

Lawrence Berkeley National Laboratory

LBL Publications

Title

TOUGH3 User's Guide, Version 1.0

Permalink

<https://escholarship.org/uc/item/0dh2w4w6>

Authors

Jung, Y

Pau, GSH

Finsterle, S

et al.

Publication Date

2023-12-06

Peer reviewed

TOUGH3 User's Guide

Version 1.0

*Yoojin Jung
George Shu Heng Pau
Stefan Finsterle
Christine Doughty*

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

January, 2018

This work was supported by the Director, Office of Science, Office of Basic Energy Sciences,
Materials Sciences and Engineering Division of the U.S. Department of Energy
under Contract no. DE-AC02-05CH11231.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or The Regents of the University of California.

Ernest Orlando Lawrence Berkeley National Laboratory
is an equal opportunity employer.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1. ABOUT THIS MANUAL	1
1.2. SCOPE AND METHODOLOGY	3
2. WHAT'S NEW IN TOUGH3	5
3. CODE INSTALLATION	7
4. GOVERNING EQUATIONS	8
4.1. MASS-BALANCE EQUATION.....	8
4.2. ACCUMULATION TERMS	8
4.3. FLUX TERMS	9
4.4. SINK AND SOURCE TERMS	12
4.5. SEMI-ANALYTICAL CONDUCTIVE HEAT EXCHANGE.....	16
5. NUMERICAL METHODS.....	19
5.1. SPACE AND TIME DISCRETIZATION	19
5.2. INTERFACE WEIGHTING SCHEMES	23
5.3. LINEAR EQUATION SOLVERS	26
5.3.1. <i>Internal serial solvers</i>	26
5.3.2. <i>PETSc solvers</i>	27
5.3.3. <i>Aztec solvers</i>	29
6. INITIAL AND BOUNDARY CONDITIONS	30
6.1. INITIAL CONDITIONS AND RESTARTING	30
6.2. NEUMANN BOUNDARY CONDITIONS.....	31
6.3. DIRICHLET BOUNDARY CONDITIONS	31
6.4. ATMOSPHERIC BOUNDARY CONDITIONS.....	33
6.5. CONSTANT TEMPERATURE BOUNDARY CONDITIONS.....	34
7. ARCHITECTURE OF TOUGH3	35
7.1. PROGRAM STRUCTURE.....	35
7.2. ARRAY STRUCTURE	36
7.2.1. <i>Primary Thermodynamic Variables</i>	36
7.2.2. <i>Secondary Parameters</i>	38

7.2.3. <i>Linear Equation Setup</i>	39
7.3. EXECUTION	42
8. PREPARATION OF INPUT DATA	45
8.1. TOUGH3 INPUT DATA	45
8.2. TOUGH3 INPUT FORMAT	50
8.3. GEOMETRY DATA	88
8.4. INPUT FORMATS FOR MESHMAKER	94
9. OUTPUT FROM TOUGH3	101
10. SAMPLE PROBLEMS	105
10.1. PROBLEM NO. 1 (*TUTORIAL*) – INFILTRATION AND DRAINAGE OF WATER	105
10.1.1. <i>Mesh Generation</i>	106
10.1.2. <i>Material Properties</i>	107
10.1.3. <i>Initialization of the Problem</i>	108
10.1.4. <i>Simulating Drainage and Infiltration</i>	110
10.1.5. <i>Analysis of the Output</i>	112
10.1.6. <i>Parallel Execution of the Simulation</i>	114
10.2. PROBLEM NO. 2 (*HYS TEST*) – TESTING HYSTERESIS	115
10.2.1. <i>Problem Setup</i>	115
10.2.2. <i>Results</i>	120
11. CONCLUDING REMARKS	126
ACKNOWLEDGEMENT	127
REFERENCES	128
APPENDIX A. RELATIVE PERMEABILITY FUNCTIONS	135
APPENDIX B. CAPILLARY PRESSURE FUNCTIONS	147
APPENDIX C. ACTIVE FRACTURE MODEL	159
C.1. ACTIVE FRACTURE CONCEPT	159
C.2. REDUCTION OF FRACTURE-MATRIX INTERFACE AREA	161
APPENDIX D. INPUT FORMAT OR OPTION CHANGES IN TOUGH3	162

LIST OF FIGURES

Figure 1. Space discretization and geometry data in the integral finite difference method.....	20
Figure 2. Weighting procedures for absolute (k) and relative permeability (k_r) at grid block interfaces.	24
Figure 3. An example of the PETSc option file (<code>.petscrc</code>).....	28
Figure 4: An example of the Aztec option file (<code>.aztecrc</code>).....	29
Figure 5. Modular architecture of TOUGH3.	36
Figure 6. Structure of thermophysical property arrays in TOUGH3. The above structure is for a serial implementation. In a parallel implementation, the order of the variables in each volume element remains the same but the order of the volume elements is different, depending on how the volume elements are distributed across multiple processors.	38
Figure 7. Linear equation structure.	41
Figure 8. User options for supplying geometry data.....	89
Figure 9. MESHMaker input formats.	91
Figure 10. Subgridding in the method of "multiple interacting continua" (MINC).	93
Figure 11. Example data block OUTPU	103
Figure 12. Element-related output variables in CSV format based on OUTPU block shown in Figure 11..	103
Figure 13. Connection-related output variables in CSV format based on OUTPU block shown in Figure 11.	103
Figure 14. Generation-related output variables in CSV format based on OUTPU block shown in Figure 11.	104
Figure 15. Schematic diagram of infiltration experiment in a vertical column.	105
Figure 16. Input file for problem P1a – generation of the 1D mesh for the column experiment. .	106
Figure 17. File MESH generated by running P1a..	107
Figure 18. Input file for problem P1b to create initial hydrostatic pressure profile.	108
Figure 19. Selected output at steady state for P1b.	109
Figure 20. Input file for problem P1c – drainage of water.	111
Figure 21. Input file for problem P1d.	112

Figure 22. Data block OUTPU	113
Figure 23. File <code>OUTPUT_ELEME.csv</code> based on OUTPU block shown in Figure 22.	113
Figure 24. Temporal change of water saturation distribution in the column.....	114
Figure 25. Illustration of the grid, boundary conditions, and source and observation locations. ...	116
Figure 26. First part of input file for regular hysteresis sample problem.	117
Figure 27. Characteristic curve specifications for regular hysteresis (IRP=ICP=12), simple hysteresis (IRP=ICP=13), and no hysteresis (IRP=ICP=7).	119
Figure 28. Saturation distributions in the model at various times during the five-step injection sequence using regular hysteresis (IRP=ICP=12). The current injection location is identified with a filled black dot; previous injection locations are identified with filled grey dots; the location of observation element A1140 is shown as an open dot.	121
Figure 29. (a) Gas saturation S_g , residual gas saturation S_{gr} , and ICURV versus time at element A1140, for regular hysteresis (curves labelled 12), simple hysteresis with the same parameters (curves labelled 13), and simple hysteresis with parameters modified to match regular hysteresis (curves labelled 13L); (b) S_g and S_{gr} versus time at element A1140, for no-hysteresis cases with large residual gas saturation (curves labelled 7A) and essentially zero residual gas saturation (curves labelled 7C).	123
Figure 30. Regular hysteresis characteristic curves: (a) Capillary pressure versus liquid saturation at element A1140. The plot is divided into two parts, based on time, to enable the different branches of the capillary pressure curve to be seen more easily. The branch switches (turning points) are marked by black dots, with ICURV for each switch labelled. (b) Relative permeability versus liquid saturation at element A1140. Branches 2, 3, and 4 all overlie one another.	124
Figure 31. (a) Capillary pressure and (b) gas relative permeability versus liquid saturation for element A1140 for regular hysteresis (case 12), simple hysteresis (cases 13 and 13L), and no hysteresis (cases 7A and 7C). For cases 7A and 7C, the capillary pressure curve follows the primary drainage branch of the regular hysteresis curve. All liquid relative permeability curves follow the primary drainage branch of the regular hysteresis curve shown in Figure 30.....	125

LIST OF TABLES

Table 1. Fluid property modules for TOUGH3.	2
Table 2. Commonly used options for PETSc linear solvers	28
Table 3. Key Subroutines in TOUGH3.....	43
Table 4. TOUGH3 input data blocks	46
Table 5. TOUGH3 disk files.....	49
Table 6. Flowing bottomhole pressures (in Pa) at 1000 m feed zone depth for a well of 20 cm (\approx 8 inch) inside diameter producing at 7 bar wellhead pressure (calculated from HOLA; Aunzo et al., 1991).....	77
Table 7. More Options	80
Table 8. Keywords of block OUTPU and related output variables	84
Table 9. Standard output based on absolute value of <i>KDATA</i>	102
Table 10. Option for Reducing Fracture-Matrix Interface Area.....	161

1. Introduction

1.1. About This Manual

TOUGH3 is a general-purpose numerical simulator for multi-dimensional fluid and heat flows of multiphase, multicomponent fluid mixtures in porous and fractured media. It is developed as an enhanced, more efficient version of the TOUGH2 suite of codes.

TOUGH3 consolidates the serial (TOUGH2; Pruess et al., 2012) and parallel (TOUGH2-MP; Zhang et al., 2008) implementations, enabling simulations to be performed on desktop computers and supercomputers using a single source code. New PETSc parallel linear solvers (Balay et al., 2016) are added to the existing serial solvers of TOUGH2 and the Aztec solver used in TOUGH2-MP. TOUGH3 also implements numerous enhanced features. The code inherits all the existing key processes and features of its predecessors, and is backwards compatible with a few justifiable exceptions (see Appendix D for the exceptions). However, the parallel computing capability and the additional, parallel linear solvers employed by TOUGH3 remarkably improve the code's computational efficiency.

The present report provides a summary of the mathematical models and numerical methods, discusses new user features, and gives complete specifications for preparing input data. It also includes a quick start guide to TOUGH3 that describes how to install the code, set up the problem, execute the simulation, and analyze the output. To make this report self-contained, we include much of the material that was covered in the TOUGH2 user's guide (Pruess et al., 2012).

TOUGH3 currently includes all equation-of-state (EOS) modules of the Version 2 of TOUGH2 (see Table 1), as well as ECO2N (Pruess, 2005; Pan et al., 2015), ECO2M (Pruess, 2011), EOS7C (Oldenburg et al., 2004), EOS7CA (Oldenburg, 2015), and TMVOC (Pruess and Battistelli, 2002). For each of the basic EOS modules from the Version 2 of TOUGH2, a separate addendum is available, which provides specification and usage of the EOS, as well as illustrative sample problems. These sample problems provide benchmarks for proper code installation, and can be used as templates to help develop new applications. For some of the more complex modules, detailed specialized reports (listed above) are available, which will facilitate applications.

Table 1. Fluid property modules for TOUGH3.

Module	Capabilities
EOS1	water, water with tracer
EOS2	water, CO ₂
EOS3	water, air
EOS4	water, air, with vapor pressure lowering
EOS5	water, hydrogen
EOS7	water, brine, air
EOS7R	water, brine, air, parent-daughter radionuclides
EOS7C	water, brine, CO ₂ (or N ₂), gas tracer, CH ₄
EOS7CA	water, brine, non-condensable gas, gas tracer, air
EOS8	water, “dead” oil, non-condensable gas
EOS9	variably-saturated isothermal flow according to Richards’ equation
EWASG	water, salt (NaCl), non-condensable gas
TMVOC	water, water-soluble volatile organic chemicals, non-condensable gases, with biodegradation
ECO2N ¹	water, NaCl, CO ₂
ECO2M	water, NaCl, CO ₂ , including transitions between super- and sub-critical conditions, and phase change between liquid and gaseous CO ₂

¹ For ECO2N, both the original (Pruess, 2005) and Version 2 (Pan et al., 2015) are available.

1.2. Scope and Methodology

The TOUGH3 simulator is developed for applications involving subsurface flow problems. TOUGH3 solves mass and energy balance equations that describe fluid and heat flow in general multiphase, multicomponent, and multidimensional systems. It fully accounts for the movement of gaseous, aqueous, and non-aqueous phases, the transport of latent and sensible heat, and the transition of components between the available phases, which may appear and disappear depending on the changing thermodynamic state of the system. Advective fluid flow in each phase occurs under pressure, viscous, and gravity forces according to the multiphase extension of Darcy's law, which includes relative permeability and capillary pressure effects. In addition, diffusive mass transport can occur in all phases. The code includes Klinkenberg effects in the gas phase and vapor pressure lowering due to capillary and phase adsorption effects. Heat flow occurs by conduction and convection, as well as radiative heat transfer according to the Stefan-Boltzmann equation. Local equilibrium of all phases is assumed to describe the thermodynamic conditions. TOUGH3 can simulate the injection or production of fluids and heat, and also includes different options for considering wellbore flow effects. Effects of sorption onto the solid grains, radionuclide transport, or biodegradation can be simulated as well for certain EOS modules.

TOUGH3 uses an integral finite difference method (IFDM) for space discretization. The IFDM (Edwards, 1972; Narasimhan and Witherspoon, 1976) avoids any reference to a global system of coordinates, and thus offers the advantage of being applicable to regular or irregular discretizations in one, two, and three dimensions. The IFDM also makes it possible, by means of simple preprocessing of geometric data, to implement double-porosity, dual-permeability, or multiple interacting continua (MINC) methods for fractured media (Pruess and Narasimhan, 1982, 1985; Pruess, 1983). No particular price needs to be paid for this flexibility; indeed, for systems of regular grid blocks referring to a global coordinate system, the IFDM is completely equivalent to conventional finite differences. Time is discretized fully implicitly as a first-order backward finite difference. The implicit time-stepping and the 100% upstream weighting of flux terms at interfaces are necessary to avoid impractical time step limitations in flow problems involving phase (dis-)appearances, and to achieve unconditional stability (Peaceman, 1977). The resulting strongly coupled, nonlinear algebraic equations are solved simultaneously using Newton-Raphson iterations for each time step, which involves the calculation of a Jacobian matrix and the solution of a set of linear equations. Time steps are automatically adjusted during a simulation run, depending on the convergence rate of the iteration process. Newton-Raphson increment weighting can also be adjusted if the iterations oscillate.

TOUGH3 can simulate various fluid mixtures by means of separate EOS modules, which internally calculate the thermophysical properties of specific fluid mixtures, e.g., fluid density, viscosity, and enthalpy. Due to this flexibility to handle a variety of flow systems, TOUGH3 can be used for diverse application areas, such as geothermal reservoir engineering, geological carbon sequestration, natural gas reservoirs, nuclear waste isolation, environmental assessment and remediation, and flow and transport in variably saturated media and aquifers, among other applications that involve nonisothermal multiphase flows.

2. What's New in TOUGH3

TOUGH3 incorporates numerous new, enhanced features. Here, we summarize the key capabilities newly available in TOUGH3.

- New Equation-of-state (EOS) modules

TOUGH3 incorporates several new EOS modules, which were either developed after TOUGH2 V2, or were separate, stand-alone codes: TMVOC, ECO2N, ECO2M, EOS7C, and EOS7CA.

- Consolidation of serial (TOUGH2) and parallel (TOUGH2-MP) implementations

The same code can perform simulations in serial mode on desktop computers as well as in parallel on multicore machines and supercomputers.

- Dynamic memory allocation

The problem-size dependent arrays are dynamically allocated during the initialization phase. The T2 file previously used to define the dimensions of major arrays is no longer needed, and the code does not need to be recompiled if the problem size changes.

- PETSc solvers

TOUGH3 includes an interface to PETSc's parallel linear solvers in addition to the existing TOUGH2 serial linear solvers and TOUGH2-MP's Aztec solvers. PETSc provides efficient implementations of linear solvers and interfaces to popular external linear solver packages. The available linear solvers in PETSc are listed on the following webpage: <https://www.mcs.anl.gov/petsc/documentation/linearsolvertable.html>. Selection of a solver and its options is specified through a configuration file, allowing users to keep abreast of PETSc developments (see Section 5.2).

- Consistent relative permeability and capillary functions

TOUGH3 implements a consistent set of two- and three-phase relative permeability and capillary pressure functions across all EOS modules (see Appendices A and B).

- Hysteresis

Hysteretic behavior of relative permeability and capillary pressure functions in two-phase flow systems can be simulated (Doughty, 2013; Patterson and Falta, 2012), which is critical to predict, for example, phase trapping of supercritical CO₂ (Doughty, 2007) or the mobilization of NAPLs during remediation (see Appendices A and B).

- Brine properties

TOUGH3 includes new brine properties that are internally consistent, derived from Driesner and Heinrich (2007) and Driesner (2007). These were originally implemented in EWASG (Battistelli, 2012), but are available as an option in TOUGH3 in all the EOS modules that include NaCl (“salt”) as one of the fluid components. The options for the brine properties can be selected in data block **SELEC**, and are described in the addendum for EWASG.

- Biodegradation

The capability of simulating multiple biodegradation reactions in TMVOCBio (Battistelli, 2004) is implemented in the TMVOC module in TOUGH3. Detailed description of the input format for biodegradation reactions is presented along with sample problems in Jung and Battistelli (2016).

- Printout format (Block **OUTPU**)

A new input block **OUTPU** allows the user can specify variables for printout, and choose the format of output files (CSV or TECPLOT) that can either directly be used by standard visualization software, or be easily parsed by post-processing programs (see Section 8.2).

- More options (Block **MOMOP**)

TOUGH3 provides additional options invoked by integer flags on a line following keyword **MOMOP** (see Section 8.2).

- Standardized printouts

If the data block **OUTPU** is *not* used to specify variables for printout, the amount of printout is controlled by variable *KDATA*. The types of element, connection, and generation variables printed out are standardized in TOUGH3, and are identical for all EOS modules.

- Controlled software development environment

TOUGH3 adopts modern software management strategies, in particular the use of source control management (Mercurial, <http://www.mercurial-scm.org>), bug tracking (Bitbucket, <http://bitbucket.org>), and cross-platform tools to automatically build, test, and package software (CMake, <http://www.cmake.org>). The use of an advanced numerical library such as PETSc makes compiling the code a multi-step process; however, TOUGH3 includes an automatic build system to simplify this process. A shell script handles the compilation process, and additional options can be specified as well for non-standard installation.

3. Code Installation

TOUGH3's source codes are distributed in two directories: ESD-TOUGH3 and ESD-TOUGHLIB. ESD-TOUGH3 contains the source code that provides the main functionalities of TOUGH3, while ESD-TOUGHLIB provides an expanded list of linear solver options compared to previous versions of TOUGH. Below are descriptions on how the code can be built and installed on different platforms. It should be noted that installation instructions may change with platform and modules (also as we enhance and update TOUGH3). Details *should* be taken from the read.me files that come with the distributions. Both C and Fortran compilers are required to compile TOUGH3 using the shell script. If only a Fortran compiler is available, TOUGH3 can be manually compiled in serial.

On Linux-like platforms (e.g., Linux, Mac, Cygwin, and Ubuntu on Windows), TOUGH3 provides an automatic build system that utilizes CMake (<http://www.cmake.org/>) and a shell script `configure.sh` (in folder ESD-TOUGH3). CMake uses simple configuration files placed in each source directory (called `CMakeLists.txt` files) to generate standard build files. The use of an advanced numerical library such as PETSc makes compiling the code a multi-step process; the automatic build system simplifies this process. Executing the provided script file for Linux-like platforms (e.g. `compile_T3_Linux.sh` for Linux) with the EOS option (e.g., 9 for EOS9) will successfully build the code by passing appropriate arguments to the shell script `configure.sh` that does the detailed work of compiling TOUGH3. If necessary, users may need to edit arguments or specify additional options for untested platforms; a list of available options can be displayed by executing `./configure.sh --help`. If TOUGH3 is successfully built, the executable (e.g., `tough3-eos9` for EOS9) will be located in `tough3-install/bin`. The executable takes two arguments in the following order: input file and output file (e.g., `tough3-eos9 input.txt output.txt` for serial execution and `mpiexec -n 4 tough3-eos9 input.txt output.txt` for parallel execution). If no argument is specified, TOUGH3 assumes the input file is `INFILE` and the output file is `OUTPUT`.

4. Governing Equations

4.1. Mass-Balance Equation

Here we describe the basic mass- and energy-balance equations solved by TOUGH3. The equations for nonisothermal, multiphase, multicomponent flows in porous and fractured media can be written in the general form

$$\frac{d}{dt} \int_{V_n} M^\kappa dV_n = \int_{\Gamma_n} \mathbf{F}^\kappa \cdot \mathbf{n} d\Gamma_n + \int_{V_n} q^\kappa dV_n \quad (1)$$

The integration is over an arbitrary subdomain V_n of the flow system under study, which is bounded by the closed surface Γ_n . The quantity M appearing in the accumulation term (left-hand side) represents the mass of component κ (e.g., water, brine, air, CO₂, tracer, radionuclides, VOC) or energy ($\kappa = h$) per volume. \mathbf{F} denotes mass or heat flux, and q denotes sinks and sources. \mathbf{n} is a normal vector on the surface element $d\Gamma_n$, pointing inward into V_n . Each term constituting the equations is described in detail in the following subsections.

4.2. Accumulation Terms

The general form of the mass accumulation term that includes equilibrium sorption on the solid grains is

$$M^\kappa = \phi \sum_{\beta} S_{\beta} \rho_{\beta} X_{\beta}^{\kappa} + (1 - \phi) \rho_R \rho_l X_l^{\kappa} K_d \quad (2)$$

where ϕ is porosity, S_{β} is the saturation of phase β (e.g., $\beta =$ gas, aqueous, non-aqueous phase liquid), ρ_{β} is the density of phase β , and X_{β}^{κ} is the mass fraction of component κ in phase β , ρ_R is the rock grain density, and K_d is the aqueous (liquid) phase distribution coefficient. The total mass of component κ is obtained by summing over the fluid phases β . The heat accumulation term in a multiphase system is

$$M^h = (1 - \phi) \rho_R C_R T + \phi \sum_{\beta} S_{\beta} \rho_{\beta} u_{\beta} \quad (3)$$

where C_R is the specific heat of the rock grains, T is temperature, and u_β is the specific internal energy of phase β .

4.3. Flux Terms

Advective mass flux is a sum of phase fluxes,

$$\mathbf{F}^\kappa \Big|_{adv} = \sum_{\beta} X_{\beta}^{\kappa} \mathbf{F}_{\beta} \quad (4)$$

where individual phase fluxes are given by a multiphase version of Darcy's law:

$$\mathbf{F}_{\beta} = \rho_{\beta} \mathbf{u}_{\beta} = -k \frac{k_{r\beta} \rho_{\beta}}{\mu_{\beta}} (\nabla P_{\beta} - \rho_{\beta} \mathbf{g}) \quad (5)$$

Here, u_{β} is the Darcy velocity (volume flux) of phase β , k is absolute permeability, $k_{r\beta}$ is relative permeability to phase β , μ_{β} is dynamic viscosity, and

$$P_{\beta} = P + P_{c\beta} \quad (6)$$

is the fluid pressure in phase β , which is the sum of the pressure P of a reference phase (usually taken to be the gas phase) and the capillary pressure $P_{c\beta}$ (≤ 0). \mathbf{g} is the vector of gravitational acceleration. Vapor pressure lowering due to capillary and phase adsorption effects can be considered, and is modeled by Kelvin's equation (Edlefsen and Anderson, 1943),

$$P_v(T, S_l) = f_{VPL}(T, S_l) P_{sat}(T) \quad (7)$$

where

$$f_{VPL} = \exp \left[\frac{M_w P_{cl}(S_l)}{\rho_l R (T + 273.15)} \right] \quad (8)$$

is the vapor pressure lowering factor. P_v is the vapor pressure, P_{sat} is the saturated vapor pressure of bulk aqueous phase, P_{cl} is the capillary pressure (i.e., the difference between aqueous and gas phase pressures), M_w is the molecular weight of water, and R is the universal gas constant.

Absolute permeability of the gas phase increases at low pressures according to the relation given by Klinkenberg (1941)

$$k=k_{\infty}\left(1+\frac{b}{P}\right) \quad (9)$$

where k_{∞} is the permeability at “infinite” pressure, and b is the Klinkenberg parameter.

In addition to Darcy flow, mass transport can also occur by diffusion. Diffusive flux is modeled as follows:

$$\mathbf{f}_{\beta}^{\kappa}=-\phi\tau_0\tau_{\beta}\rho_{\beta}d_{\beta}^{\kappa}\nabla X_{\beta}^{\kappa} \quad (10)$$

where d_{β}^{κ} is the molecular diffusion coefficient for component κ in phase β , and $\tau_0\tau_{\beta}$ is the tortuosity, which includes a porous medium dependent factor τ_0 and a coefficient that depends on phase saturation S_{β} , $\tau_{\beta} = \tau_{\beta}(S_{\beta})$. For general two-phase conditions, the total diffusive flux is then given by

$$\mathbf{f}^{\kappa}=-\Sigma_l^{\kappa}\nabla X_l^{\kappa}-\Sigma_g^{\kappa}\nabla X_g^{\kappa} \quad (11)$$

where $\Sigma_{\beta}^{\kappa}=\phi\tau_0\tau_{\beta}\rho_{\beta}d_{\beta}^{\kappa}$ is an effective diffusion coefficient in phase β . We have used this pragmatic approach because it is not possible to formulate a model for multiphase diffusion that would be accurate under all circumstances. The basic Fick law works well for diffusion of tracer solutes that are present at low concentrations in a single-phase aqueous solution at rest with respect to the porous medium².

² Many subtleties and complications can arise when multiple components diffuse in a multiphase flow system. Effective diffusivities in general may depend on all concentration variables, leading to nonlinear behavior especially when some components are present in significant (non-tracer) concentrations. Additional nonlinear effects arise from the dependence of tortuosity on phase saturations, and from coupling between advective and diffusive transport. For gases, the Fickian model has serious limitations even at low concentrations, which prompted the development of the “dusty gas” model that entails a strong coupling between advective and diffusive transport (Mason and Malinauskas, 1983; Webb, 1998) and accounts for molecular streaming effects (Knudsen diffusion) that become very important when the mean free path of gas molecules is comparable to pore sizes. Further complications arise for components that are both soluble and volatile, in which case diffusion in aqueous and gaseous phases may be strongly coupled via phase partitioning effects. An extreme case is the well-known enhancement of vapor diffusion in partially saturated

Several models are available to describe the dependence of tortuosity on porous medium properties and phase saturation. For the relative permeability model, tortuosity will be taken as $\tau_0 \tau_\beta(S_\beta) = \tau_0 k_{r\beta}(S_\beta)$ with the user-specified porous medium dependent factor τ_0 . The Millington and Quirk (1961) model, which has frequently been used for soils (Jury et al., 1983; Falta et al., 1989), yields non-zero tortuosity coefficients as long as phase saturation is non-zero³.

$$\tau_0 \tau_\beta = \phi^{1/3} S_\beta^{10/3} \quad (12)$$

For the constant diffusivity formulation, $\tau_0 \tau_\beta = S_\beta$ will be used. This alternative corresponds to the formulation for gas diffusion in the original version of TOUGH2. In the absence of phase partitioning and adsorptive effects, it amounts to effective diffusivity being approximately equal to d_β^κ , independent of saturation. This can be seen by noting that the accumulation term in the phase β contribution to the mass balance equation for component κ is given by $\phi S_\beta \rho_\beta X_\beta^\kappa$, approximately canceling out the $\phi S_\beta \rho_\beta$ coefficient in the diffusive flux.

TOUGH3 can model the pressure and temperature dependence of gas phase diffusion coefficients by the following equation (Vargaftik, 1975; Walker et al., 1981).

$$d_g^\kappa(P, T) = d_g^\kappa(P_0, T_0) \frac{P_0}{P} \left[\frac{T + 273.15}{273.15} \right]^\theta \quad (13)$$

media, which is attributed to pore-level phase change effects (Cass et al., 1984; Webb and Ho, 1998a, b). These alternative models are not implemented in TOUGH3.

³ It stands to reason that diffusive flux should vanish when a phase becomes discontinuous at low saturations, suggesting that saturation-dependent tortuosity should be related to relative permeability, i.e., $\tau_\beta(S_\beta) \approx k_{r\beta}(S_\beta)$. However, for components that partition between liquid and gas phases, a more complex behavior may be expected. For example, consider the case of a volatile and water-soluble compound diffusing under conditions of low gas saturation where the gas phase is discontinuous. In this case we have $k_{rg}(S_g) = 0$ (because $S_g < S_{gr}$), and $k_{rl}(S_l = 1 - S_g) < 1$, so that a model equating saturation-dependent tortuosity to relative permeability would predict weaker diffusion than in single-phase liquid conditions. For compounds with “significant” volatility this would be unrealistic, as diffusion through isolated gas pockets would tend to enhance overall diffusion relative to single-phase liquid conditions.

At standard conditions of $P_0 = 1 \text{ atm} = 1.01325 \text{ bar}$ and $T_0 = 0^\circ\text{C}$, the diffusion coefficient for vapor-air mixtures has a value of $2.13 \times 10^{-5} \text{ m}^2/\text{s}$; parameter θ for the temperature dependence is 1.80. Presently there are no provisions for inputting different values for the parameter θ of temperature dependence for different gas phase components. Diffusion coefficients for the non-gaseous phases are taken as constants, with no provisions for temperature dependence of these parameters.

Heat flux includes conductive, convective, and radiative components:

$$\mathbf{F}^h = -\lambda \nabla T + \sum_{\beta} h_{\beta} \mathbf{F}_{\beta} + f_{\sigma} \sigma_0 \nabla T^4 \quad (14)$$

where λ is the effective thermal conductivity, and h_{β} is the specific enthalpy in phase β , f_{σ} is the radiant emittance factor, and σ_0 is the Stefan-Boltzmann constant.

4.4. Sink and Source Terms

Sinks and sources are introduced by specifying the mass production ($q < 0$) or injection ($q > 0$) rates of fluids as well as heat flow. Any of the mass components may be injected in an element at a constant or time-dependent mass rate, and the specific enthalpy of the injected fluid may be specified as well. Heat sources/sinks (with no mass injection) may be either constant or time dependent.

In the case of fluid production, a total mass production rate needs to be specified. The phase composition of the produced fluid may be determined by the relative phase mobilities in the source element. Alternatively, the produced phase composition may be specified to be the same as the phase composition in the producing element. In either case, the mass fractions of the components in the produced phases are determined by the corresponding component mass fractions in the producing element.

TOUGH3 also includes different options for considering wellbore flow effects: a well on deliverability against specified bottomhole or wellhead pressure, or coupled wellbore flow. Details are discussed below.

