during time step (positive for increased load)

The second term on the right-hand side of Eq. (32), i.e., $\phi_{t-1} c_\phi dp$, represents the change in porosity due to the change in pore pressure during a time step; this term is implemented in standard TOUGH2. The third term, $\phi_{t-1} S_{S-1D} \zeta d\sigma_{zz}$, is the hydromechanical term that represents the change in porosity due to the change in vertical load applied during a time step. The variables that are unique to the hydromechanical formulation are the one-dimensional loading efficiency (ζ), the change in vertical load ($d\sigma_{zz}$), and the one-dimensional (uniaxial) specific storage coefficient (S_{S-1D}). The hydromechanical capability requires the one-dimensional loading efficiency to be defined for each material type. This parameter is used to determine what percentage of the applied vertical stress is borne by the pore-fluids; it is defined as:

$$\zeta = \frac{\beta(1+\nu)}{3(1-\nu) - 2\alpha\beta(1-2\nu)} \tag{33}$$

where

α : Biot-Willis coefficient (-)
β : Skempton coefficient (-)
ν : Poisson ratio (-)

The Skempton coefficient can be related to geomechanical parameters as follows:

$$\beta = \frac{\left(\frac{1}{K} - \frac{1}{K_s}\right)}{\left(\frac{1}{K} - \frac{1}{K_s}\right) + \phi\left(c_w - \frac{1}{K_s}\right)} \tag{34}$$

where

K : drained bulk modulus (Pa)
K_s : solid grain bulk modulus (Pa)
c_w : water compressibility (Pa^{-1}), calculated internally

The three-dimensional specific storage coefficient is given by:

$$S_{S-3D} = \left(\frac{1}{K} - \frac{1}{K_s}\right) + \phi\left(c_w - \frac{1}{K_s}\right) \tag{35}$$

The one-dimensional specific storage coefficient is given by:

$$S_{S-1D} = S_{S-3D}(1 - \lambda\beta) \tag{36}$$

where

$$\lambda = \frac{2\alpha(1-2\nu)}{3(1-\nu)} \tag{37}$$

The product of the one-dimensional loading efficiency and the specific storage coefficient, $S_{S-1D}\xi$, is independent of fluid composition and can be approximated as

$$S_{S-1D}\xi = \frac{\left(\frac{1}{K} - \frac{1}{K_s}\right)(1+\nu)}{3(1-\nu)} = \phi\left(c_\phi + c_w\right) \tag{38}$$

where

c_ϕ : pore compressibility

The pore compressibility should be chosen judiciously as

$$c_\phi = \frac{S_{S-1D}}{\phi} - c_w \tag{39}$$

In TOUGH2, there are two options to provide the one-dimensional loading efficiency ξ. Material-dependent parameters must be given in Columns 51–80 of block `ROCKS.1.2`,

which is invoked by setting NAD ≥ 1.

Option 1: The one-dimensional loading coefficient ζ is internally calculated as a function of drained bulk modulus K [Pa], the Poisson ratio ν, and porosity using Eq. (33). The Biot-Willis coefficient α is assumed to be 1.0 (incompressible grains). Moreover, the pore compressibility c_ϕ is also internally calculated using Equ. (39). Option 1 is invoked if non-zero values are provided for K and ν.

K : Drained bulk modulus [Pa], Columns 51–60
ν : Poisson ratio [-], Columns 61–70

Option 2: The one-dimensional loading coefficient ζ is directly provided as a user-specified material property.

ζ : One-dimensional loading coefficient [-], Columns 71–80

To invoke one-dimensional hydromechanical coupling, the user must specify, for each material, the loading efficiency ζ in record ROCKS.2, columns 71–80 (set *NAD* ≥ 1). Moreover, the time-dependent load must be specified in the input file after keyword LOAD. (Note that block LOAD must be specified *after* the optional block TIMBC described in Section 6.1.)

Record LOAD.1

Free format
LOADDEF

LOADDEF : Number of loading functions defined. The total load is the sum of all loading functions. Defining multiple functions is convenient as certain loads (e.g., Earth tides) are a superposition of multiple harmonic functions.
Repeat the following entries *LOADDEF* times.