Deliverability Model

Production wells may operate on deliverability against a prescribed flowing bottomhole pressure, P_{wb} , with a productivity index PI (Coats, 1977). With this option, the mass production rate of phase β from a grid block with phase pressure $P_\beta > P_{wb}$ is

$$q_\beta = \frac{k_{r\beta}}{\mu_\beta} \rho_\beta \cdot PI \cdot (P_\beta - P_{wb}) \quad (15)$$

For steady radial flow, the productivity index of layer l is given by (Coats, 1977; Thomas, 1982)

$$(PI)_l = \frac{2\pi(k\Delta z_l)}{\ln(r_e/r_w) + s - 1/2} \quad (16)$$

Here, Δz_l denotes the layer thickness, $(k\Delta z_l)$ is the permeability-thickness product in layer l , r_e is the grid block radius, r_w is the well radius, and s is the skin factor. If the well is producing from a grid block which does not have cylindrical shape, an approximate PI can be computed by using an effective radius

$$r_e = \sqrt{A/\pi} \quad (17)$$

where A is the grid block area; e.g., $A = \Delta x \cdot \Delta y$ for an areal Cartesian grid. More accurate expressions for specific well patterns and grid block shapes have been given in the literature (e.g., Peaceman, 1977, 1982; Coats and Ramesh, 1982).

The rate of production for mass component κ is

$$\hat{q}^\kappa = \sum_\beta X_\beta^\kappa q_\beta \quad (18)$$

For wells that are screened in more than one layer (element), the flowing wellbore pressure P_{wb} can be corrected to approximately account for gravity effects according to the depth-dependent flowing density in the wellbore. Assume that the open interval extends from layer $l = 1$ at the bottom to $l = L$ at the top. The flowing wellbore pressure in layer l , $P_{wb,l}$, is obtained from the wellbore pressure in layer $l + 1$ immediately above it by means of the following recursion formula

$$P_{wb,l} = P_{wb,l+1} + \frac{g}{2} \left(\rho_l^f \Delta z_l + \rho_{l+1}^f \Delta z_{l+1} \right) \quad (19)$$

Here, g is acceleration of gravity, and ρ_l^f is the flowing density in the tubing opposite layer l . Flowing densities are computed using a procedure given by Coats (private communication, 1982). If wellbore pressure were zero, we would obtain the following volumetric production rate of phase β from layer l .

$$r_{l,\beta} = \left(\frac{k_{r\beta}}{\mu_\beta} \right)_l (PI)_l P_{l,\beta} \quad (20)$$

The total volumetric flow rate of phase β opposite layer l is, for zero wellbore pressure

$$r_{l,\beta}^T = \sum_{m=1}^l r_{m,\beta} \quad (21)$$

From this, we obtain an approximate expression for flowing density opposite layer l which can be used in Eq. (19).

$$\rho_l^f = \sum_{\beta} \rho_{l,\beta} r_{l,\beta}^T / \sum_{\beta} r_{l,\beta}^T \quad (22)$$

During fluid production or injection, the rate of heat removal or injection is determined by

$$\hat{q}^h = \sum_{\beta} q_{\beta} h_{\beta} \quad (23)$$

where h_{β} is the specific enthalpy of phase β .

Coupled Wellbore Flow

Geothermal production wells typically operate at (nearly) constant wellhead pressures. However, as flow rate and flowing enthalpy change with time, wellbore pressure gradients and flowing bottomhole pressures will also change. TOUGH3 cannot directly describe production from geothermal wells by solving equations for flow in the reservoir and flow in the wellbore in a

fully coupled manner⁴. TOUGH3 uses an alternative approach (Murray and Gunn, 1993) in which the wellbore and reservoir simulations are performed separately. This can be accomplished by running a wellbore flow simulator prior to the reservoir simulation for a range of flow rates q and flowing enthalpies h in order to generate a table of flowing bottomhole pressures P_{wb} .

$$P_{wb} = P_{wb}(q, h; P_{wh}, z, r_w) \quad (24)$$

In addition to the functional dependence on q and h , flowing bottomhole pressure is dependent on a number of well parameters. These include wellhead pressure P_{wh} , feed zone depth z , wellbore radius r_w , friction factors, and possibly others. By interpolating on these tabular data, Eq. (24) can be directly inserted into the well source term, Eq. (15). Reservoir flow equations that include a quasi-steady approximation to wellbore flow can then be solved with little added computational expense compared to the case where no wellbore flow effects are considered. Advantages of representing wellbore flow effects through tabular data include increased robustness and computational efficiency. It also makes it possible to use different wellbore simulators and two-phase flow correlations without any programming changes in the reservoir simulation code.

We have incorporated a tabular interpolation scheme for dynamic changes of flowing bottomhole pressure into TOUGH3. Flowing enthalpy at the feed zone is known from phase mobilities and enthalpies calculated by the reservoir simulator. The unknown well flow rate and flowing bottomhole pressure are obtained by Newton-Raphson iteration on

$$R(q) \equiv q - \left(\sum \frac{k_{r\beta}}{\mu_\beta} \rho_\beta \right) \cdot PI \cdot (P - P_{wb}(q, h)) = 0 \quad (25)$$

The iterative solution of Eq. (25) was embedded in the outer (Newtonian) iteration performed by TOUGH3 on the coupled mass and heat balance equations. Additional computational work in comparison to conventional simulations with constant downhole pressure is insignificant.

⁴ The fully coupled approach was taken by Hadgu et al. (1995) who coupled the reservoir simulator TOUGH (Pruess, 1987) with the wellbore simulator WFS (Hadgu and Freeston, 1990). T2Well (Pan and Oldenburg, 2014) is also a coupled wellbore-reservoir simulator, which extends TOUGH2 to calculate the flow in both the wellbore and the reservoir simultaneously by introducing a special wellbore sub-domain. T2Well uses the drift-flux model and related conservation equations for describing transient two-phase nonisothermal flow in the wellbore sub-domain. T2Well will be implemented in TOUGH3 at a future date.

The coupled wellbore flow capability as coded in TOUGH3 is limited to wells with a single feed zone and can only handle wellbore pressure effects from changing flow rates and enthalpies. Effects from changing fluid composition, as, e.g., variable non-condensable gas content, are not modeled at present.

4.5. Semi-Analytical Conductive Heat Exchange

TOUGH3 provides options for modeling linear or radial conductive heat exchange with geologic formations where no fluid exchange is considered, using semi-analytical methods. This is a much more efficient alternative to the approach that simply extends the computational grid into those formations and assigns small or vanishing permeability to them. Even to achieve modest accuracy, the number of grid blocks in the heat flow domain could easily become comparable to, or even larger than, the number of grid blocks in the fluid flow domain, leading to a very inefficient calculation. The semi-analytical methods require no grid blocks outside of the fluid flow domain, and permit better accuracy for short- and long-term heat exchange. Note that radial *and* linear semi-analytical heat exchange cannot be combined in the current version of TOUGH3.

Linear Heat Exchange between a Reservoir and Confining Beds

TOUGH3 uses the method of Vinsome and Westerveld (1980), which gives excellent accuracy for heat exchange between reservoir fluids and confining beds⁵ such as may arise in geothermal injection and production operations. Observing that the process of heat conduction tends to dampen out temperature variations, Vinsome and Westerveld suggested that caprock or baserock temperatures would vary smoothly even for strong and rapid temperature changes at the boundary of the conduction zone. Arguing that heat conduction perpendicular to the conductive boundary is more important than parallel to it, they proposed to represent the temperature profile in a semi-infinite conductive layer by means of a simple trial function, as follows:

$$T(x,t) - T_i = (T_f - T_i + px + qx^2) \exp(-x/d) \quad (26)$$

⁵ This method is developed for calculating heat losses from the reservoir to caprock or baserock and predicting the temperature profile in a semi-infinite, homogeneous, conductive half-space confining bed.

Here, x is the distance from the boundary, T_i is initial temperature in the cap- or base-rock (assumed uniform), T_f is the time-varying temperature at the cap- or base-rock boundary, p and q are time-varying fit parameters, and d is the penetration depth for heat conduction, given by

$$d = \sqrt{\Theta t} / 2 \quad (27)$$

where $\Theta = \lambda / \rho C$ is the thermal diffusivity, λ the thermal conductivity, ρ the density of the medium, and C the specific heat. In the context of a finite-difference simulation of nonisothermal flow, each grid block in the top and bottom layers of the computational grid will have an associated temperature profile in the adjacent impermeable rock as given by Eq. (26). The coefficients p and q will be different for each grid block; they are determined concurrently with the flow simulation from the physical constraints of (1) continuity of heat flux across the boundary, and (2) energy conservation for the reservoir/confining layer system.

Radial Heat Exchange between Fluids in a Wellbore and the Surrounding Formation

Radial, conductive heat exchange between fluids in a discretized wellbore and the formation is calculated using a semi-analytical, time-convolution method. Note that the time-dependent temperature evolution in the fully-discretized wellbore is calculated numerically. 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) 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 as the product of a heat transfer function and the temperature using Eq. (28) for small values of the dimensionless time $t_d = \alpha t / r_0^2$, where α is the thermal diffusivity and r_0 is the wellbore radius (m), and Eq. (29) for large values of t_d :

$$q = f_1(t_d) \cdot \Delta T = \frac{\lambda \Delta T}{r_0} \left\{ (\pi t_d)^{-0.5} + \frac{1}{2} - \frac{1}{4} \left(\frac{t_d}{\pi} \right)^{0.5} + \frac{1}{8} t_d - \dots \right\} \quad (28)$$

$$q = f_2(t_d) \cdot \Delta T = \frac{2\lambda \Delta T}{r_0} \left\{ \frac{1}{\ln(4t_d) - 2\gamma} - \frac{\gamma}{[\ln(4t_d) - 2\gamma]^2} - \dots \right\} \quad (29)$$

Here, λ is thermal conductivity ($\text{W m}^{-1} \text{K}^{-1}$), 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 as

$$q_{total} = \sum_{i=1}^{d-1} f(t_d - t_i) \cdot \Delta T(t_i) \quad (30)$$

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 $t_d - t_i \leq 2.8$, 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., $\Delta T(t_i) = T_w(t_i) - T_f(t_{i-1})$.

5. Numerical Methods

5.1. Space and Time Discretization

TOUGH3 uses an integral finite difference method for space discretization, and first-order fully implicit time differencing. The resulting strongly coupled, nonlinear algebraic equations are solved simultaneously using Newton-Raphson iterations for each time step, which involves the calculation of a Jacobian matrix and the solution of a set of linear equations. Time steps are automatically adjusted during a simulation run, depending on the convergence rate of the iteration process. Newton-Raphson increment weighting can also be adjusted if the iterations oscillate.

The continuum equations Eq. (1) are discretized in space using the integral finite difference method (IFD; Edwards, 1972; Narasimhan and Witherspoon, 1976). Introducing appropriate volume averages, we have

$$\int_{V_n} M dV = V_n M_n \quad (31)$$

where M is a volume-normalized extensive quantity, and M_n is the average value of M over V_n .

Surface integrals are approximated as a discrete sum of averages over surface segments A_{nm} :

$$\int_{\Gamma_n} \mathbf{F}^k \cdot \mathbf{n} d\Gamma = \sum_m A_{nm} F_{nm} \quad (32)$$

Here, F_{nm} is the average value of the (inward) normal component of \mathbf{F} over the surface segment A_{nm} between volume elements V_n and V_m . The discretization approach used in the integral finite difference method and the definition of the geometric parameters are illustrated in Figure 1.

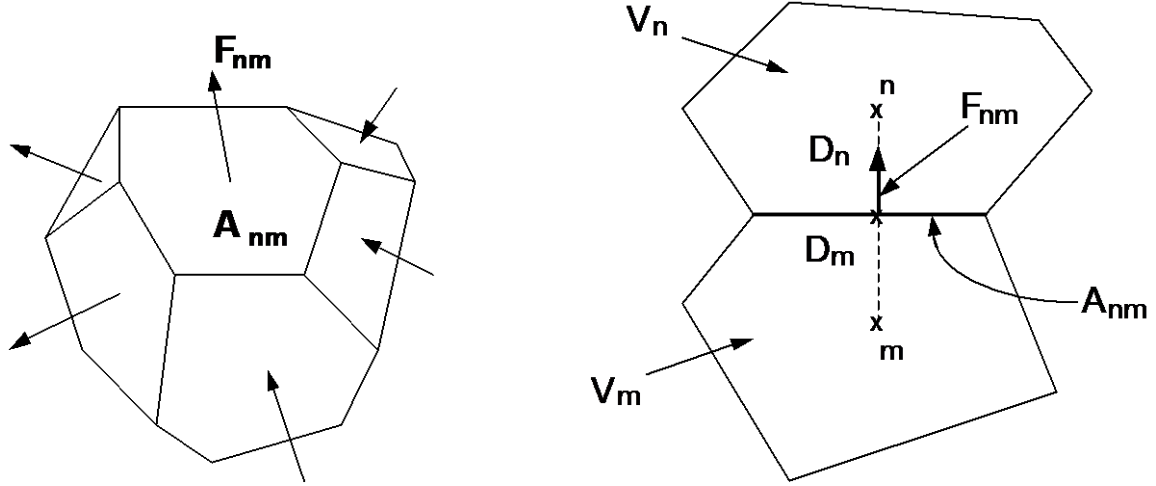


Figure 1. Space discretization and geometry data in the integral finite difference method.

The discretized flux is expressed in terms of averages over parameters for elements V_n and V_m . For the basic Darcy flux term, Eq. (5), we have

$$F_{\beta, nm} = -k_{nm} \left[\frac{k_{r\beta} \rho_\beta}{\mu_\beta} \right]_{nm} \left[\frac{P_{\beta, n} - P_{\beta, m}}{D_{nm}} - \rho_{\beta, nm} g_{nm} \right] \quad (33)$$

where the subscripts (nm) denote a suitable averaging at the interface between grid blocks n and m (such as interpolation, harmonic weighting, and upstream weighting, which will be discussed in Section 5.2). $D_{nm} = D_n + D_m$ is the distance between the nodal points n and m , and g_{nm} is the component of gravitational acceleration in the direction from m to n .

Space discretization of diffusive flux in multiphase conditions raises some subtle issues. A finite difference formulation for total diffusive flux, Eq. (11), may be written as

$$(\mathbf{f}^\kappa)_{nm} = -(\Sigma_l^\kappa)_{nm} \frac{(X_l^\kappa)_m - (X_l^\kappa)_n}{D_{nm}} - (\Sigma_g^\kappa)_{nm} \frac{(X_g^\kappa)_m - (X_g^\kappa)_n}{D_{nm}} \quad (34)$$

This expression involves the as yet unknown diffusive strength coefficients $(\Sigma_l^\kappa)_{nm}$ and $(\Sigma_g^\kappa)_{nm}$ at the interface, which must be expressed in terms of the strength coefficients in the participating grid blocks. Invoking conservation of diffusive flux across the interface between two grid blocks leads

in the usual way to the requirement of harmonic weighting of the diffusive strength coefficients. However, such weighting may in general not be applied separately to the diffusive fluxes in gas and liquid phases, because these may be strongly coupled by phase partitioning effects. This can be seen by considering the extreme case of diffusion of a water-soluble and volatile compound from a grid block in single-phase gas conditions to an adjacent grid block which is in single-phase liquid conditions. Harmonic weighting applied separately to liquid and gas diffusive fluxes would result in either of them being zero, because for each phase effective diffusivity is zero on one side of the interface. Thus total diffusive flux would vanish in this case, which is unphysical. In reality, tracer would diffuse through the gas phase to the gas-liquid interface, would establish a certain mass fraction in the aqueous phase by dissolution, and would then proceed to diffuse away from the interface through the aqueous phase. Similar arguments can be made in the less extreme situation where liquid saturation changes from a large to a small value rather than from 1 to 0, as may be the case in the capillary fringe, during infiltration events, or at fracture-matrix interfaces in variably saturated media.

TOUGH3 features a fully coupled approach in which the space-discretized version Eq. (34) of the total multiphase diffusive flux Eq. (11) is re-written in terms of an effective multiphase diffusive strength coefficient and a single mass fraction gradient. Choosing the liquid mass fraction for this we have

$$(\mathbf{f}^\kappa)_{nm} = \left\{ \Sigma_l^\kappa + \Sigma_g^\kappa \frac{(X_g^\kappa)_m - (X_g^\kappa)_n}{(X_l^\kappa)_m - (X_l^\kappa)_n} \right\} \frac{(X_l^\kappa)_m - (X_l^\kappa)_n}{D_{nm}} \quad (35)$$

where the gas phase mass fraction gradient has been absorbed into the effective diffusive strength term (in braces). As is well known, flux conservation at the interface then leads to the requirement of harmonic weighting for the full effective strength coefficient. In order to be able to apply this scheme to the general case where not both phases may be present on both sides of the interface, we always define both liquid and gas phase mass fractions in all grid blocks, regardless of whether both phases are present. Mass fractions are assigned in such a way as to be consistent with what would be present in an evolving second phase. This procedure is applicable to all possible phase combinations, including the extreme case where conditions at the interface change from single-phase gas to single-phase liquid. Note that, if the diffusing tracer exists in just one of the two phases, harmonic weighting of the strength coefficient in Eq. (35) will reduce to harmonic weighting of either Σ_l^κ or Σ_g^κ . The simpler scheme of separate harmonic weighting for individual phase diffusive fluxes is retained as an option.

Substituting Eqs. (31) and (32) into the governing Eq. (1), a set of first-order ordinary differential equations in time is obtained:

$$\frac{dM_n^\kappa}{dt} = \frac{1}{V_n} \sum_m A_{nm} F_{nm}^\kappa + q_n^\kappa \quad (36)$$

Time is discretized as a first-order finite difference, and the flux and sink and source terms on the right-hand side of Eq. (36) are evaluated at the new time level, $t^{k+l} = t^k + \Delta t$, to obtain the numerical stability needed for an efficient calculation of multiphase flow. This treatment of flux terms is known as “fully implicit,” because the fluxes are expressed in terms of the unknown thermodynamic parameters at time level $k+1$, so that these unknowns are only implicitly defined in the resulting equations (see, e.g., Peaceman, 1977). The time discretization results in the following set of coupled nonlinear, algebraic equations

$$\begin{aligned} R_n^{\kappa,k+1} &= M_n^{\kappa,k+1} - M_n^{\kappa,k} - \frac{\Delta t}{V_n} \left\{ \sum_m A_{nm} F_{nm}^{\kappa,k+1} + V_n q_n^{\kappa,k+1} \right\} \\ &= 0 \end{aligned} \quad (37)$$

where we have introduced residuals $R_n^{\kappa,k+1}$. For each volume element (grid block) V_n , there are NEQ mass and heat balance equations ($\kappa = 1, 2, \dots, NEQ$, where NEQ represents the number of equations per grid block). For a flow system with NEL grid blocks, Eq. (37) represents a total of $NEL \times NEQ$ coupled nonlinear equations. The unknowns are the $NEL \times NEQ$ independent primary variables $\{x_i; i = 1, \dots, NEL \times NEQ\}$ which completely define the state of the flow system at time t^{k+l} . These equations are solved by Newton-Raphson iteration, which is implemented as follows. We introduce an iteration index p and expand the residuals $R_n^{\kappa,k+1}$ in Eq. (37) at iteration step $p + 1$ in a Taylor series in terms of the residuals at index p .

$$R_n^{\kappa,k+1}(x_{i,p+1}) = R_n^{\kappa,k+1}(x_{i,p}) + \sum_i \left. \frac{\partial R_n^{\kappa,k+1}}{\partial x_i} \right|_p (x_{i,p+1} - x_{i,p}) + \dots = 0 \quad (38)$$

Retaining only terms up to first order, we obtain a set of $NEL \times NEQ$ linear equations for the increments $(x_{i,p+1} - x_{i,p})$:

$$-\sum_i \left. \frac{\partial R_n^{\kappa,k+1}}{\partial x_i} \right|_p (x_{i,p+1} - x_{i,p}) = R_n^{\kappa,k+1}(x_{i,p}) \quad (39)$$

All terms $\partial R_n / \partial x_i$ in the Jacobian matrix are evaluated by numerical differentiation. Eq. (39) is solved by the linear equations solver selected. Iteration is continued until the residuals $R_n^{\kappa,k+1}$ are reduced below a preset convergence tolerance.

$$\left| \frac{R_{n,p+1}^{\kappa,k+1}}{M_{n,p+1}^{\kappa,k+1}} \right| \leq \varepsilon_1 \quad (40)$$

The default (relative) convergence criterion is $\varepsilon_1 = 10^{-5}$. When the accumulation terms are smaller than ε_2 (default $\varepsilon_2 = 1$), an absolute convergence criterion is imposed,

$$|R_n^{\kappa,k+1}| \leq \varepsilon_1 \cdot \varepsilon_2 \quad (41)$$

Convergence is usually attained in 3 to 4 iterations. If convergence cannot be achieved within a certain number of iterations (default 8), the time step size Δt is reduced and a new iteration process is started.

It is appropriate to add some comments about this space discretization technique. The entire geometric information of the space discretization in Eq. (37) is provided in the form of a list of grid block volumes V_n , interface areas A_{nm} , nodal distances D_{nm} and components g_{nm} of gravitational acceleration along nodal lines. There is no reference whatsoever to a global system of coordinates, or to the dimensionality of a particular flow problem. The discretized equations are in fact valid for arbitrary irregular discretizations in one, two or three dimensions, and for porous as well as for fractured media. This flexibility should be used with caution, however, because the accuracy of solutions depends upon the accuracy with which the various interface parameters in equations such as Eq. (33) can be expressed in terms of average conditions in grid blocks. A general requirement is that there exists approximate thermodynamic equilibrium in (almost) all grid blocks at (almost) all times (Pruess and Narasimhan, 1985). For systems of regular grid blocks referenced to global coordinates (such as $r - z$ or $x - y - z$), Eq. (37) is identical to a conventional finite difference formulation (e.g., Peaceman, 1977; Moridis and Pruess, 1992).

5.2. Interface Weighting Schemes

To obtain a reasonably accurate and efficient solution, particularly for multiphase fluid and heat flow problems, a proper interface weighting scheme should be used. It is well known that for

single-phase flow, the appropriate interface weighting scheme for absolute permeability is harmonic weighting. For two-phase flow, the added problem of relative permeability weighting arises. It has been established that for transient flow problems in homogeneous media, relative permeability must be upstream weighted, or else phase fronts may be propagated with erroneous speed (Aziz and Settari, 1979). Studies at the Lawrence Berkeley National Laboratory have shown that for transient two-phase problems in composite (layered) media, both absolute and relative permeability must be fully upstream weighted to avoid the possibility of gross errors (Tsang and Pruess, 1990; Wu et al., 1993). The applicable weighting schemes for different flow problems are summarized in Figure 2. There is no single weighting scheme for general two-phase flows in composite media that would at the same time preserve optimal accuracy for single-phase or steady two-phase flows.

transient two-phase flow	
uniform medium	composite medium
k (constant) k_r (upstream)	k k_r } upstream
steady two-phase flow	
k k_r harmonic	
single-phase flow	
k harmonic	
k_r (none)	

Figure 2. Weighting procedures for absolute (k) and relative permeability (k_r) at grid block interfaces.

For modeling of fracture-matrix interaction, a different weighting scheme can be applied. If a zero nodal distance is specified, the absolute (and relative, if $MOP2(7) > 0$) permeability from the other element is used for assignment of interface mobilities.

Another interesting problem is the weighting scheme for interface densities. For proper modeling of gravity effects, it is necessary to define interface density as the arithmetic average between the densities of the two adjacent grid blocks, regardless of nodal distances from the interface. An unstable situation may arise when phases (dis-)appear, because interface density may then have to be “switched” to the upstream value when the phase in question is not present in the downstream block. For certain flow problems, spatial interpolation of densities may provide more accurate answers.

Issues of interface weighting and associated discretization errors are especially important when non-uniform or irregular grids are used. Additional complications related to interface weighting arise in flow problems that involve hydrodynamic instabilities. Examples include immiscible displacements with unfavorable mobility ratio where a less viscous fluid displaces a fluid of higher viscosity (viscous instability), and flow problems where a denser fluid invades a zone with less dense fluid from above (gravity instability). These instabilities can produce very large grid orientation errors, i.e., simulated results can depend strongly on the orientation of the computational grid (Yanosik and McCracken, 1979; Pruess and Bodvarsson, 1983; Pruess, 1991c; Brand et al., 1991).

In some cases, it may be advisable to use higher-order differencing schemes. Grid orientation effects can be reduced by using 7-point or 9-point differencing instead of the common 5-point “stencil,” to achieve a higher degree of rotational invariance in the finite difference approximations of the fundamental differential operators. In the integral finite difference method, these higher-order schemes can be implemented through preprocessing of geometry data, without any coding changes, by assigning additional flow connections with appropriate weighting factors between elements of the computational grid (Pruess and Bodvarsson, 1983; Pruess, 1991c). However, reduction or elimination of grid orientation effects as such does not necessarily achieve a better numerical approximation. It may just amount to reducing the anisotropy of space discretization errors but not their magnitude, creating an illusion of a better approximation by making space discretization effects less obvious (Pruess, 1991c).

5.3. Linear Equation Solvers

TOUGH3 offers a choice of linear solvers: the internal serial linear solvers of TOUGH2, the parallel Aztec solvers (Tuminaro et al., 1999; used in TOUGH2-MP), and all the serial and parallel solvers available in Portable, Extensible Toolkit for Scientific Computation (PETSc) (Balay et al., 2016). TOUGH3 provides a common interface to all three sources of linear solvers, allowing users to experiment with different solvers and their preconditioners, and to determine the most efficient method for the problem of interest. In addition, there is no restriction on using the Aztec and PETSc solvers in serial mode. Moreover, the internal serial solvers included in TOUGH3 are updated with a new sorting algorithm (Navarro, 2015), which results in a performance gain up to 30% when compared with the serial solvers in TOUGH2 (actual performance gains are problem-specific). The different linear equation solvers included in the TOUGH3 package can be selected by means of parameter *MOP(21)* in data block **PARAM.1** (see Section 8.2).

5.3.1. Internal serial solvers

The most reliable linear equation solvers are based on direct methods, while the performance of iterative techniques tends to be problem-specific and lacks the predictability of direct solvers. The robustness of direct solvers comes at the expense of large storage requirements and execution times that typically increase proportional to N^3 , where N is the number of equations to be solved. In contrast, iterative solvers have much lower memory requirements, and their computational work increases much less rapidly with problem size, approximately proportional to N^ω , with $\omega \approx 1.4 - 1.6$ (Moridis and Pruess, 1995). For large problems (especially 3D problems with more than several thousand volume elements), iterative conjugate gradient (CG) type solvers are therefore the method of choice. Technical details of the serial methods and their performance can be found in Moridis and Pruess (1998).

TOUGH3 includes several internal serial solvers: one direct solver, LUBAND, and four iterative solvers, DSLUBC (bi-conjugate gradient solver), DSLUCS (Lanczos-type bi-conjugate gradient solver), DSLUGM (generalized minimum residual solver), and DLUSTB (stabilized bi-conjugate gradient solver). By default, TOUGH3 uses DSLUCS with incomplete LU-factorization as preconditioner for serial runs. Users need to beware that iterative methods may fail for matrices with special features, such as many zeros on the main diagonal, large numerical range of matrix elements, and nearly linearly dependent rows or columns. Depending on features of the problem at hand, appropriate matrix preconditioning may be essential to achieve convergence. Poor accuracy

of the linear equation solution will lead to deteriorated convergence rates for the Newton-Raphson iteration, and an increase in the number of iterations for a given time step. In severe cases, time steps may fail with residuals either stagnating or wildly fluctuating. Users experiencing difficulties with the default settings are encouraged to experiment with the various solvers and preconditioners included in the TOUGH3 package.

All iterative solvers use incomplete LU-factorization as a preconditioner. Alternative preconditioners and iteration parameters can be selected by means of an optional data block **SOLVR** in the TOUGH3 input file. If **SOLVR** is present, its specifications will override the choices made by *MOP* (21). The default preconditioning with **SOLVR** is also incomplete LU-factorization.

The solver DLUSTB implements the BiCGSTAB(m) algorithm (Sleijpen and Fokkema, 1993), an extension of the BiCGSTAB algorithm of van der Vorst (1992). DLUSTB provides improved convergence behavior when iterations are started close to the solution, i.e., near steady state. The preconditioning algorithms (the Z-preprocessors) can cope with difficult problems in which many of the Jacobian matrix elements on the main diagonal are zero. An example is the “two-waters” problem with EOS1 in which typically 2/3 of the elements in the main diagonal are zero. Moridis and Pruess (1998) show that this type of problem can be solved by choosing appropriate pre-processing options.

5.3.2. PETSc solvers

TOUGH3 provides access to PETSc-based sparse and dense linear solvers. In many cases, the PETSc solvers are more efficient than the original serial solvers and the Aztec solvers. Since PETSc is being continually updated, users should consult PETSc’s documentation (<http://www.mcs.anl.gov/petsc/documentation/index.html>) and the summary of the linear solvers available from PETSc (<http://www.mcs.anl.gov/petsc/documentation/linearsolvertable.html>) for a list of updated Krylov subspace algorithms and preconditioners⁶.

The current default PETSc solver is the bi-conjugate gradient method with incomplete LU factorization as the preconditioner. Different Krylov subspace algorithms and preconditioners can be selected through the specification of a PETSc option file (`.petscrc`), an example of which is shown in Figure 3. Commonly used options are shown in

⁶ In TOUGH3, the PETSc libraries are configured without any external linear solver packages such as MUMPS and SuperLU. However, these packages can be easily included through minor changes to the configuration script.

Table 2. We do not provide the ability to set these options through the **SOLVR** block since it is difficult to encode the various options into the structure of the **SOLVR** block. In addition, the use of a separate option file ensures easy access to PETSc’s latest linear solvers and its diagnostic tools.

A special case is when using dense linear solver. Then, the user will set `ksp_type` to `none` and `pc_type` to `lu`. Note that LU decomposition only works in serial mode. To use dense linear solvers in parallel, changes can be made to the PETSc configuration scripts to include the parallel dense linear solver MUMPS and SuperLU.

```
# monitor solves
-ksp_monitor
# biconjugate gradient
-ksp_type bicg
# additive Schwarz preconditioner
-pc_type asm
# relative tolerance
-ksp_rtol 1e-7
```

Figure 3. An example of the PETSc option file (`.petscrc`).

Table 2. Commonly used options for PETSc linear solvers

Keywords	Description	Options
<code>ksp_type</code>	Solvers	<code>bicg</code> (bi-conjugate gradient method) <code>bcgsl</code> (stabilized version of bi-conjugate gradient method) <code>cg</code> (conjugate gradient method) <code>minres</code> (minimum residual gradient) <code>gmres</code> (generalized minimal residual method) <code>fgmres</code> (flexible generalized minimal residual method)
<code>pc_type</code>	preconditioner	<code>jacobi</code> (point Jacobi preconditioner) <code>pbjacobi</code> (point block Jacobi preconditioner) <code>bjacobi</code> (block Jacobi preconditioner) <code>asm</code> (restricted additive Schwarz method) <code>ilu</code> (incomplete factorization preconditioner) <code>icc</code> (incomplete Cholesky factorization preconditioner) <code>jacobi</code> (diagonal scaling preconditioning)
<code>ksp_monitor</code>	monitor convergence	N/A. Refer to http://www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/KSP/KSPMonitorSet.html for alternatives.
<code>ksp_rtol</code>	relative convergence	Any real number.

	tolerance	
--	-----------	--

5.3.3. Aztec solvers

TOUGH3 provides an interface to Aztec solvers, but users should note that the version of the Aztec solver in TOUGH3 is no longer actively maintained. The PETSc solvers are in general more efficient than the Aztec solvers, but the Aztec solvers can be more efficient when the number of variables per core is small.

Consistent with the implementation of the PETSc solvers, the options available in Aztec can be set through the specification of an Aztec option file (`.aztecrc`). An example is shown in Figure 4; the format of the Aztec option file is inherited from TOUGH2-MP. TOUGH3 allows the user to modify nearly all Aztec options and parameters described in Section 2 of the Aztec manual (http://www.cs.sandia.gov/CRF/pspapers/Aztec_ug_2.1.ps) through `.aztecrc`. User can also refer to the TOUGH2-MP manual for more details.

```
AZ_solver AZ_bicgstab
AZ_precond AZ_dom_decomp
AZ_tol 1e-6
```

Figure 4: An example of the Aztec option file (`.aztecrc`).

6. Initial and Boundary Conditions

6.1. Initial Conditions and Restarting

Flow systems are initialized by assigning a complete set of primary thermodynamic variables to all grid blocks into which the flow domain is discretized. Various options are available in a hierarchical system, as follows. During the initialization of a TOUGH3 run, all grid blocks are first assigned to default thermodynamic conditions specified in data block **PARAM.4**. The defaults can be overwritten for selected reservoir domains by assigning domain-specific conditions in data block **INDOM**. These in turn may be superseded by thermodynamic conditions assigned to individual grid blocks in data block **INCON**. A disk file `INCON` written to the same specifications as data block **INCON** may also be used.

Different EOS modules use different sets of primary variables, and some EOS modules allow different choices of primary variables for initialization. For a given EOS module, the primary variables used depend on the fluid phase composition. During phase change, primary variables will be automatically switched from one set to another. In multiphase flow systems, therefore, different grid blocks will in general have different sets of primary variables, and must be initialized accordingly.

For many applications, special initial conditions are needed, such as gravity-capillary equilibrium, or steady state corresponding to certain mass and heat flows. This can be realized by performing a series of TOUGH3 runs, in which thermodynamic conditions obtained in one run, and written to disk file `SAVE`, are used as initial conditions in a subsequent continuation run. For example, in a geothermal reservoir simulation, a first run may be made to obtain hydrostatic pressure conditions. These may subsequently be used as boundary conditions in a second run segment to simulate undisturbed natural state conditions with throughflow of mass and heat. This could be followed by a third run segment with fluid production and injection. Restarting of a TOUGH3 run is accomplished by providing file `SAVE` generated in a previous run as file `INCON` for initialization. Usually additional (often minor) adjustments will be made for a restart. For example, different specifications for the number of time steps and desired printout times may be made. Some editing of the `SAVE` and/or `MESH` files may be needed to implement boundary conditions on certain grid blocks. Then, previously calculated pressures on those grid blocks can serve as boundary conditions. In a continuation run, simulation time and time step counters may be continuously incremented, or they may be reset to zero. For example, the latter option will be used when simulating production and injection operations following preparation of a “natural” initial state, which may correspond to a large simulation time.

As far as the internal workings of the code is concerned, there is no difference between a fresh start of a simulation and a restart. The only feature that makes a simulation a continuation run is that the INCON data were generated by a previous TOUGH3 run, rather than having them explicitly provided by the user. File SAVE always ends with a data record with ‘+++’ in the first three columns, followed by one record with restart information (time step and iteration counters, simulation time). To reset all counters to zero, users should simply replace the ‘+++’ with a blank record when using SAVE as file INCON for another TOUGH3 run.

6.2. Neumann Boundary Conditions

Neumann conditions prescribe fluxes of mass or heat crossing boundary surfaces. A special case of Neumann boundary conditions is “no flux,” which in the integral finite difference framework is handled simply by not specifying any flow connections across the boundary. More general flux conditions are prescribed by introducing sinks or sources of appropriate strength into the elements adjacent to the boundary. Time-dependent Neumann conditions can be specified by making sink and source rates time dependent.

6.3. Dirichlet Boundary Conditions

Dirichlet conditions prescribe fixed thermodynamic conditions, such as pressure, temperature, or saturation. Dirichlet conditions can be implemented by assigning very large volumes (e.g., $V = 10^{50} \text{ m}^3$) to grid blocks adjacent to the boundary, so that their thermodynamic conditions do not change at all from fluid or heat exchange with finite-size blocks in the flow domain. For those elements with very large volumes, no mass or energy balance equations are set up, and their primary thermodynamic variables are not included in the list of unknowns. In addition, a small value (such as $D = 10^{-9} \text{ m}$) should be specified for the nodal distance of such blocks, so that boundary conditions are in fact maintained in close proximity to the surface where they are desired, and not at some distance from it.

TOUGH3 no longer supports the so-called “inactive” elements allowed in TOUGH2. If an element with a zero or negative volume is used to indicate that the element and all subsequent elements are to be used for Dirichlet boundary conditions, then the code will internally assign a very large volume for that and all subsequent elements (to be backward compatible with TOUGH2) and print out a warning message. However, if a dummy element with no volume *and* no material

name is used just to separate boundary elements, the code will generate an error message and stop the simulation. In TOUGH3, the user must specify a material name for every element.

Time-dependent Dirichlet conditions can be implemented using two different approaches. The first method consists of placing appropriate, large sinks or sources in boundary elements with large volumes (Moridis and Pruess, 1992). As an example, consider a laboratory experiment reported by Kruger and Ramey (1974) that involved flashing (vaporizing) flow from a sandstone core, with a time-dependent gas pressure boundary condition

$$P_b = P_0 + P_1 t + P_2 t^2 \quad (42)$$

maintained at the outflow end. According to the ideal gas law, the pressure behavior of Eq. (42) can be associated with a time-dependent gas inventory of mass M_b in a volume V as follows.

$$M_b = P_b (mV / RT) \quad (43)$$

where m is the molar weight of the gas, R the universal gas constant, and T absolute temperature. The required time dependence of M_b can be realized by means of a sink/source rate of

$$Q_b = dM_b / dt = (mV / RT) dP_b / dt = (mV / RT) (P_1 + 2P_2 t) \quad (44)$$

Therefore, the desired boundary condition can be implemented by means of a grid block with volume V that is initialized in single-phase gas (e.g., air) conditions, with pressure P_0 , and with a time-dependent sink/source term given by Eq. (44) that would be specified through tabular generation data in data block **GENER**. In order for the pressure conditions in this block to be negligibly affected by heat and mass exchange with the flow domain, the volume V should be made very large, e.g., $V = 10^{50} \text{ m}^3$, just as for time-independent Dirichlet boundary conditions. The time-dependent generation rates from Eq. (44) will then also be very large. The same approach as just outlined for a time-dependent gas pressure boundary can be used to realize other time-dependent Dirichlet conditions, such as prescribed temporal variation of temperature, capillary pressure, and others.

The second, newly implemented method to specify time-dependent Dirichlet boundary conditions is to read a set of time and primary variable data at boundary elements from the input file or a file whose name is given in data block **TIMBC**. Boundary values will be linearly

interpolated between table entries. The volume for these elements should be made very large as well.

6.4. Atmospheric Boundary Conditions

Often flow models need to include the atmosphere as the top boundary. The atmospheric boundary conditions can be specified using Dirichlet boundary elements with very large volumes. For EOS9, no atmospheric boundary element is needed since EOS9 uses Richards' equation to describe variably saturated flow of a single aqueous phase and treats the gas phase as a passive bystander at constant pressure. A single atmospheric element can be connected to all elements at the ground surface, and users may use `AddBound.exe` for this purpose (free software available from the TOUGH website).

Atmospheric pressure and temperature are used as initial condition in atmospheric elements. A relative humidity of 100% is conveniently specified by initializing the atmospheric block as a two-phase point with a liquid saturation smaller than the residual liquid saturation (so relative permeability is zero, preventing liquid flow into soil). For relative humidities less than 100%, a single-phase gas point must be specified with an appropriate air mass fraction (1.0 for dry air; the minimum value depends on vapor pressure, which is a function of temperature; intermediate values determine relative humidity). The material properties, such as relative permeability and capillary pressure functions, for the atmospheric boundary element should be appropriately selected so that (1) liquid relative permeability is zero, (2) gas relative permeability is one, and (3) capillary pressure is zero. In addition, mobilities need to be upstream weighted.

Infiltration can be simulated by specifying the infiltration rate in a row of elements below the atmospheric boundary element using the **GENER** block. Evaporation can be simulated (a) as a binary diffusion process (atmosphere at < 100% relative humidity), (b) by specifying the evaporation rate in a row of elements below the atmospheric boundary element using the **GENER** block, or (c) by assigning a capillary pressure according to Kelvin's equation in the atmospheric element (Ghezzehei et al., 2004).

6.5. Constant Temperature Boundary Conditions

Constant temperature boundary conditions but variable pressure, saturation, and mass fractions can be specified by setting rock grain density (*DROK*) to a very large value (e.g., 1.0E50). This option is useful for cases where the temperature of the injected fluid is maintained constant while the other primary variables defining the system state may vary with time.

7. Architecture of TOUGH3

7.1. Program Structure

The governing equations for multiphase fluid and heat flow have the same mathematical form, regardless of the nature and number of fluid phases and components present. Based on this recognition, TOUGH3 is set up with a modular architecture as shown in Figure 5. The core handles the input and output, sets up the mass and energy balance equations, and solves the strongly coupled, nonlinear algebraic equations using Newton-Raphson iterations for each time step. At each time step, the main flow and transport module interfaces with the EOS module (to obtain secondary parameters, which are thermophysical properties, as a function of the primary variables), and evaluates mass accumulation, flow, and sinks/sources terms (and their derivatives) for a given set of primary variables, which are the solution variables. This modular architecture gives TOUGH3 the flexibility to handle a wide variety of multicomponent, multiphase flow systems, and encourages users to further develop and enhance the code. The nature and properties of specific fluid mixtures enter into the governing equations only through thermophysical parameters, such as fluid density, viscosity, enthalpy, etc., which are provided by an appropriate EOS module.

The coding of TOUGH3 is structured around two large arrays which hold, respectively, the primary thermodynamic variables for all grid blocks (see Section 7.2.1), and the secondary thermophysical parameters needed to assemble the governing flow and transport equations (see Section 7.2.2). These arrays are one-dimensional and hold data for all grid blocks of the computational mesh in sequential fashion. The solution of a flow problem essentially consists of a complete set of thermodynamic variables as a function of time.

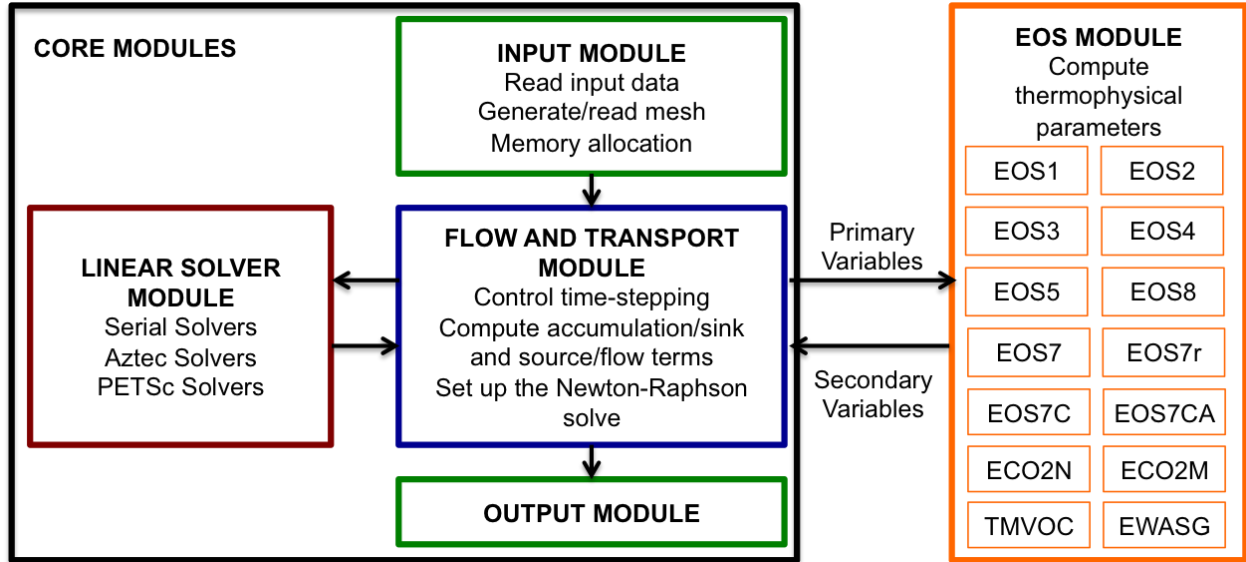


Figure 5. Modular architecture of TOUGH3.

7.2. Array Structure

7.2.1. Primary Thermodynamic Variables

TOUGH3 implements the assumption that locally all phases are in thermodynamic equilibrium. Let us now consider the number of primary thermodynamic variables that are needed to completely specify the thermodynamic state of a flow system consisting of NK components, which are distributed according to local thermodynamic equilibrium among NPH phases. From Gibbs's phase rule it follows that the number of thermodynamic degrees of freedom in such a system is

$$f = NK + 2 - NPH \quad (45)$$

In addition, there are $(NPH - 1)$ saturation degrees of freedom, because the NPH phase saturations (or phase volume fractions) S_β are constrained by the relationship

$$\sum_{\beta=1}^{NPH} S_\beta = 1 \quad (46)$$

The total number of primary thermodynamic variables (degrees of freedom) is therefore

$$\begin{aligned}
 NK1 &= f + NPH - 1 \\
 &= NK + 1
 \end{aligned}
 \tag{47}$$

which is equal to the total number of balance equations per grid block, namely, NK mass balance equations and one energy balance equation. The thermodynamic state of a discretized flow system consisting of NEL volume elements, is then completely specified by a set of $NEL \times NK1$ primary thermodynamic variables, to which correspond an equal number of mass and energy balance equations. For transient flow systems, these primary variables are time-dependent, and they represent the unknowns to be calculated in each time step. All of the $NEL \times NK1$ primary variables are stored sequentially in a one-dimensional array X : first the $NK1$ variables for grid block # 1, then the $NK1$ variables for grid block # 2, and so on (see Figure 6). There are two additional arrays DX and $DELX$ with structure identical to X . While X holds primary variables corresponding to the last successful (converged) time step, DX holds the increments calculated during the Newton-Raphson iteration process. Thus, the latest updated primary variables are $X + DX$. $DELX$ contains the increment to each primary variable used for numerically calculating derivatives.

In the modeling of fluid and heat flow processes with phase change, we need to choose the primary variables that define the thermodynamic state of the system. When a phase appears or disappears, the set of appropriate thermodynamic variables may change. In single-component flows involving water, for example, appropriate thermodynamic variables for describing single-phase conditions (subcooled liquid or superheated steam) are temperature T and pressure P . However, in two-phase conditions, pressure and temperature are not independent, but are related by the vapor-pressure relationship $P = P_{sat}(T)$. (When vapor pressure lowering effects are considered, the more complicated relationship Eq. 7 applies.) To deal with this problem, TOUGH3 uses the variables (pressure, temperature) only for single-phase conditions, and switches to the variables (pressure, saturation) when a transition to two-phase conditions occurs. Experience has proven variable-switching to be a very robust method for treating multiphase systems, and this is the approach used in most TOUGH3 modules, which need to distinguish different phase combinations. The type of primary variables for each phase are simply identified by their positions and numerical values. When the distinction of single-phase from two-phase conditions is not clear, the values in the two-phase condition are shifted (typically by 10). TMVOC and ECO2M also switch the primary variables for each phase combination, but use a separate numerical index to distinguish the seven different phase combinations. Each EOS uses different primary variables and different switching procedures for phase transitions; users need to refer to the respective addenda for the EOS of interest.

7.2.2. Secondary Parameters

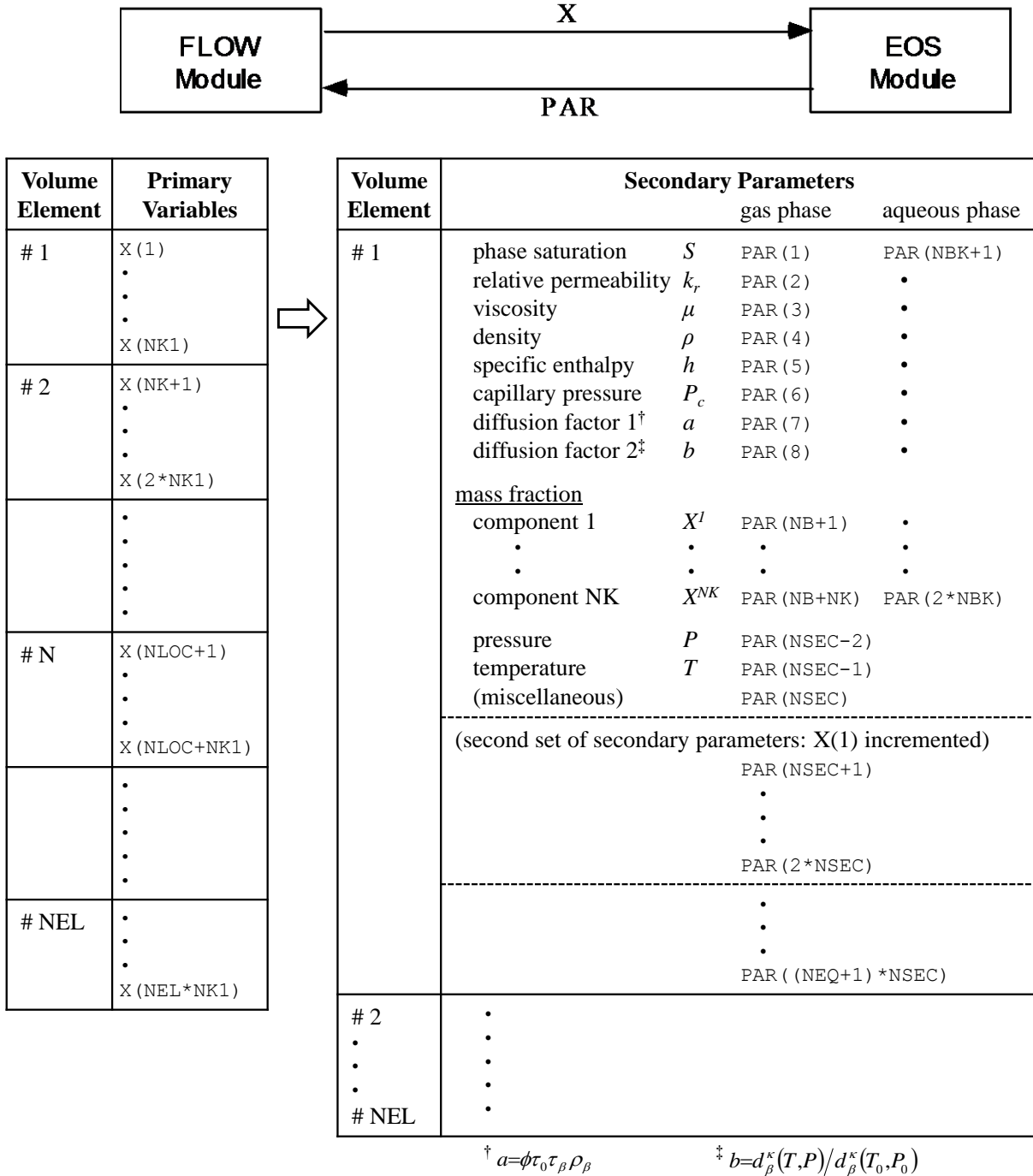


Figure 6. Structure of thermophysical property arrays in TOUGH3. The above structure is for a serial implementation. In a parallel implementation, the order of the variables in each volume element remains the same but the order of the volume elements is different, depending on how the volume elements are distributed across multiple processors.

The thermophysical properties (“secondary parameters”) needed to assemble the mass- and energy-balance equations for all volume elements are stored sequentially in a large array *PAR* (see Figure 6). These parameters are calculated in the EOS module for the latest updated primary variables during the iteration process. The first group of *NB* (= 6 or 8) secondary parameters includes 6 parameters needed for the accumulation and advective flow terms. If parameter *NB* is set equal to 8, these are followed by two parameters for the dependence of diffusion coefficients on primary variables (see Section 4.2). Starting in location *NB+1*, there are *NK* component mass fractions, so that the total number of secondary parameters per fluid phase is $NBK = NB + NK$. By convention, the *NBK* gas phase parameters come first, followed by *NBK* parameters for the aqueous phase, except EOS9 and ECO2M. EOS9 only considers a single aqueous phase and treats the gas phase as a passive bystander at constant pressure. ECO2M uses the following order: aqueous, liquid CO₂, gaseous CO₂, and solid salt. Figure 6 shows the *PAR* array structure for the case of two fluid phases. For *NPH*-phase systems, the $NPH \times NBK$ phase-specific parameters are followed by pressure *P*, temperature *T* and an additional array member that is used for different purposes by different EOS modules. The total number of secondary parameters is $NSEC = NPH \times NBK + 3$.

Thermophysical properties are needed not only for calculating the residuals of the mass- and energy-balance equations (Eq. 37), but also for calculating their derivatives in the Jacobian matrix $\mathbf{J} = -\partial \mathcal{R}_n / \partial x_i$ (see Eqs. 38 and 39; x_i denotes the collection of all primary independent thermodynamic variables). In order to avoid any restrictions in the manner that thermophysical parameters could be functionally dependent on primary variables, these derivatives are obtained by numerical differentiation. In order to calculate derivatives, the *PAR* array must provide secondary parameters not only at the “state point” (latest $X + DX$), but also for *NEQ* additional sets of primary variables in which one of the primary variables at a time is incremented by a small amount, *DELX*. Therefore, the total number of secondary parameters per grid block is $(NEQ + 1) \times NSEC$.

7.2.3. Linear Equation Setup

The data provided by the *PAR*-array are used in the flow module of TOUGH3 to assemble the linear equations (Eq. 39) that are solved at each step of the Newton-Raphson iteration procedure. The structure of the coupled linear equations at iteration step (*p+1*) is shown in Figure 7. When a time step is converged, the primary variables *X* are updated, $X \rightarrow X + DX$. These equations are arranged and numbered sequentially, with the first *NK* equations per grid block representing component mass balances, while the last equation (# *NK1*) represents the energy

balance. The row indices of the Jacobian matrix correspond to the component balance equations, while the column indices correspond to the sequence of primary variables in array X .

In many applications, heat effects and temperature changes may be small, so that it would be sufficient to consider just the mass balance equations. To force temperature changes to zero, we can assign the solid matrix an overwhelmingly large heat capacity, so that finite rates of heat exchange will cause negligible temperature change. This approach is perfectly acceptable and useable with TOUGH3, but it has the disadvantage that the full number of balance equations must be solved, even though the energy balance reduces to the trivial statement $\partial T/\partial t = 0$. TOUGH3 offers an option to run problems without temperature changes, at considerable savings of computing time. TOUGH3 uses a parameter NEQ , distinct from NK , to count the number of balance equations to be solved per grid block. Normally we would have $NEQ = NK + 1$. However, for certain EOS modules that have temperature as primary variable # $NK1$, users can choose to assign $NEQ = NK$ in the input file; then no energy equations are set up or solved and the number of coupled equations is reduced by one per grid block. In this case, only the first NK primary variables per grid block will contribute matrix columns, while variable # $NK1$, which must be temperature, remains passive and is not engaged or altered in the linear equation handling. However, all thermophysical parameters will be calculated at the temperature values specified in variable # $NK1$.

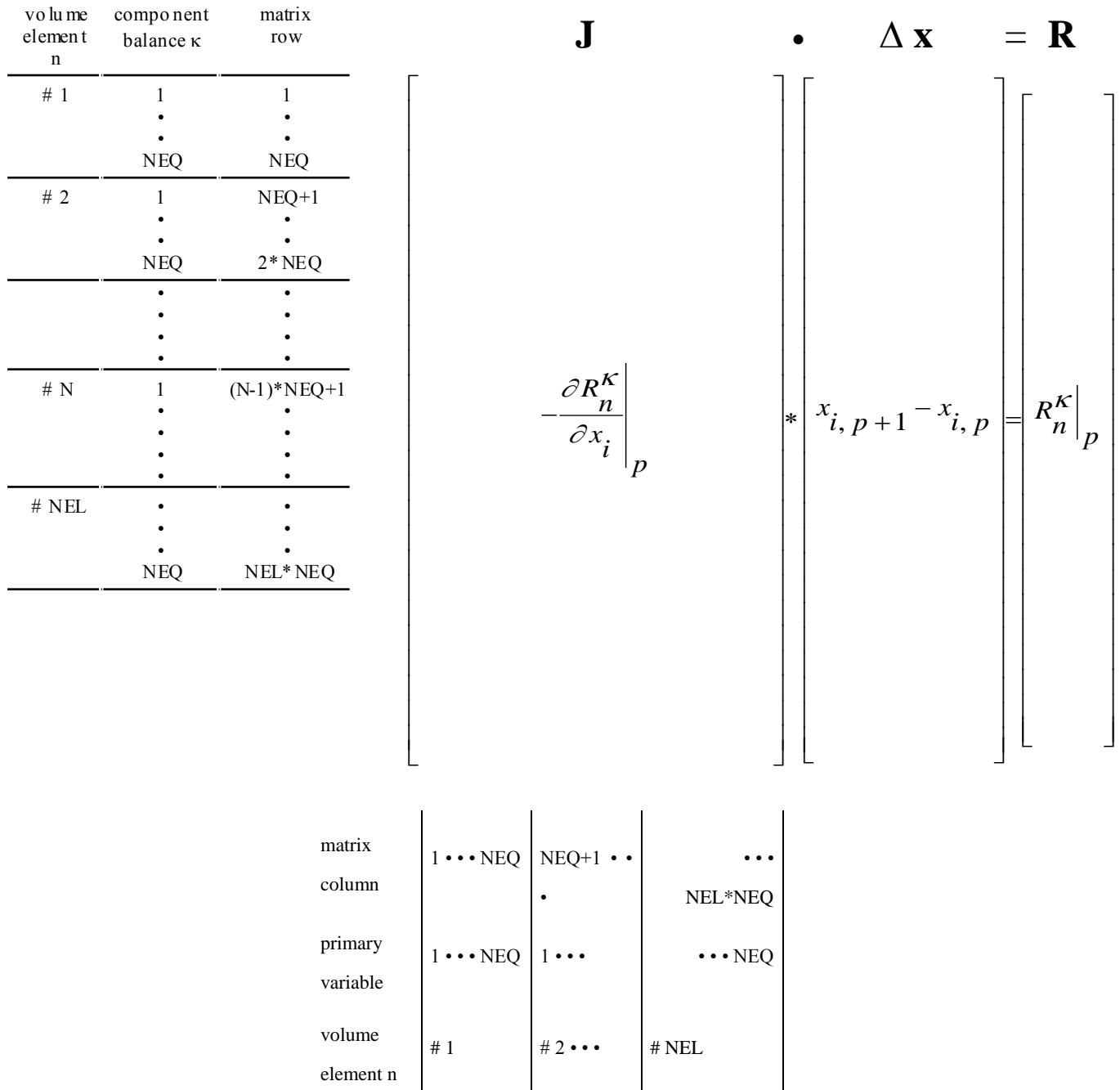


Figure 7. Linear equation structure.

Note that the accumulation terms of the balance equations depend only on primary variables for one grid block, so that they will generate non-zero derivative terms only in an $NEQ \times NEQ$ submatrix that is located on the diagonal of the Jacobian \mathbf{J} . The flow terms, being dependent on primary variables of two grid blocks, will generate two non-zero $NEQ \times NEQ$ submatrices of

derivatives, which are located in the off-diagonal matrix locations corresponding to the two grid blocks.

In TOUGH3 all Jacobian matrix elements as well as the entries in the vector R of residuals are calculated in subroutine MULTI. The calculation first assigns all matrix elements arising from the accumulation terms, of which there are $NEQ \times NEQ$. Calculation of the derivatives demands that each accumulation term is calculated $NEQ + 1$ times; once for the state point $(X + DX)$, and NEQ times for each of the NEQ primary variables incremented $(X + DX + DELX)$. Additional contributions to diagonal terms in the Jacobian \mathbf{J} may arise from sink and source terms, if present; these are assigned in subroutine QU, which is called from MULTI. Subsequently all flux terms are evaluated. These depend in general on the $2 \cdot NEQ$ primary variables of the two connected grid blocks, so that a total of $2 \cdot NEQ + 1$ flux terms need to be evaluated for calculation of the state point as well as of all derivative terms.

After all matrix elements and members of the right-hand side vector of residuals have been assembled, the linear equations (Eq. (39)) are solved by one of several user-selectable algorithms (see Section 5.2). The resulting increments in the primary variables are added to the array DX , and the process of linear equation setup and solution is repeated for the primary variables $X + DX$. This process continues until the residuals are reduced below a preset convergence tolerance (Eq. (40)). If convergence is not achieved within a specified maximum number of iterations (usually 8, see NOITE), the time step is repeated with reduced time increment.

7.3. Execution

Table 3 shows the key subroutines contained in the TOUGH3 package. The subroutines INPUT and RFILE initialize a simulation run. Most of the necessary data are supplied from disk files, which can be either directly provided by the user, or can be generated internally from data provided in the input file. The initialization stage also can generate a variety of computational grids by calling subroutine MESHM. CYCIT controls the iteration sequence for time stepping. On the first time step, all thermophysical properties are initialized (subroutine EOS), and time step counter $KCYC$, iteration counter $ITER$, and convergence flag KON are defined. The iteration counter is incremented, and the accumulation and flow terms for all equations are assembled in subroutine MULTI. If sinks or sources are present, MULTI calls subroutine QU, which in turn can call a series of satellite routines to perform functions such as calculating phase compositions of produced fluids, interpolating from tables, and obtaining flowing wellbore pressure corrections.

MULTI then computes all residuals, identifies the element index (NER) and equation index (KER) for the largest residual, and checks for convergence. If convergence is achieved ($KON = 2$), CONVER is called to update the primary variables. Otherwise control shifts to LINEQ, which functions as executive routine for the linear equation solution, and calls a variety of user-selectable linear equation solvers. FINALOUT generates printed output subject to different user controls, and BALLA prints out volume and mass balances at specified times. WRIFI generates a disk file SAVE to be used for restarting. Depending on the value of the convergence flag KON , the program will proceed to the next Newton-Raphson iteration ($KON = 1$), or to the next time step ($KON = 2$). If anything goes wrong - failure in computing thermophysical parameters, failure in solving the linear equations, failure to converge within a given maximum number of iterations (see variable $NOITE$) - the time step will be repeated with a reduced time increment Δt (see variable $REDLT$).