Record LOAD.2

Format(A9,1X,I10)
BCELM, NBCP

BCELM : String indicating type of loading function (start in Column 1)

TABLE : Table with *NBCP* rows of time t (s) vs. load σ (Pa). Loads will be linearly interpolated between table entries. For simulation times before the first or after the last time specified in the table, the load is set equal to the first or last load, respectively. Loading times specified in this block can be honoured by setting *MOP2*(21) = 1 in block MOMOP (see Section 2.1).

POLYNOM : Polynomial of degree (*NBCP* – 1):

$$\sigma_{zz}(t) = \sum_{i=0}^{NBCP-1} a_i \cdot t^i \tag{40}$$

SINE : Sinusiodal function, i.e.,

$$\sigma_{zz}(t) = \sigma_0 \cdot \sin\left(2\pi \frac{t - t_0}{t_p}\right) \quad \text{for} \quad t_B \le t \le t_E \tag{41}$$

For simulation times before t_B or after t_E, the load is set equal to $\sigma_{zz}(t_B)$ or $\sigma_{zz}(t_E)$, respectively.

SQRT : Square-root function, i.e.,

$$\sigma_{zz}(t) = \sigma_0 \cdot sqrt\left(\frac{t - t_0}{t_p}\right) \quad \text{for} \quad t_B \le t \le t_E \tag{42}$$

For simulation times before t_B or after t_E, the load is set equal to $\sigma_{zz}(t_B)$ or $\sigma_{zz}(t_E)$, respectively.

EXP : Exponential function, i.e.,

$$\sigma_{zz}(t) = \sigma_0 \cdot \exp\left(t_p(t - t_0)\right) \quad \text{for} \quad t_B \le t \le t_E \tag{43}$$

For simulation times before t_B or after t_E, the load is set equal to $\sigma_{zz}(t_B)$ or $\sigma_{zz}(t_E)$, respectively.

NBCP : Number of table entries if *BCELM* = 'TABLE', otherwise number of function coefficients provided in record LOAD.3.

Record LOAD.3 (for *BCELM* = 'TABLE')

Free format
(*TIMBCV*(*I*), *PGBCEL*(*I*)), *I*=1,*NBCP*

TIMBCV(I) : Time (s)
PGBCEL(I) : Load (Pa)

Repeat *NBCP* times.

Record LOAD.3 (for *BCELM* ≠ 'TABLE')

Free format
PGBCEL(*I*)), *I*=1,*NBCP*

PGBCEL(I) : Function coefficients

For *BCELM* = 'PLOYNOM', provide *NBCP* coefficients a_i
For *BCELM* = 'SINE', 'SQRT' or 'EXP', the coefficients are:

PGBCEL(1) = σ_0 (Pa)
PGBCEL(2) = t_p (sec)
PGBCEL(3) = t_0 (sec) (optional)
PGBCEL(4) = t_B (sec) (optional)
PGBCEL(5) = t_E (sec) (optional)

An example is given in Figure 27. Relevant input parameters are highlighted in bold.

```
Terzaghi
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
SHALE    1     2650.     .1000                     2.040E-15      2.51      920.
 1.860E-07                                                                 0.8
BOUND    0     2650.     .9900                     2.040E-15      2.51   100000.

PARAM----1--MOP:123456789012345678901234----*----5----*----6----*----7----*----8
   1 100       110000090000000400003000
 0.000E+00 3.1536E11 1.000E+02
    1.0E-7
 100000.000000000000   0.000000000000000  20.000000000000000
MOMOP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
2                   1
MULTI----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    2    2    2    6
ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
IN             BOUND .1000E+50                     .0000E+00
    1   49    1SHALE .2000E+02                     .2000E+02

CONNE----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
IN       1                   3 .1000E-10 .1000E+02 .1000E+01
    1    2   48    1    1    3 .1000E+02 .1000E+02 .1000E+01

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
INCON----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8

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

LOAD ----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   1
TABLE              2
0.0   0.0
1.0   3E6

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

Figure 27. TOUGH2 input file for simulating one-dimensional hydromechanical coupling.

13 Non-Darcy Flow Using Forchheimer Equation (Block FORCH)

The equation governing flow in partially saturated porous media as implemented in TOUGH2 is based on the multiphase version of the Darcy-Buckingham law:

$$\mathbf{F} = \rho\mathbf{u} = -k\frac{k_r\rho}{\mu}\left(\nabla P - \rho\mathbf{g}\right) \tag{44}$$

Here, $\mathbf{F}$ [kg s^{-1} m^{-2}] is the Darcy mass flux, $\mathbf{u}$ [m s^{-1}] is the average velocity, ρ [kg m^{-3}] is the density, k [m^2] is the absolute permeability, k_r [-] is the relative permeability, which is a function of saturation, μ [Pa s] is the dynamic viscosity, ∇P [Pa m^{-1}] is the pressure gradient, and $\mathbf{g}$ [m s^{-2}] is the gravity vector.

Forchheimer [1901] empirically found the relationship between the gradient of the water potential $\nabla\Phi = (\nabla P - \rho\mathbf{g})$ [Pa m^{-1}] and average velocity to be:

$$-\nabla\Phi = \frac{\mu}{k\cdot k_r}\mathbf{u} + \beta\rho|\mathbf{u}|\mathbf{u} \tag{45}$$

The second term on the right-hand side of Eq. (45) accounts for inertial forces. On the pore scale, as velocity increases, flow tends to locally separate from the tortuous pore walls, leading to local vortices and countercurrent flow regimes. The coefficient β [m^{-1}] is a function of effective permeability ($k\cdot k_r$), phase content ($\theta = \phi S$, where ϕ is porosity and S is the saturation of the considered fluid phase), and tortuosity ($\tau = L/l$, where L is the macroscopic flow distance, and l is the length of the corresponding microscopic, tortuous flow path). This dependence indicates a transition from viscous-dominated to form-dominated flow as the Reynolds number ($\mathrm{Re} = \rho u\sqrt{K}/\mu$, where K is a form factor) increases beyond a critical value. The parameter β has been referred to as the non-Darcy flow coefficient, the Forchheimer or inertial resistance coefficient, or the turbulence factor. Note, however, that deviations from Darcy's law are observed at Reynolds numbers that are at least one order of magnitude smaller than those where actual turbulence occurs. We will refer to β as the non-Darcy flow coefficient, which can be written in a general, parameterized form as follows:

$$\beta = A_1\cdot\frac{1}{(k\cdot k_r)^{A_2}}\cdot\frac{1}{\theta^{A_3}}\cdot\frac{1}{\tau^{A_4}} \tag{46}$$