Table 3. Key Subroutines in TOUGH3.

subroutine(s)	Function
TOUGH3 (main program)	executive routine; define problem-size dependent arrays, generate internal documentation
INPUT	problem initialization; read data provided through the input file
RFILE (and satellite subroutines)	read input data from disk files
MESHM (and satellites)	mesh generation
CYCIT	executive routine for time stepping
EOS (and satellites)	thermophysical properties and phase diagnostics
MULTI (and satellites)	assembly of mass and energy balance equations
QU (and satellites)	sink and source terms
LINEQ (and satellites)	solution of linear equations, provisional updating of thermodynamic variables
CONVER	conclusion of converged time steps, updating of thermodynamic variables and iteration counters
FINALOUT, BALLA, WRIFI	output of results

Execution will continue until one of several termination criteria is reached, such as a user-specified total number of time steps, or a specified simulation time. In some cases, a simulation may proceed with convergence achieved on the first iteration ($ITER = 1$). This can occur when time steps are chosen too small, conditions are so close to steady state, or the convergence criteria $RE1$ and $RE2$ are too loose, so that the convergence tolerances are satisfied without requiring any updating of thermodynamic variables. If convergence on $ITER = 1$ occurs for 10 consecutive time steps, TOUGH3 will take this to indicate attainment of steady state, write out the current conditions, and terminate the execution.

Another termination criterion is monitored when automatic time step control is in effect. In this case, execution will be stopped when convergence failure is encountered following two time steps for which convergence on $ITER = 1$ was achieved. One benign situation where this case occurs is when a simulation approaches steady state. As conditions approach steady state, the rates of change in thermodynamic variables become small, and time steps will grow to very large values when automatic time step control is in effect. The off-diagonal elements in the Jacobian matrix are proportional to time step size, and may become so large that convergence of the linear equation system may no longer be achieved because of numerical roundoff. Thus, we may have a situation where for a somewhat reduced time step convergence is achieved without requiring a linear equation solution ($ITER = 1$), while no convergence at all can be obtained when a linear equation solution is required. The aforementioned stopping criterion was designed to prevent such “no progress” calculations to go on for a potentially large number of user-specified time steps. This termination criterion can also be encountered when the time steps become very small due to convergence issues.

8. Preparation of Input Data

To characterize a flow system, users need to prepare input data, including hydrogeologic and thermal parameters and constitutive relations of the permeable medium (absolute and relative permeability, porosity, capillary pressure, heat conductivity, etc.), thermophysical properties of the fluids (mostly provided internally), initial and boundary conditions of the flow system, and sinks and sources. In addition, TOUGH3 requires specification of space-discretized geometry, and various program options, computational parameters, and time-stepping information. Here we summarize the basic conventions adopted for TOUGH3 input data (Section 8.1), and provide a detailed explanation of input data formats (Section 8.2); the summary and input formats for internal grid generation through the MESHMaker module are explained in Sections 8.3 and 8.4, respectively. Some of the EOS modules have specialized input requirements (discussed in the addenda for those modules). The TMVOC, ECO2N, and ECO2M modules have special options and input requirements, which are explained in separate reports (TMVOC: Pruess and Battistelli, 2002; ECO2N: Pruess, 2005, Pan et al., 2015; ECO2M: Pruess, 2013).

8.1. TOUGH3 Input Data

Input can be provided to TOUGH3 in a flexible manner by means of one or several ASCII data files. All TOUGH3 input is in fixed format (a few exceptions, such as data blocks **GENER** and **TIMBC**) and standard metric (SI) units, such as meters, seconds, kilograms, °C, and the corresponding derived units, such as Newtons, Joules, and Pascal = N/m² for pressure.

In the standard TOUGH3 input file data are organized in blocks which are defined by five-character keywords typed in columns 1-5; see Table 4. The first record must be a problem title of up to 80 characters. The last record usually is **ENDCY**; alternatively, **ENDFI** may be used if no flow simulation is to be carried out (useful for mesh generation). Data records beyond **ENDCY** (or **ENDFI**) will be ignored. Some data blocks, such as those specifying reservoir domains (**ROCKS**), volume elements (**ELEME**), connections (**CONNE**), sinks/sources (**GENER**), and initial conditions (**INDOM** and **INCON**), have a variable number of records, while others have a fixed number of records. The end of variable-length data blocks is indicated by a blank record. The data blocks can be provided in arbitrary order; exceptions are (1) the first line must be the title line, (2) **ELEME**, if present, must precede **CONNE**, (3) **START**, if present, must be specified before **INCON**, and (4) **MULTI** must be specified before **PARAM**, **INDOM**, **INCON** (only if $NKIN \neq NK$), and **OUTPU**. The blocks **ELEME** and **CONNE** must either be both provided through the input file, or must both be

absent, in which case alternative means for specifying geometry data will be employed (see Section 8.3, below). If **START** is present, the block **INCON** can be incomplete, with elements in arbitrary order, or can be absent altogether. Elements for which no initial conditions are specified in **INCON** will then be assigned domain-specific initial conditions from block **INDOM**, if present, or will be assigned default initial conditions given in block **PARAM**, along with default porosities given in **ROCKS**. If **START** is not present, **INCON** must contain information for all elements, in exactly the same order as they are listed in block **ELEME** (note that initial conditions will be assigned to elements according to their order, not the element name).

Table 4. TOUGH3 input data blocks

Keyword	Function
TITLE	first record, single line with a title for the simulation problem
MESHM	parameters for internal grid generation through MESHMaker
ROCKS	hydrogeologic parameters for various reservoir domains
MULTI	specifies number of fluid components and balance equations per grid block
SELEC	used with certain EOS-modules to supply thermophysical property data
START	one data record for more flexible initialization
PARAM	computational parameters; time stepping and convergence parameters; program options; default initial conditions
MOMOP	more program options
SOLVR	parameters for linear equation solver
DIFFU	diffusivities of mass components
FOFT	specifies grid blocks for which time series data are desired
COFT	specifies connections for which time series data are desired
GOFT	specifies sinks/sources for which time series data are desired
RPCAP	parameters for relative permeability and capillary pressure functions
HYTE	parameters for hysteretic characteristic curves; required only if non-default values of parameters are used
TIMES	specification of times for generating printout

*ELEM	list of grid blocks (volume elements)
*CONNE	list of flow connections between grid blocks
*GENER	list of mass or heat sinks and sources
INDOM	list of initial conditions for specific reservoir domains
*INCON	list of initial conditions for specific grid blocks
TIMBC	specifies time-dependent Dirichlet boundary conditions
OUTPU	specifies variables/parameters for printout
ENDCY	last record to close the TOUGH3 input file and initiate the simulation
ENDFI	alternative to ENDCY for closing a TOUGH3 input file; will cause flow simulation to be skipped; useful if only mesh generation is desired

* These blocks can be provided on separate disk files, in which case they would be omitted from the INPUT file.

Between data blocks, an arbitrary number of records with arbitrary data may be present that do not begin with any of the TOUGH3 keywords. This is useful for inserting comments about problem specifications directly into the input file. TOUGH3 will gather all these comments and will print the first 50 such records in the output file.

Much of the data handling in TOUGH3 is accomplished by means of disk files, which are written in a format of 80 characters per record, so that they can be edited and modified with any text editor.

Table 5 summarizes the disk files other than (default) INPUT and OUTPUT used in TOUGH3. The initialization of the arrays for geometry, generation, and initial condition data is always made from the disk files MESH (or MINC), GENER, and INCON. A user can either provide these files at execution time, or they can be written from TOUGH3 input data during the initialization phase of the program. During this initialization phase, two binary files MESH A and MESH B are created from the disk file MESH. If MESH A and MESH B exist in the working folder, the code will ignore the MESH file *and the **ELEME** and **CONNE** blocks in the input file*, and read geometric information directly from these two files, which will reduce the memory requirement for the master processor and enhance I/O efficiency. If the mesh is changed, MESH A and MESH B *must* be deleted from the working folder to make the changes take effect. If no data blocks **GENER** and **INCON** are provided in the input file, and if no disk files GENER and INCON are present, defaults will take effect (no generation; domain-specific initial conditions from block **INDOM**, or defaults from block **PARAM**). If a user intends to use these defaults, the user has to make sure that at execution time no disk files INCON or GENER are present from a previous run (or perhaps from a different problem). A safe way to use default GENER and INCON is to specify “dummy” data blocks in the input file, consisting of one record with **GENER** or **INCON**, followed by a blank record.

The format of data blocks **ELEME**, **CONNE**, **GENER**, and **INCON** is basically the same when these data are provided as disk files as when they are given as part of the input file. However, specification of these data through the input file rather than as disk files offers some added conveniences, which are useful when a new simulation problem is initiated. For example, a sequence of identical items (volume elements, connections, sinks or sources) can be specified through a single data record. Also, indices used internally for cross-referencing elements, connections, and sources will be generated internally by TOUGH3 rather than having them provided by the user. MESH, GENER, and INCON disk files written by TOUGH3 can be merged into an input file without changes, keeping the cross-referencing information.

Table 5. TOUGH3 disk files

File	Use
MESH	written in subroutine INPUT from ELEME and CONNE data, or in module MESHMaker from mesh specification data; read in RFILE to initialize all geometry data arrays used to define the discretized flow problem
GENER	written in subroutine INPUT from GENER data; read in RFILE to define nature, strength, and time-dependence of sinks and sources
INCON	written in subroutine INPUT from INCON data; read in RFILE to provide a complete specification of initial thermodynamic conditions
SAVE	written in subroutine WRIFI to record thermodynamic conditions at the end of a TOUGH3 simulation run; compatible with formats of file or data block INCON for initializing a continuation run
MINC	written in module MESHMaker with MESH-compatible specifications, to provide all geometry data for a fractured-porous medium mesh (double porosity, dual permeability, multiple interacting continua); read (optionally) in subroutine RFILE to initialize geometry data for a fractured-porous system
TABLE	written in subroutine CYCIT to record coefficients of semi-analytical linear heat exchange with confining beds at the end of a TOUGH3 simulation run; read (optionally) in subroutine QLOSS to initialize heat exchange coefficients in a continuation run

8.2. TOUGH3 Input Format

Here we cover the common input data for all EOS modules of TOUGH3. Additional EOS-specific input data are discussed in the addendum for each EOS module. The format described is for the default five-character element names.

TITLE is the first record of the input file, containing a header of up to 80 characters, to be printed on output. This can be used to identify a problem. If no title is desired, leave this record blank.

MESHM introduces parameters for internal mesh generation and processing. The MESHMaker input has a modular structure which is organized by keywords. Detailed instructions for preparing MESHMaker input are given in Section 8.4.

Record MESHM.1

Format (A5)

WORD

WORD enter one of several keywords, such as RZ2D, RZ2DL, XYZ, MINC, to generate different kinds of computational meshes.

Record MESHM.2 A blank record closes the MESHM data block.

ENDFI is a keyword that can be used to close a TOUGH3 input file when no flow simulation is desired. This will often be used for a mesh generation run when some hand-editing of the mesh will be needed before the actual flow simulation.

ROCKS introduces material parameters for different reservoir domains.

Record ROCKS.1

Format (A5, I5, 7E10.4)

MAT, NAD, DROK, POR, (PER (I), I = 1,3), CWET, SPHT

*MAT** material name (rock type).

<i>NAD</i>	if zero or negative, defaults will take effect for a number of parameters (see below); ≥1: will read another data record to override defaults. ≥2: will read two more records with domain-specific parameters for relative permeability and capillary pressure functions.
<i>DROK</i>	rock grain density (kg/m ³). If <i>DROK</i> is set to a very large value (typically 1.0E50), a constant <i>temperature</i> boundary condition can be specified (but variable pressure/saturation).
<i>POR</i>	default porosity (void fraction) for all elements belonging to domain “ <i>MAT</i> ” for which no other porosity has been specified in block INCON . Option “ START ” is necessary for using default porosity.
<i>PER (I)</i>	I = 1,3 absolute permeabilities along the three principal axes, as specified by <i>ISOT</i> in block CONNE .
<i>CWET</i>	formation heat conductivity under fully liquid-saturated conditions (W/m °C).
<i>SPHT</i>	rock grain specific heat (J/kg °C). Domains with <i>SPHT</i> > 10 ⁴ J/kg °C will not be included in global material balances. This provision is useful for boundary nodes, which are given very large volumes so that their thermodynamic state remains constant. Because of the large volume, inclusion of such nodes in global material balances would make the balances useless.

* When a (dummy) domain named “SEED” is specified, the absolute permeabilities specified in *PER (I)* are modified by the block-by-block permeability modifiers (PM) according to

$$k_n \rightarrow k_n' = k_n \cdot \zeta_n \quad (48)$$

Here, k_n is the absolute (“reference”) permeability of grid block n , as specified in data block **ROCKS** for the domain to which that grid block belongs, k_n' is the modified permeability, and ζ_n is the PM coefficient. When PM is in effect, the strength of capillary pressure will be automatically scaled according to (Leverett, 1941)

$$P_{cap,n} \rightarrow P'_{cap,n} = P_{cap,n} / \sqrt{\zeta_n} \quad (49)$$

No grid blocks should be assigned to this domain; the presence of domain 'SEED' simply serves as a flag to put permeability modification into effect. Random (spatially uncorrelated) PM data can be internally generated in TOUGH3. Alternatively, externally defined permeability modifiers may be provided as part of the geometry data (*PMX*) in block **ELEME**. Available PM options are:

- (1) externally supplied : $\zeta_n = PMX - PER(2)$
- (2) "linear" : $\zeta_n = PER(1) \times s - PER(2)$
- (3) "logarithmic" : $\zeta_n = \exp(-PER(1) \times s) - PER(2)$

where *s* is a random number between 0 and 1

Data provided in domain 'SEED' are used to select the following options.

DROK random number seed for internal generation of "linear" permeability modifiers.

= 0: (default) no internal generation of "linear" permeability modifiers.

> 0: perform "linear" permeability modification; random modifiers are generated internally with *DROK* as seed.

POR random number seed for internal generation of "logarithmic" permeability modifiers

= 0: (default) no internal generation of "logarithmic" permeability modifiers.

> 0: perform "logarithmic" permeability modification; random modifiers are generated internally with *POR* as seed.

Note if both *DROK* and *POR* are specified as non-zero, *DROK* takes precedence. And if both *DROK* and *POR* are zero, permeability modifiers as supplied through **ELEME** data will be used.

PER(1) scale factor (optional) for internally generated permeability modifiers.

= 0: (defaults to *PER(1) = 1*): permeability modifiers are generated as random numbers in the interval (0, 1).

> 0: permeability modifiers are generated as random numbers in the interval (0, $PER(1)$).

$PER(2)$ shift (optional) for internal or external permeability modifiers.

= 0: (default) no shift is applied to permeability modifiers.

> 0: permeability modifiers are shifted according to $\zeta'_n = \zeta_n - PER(2)$.

All $\zeta'_n < 0$ are set equal to zero.

Note that the domain 'SEED' is not required in TOUGH3 if externally defined permeability modifiers in block **ELEME** are used without any shift.

Record **ROCKS.1.1** (optional, $NAD \geq 1$ only)

Format (7E10.4)

COM, EXPAN, CDRY, TORTX, GK, XKD3, XKD4

COM pore compressibility (Pa^{-1}), $(1/\phi)(\partial\phi/\partial P)_T$ (default is 0).

$EXPAN$ pore expansivity ($1/^\circ\text{C}$), $(1/\phi)(\partial\phi/\partial T)_P$ (default is 0).

$CDRY$ formation heat conductivity under desaturated conditions ($\text{W/m } ^\circ\text{C}$), (default is $CWET$).

$TORTX$ tortuosity factor for binary diffusion. If $TORTX = 0$, a porosity and saturation-dependent tortuosity will be calculated internally from the Millington and Quirk (1961) model, Eq. (12). When diffusivities, $FDDIAG$ in data block **DIFFU**, are specified as negative numbers, $\tau_0\tau_\beta = S_\beta$ will be used.

GK Klinkenberg parameter b (Pa^{-1}) for enhancing gas phase permeability according to the relationship $k_{gas} = k * (1 + b/P)$.

$XKD3$ Distribution coefficient for parent radionuclide, Component 3, in the aqueous phase, $\text{m}^3 \text{kg}^{-1}$ (EOS7R only).

$XKD4$ Distribution coefficient for daughter radionuclide, Component 4, in the aqueous phase, $\text{m}^3 \text{kg}^{-1}$ (EOS7R only).

Record **ROCKS.1.2** (optional, $NAD \geq 2$ only)

Format (I5, 5X, 7E10.4)

IRP, (RP(I), I= 1,7)

IRP integer parameter to choose type of relative permeability function (see Appendix A).

RP(I) I = 1, ..., 7 parameters for relative permeability function (Appendix A).

Record **ROCKS.1.2.1** (optional, *IRP* = 12 only)

Format (3E10.4)

RP(I), I = 8,10

RP(I) I = 8, ..., 10 parameters for relative permeability function (Appendix A).

Record **ROCKS.1.3** (optional, *NAD* ≥ 2 only)

Format (I5, 5X,7E10.4)

ICP, (*CP(I)*, I = 1,7)

ICP integer parameter to choose type of capillary pressure function (see Appendix B).

CP(I) I = 1, ..., 7 parameters for capillary pressure function (Appendix B).

Record **ROCKS.1.3.1** (optional, *ICP* = 12 only)

Format (6E10.4)

CP(I), I = 8,13

CP(I) I = 8, ..., 13 parameters for capillary pressure function (Appendix B).

Repeat records 1, 1.1, 1.2, 1.2.1., 1.3, and 1.3.1 for different reservoir domains.

Record **ROCKS.2** A blank record closes the **ROCKS** data block.

MULTI permits the user to select the number and nature of balance equations that will be solved. For most EOS modules this data block is not needed, as default values are provided internally. Available parameter choices are different for different EOS modules (see the addendum of the EOS module of interest).

Record **MULTI.1**

Format (5I5)

NK, NEQ, NPH, NB, NKIN

NK number of mass components.

NEQ number of balance equations per grid block. Usually we have $NEQ = NK + 1$, for solving NK mass and one energy balance equation. Some EOS modules allow the option $NEQ = NK$, in which case only NK mass balances and no energy equation will be solved.

NPH number of phases that can be present (2 for most EOS modules).

NB number of secondary parameters in the *PAR*-array (see Figure 6) other than component mass fractions. Available options include $NB = 6$ (no diffusion) and $NB = 8$ (include diffusion).

NKIN number of mass components in **INCON** data (default is $NKIN = NK$). This parameter can be used, for example, to initialize an EOS7R simulation ($NK = 4$ or 5) from data generated by EOS7 ($NK = 2$ or 3). If a value other than the default is to be used, then data block **MULTI** must appear before any initial conditions in data blocks **PARAM**, **INDOM**, or **INCON**.

START (optional) A record with **START** typed in columns 1-5 allows a more flexible initialization. More specifically, when **START** is present, **INCON** data can be in arbitrary order, and need not be present for all grid blocks (in which case defaults will be used). Without **START**, there must be a one-to-one correspondence between the data in blocks **ELEME** and **INCON**.

PARAM introduces computation parameters, time stepping information, and default initial conditions.

Record **PARAM.1**

Format (2I2, 3I4, 24I1, 10X, 2E10.4).

NOITE, KDATA, MCYC, MSEC, MCYPR, (MOP(I), I = 1, 24), TEXP, BE

NOITE specifies the maximum number of Newtonian iterations per time step (default is 8)

KDATA specifies amount of printout (default is 1).
 0 or 1: print a selection of element variables.
 2: in addition, print a selection of connection variables.
 3: in addition, print a selection of generation variables.

If the above values for *KDATA* are increased by 10, printout will occur after each Newton-Raphson iteration (not just after convergence).

If negative values are used, printout for the selection of variables will be generated in the file format of TECPLOT. The amount of printout is given by the absolute value of *KDATA*.

MCYC maximum number of time steps to be calculated. If a negative value is used, $MCYC=2^{-MCYC}$.

MSEC maximum duration, in CPU seconds, of the simulation (default is infinite).

MCYPR printout will occur for every multiple of *MCYPR* steps (default is 1). If a negative value is used, $MCYPR=2^{-MCYPR}$.

MOP (I) I = 1,24 allows choice of various options, which are documented in printed output from a TOUGH3 run.

MOP (1) if unequal 0, a short printout for non-convergent iterations will be generated.

MOP (2) through *MOP (6)* generate additional printout in various sub-routines, if set unequal 0. This feature should not be needed in normal applications, but it will be convenient when a user suspects a bug and wishes to examine the inner workings of the code. The amount of printout increases with *MOP (I)* (consult source code listings for details).

MOP (2) CYCIT (main subroutine).

MOP (3) MULTI (flow- and accumulation-terms).

MOP (4) QU (sinks/sources). No longer supported in TOUGH3.

MOP (5) EOS (equation of state).

MOP (6) LINEQ (linear equations). No longer supported in TOUGH3.

MOP (7) if unequal 0, a printout of input data will be provided.

- MOP (8)* if $ISOT < 0$, chooses the option for reducing fracture-matrix interface area (See Appendix C2).
- MOP (9)* determines the composition of produced fluid with the MASS option (see **GENER**, below). The relative amounts of phases are determined as follows:
- 0: according to relative mobilities in the source element.
 - 1: produced source fluid has the same phase composition as the producing element.
- MOP (10)* chooses the interpolation formula for heat conductivity as a function of liquid saturation (S_l)
- 0: $C(S_l) = C_{DRY} + \text{SQRT}(S_l) * [C_{WET} - C_{DRY}]$
 - 1: $C(S_l) = C_{DRY} + S_l * (C_{WET} - C_{DRY})$
- MOP (11)* determines evaluation of mobility and permeability at interfaces.
- 0: mobilities are upstream weighted with WUP (see **PARAM.3**), permeability is upstream weighted.
 - 1: mobilities are averaged between adjacent elements, permeability is upstream weighted.
 - 2: mobilities are upstream weighted, permeability is harmonic weighted.
 - 3: mobilities are averaged between adjacent elements, permeability is harmonic weighted.
 - 4: mobility and permeability are both harmonic weighted.
- MOP (12)* determines interpolation procedure for time dependent sink/source data (flow rates and enthalpies).
- 0: triple linear interpolation; tabular data are used to obtain interpolated rates and enthalpies for the beginning and end of the time step; the average of these values is then used.
 - 1: step function option; rates and enthalpies are taken as averages of the table values corresponding to the beginning and end of the time step.
 - 2: rigorous step rate capability for time dependent generation data.

A set of times t_i and generation rates q_i provided in data block **GENER** is interpreted to mean that sink/source rates are piecewise constant and change in discontinuous fashion at table points. Specifically, generation is assumed to occur at constant rate q_i during the time interval $[t_i, t_{i+1})$, and changes to q_{i+1} at t_{i+1} . Actual rate used during a time step that ends at time t , with $t_i \leq t \leq t_{i+1}$, is automatically adjusted in such a way that total cumulative exchanged mass at time t

$$Q(t) = \int_0^t q dt' = \sum_{j=1}^{i-1} q_j (t_{j+1} - t_j) + q_i (t - t_i) \quad (50)$$

is rigorously conserved. If also tabular data for enthalpies are given, an analogous adjustment is made for fluid enthalpy, to preserve $\int q h dt$.

MOP (13) determines content of INCON and SAVE files.

- 0: standard content.
- 1: writes user-specified initial conditions to file SAVE.
- 2: reads parameters of hysteresis model from file INCON.

MOP (14) (void)

MOP (15) determines conductive heat exchange with impermeable geologic formations using semi-analytical methods (see Section 4.5).

- 0: heat exchange is off.
- 1: linear heat exchange *between a reservoir and confining beds* is on (for grid blocks that have a non-zero heat transfer area; see data block **ELEME**). Initial temperature in cap- or base-rock is assumed uniform and taken as the temperature with which the last element in data block **ELEME** is initialized.
- 2: linear heat exchange *between a reservoir and confining beds* is on. Initial temperature for the confining bed adjacent to an element that has a non-zero heat transfer area is taken as the temperature of that element in data block **INCON**.

Heat capacity and conductivity of the confining beds are specified in block **ROCKS** for the particular domain to which the very last volume element in data block **ELEME** belongs. Thus, if a semi-analytical heat exchange calculation is desired, the user would append an additional dummy element of a very large volume at the end of the **ELEME** block, and provide the desired parameters as initial conditions and domain data, respectively, for this element. It is necessary to specify which elements have an interface area with the confining beds, and to give the magnitude of this interface area. This information is input as parameter *AHTX* in columns 31-40 of volume element data in block **ELEME**. Volume elements for which a zero-interface area is specified will not be subject to heat exchange. A sample problem using $MOP(15)=1$ is included in the addendum for EOS1.

- 5: radial heat exchange *between fluids in a wellbore and the surrounding formation* is on. Constant well and formation properties are given in 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 [$^{\circ}\text{C}$]

PER(3): Geothermal gradient [$^{\circ}\text{C}/\text{m}$]

CWET: Heat conductivity [$\text{W}/\text{kg } ^{\circ}\text{C}$] of formation near well

SPHT: Rock grain specific heat [$\text{J}/\text{kg } ^{\circ}\text{C}$] of formation near well

- 6: radial heat exchange *between fluids in a wellbore and the surrounding formation* is on. Depth-dependent well and formation properties (depth, radius, temperature, conductivity, density, capacity) are provided on an external file named *radqloss.dat* with the information in the following format:

On the first line, *NMATQLOSS* is the number of elevations with geometric and thermal data, and *NMATQLOSS* lines are provided

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.

RESTART is possible for linear heat exchange between a reservoir and confining beds ($MOP(15) = 1$ or 2), but not for radial heat exchange ($MOP(15) = 5$ or 6). The data necessary for continuing the linear heat exchange calculation in a TOUGH3 continuation run are written onto a disk file TABLE. When restarting a problem, this file has to be provided under the name TABLE. If file TABLE is absent, TOUGH3 assumes that no prior heat exchange with confining beds has taken place.

- MOP(16)* provides automatic time step control.
- 0: automatic time stepping based on maximum change in saturation.
 - 1: automatic time stepping based on number of iterations needed for convergence.
 - >1: time step size will be doubled if convergence occurs within $ITER \leq MOP(16)$ Newton-Raphson iterations. It is recommended to set $MOP(16)$ in the range of 2 - 4.
- MOP(17)* handles time stepping after linear equation solver failure.
- 0: no time step reduction despite linear equation solver failure.
 - 9: reduce time step after linear equation solver failure.
- MOP(18)* selects handling of interface density.
- 0: perform upstream weighting for interface density.
 - >0: average interface density between the two grid blocks. However, when one of the two phase saturations is zero, upstream weighting will be performed.
- MOP(19)* switch used by different EOS modules for conversion of primary variables.
- MOP(20)* switch for vapor pressure lowering in EOS4; $MOP(20)=1$ optionally suppresses vapor pressure lowering effects.
- MOP(21)* selects the linear equation solver.
- 0: defaults to $MOP(21) = 3$, DSLUCS, Lanczos-type preconditioned bi-conjugate gradient solver.

- 1: (void)
- 2: DSLUBC, bi-conjugate gradient solver.
- 3: DSLUCS (default).
- 4: DSLUGM, generalized minimum residual preconditioned conjugate gradient solver.
- 5: DLUSTB, stabilized bi-conjugate gradient solver.
- 6: LUBAND, banded direct solver.
- 7: AZTEC parallel iterative solver
- 8: PETSc parallel iterative solver

All conjugate gradient solvers use incomplete LU-factorization as a default preconditioner. Other preconditioners may be chosen by means of a data block **SOLVR**.

- MOP (22)* (void)
- MOP (23)* (void)
- MOP (24)* determines handling of multiphase diffusive fluxes at interfaces.
 - 0: harmonic weighting of fully coupled effective multiphase diffusivity.
 - 1: separate harmonic weighting of gas and liquid phase diffusivities.
- TEXP* parameter for temperature dependence of gas phase diffusion coefficient (see Eq. (13)).
- BE* (optional) parameter for effective strength of enhanced vapor diffusion; if set to a non-zero value, will replace the parameter group $\phi\tau_0\tau_\beta$ for vapor diffusion (see Eq. (10)).

Record PARAM.2

Format (4E10.4, A5, 5X,3E10.4)

TSTART, TIMAX, DELTEN, DELTMX, ELST, GF, REDLT, SCALE

- TSTART* starting time of simulation in seconds (default is 0).
- TIMAX* time in seconds at which simulation should stop (default is infinite).

<i>DELTEN</i>	length of time steps in seconds. If <i>DELTEN</i> is a negative integer, <i>DELTEN</i> = - <i>NDLT</i> , the program will proceed to read <i>NDLT</i> records with time step information. Note that - <i>NDLT</i> must be provided as a floating point number, with decimal point.
<i>DELTMX</i>	upper limit for time step size in seconds (default is infinite)
<i>ELST</i>	no longer supported in TOUGH3. For printout after each time step, use FOFT instead.
<i>GF</i>	magnitude (m/sec ²) of the gravitational acceleration vector. Blank or zero gives "no gravity" calculation.
<i>REDLT</i>	factor by which time step is reduced in case of convergence failure or other problems (default is 4).
<i>SCALE</i>	scale factor to change the size of the mesh (default = 1.0).

Record **PARAM.2.1, 2.2, etc.**

Format (8E10.4)
(DLT(I), I = 1, 100)

DLT (I) Length (in seconds) of time step I.

This set of records is optional for *DELTEN* = - *NDLT*, a negative integer. Up to 13 records can be read, each containing 8 time step data. If the number of simulated time steps exceeds the number of *DLT (I)*, the simulation will continue with time steps determined by automatic time step control (*MOP (16)*).

Record **PARAM.3**

Format (8E10.4)
RE1, RE2, U, WUP, WNR, DFAC, FOR,AMRES

RE1 convergence criterion for relative error, see Appendix B (default= 10⁻⁵).
RE2 convergence criterion for absolute error, see Appendix B (default= 1).
U (void).

<i>WUP</i>	upstream weighting factor for mobilities and enthalpies at interfaces (default = 1.0 is recommended). $0 \leq WUP \leq 1$.
<i>WNR</i>	weighting factor for increments in Newton/Raphson - iteration (default = 1.0 is recommended). $0 < WNR \leq 1$.
<i>DFAC</i>	increment factor for numerically computing derivatives (default value is $DFAC = 10^{-k/2}$, where k , evaluated internally, is the number of significant digits of the floating point processor used; for 64-bit arithmetic, $DFAC \approx 10^{-8}$).
<i>FOR</i>	factor to change the size of the time step during the Newtonian iteration (default =1.0).
<i>AMRES</i>	maximum permissible residual during the Newtonian iteration. If a residual larger than AMRES is encountered, time step will automatically be reduced (default= 10^5).

Record **PARAM.4** introduces a set of primary variables which are used as default initial conditions for all grid blocks that are not assigned by means of data blocks **INDOM** or **INCON**. Option **START** is necessary to use default INCON.

Format (4E20.13)

DEP(I), I = 1, NKIN+1

The number of primary variables, $NKIN+1$, is normally assigned internally in the EOS module, and is usually equal to the number NEQ of equations solved per grid block. See data block **MULTI** for special assignments of $NKIN$. When more than four primary variables are used more than one line (record) must be provided

Different sets of primary variables are in use for different EOS modules; see the description of EOS modules, Section 6.