Table 11 summarizes various correlation models for the non-Darcy flow coefficient as implemented in iTOUGH2. Most of the original models were developed for single-phase gas flow or gas flow at immobile water saturation. They are generalized here to two-phase conditions by replacing absolute permeability k with effective permeability, $k_{eff} = k\cdot k_r(S)$, and porosity ϕ with phase content, $\theta = \phi S$. Moreover, it is assumed that the Forchheimer equation applies to both liquid and gas flow. (Note that the user can restrict its applicability to gas flow only, see Section 4.1). The appropriateness of these extensions is not examined here.

Table 11. Parameters for Non-Darcy Flow Coefficient β

#	A_1	A_2	A_3	A_4	**Reference**
1	FORCH(1)	0	0	0	β = user-specified constant
2	5.00×10^{-3}	0.5	5.5	0	*Geertsma* [1974]
3	1.60×10^{-13}	1.64	0	0	*Frederick and Graves* [1994], Eq. (46)
4	2.53×10^{-13}	1.6	0.404	0	*Frederick and Graves* [1994], Eq. (47)
5	3.81×10^{-13}	1.55	1.0	0	*Frederick and Graves* [1994], Eq. (48)
6[&]	-	-	-	-	*Frederick and Graves* [1994], Eq. (49)
7	3.18×10^{-9}	1.25	0.75	0	*Janicek and Katz* [1955]
8	1.11×10^{-12}	1.55	0	0	*Jones* [1987]
9	2.89×10^{-6}	1.0	1.0	1.0	*Liu et al.* [1995]
10	2.66×10^{-6}	0.98	0.29	3.35	*Thauvin and Mohanty* [1998]
11	2.16×10^{-16}	1.88	0.449	0	*Coles and Hartman* [1998]
12	4.55×10^{-6}	1.023	0	1.943	*Cooper et al.* [1999]
13	1.42×10^{-1}	0.5	1.5	0	*Ergun* [1952]
14	FORCH(1)	FORCH(2)	FORCH(3)	FORCH(4)	User specified

The following conversion factors were used to convert, as applicable, the correlations given in the original references to SI metric units:

Darcy $\times 9.869233\times10^{-13}$ = m^2
Feet $\times 0.3048$ = m

Most of the original correlations were developed for single-phase gas flow. An extension to two-phase conditions is implemented by replacing absolute permeability with effective permeability, $k_{eff} = k \cdot k_r$, and porosity with phase content, $\theta = \phi \cdot S$.

The parameters A_1 through A_4 apply to the following general form for the non-Darcy flow coefficient:

$$\beta = A_1 \cdot (k \cdot k_r)^{-A_2} \cdot \theta^{-A_3} \cdot \tau^{-A_4}$$

[&] Model No. 6 [*Frederick and Graves*, 1994, Eq. (6)]: $\beta = \frac{3.2808}{\theta^2} \cdot \exp\left[45 - \sqrt{3205.71 + 81 \cdot \ln\left(k \cdot k_r / \theta\right)}\right]$

The non-Darcy flow effects can be combined with the Klinkenberg gas-slip flow effect and expressed as a pressure- and velocity-dependent correction to the absolute permeability:

$$\tilde{k} = k\left[\frac{1+\left(\delta_g\, b/P\right)}{1+F_0}\right] \tag{47}$$

Here, b [Pa] is the Klinkenberg factor, which accounts for enhanced gas slip flow that occurs when the mean free path of the gas molecules is large relative to the characteristic dimension of the pores [*Klinkenberg*, 1941]. Slip flow is important at low pressures P [Pa] and in small pores, when a significant fraction of molecular collision is with the pore wall rather than with other gas molecules. The Kroenecker delta δ_g is set to 1 if the fluid is gas, and zero if the fluid is liquid.

The coefficient F_0 describes the velocity-dependent reduction in permeability as represented by the Forchheimer equation (Eq. 45), and is given by:

$$F_0 = \beta\rho\frac{k\cdot k_r}{\mu}|\mathbf{u}| \tag{48}$$

Note that the Klinkenberg effect increases gas flow rates, whereas the non-Darcy flow effects of the Forchheimer equation tends to decrease flow rates. Furthermore, conditions under which the Klinkenberg gas slip effect becomes significant are often opposite to those under which the second term of the Forchheimer equation becomes relevant.

Tortuosity effects are considered to have a porous-medium-dependent part and a saturation-dependent part. The resulting tortuosity factor τ entering some of the non-Darcy flow coefficient models (see models with $A_4 \neq 0$ in Table 11) is evaluated using one of the following options:

$$\tau = \phi^{1/3}S^{10/3} \tag{49a}$$

$$\tau = \tau_0 \cdot S \tag{49b}$$

$$\tau = \tau_0 \cdot k_r \tag{49c}$$

$$\tau = \tau_0 \tag{49d}$$

Eq. (49a) represents the *Millington and Quirk* [1961] model, which yields non-zero tortuosity coefficients as long as phase saturation is non-zero. A linear correlation between tortuosity and saturation is given by Eq. (49b). One might also argue that flow is most tortuous (i.e., the tortuosity factor approaches zero) when the fluid phase becomes discontinuous, suggesting that saturation-dependent tortuosity should be related to relative permeability (Eq. 49c). Finally, Eq. (49d) describes a constant, saturation-independent tortuosity factor.

Non-Darcy flow based on the Forchheimer equation is implemented as follows::

- The new keyword `FORCH` is followed by user-specified input for the non-Darcy flow coefficient model (Eq. 46).
- The non-Darcy flow coefficient β is evaluated for each gridblock in a connection, and then averaged according to the selected weighting scheme.
- The quadratic Forchheimer equation (Eq. 46) is solved for the absolute value of the non-Darcy velocity u using the following numerically stable formula:

$$\mathbf{u} = \frac{-2\nabla\Phi}{\frac{\mu}{k \cdot k_r} + \sqrt{\left(\frac{\mu}{k \cdot k_r}\right)^2 + 4\beta\rho|\nabla\Phi|}} \tag{50}$$

  Note that as $\beta \to 0$, Eq. (50) converges to the solution of the Darcy flow equation.

- The non-Darcy flow coefficient β is calculated as a function of absolute permeability, relative permeability, porosity, saturation, and tortuosity (Eq. 46).

The simulation of non-Darcy flow according to the Forchheimer equation is invoked by providing a new data block following keyword `FORCH` in the TOUGH2 input file:

`FORCH` Invokes flow calculation according to the Forchheimer equation (Eq. 45); introduces information on non-Darcy flow coefficient model (Eq. 46) and weighting scheme.

Record `FORCH.1`

FORMAT (2I5, 4E10.4)
`IFORCH, MFORCH, (FORCH(I), I``=1,4)`

`IFORCH` Integer parameter to choose type of non-Darcy flow coefficient model (see Table 1, Column 1). If a negative number is given, the Forchheimer equation is applied to gas flow only; liquid flow is governed by Darcy's law (Eq. 44).

`MFORCH` Determines interface weighting scheme for non-Darcy flow coefficient β:

`MFORCH` = 0: Upstream weighting (recommended)
`MFORCH` = 1: Upstream weighting with *`WUP`* (see record `PARAM.3`)
`MFORCH` = 2: Harmonic weighting
`MFORCH` = 3: Geometric average
`MFORCH` = 4: Arithmetic average

`FORCH(I)` *`I`* = 1, …, 4; user-specified parameters for non-Darcy flow coefficient model (see Table 11, Columns 2–5; only needed for models *`IFORCH`* = 1 and 14).

The porous-medium-dependent tortuosity factor τ_0 and the tortuosity model (see Eq. 49a–49d) are selected through variable *`TORTX`* (Columns 31–40 of record `ROCKS.1.1`, see *Pruess et al.* [1999]) as follows:

TORTX	= 0:	Millington-Quirk model (Eq. 49a)
	> 0:	Tortuosity depends on relative permeability (Eq. 49c)
	> 0 and *DIFF0* < 0:	Tortuosity is a linear function of saturation (Eq. 49b); *DIFF0* is provided in record PARAM.1, Columns 41–50, see *Pruess et al*. [1999].
	< 0:	Constant tortuosity factor (Eq. 49d)

The Klinkenberg factor b [Pa] for enhancing gas phase permeability k_g over liquid phase permeability k_l according to the relationship $k_g = k_l \cdot (b/P)$ (see also Eq. 48) is provided through variable *GK* (Columns 41–50 of record ROCKS.1.1, see *Pruess et al*. [1999]).

The non-Darcy flow coefficients, i.e., variable *FORCH(IPAR)*, can be estimated by iTOUGH2 using command >> FORCHHEIMER.

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.

Ergun, S., Fluid flow through packed column, *Chemical Engineering Progress, 48*(2), 89–94, 1952.

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., G. J. Moridis, and K. Pruess, *A TOUGH2 Equation-of-State Module for the Simulation of Two-Phase Flow of Air, Water, and a Miscible Gelling Liquid*, Report LBL-36086, Lawrence Berkeley Laboratory, Berkeley, Calif., May 1994.

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

Finsterle, S., and P.A. Witherspoon, *Implementation of the Forchheimer Equation into iTOUGH2*, Project report, Witherspoon Inc., Berkeley, Calif., 2001.