INDOM introduces domain-specific initial conditions. These will supersede default initial conditions specified in **PARAM.4**, and can be overwritten by element-specific initial conditions in data block **INCON**. Option **START** is needed to use **INDOM** conditions.

Record **INDOM.1**

Format (A5)

MAT

MAT name of a reservoir domain, as specified in data block **ROCKS**.

Record **INDOM.2**

Format (4E20.13)

X1, X2, X3, ...

A set of primary variables assigned to all grid blocks in the domain specified in record **INDOM.1**. Different sets of primary variables are used for different EOS modules; see the description of EOS modules in Section 6.

Record **INDOM.3**

A blank record closes the **INDOM** data block. Repeat records **INDOM.1** and **INDOM.2** for as many domains as desired. The ordering is arbitrary and need not be the same as in block **ROCKS**.

INCON introduces element-specific initial conditions.

Record **INCON.1**

Format (A3, I2, 2I5, E15.9,5E10.4)

EL, NE, NSEQ, NADD, PORX, (USRX(I), I=1,5)

EL, NE code name of element.

NSEQ number of additional elements with the same initial conditions.

NADD increment between the code numbers of two successive elements with identical initial conditions.

PORX porosity; if zero or blank, porosity will be taken as specified in block **ROCKS** if option **START** is used.

USRX (I) grid block specific parameters (optional).

Record **INCON.2** specifies primary variables.

Format (4E20.14)
X1, X2, X3, ...

A set of primary variables for the element specified in record **INCON.1**. **INCON** specifications will supersede default conditions specified in **PARAM.4**, and domain-specific conditions that may have been specified in data block **INDOM**. Different sets of primary variables are used for different EOS modules; see the description of EOS modules in their corresponding addendum.

Record **INCON.3**

A blank record closes the **INCON** data block. Alternatively, initial condition information may terminate on a record with '+++' typed in the first three columns, followed by time stepping information. This feature is used for a continuation run from a previous TOUGH3 simulation.

SOLVR (optional) introduces a data block with parameters for linear equation solvers. For the parallel solvers, only *MATSLV* is used, and the other options should be specified through the PETSc (*.petscsrc*) and Aztec (*.aztecrc*) option files.

Record **SOLVR.1**

Format (I1,2X,A2,3X,A2,2E10.4)
MATSLV, ZPROCS, OPROCS, RITMAX, CLOSUR

<i>MATSLV</i>	selects the linear equation solver
1:	(void)
2:	DSLUBC, a bi-conjugate gradient solver
3:	DSLUCS, a Lanczos-type bi-conjugate gradient solver
4:	DSLUGM, a generalized minimum residual solver
5:	DLUSTB, a stabilized bi-conjugate gradient solver
6:	direct solver LUBAND
7:	AZTEC parallel iterative solver
8:	PETSc parallel iterative solver

ZPROCS selects the Z-preconditioning (Moridis and Pruess, 1998). Regardless of user specifications, Z-preprocessing will only be performed when iterative solvers are used ($2 \leq MATSLV \leq 5$), and if there are zeros on the main diagonal of the Jacobian matrix.

Z0: no Z-preprocessing (default for $NEQ=1$)

Z1: replace zeros on the main diagonal by a small constant (1.e-25; default for $NEQ \neq 1$)

Z2: make linear combinations of equations for each grid block to achieve non-zeros on the main diagonal

Z3: normalize equations, followed by Z2

Z4: affine transformation to unit main-diagonal submatrices, without center pivoting

OPROCS selects the O-preconditioning (Moridis and Pruess, 1998).

O0: no O-preprocessing (default, also invoked for $NEQ=1$)

O1: eliminate lower half of the main-diagonal submatrix with center pivoting

O2: O1, plus eliminate upper half of the main-diagonal submatrix with center pivoting

O3: O2, plus normalize, resulting in unit main-diagonal submatrices

O4: affine transformation to unit main-diagonal submatrices, without center pivoting

RITMAX selects the maximum number of CG iterations as a fraction of the total number of equations ($0.0 < RITMAX \leq 1.0$; default is $RITMAX = 0.1$)

CLOSUR convergence criterion for the CG iterations ($1.e-12 \leq CLOSUR \leq 1.e-6$; default is $CLOSUR = 1.e-6$)

FOFT (optional) introduces a list of elements (grid blocks) for which time-dependent data are to be written out for plotting during the simulation. A separate file is generated for each element in CSV format. The name of each file starts with FOFT, and includes the element name. If regular or simple hysteresis is invoked via $IRP = ICP = 12$ or $IRP = ICP = 13$, then relevant hysteresis parameters are also written

to **FOFT**. If *MOP2* (17) > 0, print variables according to the input data in block **OUTPU**.

Record **FOFT.1**

Format (A5,5X,I5)

EFOFT, IFOFTF

EFOFT element name.

IFOFTF a flag to control the amount of printout.

0: default printout of pressure, temperature, and saturation of flowing phases.

>0: print out mass fraction of each component in the specified phase in addition to the default printout.

<0: print out mass fraction of each component in each of all the flowing phases in addition to the default printout.

Record **FOFT.2**

A blank record closes the **FOFT** data block. Repeat records **FOFT.1** for as many elements as desired.

COFT (optional) introduces a list of connections for which time-dependent data are to be written out for plotting during the simulation. A separate file is generated for each connection in CSV format. The name of each file starts with COFT, and includes the pair of two element names.

Record **COFT.1**

Format (A10,10X,I5)

ECOFT, ICOFTF

ECOFT a connection name, i.e., an ordered pair of two element names (no blank between elements).

ICOFTF a flag to control the amount of printout.

0: default printout of heat flux and flow rate of each flowing phase.

- >0: print out fractional mass flow of each component in the specified phase in addition to the default printout.
- <0: print out fractional mass flow of each component in each of all the flowing phases in addition to the default printout.

Record **COFT.2**

A blank record closes the **COFT** data block. Repeat records **COFT.1** for as many connections as desired.

GOFT (optional) introduces a list of sinks/sources for which time-dependent data are to be written out for plotting during the simulation. A separate file is generated for each sink/source in CSV format. The name of each file starts with GOFT, and includes the sink/source name.

Record **GOFT.1**

Format (A5,A5,5X,I5)

EGOFT, S_GOFT, IGOFTF

ECOFT the name of an element in which a sink/source is defined.

S_GOFT the name of the source in which a sink/source is defined. It is required when multiple sinks/sources are assigned to the same element.

IGOFTF a flag to control the amount of printout.

0: default printout of mass flow rate and flowing enthalpy.

>0: print out fractional mass flow rate of the specified phase in addition to the default printout.

<0: print out fractional mass flow rate of each of all the flowing phases in addition to the default printout.

Record **GOFT.2**

A blank record closes the **GOFT** data block. Repeat records **GOFT.1** for as many sinks/sources as desired.

DIFFUSION (optional; needed only for $NB = 8$) introduces diffusion coefficients.

Record **DIFFU.1**

Format (8E10.4)

FDDIAG(I,1), I=1,NPH

diffusion coefficients for mass component # 1 in all phases (I=1: gas; I=2: aqueous; etc.)

Record **DIFFU.2**

Format (8E10.4)

FDDIAG(I,2), I=1,NPH

diffusion coefficients for mass component # 2 in all phases (I=1: gas; I=2: aqueous; etc.)

provide a total of *NK* records with diffusion coefficients for all *NK* mass components.

SELECTION (optional) introduces a number of integer and floating point parameters that are used for different purposes in different TOUGH3 modules (EOS7, EOS7R, EWASG, ECO2N, ECO2M, TMVOC). Note that TOUGH3 includes additional *IE* options for the calculation of brine properties in EWASG, ECO2N, and ECO2M (see the addendum for EWASG or output from a TOUGH3 run).

Record **SELEC.1**

Format (16I5)

IE(I), I=1,16

IE (1) number of records with floating point numbers that will be read (default is *IE (1)* = 1; maximum values is 64).

Record **SELEC.2, SELEC.3, ..., SELEC.IE(1)*8**

Format (8E10.4)

FE(I), I=1,IE(1)*8

provide as many records with floating point numbers as specified in *IE (1)*, up to a maximum of 64 records.

RPCAP introduces information on relative permeability and capillary pressure functions, which will be applied for all flow domains for which no data were specified in records **ROCKS.1.2** and **ROCKS.1.3**. A catalog of relative permeability and capillary pressure functions is presented in Appendix A and Appendix B, respectively.

Record **RPCAP.1**

Format (I5,5X,7E10.4)

IRP, (RP(I), I = 1, 7)

IRP integer parameter to choose type of relative permeability function (see Appendix A).

RP(I) I = 1, ..., 7 parameters for relative permeability function (Appendix A).

Record **RPCAP.1.1** (optional, IRP = 12 only)

Format (3E10.4)

RP(I), I = 8, 10

RP(I) I = 8, ..., 10 parameters for relative permeability function (Appendix A).

Record **RPCAP.2**

Format (I5,5X,7E10.4)

ICP, (CP(I), I = 1, 7)

ICP integer parameter to choose type of capillary pressure function (see Appendix B).

CP(I) I = 1, ..., 7 parameters for capillary pressure function (Appendix B).

Record **RPCAP.2.1** (optional, ICP = 12 only)

Format (6E10.4)

CP(I), I = 8, 13

CP(I) I = 8, ..., 13 parameters for capillary pressure function (Appendix B).

HYTE (optional) provides numerical controls on the hysteretic characteristic curves. It is not needed if the default values of all its parameters are to be used.

Record **HYTE.1**

Format(3I5)
IEHYS(I), I=1,3

- IEHYS (1)* flag to print information about hysteretic characteristic curves
=0 no additional print out
≥1 print a one-line message to the output file every time a capillary-pressure branch switch occurs (recommended)
- IEHYS (2)* flag indicating when to apply capillary-pressure branch switching
=0 after convergence of time step (recommended)
>0 after each Newton-Raphson iteration
- IEHYS (3)* run parameter for sub-threshold saturation change
=0 no branch switch unless $|\Delta S| > CP(10)$
>0 allow branch switch after run of *IEHYS (3)* consecutive time steps for which all $|\Delta S| < CP(10)$ and all ΔS are the same sign. Recommended value 5-10. This option may be useful if the time step is cut to a small value due to convergence problems, making saturation changes very small.

TIMES (optional) permits the user to obtain printout at specified times. This printout will occur in addition to printout specified in record **PARAM.1**.

Record **TIMES.1**

Format (2I5,2E10.4)
ITI, ITE, DELAF, TINTER

- ITI* number of times provided on records **TIMES.2**, **TIMES.3**, etc.
ITE total number of times desired ($ITI \leq ITE$; default is $ITE = ITI$).

DELAF maximum time step size after any of the prescribed times have been reached (default is infinite).

TINTER time increment for times with index *ITI*, *ITI + 1*, ..., *ITE*.

Record **TIMES.2**, **TIMES.3**, etc.

Format (8E10.4)
(TIS(I), I = 1, ITI)

TIS (I) list of times (in ascending order) at which printout is desired.

ELEME introduces element (grid block) information.

Record **ELEME.1**

Format (A3, I2, 2I5, A3, A2, 9E10.4)
EL, NE, NSEQ, NADD, MA1, MA2, VOLX, AHTX, PMX, X, Y, Z,
(USERX(I), I=1,3)

EL, NE five-character code name of an element. The first three characters are arbitrary, the last two characters must be numbers.

NSEQ number of additional elements having the same volume and belonging to the same reservoir domain.

NADD increment between the code numbers of two successive elements. (Note: the maximum permissible code number $NE + NSEQ \times NADD$ is ≤ 99 .)

MA1, MA2 a five-character material identifier corresponding to one of the reservoir domains as specified in block **ROCKS**. If the first three characters are blanks and the last two characters are numbers then they indicate the sequence number of the domain as entered in **ROCKS**. If both *MA1* and *MA2* are left blank the element is by default assigned to the first domain in block **ROCKS**.

VOLX element volume (m³).

AHTX interface area (m²) for linear heat exchange with semi-infinite confining beds. Internal MESH generation via MESHMaker will automatically assign *AHTX*.

PMX block-by-block permeability modification coefficient, ζ_n (optional). The *PMX* may be used to specify spatially correlated heterogeneous fields. But users need to run their own geostatistical program to generate the fields they desire, and then use preprocessing programs to place the modification coefficients into the **ELEME** data block, as TOUGH3 provides no internal capabilities for generating such fields.

If a dummy domain 'SEED' is specified in data block **ROCKS**, it will be used as multiplicative factor for all the permeability parameters from block **ROCKS** (see Eq. (48)), and strength of capillary pressure will be scaled according to Eq. (49). With a dummy domain 'SEED' in data block **ROCKS**, $PMX = 0$ will result in an impermeable block.

In TOUGH3, *PMX* can be active *without* a dummy domain 'SEED' in the **ROCKS** block. If a dummy domain 'SEED' is not specified in data block **ROCKS**, it can be used to specify grid block permeabilities or permeability modifiers. If a positive value less than $1e-4$ 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. Alternatively, the same information can be provided through *USRX* (columns 31–40) in block **INCON**.

If *PMX* is blank for the first element, the element-by-element permeabilities are ignored. If a dummy domain 'SEED' is not specified in data block **ROCKS**, strength of capillary pressure will not be automatically scaled. Leverett scaling of capillary pressure can be applied with $MOP2(6) > 0$ in data block **MOMOP**.

X, Y, Z Cartesian coordinates of grid block centers. These may be included in the **ELEME** data to make subsequent plotting of results more convenient. Note that coordinates are not used in TOUGH3; the exceptions are for optional initialization of a gravity-capillary equilibrium with EOS9 (see the addendum for EOS9) and for optional addition of potential energy to enthalpy with $MOP2(12) > 0$ in data block **MOMOP**.

USERX(I) anisotropic permeability or permeability modifier of the X-, Y-, and Z-direction for $I = 1, 2, \text{ and } 3$, respectively. Based on the values, TOUGH3 will internally determine whether it is permeability itself or permeability modifier. To read three permeabilities or permeability modifiers,

MOP2 (20) in data block **MOMOP** should be set according to Table 7. Alternatively, the same information can be provided through *USRX* (columns 31–60) in block **INCON**. To write these element-by-element parameters to file *SAVE*, set *MOP (13) = 1*.

Repeat record **ELEME.1** for the number of elements desired.

Record **ELEME.2** A blank record closes the **ELEME** data block.

CONNE introduces information for the connections (interfaces) between elements.

Record **CONNE.1**

Format (A3, I2, A3, I2, 4I5, 5E10.4)

EL1, NE1, EL2, NE2, NSEQ, NAD1, NAD2, ISOT, D1, D2, AREAX, BETAX, SIGX

EL1, NE1 code name of the first element.

EL2, NE2 code name of the second element.

NSEQ number of additional connections in the sequence.

NAD1 increment of the code number of the first element between two successive connections.

NAD2 increment of the code number of the second element between two successive connections.

ISOT set equal to 1, 2, or 3; specifies absolute permeability to be *PER (ISOT)* for the materials in elements (*EL1, NE1*) and (*EL2, NE2*), where *PER* is read in block **ROCKS**. This allows assignment of different permeabilities, e.g., in the horizontal and vertical direction.

D1 distance (m) from first element to interface.

D2 distance (m) from second element to interface.

AREAX interface area (m²).

BETAX cosine of the angle between the gravitational acceleration vector and the line between the two elements. $GF \times BETAX > 0$ (<0) corresponds to first element being above (below) the second element.

SIGX “radiant emittance” factor for radiative heat transfer, which for a perfectly “black” body is equal to 1. The rate of radiative heat transfer between the two grid blocks is

$$G_{rad} = SIGX * \sigma_0 * AREAX * (T_2^4 - T_1^4) \quad (51)$$

where $\sigma_0 = 5.6687e-8 \text{ J/m}^2 \text{ K}^4 \text{ s}$ is the Stefan-Boltzmann constant, and T_1 and T_2 are the absolute temperatures of the two grid blocks. *SIGX* may be entered as a negative number, in which case the absolute value will be used, and heat conduction at the connection will be suppressed. *SIGX* = 0 will result in no radiative heat transfer.

Repeat record **CONNE.1** for the number of connections desired.

Record **CONNE.2** A blank record closes the **CONNE** data block. Alternatively, connection information may terminate on a record with ‘+++’ typed in the first three columns, followed by element cross-referencing information. This is the termination used when generating a MESH file with TOUGH3.

GENER introduces sinks and/or sources.

Record **GENER.1**

Format (A3, I2, A3, I2, 4I5, 5X, A4, A1, 3E10.4)

EL, NE, SL, NS, NSEQ, NADD, NADS, LTAB, TYPE, ITAB, GX, EX, HX

EL, NE code name of the element containing the sink/source.

SL, NS code name of the sink/source. The first three characters are arbitrary, the last two characters must be numbers.

NSEQ number of additional sinks/sources with the same injection/production rate (not applicable for *TYPE* = DELV).

NADD increment between the code numbers of two successive elements with identical sink/source.

NADS increment between the code numbers of two successive sinks/sources.

LTAB number of points in table of generation rate versus time. Set 0 or 1 for constant generation rate. For wells on deliverability, *LTAB* denotes the number of open layers, to be specified only for the bottommost layer.

TYPE specifies different options for fluid or heat production and injection. For example, different fluid components may be injected, the nature of which depends on the EOS module being used. Different options for considering wellbore flow effects may also be specified.

HEAT Introduces a heat sink/source

COM1 component 1 (water)

WATE

COM2 component 2

injection only

COM3 component 3

...

COMn component n

MASS mass production rate specified

DELV well on deliverability, i.e., production occurs against specified bottomhole pressure. If well is completed in more than one layer, bottommost layer must be specified first, with number of layers given in *LTAB*. Subsequent layers must be given sequentially for a total number of *LTAB* layers.

F--- or f--- well on deliverability against specified wellhead pressure.

By convention, when the first letter of a type specification is F or f, TOUGH3 will perform flowing wellbore pressure corrections using tabular data of flowing bottomhole pressure vs. flow rate and flowing enthalpy. The tabular data used for flowing wellbore correction must be generated by means of a wellbore simulator ahead of a TOUGH3 run as ASCII data of 80 characters per record, according to the format specifications below.

The first record is an arbitrary title. The second record holds the number of flow rate and flowing enthalpy data (table points), *NG* and *NH*, respectively, in Format 2I5; in Table 6 we have *NG* = 11, *NH* = 9. This is followed by *NG* flow rate data in format 8E10.4, and *NH* enthalpy data also in format 8E10.4. After this come *NG* sets of *NH* flowing bottomhole pressure data in

format 8E10.4. The data in Table 6 were generated with the HOLA wellbore simulator (Aunzo et al., 1991) for a 0.2 m (\approx 8 inch) inside diameter well of 1,000 m feed zone depth with 7 bars wellhead pressure. Formation temperature for the conductive heat loss calculation in HOLA was assumed to increase linearly from 25 °C at the land surface to 275.5 °C at 750 m depth. Flow rates cover the range from 0.5 to 90.5 kg/s, and flowing enthalpies cover the range from 1,000 to 1,400 kJ/kg. A data record with very large bottomhole pressures of 55.55 MPa was added by hand for a very large hypothetical rate of 1,000 kg/s. This was done to avoid rates going out of table range during Newton-Raphson iteration in a TOUGH3 flow simulation.

```

*f725d* - (q,h) from ( .5000E+00, .1000E+07) to ( .9050E+02, .1400E+07)
  11      9
.5000E+00 .1050E+02 .2050E+02 .3050E+02 .4050E+02 .5050E+02 .6050E+02 .7050E+02
.8050E+02 .9050E+02      1.e3
.1000E+07 .1050E+07 .1100E+07 .1150E+07 .1200E+07 .1250E+07 .1300E+07 .1350E+07
.1400E+07
.1351E+07 .1238E+07 .1162E+07 .1106E+07 .1063E+07 .1028E+07 .9987E+06 .9740E+06
.9527E+06
.1482E+07 .1377E+07 .1327E+07 .1299E+07 .1284E+07 .1279E+07 .1279E+07 .1286E+07
.1292E+07
.2454E+07 .1826E+07 .1798E+07 .1807E+07 .1835E+07 .1871E+07 .1911E+07 .1954E+07
.1998E+07
.4330E+07 .3199E+07 .2677E+07 .2280E+07 .2322E+07 .2376E+07 .2434E+07 .2497E+07
.2559E+07
.5680E+07 .4772E+07 .3936E+07 .3452E+07 .2995E+07 .2808E+07 .2884E+07 .2967E+07
.3049E+07
.6658E+07 .5909E+07 .5206E+07 .4557E+07 .4158E+07 .3746E+07 .3391E+07 .3402E+07
.3511E+07
.7331E+07 .6850E+07 .6171E+07 .5627E+07 .5199E+07 .4814E+07 .4465E+07 .4208E+07
.3957E+07
.7986E+07 .7548E+07 .7140E+07 .6616E+07 .6256E+07 .5908E+07 .5634E+07 .5399E+07
.5128E+07
.8621E+07 .8177E+07 .7820E+07 .7560E+07 .7234E+07 .6814E+07 .6624E+07 .6385E+07
.6254E+07
.8998E+07 .8732E+07 .8453E+07 .8124E+07 .7925E+07 .7671E+07 .7529E+07 .7397E+07
.7269E+07
.5555e+08 .5555e+08 .5555e+08 .5555e+08 .5555e+08 .5555e+08 .5555e+08 .5555e+08
.5555e+08

```

Table 6. Flowing bottomhole pressures (in Pa) at 1000 m feed zone depth for a well of 20 cm (\approx 8 inch) inside diameter producing at 7 bar wellhead pressure (calculated from HOLA; Aunzo et al., 1991).

The data must be provided by means of a disk file, whose name consists of the four characters of the *TYPE* specification, and the one character of the

following *ITAB* parameter. For example, to use wellbore pressure data in a disk file called *f725d*, specify *TYPE* as 'f725', and specify *ITAB* as 'd'. Different wellbore tables, representing e.g. wells with different diameter, feed zone depth, and flowing wellhead pressure, may be used simultaneously in a TOUGH3 simulation. Also, several wells completed in different grid blocks may reference the same wellbore table.

The capability for flowing wellbore pressure correction is presently only available for wells with a single feed zone.

- ITAB* unless left blank, table of specific enthalpies will be read (*LTAB* > 1 only). When time-dependent injection is used, *ITAB* must be specified non-blank, and specific enthalpy data must be given.
- GX* constant generation rate; positive for injection, negative for production; *GX* is mass rate (kg/sec) for generation types COM1, COM2, COM3, etc., and MASS; it is energy rate (J/s) for a HEAT sink/source. For wells on deliverability, *GX* is productivity index PI (m³), Eq. (30).
- EX* fixed specific enthalpy (J/kg) of the fluid for mass injection (*GX*>0). For wells on deliverability against fixed bottomhole pressure, *EX* is bottomhole pressure P_{wb} (Pa), at the center of the topmost producing layer in which the well is open.
- HG* thickness of layer (m; wells on deliverability with specified bottomhole pressure only).

Record **GENER.1.1** (optional, *LTAB* > 1 only)

Format (4E14.7)
F1(L), L=1, *LTAB*

- F1* generation times

Record **GENER.1.2** (optional, *LTAB* > 1 only)

Format (4E14.7)
F2(L), L=1, *LTAB*

- F2* generation rates.

Record **GENER.1.3** (optional, $LTAB > 1$ and $ITAB$ non-blank only; this data must be provided for injection at time-dependent rates)

Format (4E14.7)

F3(L), L=1, LTAB

F3 specific enthalpy of produced or injected fluid.

Repeat records **GENER.1**, 1.1, 1.2, and 1.3 for the number of sinks/sources desired.

Record **GENER.2** A blank record closes the **GENER** data block.
Alternatively, generation information may terminate on a record with ‘+++’ typed in the first three columns, followed by element cross-referencing information.

In addition to the standard input format, time-dependent generation rates (i.e., if $LTAB > 1$ in block **GENER.1**) can be provided as a free-format table with time in the first column, injection or production rate in the second column, and (if $ITAB$ is not left blank) specific enthalpy in the third column. The number of table rows is given by $LTAB$. The tabular format is chosen by providing the character “T” or “D” in Column 7 after keyword **GENER**. Moreover, time and rate conversion factors can be given in Columns 11–20 and 21–30. If character “D” is specified in Column 7, time can be given in (any) date format; it will be converted to seconds (relative to the first date given). These conversion factors only apply to sinks/source with time-dependent generation rates (i.e., constant rates given in Columns 41–50 of block **GENER.1** are not affected). The free-format options are only available if sinks/sources are given directly in the TOUGH3 input deck. The external file **GENER** has to be provided in the standard format.

TIMBC permits the users to specify time-dependent Dirichlet boundary conditions. All values in this data block are read in free format.

Record **TIMBC.1**

NTPTAB

NTPTAB number of elements with time-dependent boundary conditions.

Record **TIMBC.2**

NBCP, NBCPV

NBCP number of times.

NBCPV number of primary variable that is time dependent, e.g., 1 for pressure.

Record **TIMBC.3**

BCELEM

BCELEM name of boundary element (start in Column 1)

Record **TIMBC.4**

TIMBCV(I), PGBCEL(I), I = 1, NBCP

TIMBCV time of primary variable *NBCPV* at boundary element *BCELM*

PGBCEL value of primary variable *NBCPV* at boundary element *BCELM* at time
TIMBCV

Repeat records **TIMBC.2**, **TIMBC.3**, and **TIMBC.4** for *NPTTAB* times

MOMOP describes additional options as shown in Table 7.

Format(27I1)

MOP2(I), I = 1,27

Table 7. More Options

<i>MOP2</i>	Value	Description
1	0, 1 2	Minimum number of Newton-Raphson iterations Allow convergence in a single Newton-Raphson iteration Perform at least two iterations; primary variables are always updated.
2	5-9	Length of element names (default: 5 characters)

	<p>Format of blocks ELEME, CONNE, INCON, and GENER change depending on Element-name length as follows:</p> <p>ELEME</p> <p>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)</p> <p>CONNE</p> <p>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)</p> <p>INCON</p> <p>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)</p> <p>GENER</p> <p>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)</p>
3	<p>Honoring generation times in block GENER</p> <p>0 Generation times ignored >0 Time steps adjusted to match generation times</p>
4	<p>Vapor pressure reduction</p> <p>0 No vapor pressure reduction at low liquid saturation >0 Reduces vapor pressure for $S_l < 0.02$ to prevent liquid disappearance by evaporation (only certain EOS modules)</p>
5	<p>Active Fracture Model (Appendix C)</p> <p>0 Active Fracture Model applied to liquid phase only >0 Active Fracture Model applied to all phases</p>
6	<p>Leverett scaling of capillary pressure</p> <p>0 No Leverett scaling >0 Rescale capillary pressure: $P_c = P_{c.ref} \sqrt{k_{ref}/k}$ if element-specific permeabilities are specified</p>
7	<p>Zero nodal distance</p> <p>0 Take absolute permeability from other element</p>

	>0	Take absolute and relative permeability from other element
8	0	Conversion of element names
	0	No conversion
	>0	Convert non-leading spaces in element names to zeros
9		Time stepping after time-step reduction to honor printout time
	0	Continue with time step used before forced time-step reduction
	>0	Continue with time step imposed by forced time-step reduction
10		Writing SAVE file
	0	Write SAVE file only at the end of a forward run
	1	Write SAVE file after each printout time
	2	Write separate SAVE files after each printout time
11		Water properties
	0	International Formulation Committee (1967)
	1	IAPWS-IF97
12		Enthalpy of liquid water
	0	Potential energy not included in enthalpy of liquid water
	>0	Potential energy included in enthalpy of all phases (Stauffer et al., 2014)
13		Adjustment of Newton-Raphson increment weighting
	0	No adjustment
	>0	Reduce <i>WNR</i> by <i>MOP2 (13)</i> percent if Newton-Raphson iterations oscillate and time step is reduced because <i>ITER=NOITE</i>
14		Print input file to the end of output file
	0	Do not reprint input files
	1	Print TOUGH3 input file to the end of TOUGH3 output file
15		Porosity used for calculation of rock energy content
	0	Use porosity of block ROCKS ; this assumes that the porosities provided in block INCON were the result of a pore compressibility/expansivity calculation; the “original” porosity from block ROCKS is used to compensate for equivalent rock-grain density changes.
	>0	Use porosity from block INCON ; this assumes that these porosities were not the result from a pore compressibility/expansivity calculation; changes 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 <code>PER2POR</code>)
17		Variables printed on FOFT files
	0	Print variables according to input data in block FOFT
	1	Print variables according to input data in block OUTPU
18		Void
19		Void
20		Reading anisotropic permeability modifiers in block ELEME
	0	Read isotropic permeability modifiers from columns 41–50
	1	Read anisotropic permeability modifiers from columns 81–110
	2	Read anisotropic permeability modifiers for <i>ISOT=1</i> from columns 41–50

		and for <i>ISOT</i> =2 and 3 from columns 91–110
21		Honoring generation times in blocks TIMBC .
	0	Time stepping ignores times specified in block TIMBC
	>0	Time steps adjusted to match times in block TIMBC
22		Format for reading coordinates in block ELEME
	0	Read coordinates in format 3E10.4 from columns 51–80
	>0	Read coordinates in format 3E20.14 from columns 51–110
23		Void
24		Void
25		Check mesh for duplicate element names
	0	Do not check the mesh
	1	Check the mesh
26		Printout format
	0	Both traditional output and CSV formats
	1	CSV/TECPLOT format (types separated; times combined)
	2	CSV/TECPLOT format (types separated; times separated)
	3	CSV/TECPLOT format (types combined; times combined)
27		Element naming in MESHMAKER
	0	Traditional element naming scheme
	1	Element name is equal to consecutive element number

OUTPU permits the users to obtain printout of specified variables (optional). **OUTPU** specifications will supersede the default condition specified in *KDATA* in data block **PARAM**. Block **OUTPU** must be specified after block **MULTI**.

Record **OUTPU.0** (optional)
Format(A20)
COUTFM

COUTFM a keyword indicating the desired output format, currently either CSV, TECPLOT, or PETRASIM

Record **OUTPU.1**
Format(I5)
MAXOUTVAR

MOUTVAR number of variables to be printed out.

Record **OUTPU.2, OUTPU.3, ..., OUTPU.(1+MOUTVAR)**

Format(A20,2I5)

COU TLN ,ID1,ID2

COU TLN name of variable, to be chosen among the available options. They include primary variables, secondary parameters, flow terms, and more. The name of variables is all capital letters, and should be typed in the input file as shown in Table 8.

ID1 first option for the corresponding keyword in *COU TLN* , as shown in Table 8.

ID2 second option for the corresponding keyword in *COU TLN* , as shown in Table 8.