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

Finsterle, S., *iTOUGH2 Universal Optimization Using the PEST Protocol*, Report LBNL-LB3698E, Lawrence Berkeley National Laboratory, Berkeley, Calif., July 2010.

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., *iTOUGH2-EOS1nT: A Nonisothermal Two-Phase Flow Simulator for Water and Multiple Tracers—User's Guide*, Report FGC-17-01, Finsterle GeoConsulting, Kensington, Calif., January 2017.

Forchheimer, P., Wasserbewegung durch Boden, *Zeitschrift Verein Deutscher Ingenieure, 45*, 1782–1788, 1901.

Frederick, D.C., and R. M. Graves, New correlation to predict non-Darcy flow coefficients at immobile and mobile water saturation, Paper SPE 28451, presented at 1994 SPE Annual Technical Conference and Exhibition, pp. 915–921, New Orleans, LA, Sept. 25–28, 1994.

Geertsma, J., Estimating the coefficient of inertial resistance in fluid flow through porous media, *SPEJ*, 445–450, 1974.

Guðmundsdóttir, H., M.Þ. Jónsson, J.-C. Berthet, A. Arnaldsson, and S. Finsterle, *iTOUGH2-FloWell: Simulating Coupled Reservoir and Wellhead Conditions, User's Guide*, Report FGC-17-08 and LBNL-7017E, 2017.

Janicek, J. D., and D. L. Katz, Applications in unsteady state gas flow calculations, *Proc*., Univ. of Michigan Research Conference, June 20, 1955.

Jones, S. C., Using the inertial coefficient, b, to characterize heterogeneity in reservoir rock, Paper SPE 16949, presented at the 1987 SPE Annual Technical Conference and Exhibition, New Orleans, LA, Oct. 5–8, 1987.

Klinkenberg, L. J., The permeability of porous media to liquids and gases, in *Drilling Production Practice*, pp. 200–213, Am. Pet. Inst., Washington, D.C., 1941.

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.

Liu, X., F. Civan, and R. D. Evans, Correlation of the non-Darcy flow coefficient, *JCPT*, *34*(10), 50–54, 1995.

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.

Magnúsdóttir, L., and S. Finsterle, *iTOUGH2-EOS1sc: Multiphase Reservoir Simulator for Water Under Sub- and Supercritical Conditions, User's Guide*, Report LBNL-7017E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015.

Millington, R.J. and J.P. Quirk, Permeability of Porous Solids, *Trans. Faraday Soc*., 57, 1200–1207, 1961.

Magnúsdóttir, L., and S. Finsterle, *iTOUGH2-EOS1sc: Multiphase Reservoir Simulator for Water under Sub- and Supercritical Conditions*, *User's Guide*, Report LBNL-7017E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015.

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.

Oldenburg, C.M., G.J. Moridis, N. Spycher, and K. Pruess, *EOS7C Version 1.0: TOUGH2 Module for Carbon Dioxide or Nitrogen in Natural Gas (Methane) Reservoirs*,

Report LBNL-56589, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2004.

Parashar, R., and D. Reeves, Computation of flow and transport in fracture networks on a continuum grid, MODFLOW and MORE 2011: Integrated Hydrologic Modeling: Golden, Colorado, 2011.

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.

Stauffer, P.H., K.C. Lewis, J.S. Stein, B.J. Travis, P. Lichtner, and G. Zyvoloski, Joule-Thomson effects on the flow of liquid water, *Transp. Porous Media, 105*(3), 471–485, 2014.

Thauvin, F., and K.K. Mohanty, Network modeling of non-Darcy flow through porous media, *Transport in Porous Media*, *31*, 19–37, 1998.

Wainwright, H.M., and S. Finsterle, *Global Sensitivity and Data-Worth Analysis in iTOUGH2, User's Guide*, Report LBNL-TBD, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015.

Walsh, R., N. Calder, and J. Avis, A simple implementation of 1D hydromechanical coupling in TOUGH2, In: *Proceedings of the TOUGH Symposium 2012*. Report LBNL-5808E. Lawrence Berkeley National Laboratory, Berkeley, California, 425–432, 2012.

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

Wu, Y.-S., S. Finsterle, and K. Pruess, *EOS3nn: An iTOUGH2 Module for Non-Newtonian Liquid and Gas Flow*, Report LBNL-51103, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2002.

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.