Table 8. Keywords of block **OUTPU** and related output variables

Keyword	ID1	ID2	Output Variable	Header
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	
Element-related output variables				
MATERIAL ROCK ROCK TYPE	-	-	Material number	ROCK
MATERIAL NAME ROCK NAME ROCK TYPE NAME	-	-	Material name	ROCK
ABSOLUTE	ISOT	-	Absolute permeability in direction ISOT; if ISOT = 0, permeabilities related to all three directions are printed	ABS
POROSITY	-	-	Porosity	POR
TEMPERATURE	-	-	Temperature	TEMP
PRESSURE	IPH [#]	-	Pressure of phase IPH	PRES
SATURATION	IPH [#]	-	Saturation of phase IPH	SAT
RELATIVE	IPH [#]	-	Relative permeability of phase IPH	REL
VISCOSITY	IPH [#]	-	Viscosity of phase IPH	VIS
DENSITY	IPH [#]	-	Density of phase IPH	DEN
ENTHALPY	IPH [#]	-	Enthalpy of phase IPH	ENT
CAPILLARY	IPH [#]	-	Capillary pressure; if IPH = 1, difference between gas and aqueous phase pressures (for ECO2M, difference between gaseous CO ₂ and aqueous phase pressures); if IPH=2, difference between gas-NAPL phase pressures for TMVOC, and difference between gaseous and liquid CO ₂ phase pressures for ECO2M	PCAP
MASS FRACTION	IPH	IC	Mass fraction of component IC in phase IPH	X1,X2, ...

DIFFUSION1	IPH [#]	-	Diffusion parameter group 1 ($\phi\tau_0\tau_\beta\rho_\beta$) of phase $\beta = \text{IPH}$	DIFF1
DIFFUSION2	IPH [#]	-	Diffusion parameter group 2 ($\frac{P_0}{P} \left[\frac{T+273.15}{273.15} \right]^\theta$) of phase $\beta = \text{IPH}$	DIFF2
PSAT	-	-	Saturated vapor pressure	PSAT
BIOMASS	-	-	Biomass concentration	BIO
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	
Connection-related output variables				
TOTAL FLOW TOTAL FLOW RATE	-	-	Total flow rate	FLOW
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	FLOW
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 IPH IPH>0; IC<0: The flow of each component in phase IPH	FDIFF
HEAT FLOW	-	-	Heat flow rate	HEAT
VELOCITY	IPH [#]	-	Flow velocity of phase IPH	VEL
Generation-related output variables				

GENERATION GENERATION RATE	-	-	Production or injection rate	GEN
FLOWING ENTHALPY	-	-	Flowing enthalpy	ENTG
FRACTIONAL FLOW	IPH [#]	-	Fractional flow of phase IPH (production only)	FF
WELLBORE PRESSURE	-	-	Wellbore pressure (wells on deliverability only)	PWB
[#] If IPH = 0, the output variables of all phases are printed				

ENDCY closes the TOUGH3 input file and initiates the simulation.

8.3. Geometry Data

General Concepts

Flow geometry in TOUGH3 is defined by means of a list of volume elements ("grid blocks") and a list of flow connections between them, as in other "integral finite difference" codes (Edwards, 1972; Narasimhan and Witherspoon, 1976). This formulation can handle regular and irregular flow geometries in one, two, and three dimensions. Single- and multiple-porosity systems (porous and fractured media) can be specified, and higher-order methods, such as seven- and nine-point differencing, can be implemented by means of appropriate specification of geometric data (Pruess and Bodvarsson, 1983).

In TOUGH3, volume elements are identified by names that consist of a string of by default five characters, '12345', unless a different length of element names is specified in data block **MOMOP** ($MOP2(2) > 5$). These are arbitrary, except that the last two characters (#4 and 5) must be numbers if grids are generated using *NSEQ* in data block **ELEME** and **CONNE**; an example of a valid element name is "ELE10". Flow connections are specified as ordered pairs of elements, such as "(ELE10, ELE11)". A variety of options and facilities are available for entering and processing geometric data (see Figure 8). Element volumes and domain identification can be provided by means of a data block **ELEME** in the input file, while a data block **CONNE** can be used to supply connection data, including interface area, nodal distances from the interface, and orientation of the nodal line relative to the vertical. These data are internally written to a disk file MESH, which in turn initializes the geometry data arrays used during the flow simulation. It is also possible to omit the **ELEME** and **CONNE** data blocks from the input file, and provide geometry data directly on a disk file MESH.

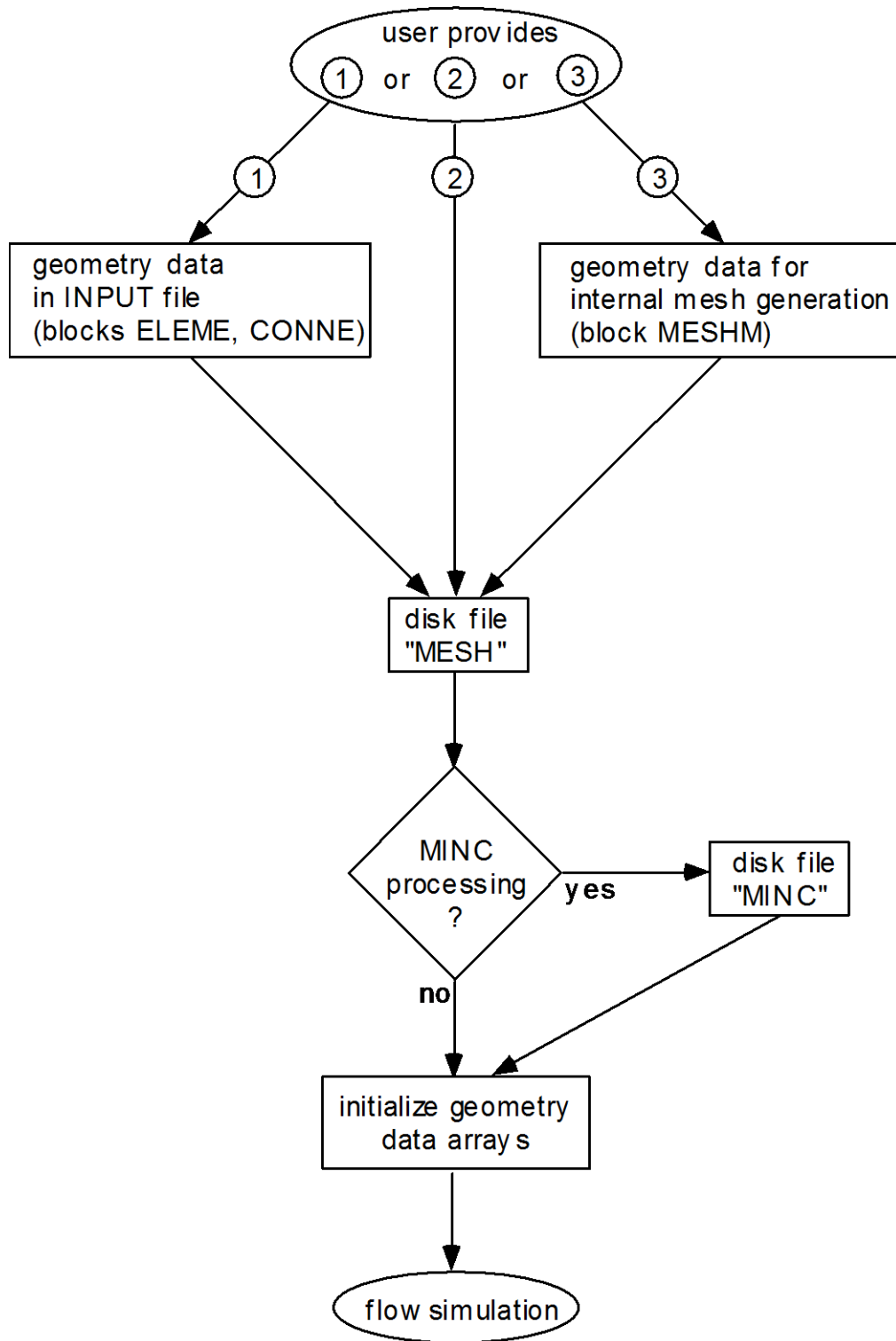


Figure 8. User options for supplying geometry data.

Meshmaker

TOUGH3 offers an additional way for defining flow system geometry by means of a special program module MESHMaker. This can perform a number of mesh generation and processing operations and is invoked with the keyword **MESHM** in the input file (see Figure 9). The MESHMaker module has a modular structure. At the present time, there are three sub-modules available in MESHMaker: keywords RZ2D or RZ2DL invoke generation of a one or two-dimensional radially symmetric R-Z mesh; XYZ initiates generation of a one, two, or three dimensional Cartesian X-Y-Z mesh; and MINC calls a modified version of the GMINC program (Pruess, 1983) to sub-partition a primary porous medium mesh into a secondary mesh for fractured media, using the method of “multiple interacting continua” (Pruess and Narasimhan, 1982, 1985), which will be described in detail below.

Several naming conventions for the elements created with keywords RZ2D (or RZ2DL) and XYZ have been adopted in the internal mesh generation process. In addition to the traditional TOUGH2 conventions, the other conventions are adopted to accommodate large numbers of grid blocks in cases with $MOP2(2) > 5$ or $MOP2(27) > 0$. Both RZ2D and XYZ assign all grid blocks to domain #1 (first entry in block **ROCKS**); a user desiring changes in domain assignments must do so by hand, either through editing of the MESH file with a text editor, or by means of preprocessing with an appropriate utility program, or by appropriate source code changes in subroutines WRZ2D and GXYZ. TOUGH3 runs that involve RZ2D or XYZ mesh generation will produce a special printout, showing element names arranged in their actual geometric pattern. An example is given in Section 3.2 in the addendum for EOS1.

The naming conventions for the MINC process are as follows. For a primary grid block with name '12345,' the corresponding fracture subelement in the secondary mesh is named ' 2345' (character #1 replaced with a blank for easy recognition). The successive matrix continua are labeled by running character #1 through 2, ..., 9, A, B, ..., Z. The domain assignment is incremented by 1 for the fracture grid blocks, and by 2 for the matrix grid blocks. Thus, domain assignments in data block **ROCKS** should be provided in the following order: the first entry is the single (effective) porous medium, then follows the effective fracture continuum, and then the rock matrix. Users should be aware that the MINC process may lead to ambiguous element names when the inactive element device is used to keep a portion of the primary mesh as unprocessed porous medium. Also, the MINC process may generate duplicate element names. TOUGH3 will check the element names after reading disk file MESH, and abort the simulation if duplicate element names are found.

MESHMAKER - Two-dimensional R-Z Grids							
-----1-----2-----3-----4-----5-----6-----7-----8							
MESHMAKER							
RIZ2D							
RAAD11							
NRAD							
RC(1)	RC(2)	RC(3)			RC(NRAD)		
EQUILID							
NEQU	<input type="checkbox"/>	DR					
LOGIAR							
NLOG	<input type="checkbox"/>	RLOG	DR				
LAYER							
NLAY							
H(1)	H(2)	H(3)			H(NLAY)		

MESHMAKER-Rectilinear Grids							
-----1-----2-----3-----4-----5-----6-----7-----8							
MESHMAKER							
XIYZI							
DEG							
<input type="checkbox"/>	N01	DEL					
NX	DX(1)	DX(2)	DX(3)			DX(N01)	
<input type="checkbox"/>	N02	DEL					
NX	DX(N01 + 1)	DX(N01 + 2)			DX(N01 + N02)		
<input type="checkbox"/>	N0	DEL					
NY	DY(1)	DY(2)	DY(3)			DY(N0)	
<input type="checkbox"/>	N0	DEL					
NZ	DZ(1)	DZ(2)	DZ(3)			DZ(N0)	
<input type="checkbox"/>							

MESHMAKER - MINC Processing for Fractured Media							
-----1-----2-----3-----4-----5-----6-----7-----8							
MESHMAKER							
MINC							
TYPE	<input type="checkbox"/>	DUAL					
PAR(1)	PAR(2)	PAR(3)	PAR(4)	PAR(5)	PAR(6)	PAR(7)	
J NUOLWHERE	PAR(1)	PAR(2)	PAR(3)	PAR(4)	PAR(5)	PAR(6)	PAR(7)
VOL(1)	VOL(2)	VOL(3)			VOL(NVOL)		
					VOL(NVOL)		

Figure 9. MESHMAKER input formats.

As a convenience for users desiring graphical display of data, the internal mesh generation process will also write nodal point coordinates on file MESH. By default these data are written in 3E10.4 format into columns 51-80 of each grid block entry in data block **ELEME**, unless a longer effective digit of 3E20.14 format into columns 51–110 is specified in data block **MOMOP** ($MOP2(22) > 0$). No internal use is made of nodal point coordinates in TOUGH3, except for optional initialization of a gravity-capillary equilibrium with EOS9 (see the addendum for EOS9) and for optional addition of potential energy to enthalpy with $MOP2(12) > 0$ in data block **MOMOP**.

Mesh generation and/or MINC processing can be performed as part of a simulation run. Alternatively, by closing the input file with the keyword **ENDFI** (instead of **ENDCY**), it is possible to skip the flow simulation and only execute the MESHMaker module to generate a MESH or MINC file. These files can then be used, with additional user-modifications by hand if desired, in subsequent flow simulations. Execution of MESHMaker produces printed output which is self-explanatory.

Multiple-porosity processing

Multiple-porosity processing for simulation of flow in naturally fractured reservoirs can be invoked by means of a keyword MINC, which stands for “multiple interacting continua” (Pruess and Narasimhan, 1982, 1985). The MINC-process operates on the data of the primary (porous medium) mesh as provided on disk file MESH, and generates a “secondary” mesh containing fracture and matrix elements with identical data formats on file MINC. By appropriate subgridding of the matrix blocks, as shown in Figure 10, and therefore by resolving the driving pressure, temperature, and mass fraction gradients at the matrix and fracture interface, the transient, multiphase interporosity flows between rock matrix and fractures can accurately be described. The MINC concept is based on the notion that changes in fluid pressures, temperatures, phase compositions, etc. due to the presence of sinks and sources (production and injection wells) will propagate rapidly through the fracture system, while invading the tight matrix blocks only slowly. Therefore, changes in matrix conditions will (locally) be controlled by the distance from the fractures. Fluid and heat flow from the fractures into the matrix blocks, or from the matrix blocks into the fractures, can then be modeled by means of one-dimensional strings of nested grid blocks, as shown in Figure 10. The MINC-method lumps all fractions within a certain reservoir subdomain into continuum # 1, all matrix material within a certain distance from the fractures into continuum # 2, matrix material at larger distance into continuum # 3, and so on. Quantitatively, the subgridding is specified by means of a set of volume fractions, into which the primary porous medium grid blocks are partitioned. The MINC-process in the MESHMaker module operates on

the element and connection data of a porous medium mesh to calculate, for given data on volume fractions, the volumes, interface areas, and nodal distances for a secondary fractured medium mesh. The information on fracturing (spacing, number of sets, shape of matrix blocks) required for this is provided by a "proximity function" $PROX(x)$ which expresses, for a given reservoir domain V_o , the total fraction of matrix material within a distance x from the fractures. If only two continua are specified (one for fractures, one for matrix), the MINC approach reduces to the conventional double-porosity method (Warren and Root, 1963). Full details are given in a separate report (Pruess, 1983). For any given fractured reservoir flow problem, selection of the most appropriate gridding scheme must be based on a careful consideration of the physical and geometric conditions of flow. The MINC approach is not applicable to systems in which fracturing is so sparse that the fractures cannot be approximated as a continuum.

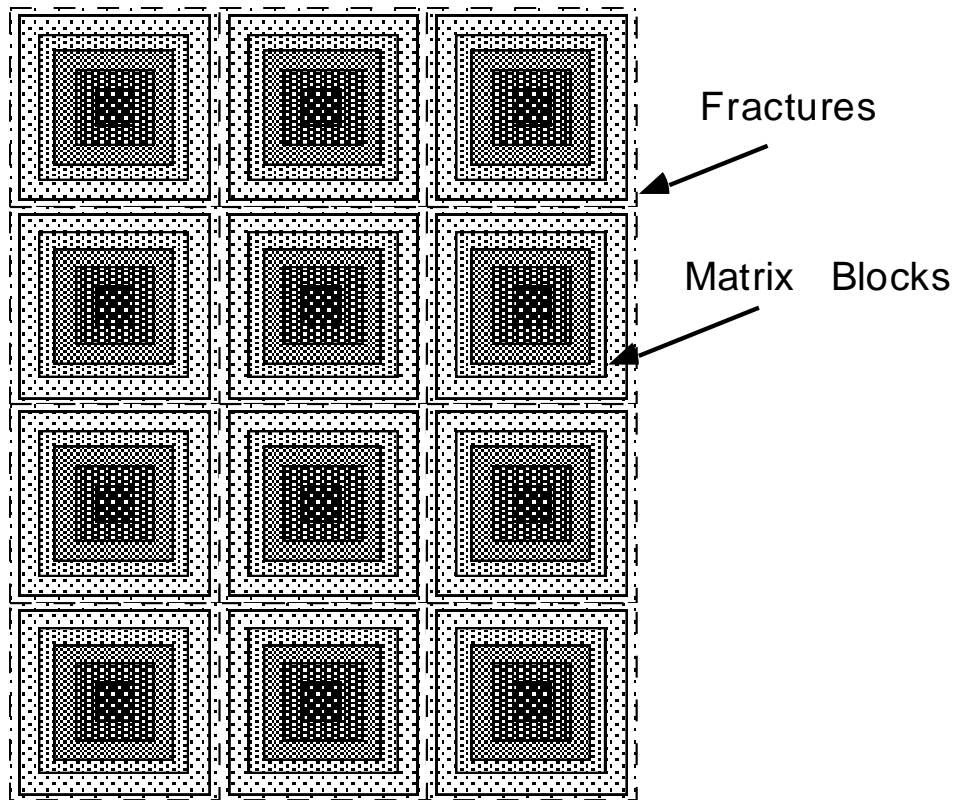


Figure 10. Subgridding in the method of "multiple interacting continua" (MINC).

The file MESH used in this process can be either directly supplied by the user, or it can have been internally generated either from data in input blocks **ELEME** and **CONNE**, or from RZ2D or XYZ mesh-making. The MINC process of sub-partitioning porous medium grid blocks into fracture and matrix continua will only operate on active grid blocks, while inactive grid blocks are left unchanged as single porous medium blocks. In TOUGH3, elements in data block **ELEME** (or file MESH) are taken to be “active” unless they have very large volumes, which are taken to be “inactive.” In order to exclude selected reservoir domains from the MINC process and make them remain single porous media, the user needs to change the volume of the corresponding blocks to a very large number before MINC partitioning is made. Note that here the concept of inactive blocks is used in an unrelated manner with respect to the one to maintain time-independent Dirichlet boundary conditions (see Section 6).

8.4. Input Formats for MESHMAKER

Generation of radially symmetric grids

Keyword RZ2D (or RZ2DL) invokes generation of a radially symmetric mesh. Nodal points will be placed half-way between neighboring radial interfaces. When RZ2D is specified, the mesh will be generated by columns; i.e., in the **ELEME** block we will first have the grid blocks at smallest radius for all layers, then the next largest radius for all layers, and so on. With keyword RZ2DL the mesh will be generated by layers; i.e., in the **ELEME** block we will first have all grid blocks for the first (top) layer from smallest to largest radius, then all grid blocks for the second layer, and so on. Apart from the different ordering of elements, the two meshes for RZ2D and RZ2DL are identical. The reason for providing the two alternatives is as a convenience to users in implementing boundary conditions by way of inactive elements (see Section 6). RZ2D makes it easy to declare a vertical column inactive, facilitating assignment of boundary conditions in the vertical, such as a gravitationally equilibrated pressure gradient. RZ2DL on the other hand facilitates implementation of areal (top and bottom layer) boundary conditions.

RADII is the first keyword following RZ2D; it introduces data for defining a set of interfaces (grid block boundaries) in the radial direction.

Record RADII.1

Format (I5)

NRAD

NRAD number of radius data that will be read. At least one radius must be provided, indicating the inner boundary of the mesh.

Record RADII.2, RADII.3, etc.

Format (8E10.4)
RC(I), I = 1, NRAD

RC (I) a set of radii in ascending order.

EQUIDistant introduces data on a set of equal radial increments.

Record EQUID.1

Format (I5, 5X, E10.4)
NEQU, DR

NEQU number of desired radial increments.

DR magnitude of radial increment.

Note: At least one radius must have been defined via block RADII before EQUID can be invoked.

LOGARithmic introduces data on radial increments that increase from one to the next by the same factor ($\Delta R_{n+1} = f \cdot \Delta R_n$).

Record LOGAR.1

Format (I5, 5X, 2E10.4)
NLOG, RLOG, DR

NLOG number of additional interface radii desired.

RLOG desired radius of the last (largest) of these radii.

DR reference radial increment: the first ΔR generated will be equal to $f \cdot DR$, with f internally determined such that the last increment will bring total radius to *RLOG*. $f < 1$ for decreasing radial increments is permissible. If *DR*

is set equal to zero, or left blank, the last increment DR generated before keyword LOGAR will be used as default.

Additional blocks RADII, EQUID, and LOGAR can be specified in arbitrary order.

Note: At least one radius must have been defined before LOGAR can be invoked. If $DR = 0$, at least two radii must have been defined.

LAYER introduces information on horizontal layers, and signals closure of RZ2D input data.

Record LAYER.1

Format (I5)

NLAY

$NLAY$ number of horizontal grid layers.

Record LAYER.2

Format (8E10.4)

H(I), I = 1, NLAY

$H(I)$ a set of layer thicknesses, from top layer downward. By default, zero or blank entries for layer thickness will result in assignment of the last preceding non-zero entry. Assignment of a zero layer thickness, as needed for inactive layers, can be accomplished by specifying a negative value.

The LAYER data close the RZ2D data block. Note that one blank record must follow to indicate termination of the **MESHM** data block. Alternatively, keyword MINC can appear to invoke MINC-processing for fractured media (see below).

Generation of rectilinear grids

XYZ invokes generation of a Cartesian (rectilinear) mesh.

Record XYZ.1

Format (E10.4)

DEG

DEG angle (in degrees) between the Y-axis and the horizontal. If gravitational acceleration (parameter GF in record **PARAM.2**) is specified positive, $-90^\circ < DEG < 90^\circ$ corresponds to grid layers going from top down. Grids can be specified from bottom layer up by setting *GF* or *BETA* negative. Default ($DEG = 0$) corresponds to horizontal Y- and vertical Z-axis. X-axis is always horizontal.

Record XYZ.2

Format (A2, 3X, I5, E10.4)

NTYPE, NO, DEL

NTYPE set equal to *NX*, *NY* or *NZ* for specifying grid increments in X, Y, or Z direction.

NO number of grid increments desired.

DEL constant grid increment for *NO* grid blocks, if set to a non-zero value.

Record XYZ.3 (optional, DEL = 0. or blank only)

Format (8E10.4)

DEL(I), I = 1, NO

DEL (I) a set of grid increments in the direction specified by *NTYPE* in record XYZ.2. Additional records with formats as XYZ.2 and XYZ.3 can be provided, with X, Y, and Z-data in arbitrary order.

Record XYZ.4 a blank record closes the XYZ data block.

Note that the end of block MESHMaker is also marked by a blank record. Thus, when MESHMaker/XYZ is used, there will be two blank records at the end of the corresponding input data block.

MINC processing for fractured media

MINC invokes postprocessing of a primary porous medium mesh from file *MESH*. The input formats in data block *MINC* are identical to those of the *GMINC* program (Pruess, 1983), with two enhancements: there is an additional facility for specifying global matrix-matrix connections (“dual permeability”); further, only active elements will be subjected to *MINC*-processing, the remainder of the *MESH* remaining unaltered as porous medium grid blocks.

PART is the first keyword following *MINC*; it will be followed on the same line by parameters *TYPE* and *DUAL* with information on the nature of fracture distributions and matrix-matrix connections.

Format (2A5, 5X, A5)

PART, *TYPE*, *DUAL*

PART identifier of data block with partitioning parameters for secondary mesh.

TYPE a five-character word for selecting one of the six different proximity functions provided in *MINC* (Pruess, 1983).

ONE-D: a set of plane parallel infinite fractures with matrix block thickness between neighboring fractures equal to *PAR (1)*.

TWO-D: two sets of plane parallel infinite fractures, with arbitrary angle between them. Matrix block thickness is *PAR (1)* for the first set, and *PAR (2)* for the second set. If *PAR (2)* is not specified explicitly, it will be set equal to *PAR (1)*.

THRED: three sets of plane parallel infinite fractures at right angles, with matrix block dimensions of *PAR (1)*, *PAR (2)*, and *PAR (3)*, respectively. If *PAR (2)* and/or *PAR (3)* are not explicitly specified, they will be set equal to *PAR (1)* and/or *PAR (2)*, respectively.

STANA: average proximity function for rock loading of Stanford large reservoir model (Lam et al., 1988).

STANB: proximity function for the five bottom layers of Stanford large reservoir model.

STANT: proximity function for top layer of Stanford large reservoir model.

Note: a user wishing to employ a different proximity function than provided in MINC needs to replace the function subprogram PROX(x) in file Mesh_Maker.f90 with a routine of the form:

```
FUNCTION PROX (x)
  PROX = (arithmetic expression in x)
  RETURN
END
```

It is necessary that PROX(x) is defined even when x exceeds the maximum possible distance from the fractures, and that PROX = 1 in this case. Also, when the user supplies his/her own proximity function subprogram, the parameter *TYPE* has to be chosen equal to ONE-D, TWO-D, or THRED, depending on the dimensionality of the proximity function. This will assure proper definition of innermost nodal distance (Pruess, 1983).

DUAL is a five-character word for selecting the treatment of global matrix matrix flow.

blank: (default) global flow occurs only through the fracture continuum, while rock matrix and fractures interact locally by means of interporosity flow (“double-porosity” model).

MMVER: global matrix-matrix flow is permitted only in the vertical; otherwise like the double-porosity model; for internal consistency this choice should only be made for flow systems with one or two predominantly vertical fracture sets.

MMALL: global matrix-matrix flow in all directions; for internal consistency only two continua, representing matrix and fractures, should be specified (“dual-permeability”).

Record PART.1

Format (2I3, A4, 7E10.4)

J, NVOL, WHERE, (PAR(I), I = 1, 7)

J total number of multiple interacting continua.

NVOL total number of explicitly provided volume fractions ($NVOL < J$). If $NVOL < J$, the volume fractions with indices $NVOL+1, \dots, J$ will be internally generated; all being equal and chosen such as to yield proper normalization to 1.

WHERE specifies whether the sequentially specified volume fractions begin with the fractures ($WHERE = 'OUT'$) or in the interior of the matrix blocks ($WHERE = 'IN'$).

PAR (I) $I = 1, 7$ holds parameters for fracture spacing (see above).

Record PART.2.1, 2.2, etc.

Format (8E10.4)

(VOL(I), I = 1, NVOL)

VOL (I) volume fraction (between 0 and 1) of continuum with index I (for $WHERE = 'OUT'$) or index $J + 1 - I$ (for $WHERE = 'IN'$). *NVOL* volume fractions will be read. For $WHERE = 'OUT'$, I = 1 is the fracture continuum, I = 2 is the matrix continuum closest to the fractures, I = 3 is the matrix continuum adjacent to I = 2, etc. The sum of all volume fractions must not exceed 1.

9. Output from TOUGH3

TOUGH3 produces a variety of printed output, most of which can be controlled by the user. In the initialization phase TOUGH3 writes out dimensions of problem-size dependent arrays and disk files in use to the standard output file (OUTPUT). This is followed by documentation on block-by-block permeability modification, on settings of the *MOP*- and *MOP2*-parameters for choosing program options, and on the EOS-module. During execution TOUGH3 can optionally generate a brief printout for Newtonian iterations and time steps. The file OUTPUT also includes the volume- and mass-balances at each specified printout times or time step.

TOUGH3 generates a pre-defined selection of element, connection, or generation variables based on *KDATA* in block **PARAM** (see Table 9). The list is not EOS-specific, except biomass, which will be included when simulating biodegradation reactions using TMVOC. Separate output files in the selected format (either CSV for positive *KDATA* or TECPLOT for negative *KDATA*) will be generated for element-, connection-, and sinks/sources-related outputs at user-specified simulation times in block **TIMES** or time step frequencies in block **PARAM**.

TOUGH3 allows the user to select the output variables to be printed using block **OUTPU**. The user can choose any number of element-, connection-, and generation-related output variables. A lumped set of primary variables or secondary parameters can be selected, and other information, such as grid-block or connection coordinates, index of elements, connection, or sinks/sources, and element names, can be included as well. The list of the output variables is shown in Table 8 in Section 8.2. The header and unit of variables/parameters selected for printout will be generated accordingly. An example **OUTPU** block is shown in Figure 11; excerpts of resulting output files for element-, connection- and generation-related output variables are shown in Figure 12, Figure

13 and Figure 14

"	ELEM", "	SOURCE", "	GEN"
"	", "	", "	(W) "
"TIME [sec]	0.31557600E+08"		
"	A1001", "	HTR 1",	0.300000000000E+04
"TIME [sec]	0.12559000E+09"		
"	A1001", "	HTR 1",	0.300000000000E+04
...			

respectively.

Table 9. Standard output based on absolute value of *KDATA*

abs (<i>KDATA</i>) = 1: a selection of element variables	
Output variable	Comment
Pressure	
Temperature	Only in nonisothermal mode
Saturation	Saturation of all phases
Mass fraction	Mass fraction of all components in all mobile phases
Relative permeability	Relative permeability of all mobile phases
Capillary pressure	Between mobile phases
Density	Density of all mobile phases
Porosity	
Biomass	Biomass of all microbial populations (only for TMVOC)
abs (<i>KDATA</i>) = 2: in addition, a selection of connection variables	
Heat flow	Only in nonisothermal mode
Total flow	
Phase flow	Flow of all mobile phases
Diffusive flow	Only when $NB = 8$
abs (<i>KDATA</i>) = 3: in addition, a selection of generation variables	
Generation rate	Mass (kg/s) or energy (J/s) rate depending on generation type
Flowing enthalpy	Only in nonisothermal mode
Fractional flow	Only for production
Wellbore pressure	Only for production wells operated on deliverability against specified bottomhole pressure

```

OUTPU-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6
8
PRESSURE
SECONDARY                2      6
SECONDARY                1      4
MASS FRACTION           2      2
FLOW                    1
VELOCITY                1
HEAT FLOW
GENERATION RATE

```

Figure 11. Example data block OUTPU.

```

"          ELEM", "          PRES_G", "          PCAP_L", "          DEN_G", "          X_AIR_L"
"          "      "          (PÄ)", "          (PÄ)", "          (KG/M**3)", "          (-)"
"TIME [sec] 0.31557600E+08"
"          A1001", 0.145733984563E+06, -0.500000000000E+09, 0.729984424325E+00, 0.000000000000E+00
"          A1002", 0.145733992876E+06, -0.333270877952E+08, 0.839804437107E+00, 0.000000000000E+00
"          A1003", 0.130069098208E+06, -0.149548290610E+06, 0.755041253102E+00, 0.000000000000E+00
"          A1004", 0.118768201640E+06, -0.375912049570E+05, 0.693490433326E+00, 0.000000000000E+00
"          A1005", 0.109739217020E+06, -0.199789863209E+05, 0.644044319825E+00, 0.293927428376E-15
"          A1006", 0.102044115154E+06, -0.109444612669E+05, 0.601696773695E+00, 0.274267136463E-10
"          A1007", 0.100043502582E+06, -0.877952984850E+04, 0.676573675262E+00, 0.349785972071E-05
...
"TIME [sec] 0.12559000E+09"
"          A1001", 0.131667455445E+06, -0.500000000000E+09, 0.579679135438E+00, 0.000000000000E+00
"          A1002", 0.131667427270E+06, -0.500000000000E+09, 0.652436733809E+00, 0.000000000000E+00
"          A1003", 0.131667376616E+06, -0.500000000000E+09, 0.697207803100E+00, 0.000000000000E+00
"          A1004", 0.131667303336E+06, -0.500000000000E+09, 0.731317756019E+00, 0.000000000000E+00
"          A1005", 0.131667206441E+06, -0.500000000000E+09, 0.759680036685E+00, 0.000000000000E+00
"          A1006", 0.125971775053E+06, -0.385373109044E+06, 0.732766156518E+00, 0.000000000000E+00
"          A1007", 0.120105966675E+06, -0.609113801028E+05, 0.700795547905E+00, 0.000000000000E+00
...

```

Figure 12. Element-related output variables in CSV format based on **OUTPU** block shown in Figure 11.

```

"          ELEM1", "          ELEM2", "          FLOW_G", "          VEL_G", "          HEAT"
"          "      "          "          (KG/S)", "          (M/S)", "          (W)"
"TIME [sec] 0.31557600E+08"
"          A1001", "          A1002", 0.314097500713E-09, 0.442085908383E-09, -0.299974415519E+04
"          A1002", "          A1003", -0.117124968441E-02, -0.815791410995E-03, -0.299910711963E+04
"          A1003", "          A1004", -0.113996116095E-02, -0.668855689283E-03, -0.300203907460E+04
"          A1004", "          A1005", -0.111083200985E-02, -0.743130677782E-03, -0.300533553035E+04
"          A1005", "          A1006", -0.107980321381E-02, -0.905005010191E-03, -0.300885963338E+04
"          A1006", "          A1007", -0.292168383775E-03, -0.394691197242E-03, -0.301452215868E+04
"          A1007", "          A1008", -0.296897257092E-07, -0.400951136232E-07, -0.297658449442E+04
...
"TIME [sec] 0.12559000E+09"
"          A1001", "          A1002", -0.536396107829E-09, -0.109093702349E-08, -0.299835837082E+04
"          A1002", "          A1003", -0.246742921890E-08, -0.220645706376E-08, -0.299318958181E+04
"          A1003", "          A1004", -0.612573151962E-08, -0.338056661611E-08, -0.298393647773E+04
"          A1004", "          A1005", -0.118590967237E-07, -0.463183822453E-08, -0.296972471428E+04
"          A1005", "          A1006", -0.937133136787E-03, -0.278903292626E-03, -0.294927355061E+04
"          A1006", "          A1007", -0.111723414932E-02, -0.302832717904E-03, -0.295107682950E+04
...

```

Figure 13. Connection-related output variables in CSV format based on **OUTPU** block shown in Figure 11.

```

"      ELEM", "      SOURCE", "      GEN"
"      ", "      ", "      (W) "
"TIME [sec] 0.31557600E+08"
"      A1001", "      HTR 1", 0.300000000000E+04
"TIME [sec] 0.12559000E+09"
"      A1001", "      HTR 1", 0.300000000000E+04
...

```

Figure 14. Generation-related output variables in CSV format based on **OUTPU** block shown in Figure 11.

By default, outputs are printed to both the default standard output file (OUTPUT) and the CSV format files. Users may opt out printing output variables to the standard output file, particularly for large-scale simulations, to avoid creating a very big file. TOUGH3 also offers an option to generate a file in a format suitable for TECPLOT. The output variables written to the standard output file can be extracted for visualization using a reformatting program (such as EXT, a free post-processing program downloadable from the TOUGH website, 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, for instance, using TECPLOT). It should be noted that EXT calculates the components of 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.

Time series of some parameters for plotting can optionally be written to files in the CSV file format using blocks **FOFT** (for elements), **COFT** (for connections), and **GOFT** (for sinks and sources). TOUGH3 will generate separate files for each element, connection, and sink/source. While TOUGH3 has several input options to select the parameters to be written out, users desiring different parameters than what is coded in FGTab should modify that subroutine.

10. Sample Problems

10.1. Problem No. 1 (*tutorial*) – Infiltration and Drainage of Water

This example is prepared to provide a quick guide to using TOUGH3, from input file preparation to analysis of the output. In this quick guide, we will use an illustrative example of water infiltration in a laboratory column experiment (see Figure 15). The column (1 m in length) is homogeneous sand and is initially saturated with liquid water. The level of the downstream water reservoir is then lowered to 0.3 m above the bottom of the column, and water is drained to re-equilibrate to the new reservoir water level. Then, an infiltration flux is applied at the top of the column. Several steps and input files are required to simulate this example: generate the mesh (Section 10.1.1), assign material properties (Section 10.1.2), define the initial and boundary conditions and obtain a hydrostatic profile (Section 10.1.3), execute the drainage and infiltration simulations (Section 10.1.4), and analyze the output (Section 10.1.5). Here, we use EOS9 to simulate variably saturated flow of a single aqueous phase using the Richards' equation (Richards, 1931). For a detailed description of the input format, users should refer to Section 8 and the addendum for EOS9.

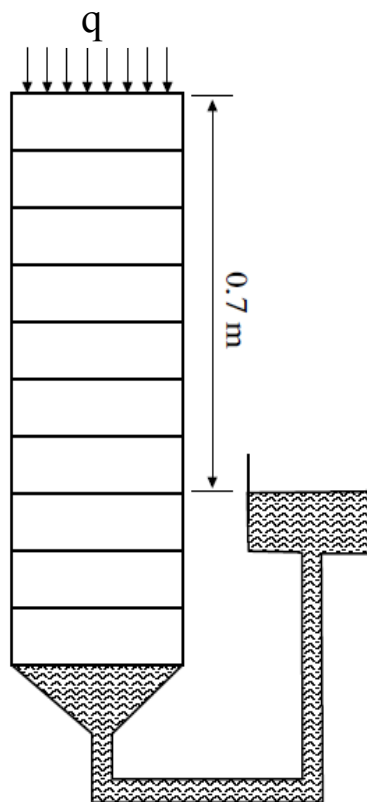


Figure 15. Schematic diagram of infiltration experiment in a vertical column.

10.1.1. Mesh Generation

To generate the mesh (problem P1a), internal or external mesh generators can be used along with manual modifications. Here we use the internal MESHMaker (block **MESHM**). This column experiment is considered as a 1D flow problem, and the input file (as shown in Figure 16) is prepared accordingly for the mesh generation. The first line is the title. Note that the user needs to specify NX , NY , and NZ , even though this is a 1D problem. Parameters dX and dY can be selected arbitrarily (as long as the product dX times dY is identical to the cross-sectional area of the column). The elements at the top and bottom boundaries are specified with a small value of dZ so that the nodal distance of these blocks to the boundary is small, setting boundary conditions right at the top and bottom of the column. As discussed in Section 6.4, no atmospheric boundary element is needed for EOS9 since EOS9 uses Richards' equation to describe variably saturated flow of a single aqueous phase and treats the gas phase as a passive bystander at constant pressure. The dummy boundary element at the top is used only to generate a hydrostatic pressure condition with the constant boundary pressure at the surface (instead of at the center of the first column element). The keyword **ENDFI** is used to close a TOUGH3 input file and to just generate the mesh without performing a flow simulation.

Next we need to identify the location of the TOUGH3 executable. If it is generated using the automated build system included in TOUGH3, it will be located in `esd-tough3/tough3-install/bin` with the name `tough3-eos9` since we will use EOS9. In a command prompt, type

```
esd-tough3/tough3-install/bin/tough3-eos9 P1a.txt P1a.out
```

An output MESH file, as shown in Figure 17, will be generated.

```
P1a: Generating a 1D mesh for simulating column experiment
MESHM----1----*-----2----*-----3----*-----4----*-----5----*-----6----*-----7----*-----8
XYZ
NX          1          1.0          Specify orientation
NY          1          1.0          Select dX and dY to yield cross section of column
NZ          1          1.0E-10       Add (dummy) top boundary element
NZ          20         0.05          Create uniform grid for 1 m long column
NZ          1          1.0E-10       Add (dummy) bottom boundary element
Needs one empty line to terminate XYZ block
Needs another empty line to terminate MESHMAKER
ENDFI----1----*-----2----*-----3----*-----4----*-----5----*-----6----*-----7----*-----8
```

Figure 16. Input file for problem P1a – generation of the 1D mesh for the column experiment.

```

ELEMEM0 NX= 1 NY= 1 NZ= 22
A11 1 10.1000E-090.1000E+01 0.5000E+000.5000E+00-.5000E-10
A21 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.2500E-01
A31 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.7500E-01
A41 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.1250E+00
.....
.....
AJ1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.8750E+00
AK1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.9250E+00
AL1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.9750E+00
AM1 1 10.1000E-090.1000E+01 0.5000E+000.5000E+00-.1000E+01

CONNE
AJ1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.8750E+00
AK1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.9250E+00
AL1 1 10.5000E-010.0000E+00 0.5000E+000.5000E+00-.9750E+00
AM1 1 10.1000E-090.1000E+01 0.5000E+000.5000E+00-.1000E+01
.....
.....
AI1 1AJ1 1 30.2500E-010.2500E-010.1000E+010.1000E+01
AJ1 1AK1 1 30.2500E-010.2500E-010.1000E+010.1000E+01
AK1 1AL1 1 30.2500E-010.2500E-010.1000E+010.1000E+01
AL1 1AM1 1 30.2500E-010.5000E-100.1000E+010.1000E+01

```

Figure 17. File MESH generated by running P1a.

10.1.2. Material Properties

Three materials are defined in block **ROCKS** (see Figure 18), which defines material names and specifies their hydrological and thermal properties. **SAND** is for the sand in the column, and **BOUND** is for the boundary elements. **REFCO** is a special domain only available for EOS9, which contains reference water properties. Block **RPCAP** is to define default relative permeability and capillary pressure curves for the system. Here, the van Genuchten (1980) relative permeability and capillary pressure functions ($IRP = ICP = 7$) are used. For more information on these and other functions available in TOUGH3, users should consult Appendices A and B. The **ELEME** and **CONNE** blocks are copied into the input file for problem P1b from the file MESH that was generated in the previous mesh generation step (P1a). The proper material is assigned to each element in block **ELEME**. Note that materials can be referred to by name or order number in block **ROCKS** (as shown in Figure 18, specifying **SAND** and 1 are equivalent; specifying **BOUND** and 2 are equivalent).

```

P1b: Specify properties and create hydrostatic pressure profile
ROCKS-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
SAND          2650.0      0.35                          1.000E-12
BOUND        2      2650.0      .9999                          1.000E-12                          1.E5

          1          0.2                          0.8
          1          0.0          0.0          1.0
REFCO          1.e5          20.      1000.0      0.001      4.5E-10

RPCAP-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
          7          0.5          0.2          1.0          0.0
          7          0.5          0.15          0.001      1.0e10          1.0
PARAM-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
          2 100          1100000100000000400001000
          86400.0 1.000E-01                          10.0
          1.000E-07
          1.0E+05

The ELEME and CONNE blocks were copied here from file MESH (output from Pla.txt)

ELEME-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
A11 1          BOUND0.1000E+500.1000E+01          0.5000E+000.5000E+00-.5000E-10
A21 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.2500E-01
A31 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.7500E-01
A41 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.1250E+00
.....
.....
AJ1 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.8750E+00
AK1 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.9250E+00
AL1 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.9750E+00
AM1 1          BOUND0.1000E-090.1000E+01          0.5000E+000.5000E+00-.1000E+01

CONNE-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
A11 1A21 1          30.5000E-100.2500E-010.1000E+010.1000E+01
A21 1A31 1          30.2500E-010.2500E-010.1000E+010.1000E+01
A31 1A41 1          30.2500E-010.2500E-010.1000E+010.1000E+01
.....
.....
AJ1 1AK1 1          30.2500E-010.2500E-010.1000E+010.1000E+01
AK1 1AL1 1          30.2500E-010.2500E-010.1000E+010.1000E+01
AL1 1AM1 1          30.2500E-010.5000E-100.1000E+010.1000E+01

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

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

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

```

Figure 18. Input file for problem P1b to create initial hydrostatic pressure profile.

10.1.3. Initialization of the Problem

Initial and boundary conditions are specified for the simulation of a hydrostatic pressure profile under fully liquid saturated conditions (problem P1b). TOUGH3 uses a set of primary

variables to define the system state, and the number of primary variables depends on the EOS module. EOS9 requires a single primary variable: pressure for saturated conditions and water saturation for unsaturated conditions. The default initial condition in block **PARAM.4** (1.e5 Pa = 1 bar) is applied to all elements. Computational parameters, printout options, and time stepping information are provided in block **PARAM**. For instance, *KDATA* and *MCYPR* in **PARAM.1** specify the amount of printout and the frequency of printout, respectively. To apply a constant-pressure boundary condition at the top of the model, the volume of the top element (A11 1) is changed from 1E-10 to 1E50, so that it maintains its initial pressure and can provide a source of water as the column is compressed.

Figure 19 shows an excerpt of the output. It took 14 time steps to reach steady state. The column is fully saturated, and the pressure profile shows an (almost⁷) linear increase with depth. Note that in this problem the magnitude of the gravitational acceleration vector *G_F* in **PARAM.2** is set as 10 m/sec² to ease the interpretation of the results.

```

OUTPUT DATA AFTER ( 14, 1)-2-TIME STEPS                                     THE TIME IS 0.473958E-02 DAYS
#####
TOTAL TIME      KCYC      ITER      ITCR      KON      DX1M      DX2M      DX3M      MAX. RES.      NER      KER      DELTEX
0.409500E+03    14         1        45         2         0.00000E+00 0.00000E+00 0.00000E+00 0.46571E-07    22         1         0.51200E+02
#####
ELEM. INDEX    PRES          SAT_L          X_WATER_L      REL_L          PCAP_L          DEN_L          POR
(PA)           (-)            (-)            (-)            (PA)           (KG/M**3)      (-)
A1101          1 0.10000E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A2101          2 0.10025E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A3101          3 0.10075E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A4101          4 0.10125E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A5101          5 0.10175E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A6101          6 0.10225E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A7101          7 0.10275E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A8101          8 0.10325E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
A9101          9 0.10375E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AA101         10 0.10425E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AB101         11 0.10475E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AC101         12 0.10525E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AD101         13 0.10575E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AE101         14 0.10625E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AF101         15 0.10675E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AG101         16 0.10725E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AH101         17 0.10775E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AI101         18 0.10825E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AJ101         19 0.10875E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AK101         20 0.10925E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AL101         21 0.10975E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.3500E+00
AM101         22 0.11000E+06 0.1000E+01 0.1000E+01 0.1000E+01 0.0000E+00 0.1000E+04 0.9999E+00
#####

```

Figure 19. Selected output at steady state for P1b.

⁷ Strictly speaking, the profile is not linear (as can be seen in the SAVE file), because water density slightly increases with depth as a function of pressure (and temperature if nonisothermal, NOT AN OPTION IN EOS9). This exemplifies the accuracy of the thermodynamics used in TOUGH3.

10.1.4. Simulating Drainage and Infiltration

After getting the hydrostatic condition, the level of the water reservoir is lowered to 0.3 m above the bottom of the column to establish an unsaturated zone (problem P1c). To simulate this drainage process, several modifications are made in the input file (see Figure 20). In block **ELEME**, the volume of the top element (A11 1) is changed back to its original value, and that of the bottom element (AM1 1) is set to be very large. The initial conditions are copied from the *SAVE* file of the previous run (P1b) to block **INCON** in the input file. The pressure of element AM1 1 is set to 1.03 bar, which is the hydrostatic pressure at the bottom of the column 0.3 m below the target water table. Water is drained based on this new bottom pressure for one day.

Then, water is infiltrated for 1 hour (problem P1d). During and after the infiltration experiment, water is drained (for a total of 24 hours). The infiltration rate is specified in the top element (A11 1) using block **GENER** (see Figure 21). To honor the generation times in block **GENER**, *MOP2 (3)* in block **MOMOP** is set to 1. Block **TIMES** is used to obtain printout at specific times in addition to printout specified in **PARAM**. The initial conditions in block **INCON** are copied from the *SAVE* file of the drainage simulation (P1c).

```

P1c: Drain column
ROCKS---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
SAND      2650.0      0.35      1.000E-12
BOUND    2      2650.0      .9999      1.000E-12      1.E5

      1      0.2      0.8
      1      0.0      0.0      1.0
REFCO      1.e5      20.      1000.0      0.001      4.5E-10

RPCAP---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
      7      0.5      0.2      1.0      0.0
      7      0.5      0.15      0.001      1.0e10      1.0
PARAM---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
      2 100      5100000100000000400001000
      86400.0 1.000E-01      10.0
      1.000E-07
      1.0E+05

ELEME---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
A11 1      10.1000E-090.1000E+01      0.5000E+000.5000E+00-.5000E-10
A21 1      10.5000E-010.0000E+00      0.5000E+000.5000E+00-.2500E-01
.....
.....
AL1 1      10.5000E-010.0000E+00      0.5000E+000.5000E+00-.9750E+00
AM1 1      BOUND0.1000E+500.1000E+01      0.5000E+000.5000E+00-.1000E+01

CONNE---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
A11 1A21 1      30.5000E-100.2500E-010.1000E+010.1000E+01
A21 1A31 1      30.2500E-010.2500E-010.1000E+010.1000E+01
.....
.....
AK1 1AL1 1      30.2500E-010.2500E-010.1000E+010.1000E+01
AL1 1AM1 1      30.2500E-010.5000E-100.1000E+010.1000E+01

INCON---1---*---2---*---3---*---4---*---5---*---6---*---7---*---8
A1101      0.99990000E+00
      0.10000000000000E+06
A2101      0.35000000E+00
      0.1002500000108E+06
A3101      0.35000000E+00
      0.1007500001159E+06
.....
.....
AK101      0.35000000E+00
      0.1092500191579E+06
AL101      0.35000000E+00
      0.1097500212948E+06
AM101      0.99990000E+00
      0.10300000000000E+06

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

```

Figure 20. Input file for problem P1c – drainage of water.

```

P1d: Infiltration for 1 hr and subsequent drainage
.....
.....

ELEME-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
A11 1          10.1000E-090.1000E+01          0.5000E+000.5000E+00-.5000E-10
A21 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.2500E-01
.....
.....
AL1 1          10.5000E-010.0000E+00          0.5000E+000.5000E+00-.9750E+00
AM1 1          BOUND0.1000E+500.1000E+01          0.5000E+000.5000E+00-.1000E+01

CONNE-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
A11 1A21 1          30.5000E-100.2500E-010.1000E+010.1000E+01
A21 1A31 1          30.2500E-010.2500E-010.1000E+010.1000E+01
.....
.....
AK1 1AL1 1          30.2500E-010.2500E-010.1000E+010.1000E+01
AL1 1AM1 1          30.2500E-010.5000E-100.1000E+010.1000E+01

START-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
GENER-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
A21 1INJ 1          4          WATE          0.01
          0.0          3600.0          3610.0          86400.0
          0.01          0.01          0.0          0.0

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

TIMES-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-----*-----8
3
3600.0          7200.0          36000.0

INCON
A1101          0.35000000E+00
          0.4604737132753E+00
A2101          0.35000000E+00
          0.4890926357798E+00
.....
.....
AL101          0.35000000E+00
          0.1027557815743E+06
AM101          0.99990000E+00
          0.10300000000000E+06

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

```

Figure 21. Input file for problem P1d.

10.1.5. Analysis of the Output

TOUGH3 produces a variety of printed output, most of which can be controlled by the user. Here, a pre-selected set of element and connection variables ($KDATA = 2$) are printed out to the default text output file (OUTPUT, as shown in Figure 19) and to the CSV format output files (OUTPUT_ELEME.csv and OUTPUT_CONNE.csv). Alternatively, users can select the output

variables to be printed using block **OUTPU**. Figure 22 shows the block **OUTPU** to print out the Z coordinate and water saturation values at each printout time. Since only element-related variables are selected, only file `OUTPUT_ELEME.csv` (see Figure 23) is created in this case. Using this output file, the change of water saturation during the infiltration and drainage experiment (problem P1d) is plotted in Figure 24.

```

OUTPU----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
3
SATURATION          1
COORDINATES        3
NO NAME

```

Figure 22. Data block **OUTPU**.

```

"          Z", "          SAT_L"
"          (M)", "          (-)"
"TIME [sec] 0.40000153E+00"
-0.500000000000E-10, 0.835830489632E+00
-0.250000000000E-01, 0.489320875121E+00
-0.750000000000E-01, 0.520986688010E+00
.....
.....
-0.925000000000E+00, 0.100000000000E+01
-0.975000000000E+00, 0.100000000000E+01
-0.100000000000E+01, 0.100000000000E+01
"TIME [sec] 0.42240000E+03"
-0.500000000000E-10, 0.914636184426E+00
-0.250000000000E-01, 0.719425065975E+00
.....
.....
.....
"TIME [sec] 0.36000000E+04"
-0.500000000000E-10, 0.999478996430E+00
-0.250000000000E-01, 0.999159494516E+00
-0.750000000000E-01, 0.996872022803E+00
.....
.....
.....
"TIME [sec] 0.72000000E+04"
-0.500000000000E-10, 0.687277037021E+00
-0.250000000000E-01, 0.758319390424E+00
-0.750000000000E-01, 0.810599762216E+00
.....
.....

```

Figure 23. File `OUTPUT_ELEME.csv` based on **OUTPU** block shown in Figure 22.

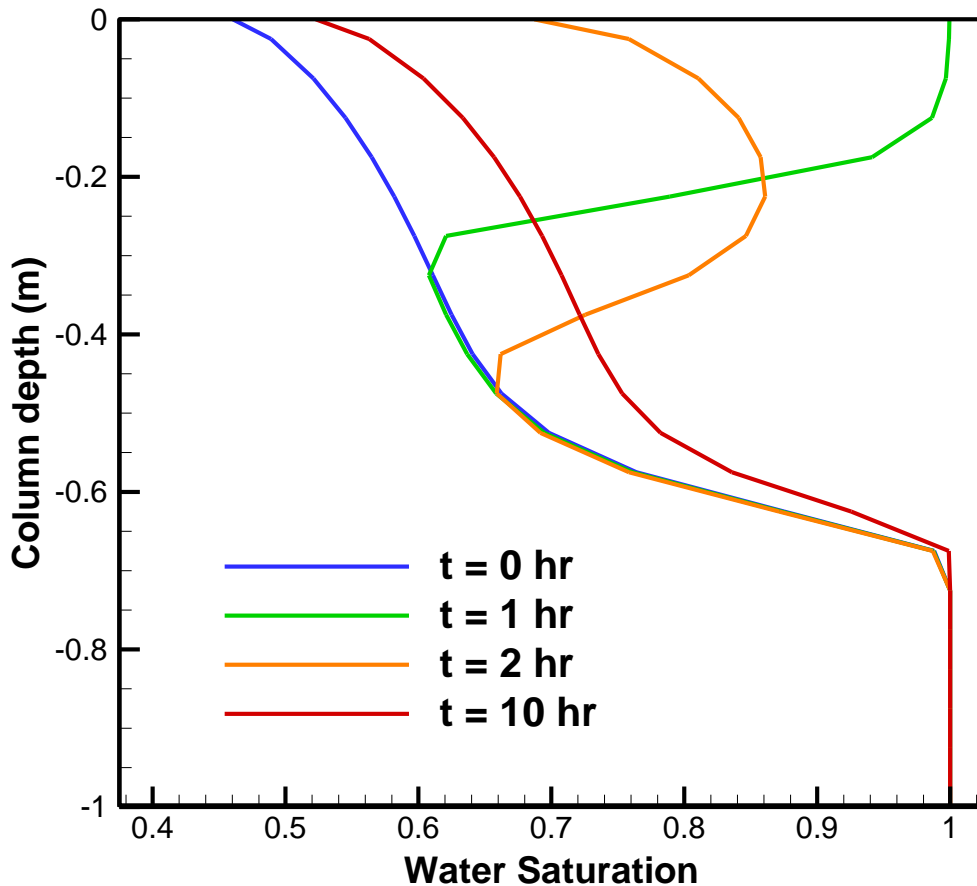


Figure 24. Temporal change of water saturation distribution in the column for problem P1d.

10.1.6. Parallel Execution of the Simulation

The input file described in Section 10.1.4 can also be executed in parallel. For a Linux system in which MPI libraries are already installed, users can directly execute the following command if they want to use 2 processors:

```
mpiexec -n 2 esd-tough3/tough3-install/bin/tough3-eos9
Pla.txt Pla.out
```

TOUGH3 will automatically switch to the default PETSc solver if a serial solver is specified in the input file when it is run with more than 1 processor. Since the volume elements are distributed across 2 different processors, the order of the elements in `OUTPUT_ELEME.csv` does not follow the order of the elements in `MESH`.

10.2. Problem No. 2 (*HysTest*) – Testing Hysteresis

10.2.1. Problem Setup

This sample problem is prepared in order to test all the branches of the hysteretic curves and the branch-switching processes, using regular hysteresis ($IRP=ICP=12$). It is also simulated with simplified hysteresis ($IRP=ICP=13$) and with non-hysteretic characteristic curves ($IRP=ICP=7$) in order to compare the three approaches. The problem (originally developed by Michele Carpita and Alfredo Battistelli of Snamprogetti, Italy) considers a series of successive CO₂ and water injections at various locations inside a 240 m long, one-dimensional, horizontal model without salt and under isothermal conditions. The whole simulation lasts 60 days, and there is no resting time between any two subsequent injections. The model is discretized into 60 elements, each 4-m wide. The entire grid is initialized at $P = 100$ bar and $T = 50$ °C conditions. A boundary element with a fixed pressure of $P = 100$ bar is added at $x = 0$ in order to induce all the injected fluids to move toward it.

Each injection element is located upgradient with respect to the previous injection element, to avoid spurious effects in the elements far from the fixed boundary. The injection sequence consists of five steps, detailed as follows, and illustrated in Figure 25:

- 1) A first CO₂ injection (2 kg/s for 2 days) in element A1146 ($x = 178$ m) in order to place the nearby elements along the primary drainage curve ($ICURV = 1$).
- 2) A first H₂O injection (0.01 kg/s for 23 days) in element A1149 ($x = 190$ m) to put some elements on the first-order imbibition curve ($ICURV = 2$).
- 3) Another CO₂ injection (0.5 kg/s for 5 days) in element A1152 ($x = 202$ m) to activate the second-order drainage curve ($ICURV = 3$).
- 4) A second H₂O injection (0.1 kg/s for 20 days) in element A1155 ($x = 214$ m) to activate the third-order imbibition curve ($ICURV = 4$). The H₂O injection is prolonged so as to let the water saturation increase above the domain of the second-order drainage curve (i.e., to $S_l > S_l^{\Delta 23}$), to check if there is a correct branch switch to the first-order imbibition curve ($ICURV = 2$).
- 5) A final CO₂ injection (1 kg/s for 10 days) in element A1158 ($x = 226$ m) to see what happens after the third-order imbibition.

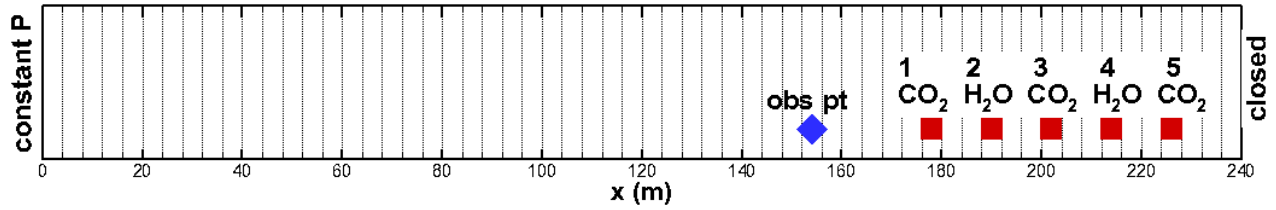


Figure 25. Illustration of the grid, boundary conditions, and source and observation locations.

Figure 26 shows the main part of the input file for the regular hysteresis case. Figure 27 compares the characteristic curve definitions for the regular hysteresis case, two versions of the simple hysteresis case, and two versions with no hysteresis. Note that for regular hysteresis ($IRP=ICP=12$), two lines of parameters are read in for both relative permeability (RP) and capillary pressure (CP) parameters, whereas for simple hysteresis ($IRP=ICP=13$) and no hysteresis ($IRP=ICP=7$), only one line is read for each. Also note that capillary pressure strength P_{c0} is input directly as $CP(3)$ for $ICP=12$, but for $ICP=13$ and $ICP=7$, $1/P_{c0}$ is read in $CP(3)$. Finally, for the no hysteresis case 7c, in order to specify zero residual gas saturation, a small non-zero value is given for $RP(4)$.

```

*HysTest12 1D model, sequence of injection of CO2 and H2O at various locations:
*CO2 injection at element A1146 = 178m +
*H2O injection at element A1149 = 190m +
*CO2 injection at element A1152 = 202m +
*H2O injection at element A1155 = 214m +
*CO2 injection at element A1158 = 226m
*ECO2N, Isothermal conditions, regular hysteresis
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
sand1  2  2600.e00      .30 2000.e-15 2000.e-15 2000.e-15      2.51      920.
      3.e-9
      12      0.917      0.150      0.200      0.500      1.000      0.00      0.97
      12      0.412      0.030      4.e3      4.e5      1.0      0.412      4.e3
      0.99      1.0

MULTI----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
      3      3      3      6
SELEC....2....3....4....5....6....7....8....9....10...11...12...13...14...15...16
      1
      .8      .8
HYSTE....2....3....4....5....6....7....8....9...10...11...12...13...14...15...16
      1      0      0
SOLVR----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
2 Z1  00      8.0e-1      1.0e-7
START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1-MOP: 123456789012345678901234----*----5----*----6----*----7----*----8
      21000      10002000301000021  4      3
      5184000.      1.e+0      A1140      9.806
      1.E-7      1.E00
      100.0e5      0.00      0.00      50.
ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A11 2      10.6000E+020.2400E+02      0.2001E+010.1500E+01-.2500E+01
A11 3      10.6000E+020.2400E+02      0.6001E+010.1500E+01-.2500E+01
A11 4      10.6000E+020.2400E+02      0.1000E+020.1500E+01-.2500E+01
A11 5      10.6000E+020.2400E+02      0.1400E+020.1500E+01-.2500E+01
      ...51 elements not shown...
A1156      10.6000E+020.2400E+02      0.2180E+030.1500E+01-.2500E+01
A1157      10.6000E+020.2400E+02      0.2220E+030.1500E+01-.2500E+01
A1158      10.6000E+020.2400E+02      0.2260E+030.1500E+01-.2500E+01
A1159      10.6000E+020.2400E+02      0.2300E+030.1500E+01-.2500E+01
A1160      10.6000E+020.2400E+02      0.2340E+030.1500E+01-.2500E+01
A1161      10.6000E+020.2400E+02      0.2380E+030.1500E+01-.2500E+01
A11 1      10.00      0.6000E-02      0.5000E-030.1500E+01-.2500E+01

CONNE
A11 1A11 2      10.5000E-030.2000E+010.1500E+02
A11 2A11 3      10.2000E+010.2000E+010.1500E+02
A11 3A11 4      10.2000E+010.2000E+010.1500E+02
A11 4A11 5      10.2000E+010.2000E+010.1500E+02
A11 5A11 6      10.2000E+010.2000E+010.1500E+02
      ...51 connections not shown...
A1156A1157      10.2000E+010.2000E+010.1500E+02
A1157A1158      10.2000E+010.2000E+010.1500E+02
A1158A1159      10.2000E+010.2000E+010.1500E+02
A1159A1160      10.2000E+010.2000E+010.1500E+02
A1160A1161      10.2000E+010.2000E+010.1500E+02

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

```

Figure 26. First part of input file for regular hysteresis sample problem.

```

TIMES----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
 25
43200.      86400.   129600.   172800.   345600.   518400.   864000.   1296000.
 1728000.  2160000.  2203200.  2246400.  2289600.  2332800.  2592000.  3024000.
 3456000.  3888000.  4320000.  4363200.  4406400.  4449600.  4492800.  4752000.
 5184000.
GENER----1----*----2-----3----*----4-----5-----6----*----7----*----8
A1146inj 1          6      COM31
      0.      172800.   2160000.   2592000.
      4320000.  5320000.
      2.00      0.00      0.00      0.00
      0.00      0.00
      .99380E+5 .99380E+5 .99380E+5 .99380E+5
      .99380E+5 .99380E+5
A1149inj 2          6      COM11
      0.      172800.   2160000.   2592000.
      4320000.  5320000.
      0.00      0.01      0.00      0.00
      0.00      0.00
      .99380E+5 .99380E+5 .99380E+5 .99380E+5
      .99380E+5 .99380E+5
A1152inj 3          6      COM31
      0.      172800.   2160000.   2592000.
      4320000.  5320000.
      0.00      0.00      0.500      0.00
      0.00      0.00
      .99380E+5 .99380E+5 .99380E+5 .99380E+5
      .99380E+5 .99380E+5
A1155inj 4          6      COM11
      0.      172800.   2160000.   2592000.
      4320000.  5320000.
      0.00      0.00      0.00      0.10
      0.00      0.00
      .99380E+5 .99380E+5 .99380E+5 .99380E+5
      .99380E+5 .99380E+5
A1158inj 5          6      COM31
      0.      172800.   2160000.   2592000.
      4320000.  5320000.
      0.00      0.00      0.00      0.00
      1.00      0.00
      .99380E+5 .99380E+5 .99380E+5 .99380E+5
      .99380E+5 .99380E+5

FOFT ----1----*----2-----3----*----4-----5-----6----*----7----*----8
A1140      0
A1146      0
A1149      0
A1152      0
A1155      0
A1158      0

COFT
A1101A1102      0

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

```

Figure 26 cont. Remainder of input file for regular hysteresis sample problem.

Case 12 (regular hysteresis) and case 13 (simple hysteresis) specify a relative permeability function with $m = 0.917$ ($RP(1)$), $S_{lr} = 0.15$ ($RP(2)$), and $S_{grmax} = 0.2$ ($RP(3)$) for $IRP=12$ and $RP(4)$ for $IRP=13$. For case 13L, m is only used for liquid relative permeability, and an independent parameter $m_{gas} = 0.68$ ($RP(5)$) is specified for gas relative permeability; additionally, S_{grmax} is increased from 0.2 to 0.3. The capillary pressure strength for case 13L is also increased compared to that for case 12, by decreasing I/P_{c0} from $2.5e-4$ to $1.75e-4$. These changes are made in order for saturation changes and the relative permeability and capillary pressure functions for case 13L to match those of case 12 as much as possible, as will be illustrated below. The two non-hysteretic cases ($IRP=7$) also use $m = 0.917$ ($RP(1)$) and $S_{lr} = 0.15$ ($RP(2)$), but use different values of residual gas saturation $RP(4)$: in case 7A, $S_{gr} = 0.2$, comparable to the value found during imbibition for the hysteretic cases, and in case 7C, $S_{gr} = 0.001$ (essentially zero), comparable to the value used during drainage for regular hysteresis.

Regular Hysteresis (Case 12)							
12	0.917	0.150	0.200	0.500	1.000	0.00	0.97
12	0.412	0.030	4.e3	4.e5	1.0	0.412	4.e3
0.99	1.0						
Simple Hysteresis (Case 13, same parameters as for regular hysteresis)							
13	0.917	0.150	1.000	0.200			
13	0.412	0.030	2.5e-4	4.e5	1.0	0.000	0.20
Simple Hysteresis (Case 13L, modified parameters to match relative permeability and capillary pressure functions of regular hysteresis)							
13	0.917	0.150	1.000	0.300	0.680		
13	0.412	0.030	1.75e-4	4.e5	1.0	0.000	0.20
No hysteresis (Case 7A, large residual gas saturation)							
07	0.917	0.150	1.000	0.200			
07	0.412	0.030	2.5e-4	4.e5	1.0		
No hysteresis (Case 7C, essentially zero residual gas saturation)							
07	0.917	0.150	1.000	0.001			
07	0.412	0.030	2.5e-4	4.e5	1.0		

Figure 27. Characteristic curve specifications for regular hysteresis ($IRP=ICP=12$), simple hysteresis ($IRP=ICP=13$), and no hysteresis ($IRP=ICP=7$).

10.2.2. Results

Figure 28 shows a series of snapshots of the saturation distribution in the model using regular hysteresis. During Step 1, CO₂ injection creates a CO₂ plume that moves toward the constant-pressure boundary at $x = 0$, reaching it between 1 and 1.5 days. During Step 2, the small injection of water pushes the trailing edge of the CO₂ plume slightly toward $x = 0$, but the residual CO₂ saturation is about 0.2, so the CO₂ saturation remains above this value throughout most of the model. During Step 3, the injection of CO₂ pushes the trailing edge of the original CO₂ plume toward $x = 0$, while creating a new CO₂ plume, which eventually overtakes and merges with the original plume. During Step 4, the moderate injection of water pushes the trailing edge of the CO₂ plume toward $x = 0$, again the CO₂ saturation remains above 0.2 throughout most of the model. However, this larger water injection is sufficient to enable complete dissolution of CO₂ close to the water source, where CO₂ saturation drops to zero. During Step 5, the injection of CO₂ initially pushes undersaturated water toward $x = 0$, where CO₂ dissolution results in CO₂ saturation dropping below 0.2. Eventually this region of low CO₂ saturation is replaced with another CO₂ plume.

Element A1140, which is located at $x = 154$ m, just downstream of all the injection elements, is chosen as the primary location at which to examine the saturation, relative permeability, and capillary pressure variations more closely. Figure 29 shows gas saturation S_g , residual gas saturation S_{gr} , and $ICURV$, the variable identifying the branch of the capillary pressure function, as a function of time at element A1140. Values of $ICURV$ are 1 for primary drainage, 2 for primary imbibition, 3 for secondary drainage, and 4 for secondary imbibition. Figure 29a compares results for hysteretic cases 12, 13, and 13L, while Figure 29b compares results for hysteretic case 12 and non-hysteretic cases 7A and 7C. Parameters for case 13 are identical to those for case 12, whereas parameters for case 13L are adjusted so that gas saturation versus time at element A1140 agrees better with that for case 12.

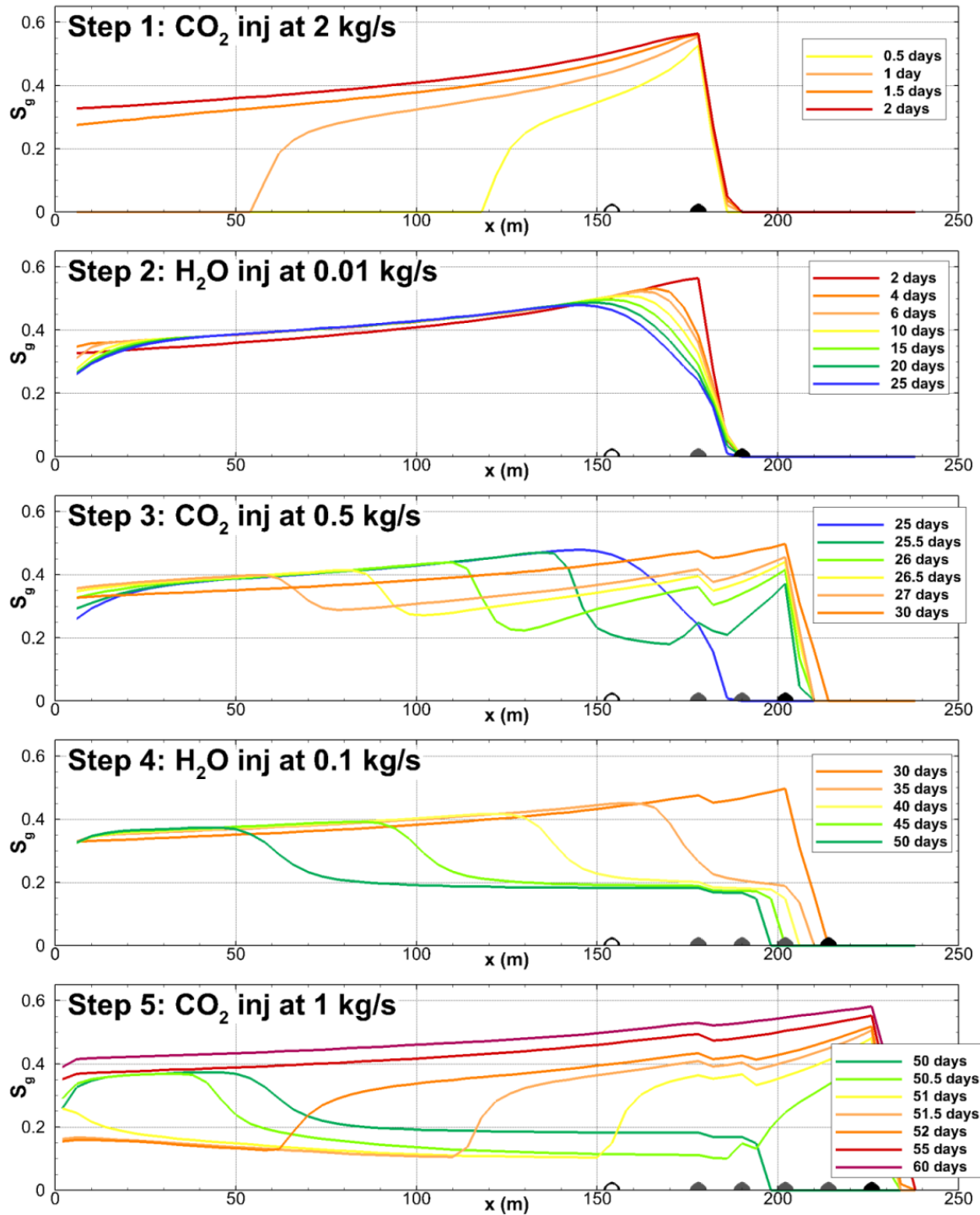


Figure 28. Saturation distributions in the model at various times during the five-step injection sequence using regular hysteresis ($IRP=ICP=12$). The current injection location is identified with a filled black dot; previous injection locations are identified with filled grey dots; the location of observation element A1140 is shown as an open dot.

Figure 30 shows capillary pressure and relative permeabilities as a function of liquid saturation at element A1140 for regular hysteresis, illustrating the different branches separated by turning points. Figure 31 compares capillary pressure and gas relative permeability curves for the regular hysteresis, simple hysteresis, and no-hysteresis cases. A key feature shown in Figure 29 is that S_{gr} begins increasing during the drainage period for simple hysteresis, whereas for regular hysteresis S_{gr} is zero until imbibition begins, and for the non-hysteretic cases S_{gr} is constant. This results in rather different saturation variations (Figure 29) and different capillary pressure and relative permeability curves (Figure 31). Simple hysteresis parameters can be adjusted to mimic regular hysteresis curves to a great extent, as shown for case 13L in Figure 31. Consequently, the saturation variation is also reasonably well represented, as shown in Figure 29a. However, non-hysteretic cases cannot capture the behaviour exhibited by hysteretic cases, as shown in Figure 29b: when S_{gr} is large as in case 7A, S_g is uniformly too large, and when S_{gr} is near zero as in case 7C, far too little trapping occurs during imbibition.

Note that for this sample problem, the characteristic curves for simple hysteresis were modified by comparing numerical simulation results for simple and regular hysteresis for a given element, as presented in the FOFT files. However, such modifications can be made without doing any numerical simulations, by entering the functional forms for regular and simple hysteretic characteristic curves in a spreadsheet, and varying parameters until an optimal match is obtained.

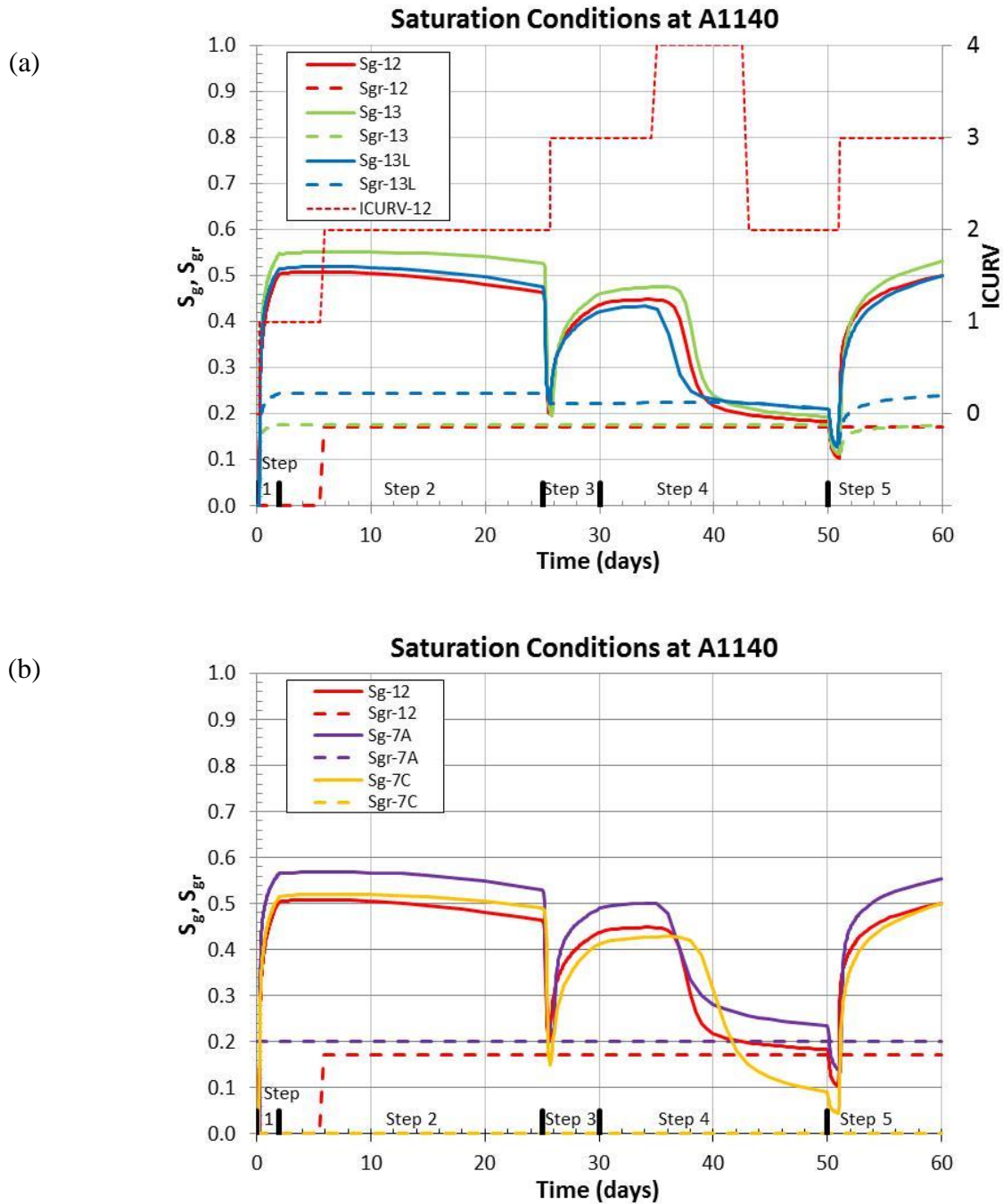


Figure 29. (a) Gas saturation S_g , residual gas saturation S_{gr} , and $ICURV$ versus time at element A1140, for regular hysteresis (curves labelled 12), simple hysteresis with the same parameters (curves labelled 13), and simple hysteresis with parameters modified to match regular hysteresis (curves labelled 13L); (b) S_g and S_{gr} versus time at element A1140, for no-hysteresis cases with large residual gas saturation (curves labelled 7A) and essentially zero residual gas saturation (curves labelled 7C).

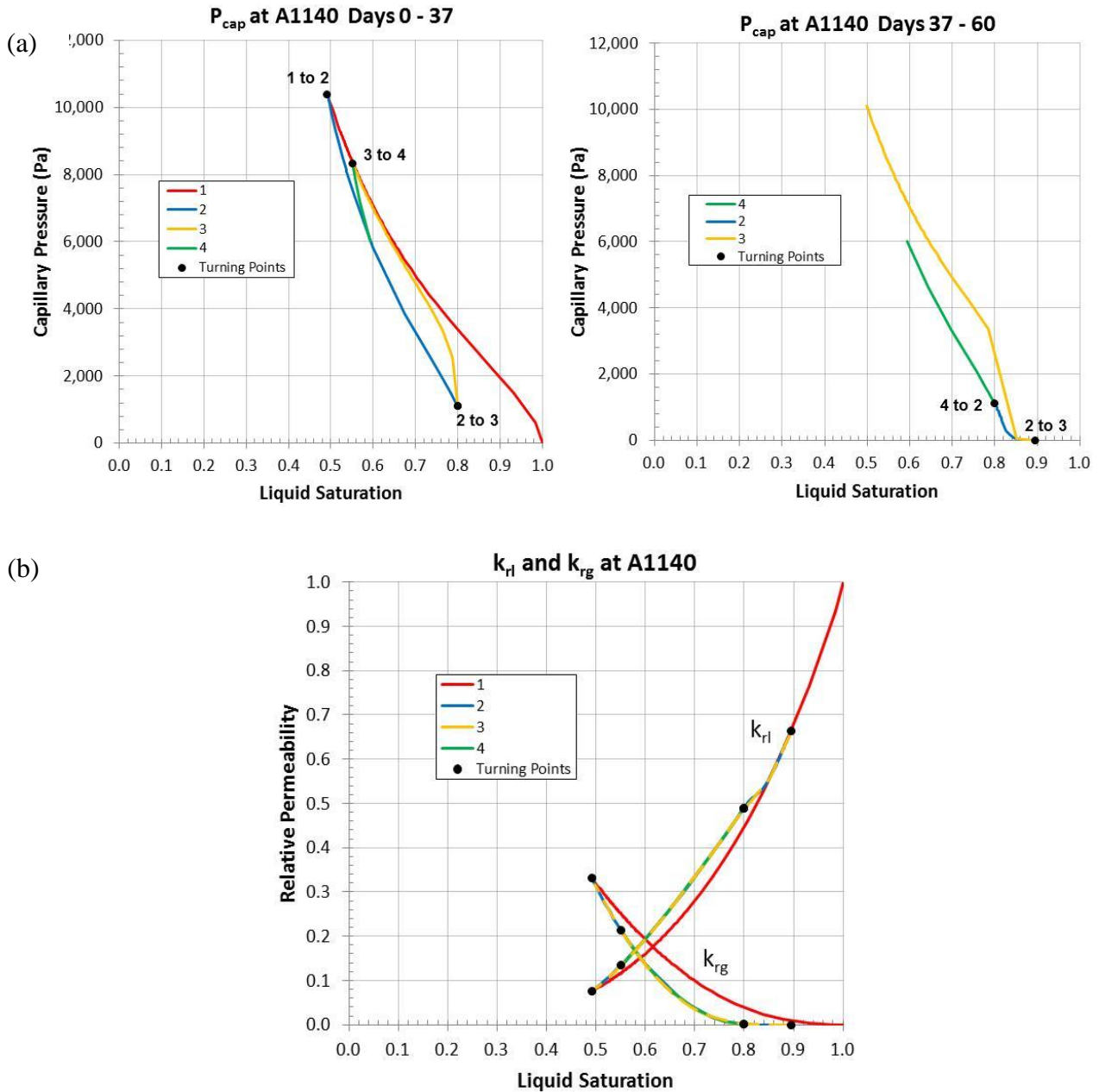


Figure 30. Regular hysteresis characteristic curves: (a) Capillary pressure versus liquid saturation at element A1140. The plot is divided into two parts, based on time, to enable the different branches of the capillary pressure curve to be seen more easily. The branch switches (turning points) are marked by black dots, with $ICURV$ for each switch labelled. (b) Relative permeability versus liquid saturation at element A1140. Branches 2, 3, and 4 all overlie one another.

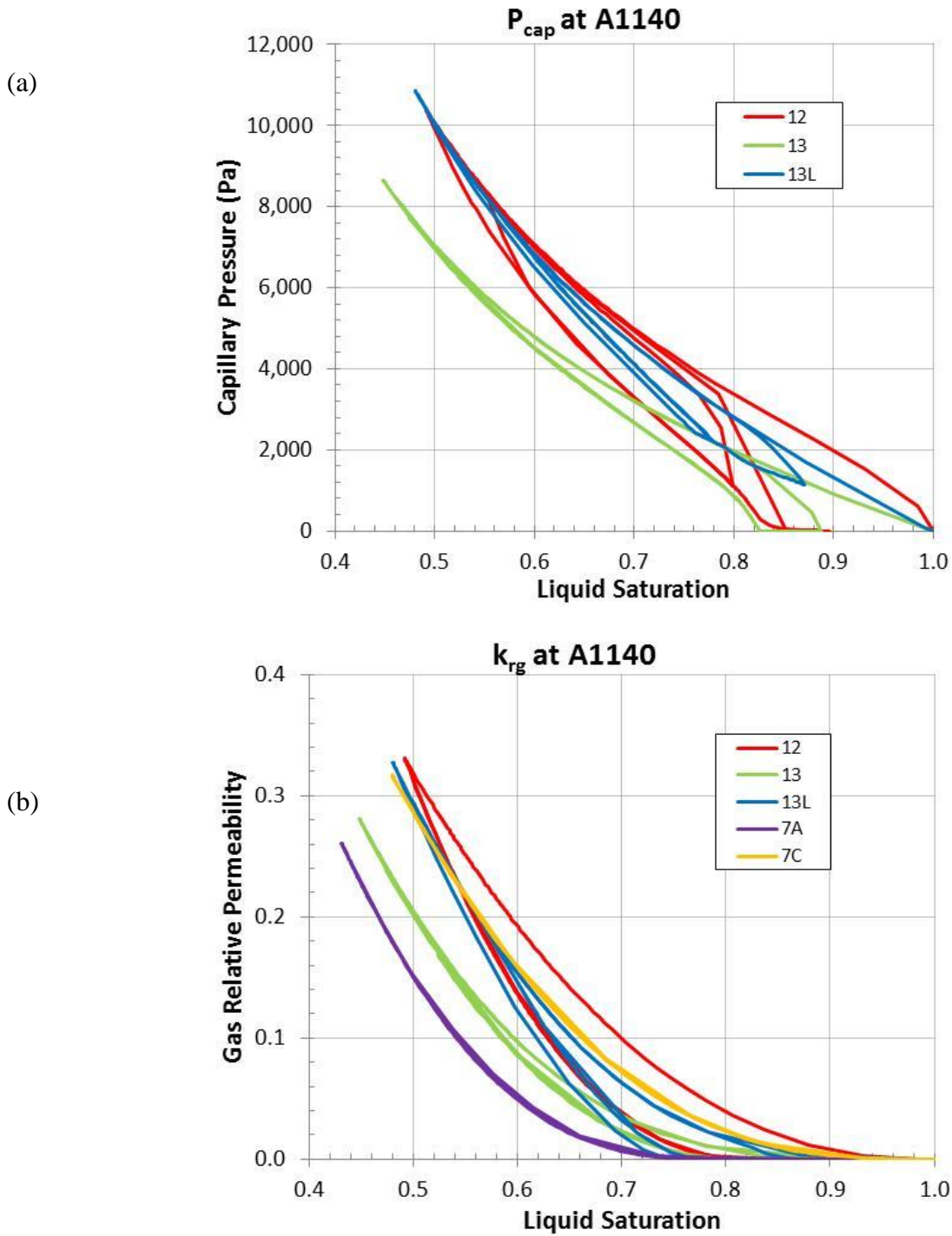


Figure 31. (a) Capillary pressure and (b) gas relative permeability versus liquid saturation for element A1140 for regular hysteresis (case 12), simple hysteresis (cases 13 and 13L), and no hysteresis (cases 7A and 7C). For cases 7A and 7C, the capillary pressure curve follows the primary drainage branch of the regular hysteresis curve. All liquid relative permeability curves follow the primary drainage branch of the regular hysteresis curve shown in Figure 30.

11. Concluding Remarks

TOUGH3 is developed to improve the capabilities, usability, and efficiency of the TOUGH simulators. The serial (TOUGH2 V2.1) and parallel (TOUGH2-MP V2.01) implementations are consolidated, and numerous enhanced features are implemented. Specifically the newly added PETSc parallel linear solvers can significantly improve the efficiency of the parallel (and serial as well, when a single processor is used) computations. Note that the parallel capabilities of TOUGH3 are not limited to the linear solver module. All other modules in Figure 5 have been efficiently parallelized. For example, various operations within the flow and transport module, such as the evaluations of the Jacobian matrix and the EOS parameters in each grid block, have been efficiently parallelized. Grid-based data are distributed, allowing high-resolution models (up to 100 million grid blocks) to be simulated. TOUGH3 includes additional EOS modules (EOS7C, EOS7CA, ECO2N, ECO2M, and TMVOC). Therefore, the broad user community can use TOUGH3 as a robust tool for diverse scientific and practical applications related to subsurface flow problems. In our future development effort, we intend to introduce parallel IO and automatic testing capabilities, and expand the processes that can be modeled by TOUGH3, including the coupled wellbore flow simulation feature.

Acknowledgement

This work was funded by the TOUGH Development Grant through the U.S. Dept. of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under the Contract No. DE-AC02-05CH11231. The authors would like to thank Y. Zhang for her careful review.

References

- Aunzo, Z.P., G. Bjornsson and G.S. Bodvarsson, Wellbore Models GWELL, GWNACL, and HOLA, Lawrence Berkeley National Laboratory Report LBL-31428, Berkeley, CA, October 1991.
- Balay, S., S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang, PETSc Web Page, <http://www.mcs.anl.gov/petsc>, 2016.
- Battistelli, A., C. Calore and K. Pruess, The Simulator TOUGH2/EWASG for Modeling Geothermal Reservoirs with Brines and Non-Condensable Gas, *Geothermics*, Vol. 26, No. 4, pp. 437 - 464, 1997.
- Battistelli, A., Modeling biodegradation of organic contaminants under multiphase conditions with TMVOCBio. *Vadose Zone Journal*, 3, 875–883, 2004.
- Battistelli A., Improving the treatment of saline brines in EWASG for the simulation of hydrothermal systems. In: Finsterle et al. (Eds.), Proceedings of the TOUGH Symposium 2012. Report LBNL-5808E. Lawrence Berkeley National Laboratory, Berkeley, Calif., 2012.
- Brooks, R.H., and A.T. Corey, Hydraulic properties of porous media: Hydrology Papers, Colorado State University, 24 p., 1964.
- Carslaw, H.S. and J.C. Jaeger, *Conduction of Heat in Solids, 2nd edition*. Oxford University Press, Oxford, pp. 336, 1959.
- Cass, A., G.S. Campbell and T.L. Jones, Enhancement of Thermal Water Vapor Diffusion in Soil. *Soil Sci. Soc. Am. J.*, Vol. 48, no. 1, pp. 25 - 32, 1984.
- Charbeneau, R. LNAPL Distribution and Recovery Model (LDRM) Volume 1: Distribution and Recovery of Petroleum Hydrocarbon Liquids in Porous Media, American Petroleum Institute API Publication 4760, 2007.
- Coats, K.H., Geothermal Reservoir Modeling, paper SPE-6892, presented at the 52nd Annual Fall Technical Conference and Exhibition of the SPE, Denver, Colorado, October 1977.
- Coats, K.H. and A.B. Ramesh, Effects of Grid Type and Difference Scheme on Pattern Steamflood Simulation Results, paper SPE-11079, presented at the 57th Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers, New Orleans, LA, September 1982.

- Corey, A.T., The Interrelation Between Gas and Oil Relative Permeabilities, *Producers Monthly*, 38-41, November 1954.
- Doughty C. Modeling geologic storage of carbon dioxide: comparison of non-hysteretic and hysteretic characteristic curves. *Energy Conversion and Management*, 48(6), 1768-1781, 2007.
- Doughty, C., User's Guide for Hysteresis Capillary Pressure and Relative Permeability Functions in TOUGH2, Report LBNL-6533E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2013.
- Driesner T., and C.H. Heinrich, The system H₂O–NaCl. Part I: Correlation formulae for phase relations in temperature–pressure–composition space from 0 to 1000°C, 0 to 5000 bar, and 0 to 1 XNaCl. *Geoch. Cosm. Acta*, 71, 4880–4901, 2007.
- Driesner T., The system H₂O–NaCl. Part II: Correlations for molar volume, enthalpy, and isobaric heat capacity from 0 to 1000°C, 1 to 5000 bar, and 0 to 1 XNaCl. *Geoch. Cosm. Acta*, 71, 4902–4919, 2007.
- Edlefsen, N.E. and A.B.C. Anderson., Thermodynamics of Soil Moisture, *Hilgardia*, 15 (2), 31–298, 1943.
- Edwards, A.L., TRUMP: A Computer Program for Transient and Steady State Temperature Distributions in Multidimensional Systems, National Technical Information Service, National Bureau of Standards, Springfield, VA, 1972.
- Fatt, I. and W.A. Klikoff, Effect of Fractional Wettability on Multiphase Flow Through Porous Media, *AIME Transactions*, 216, 246, 1959.
- Faust, C.R. Transport of immiscible fluids within and below the unsaturated zone: A numerical model, *Water Resour. Res.*, Vol. 21, No. 4, pp. 587 - 596, 1985.
- Grant, M.A., Permeability Reduction Factors at Wairakei, paper 77-HT-52, presented at AICHE-ASME Heat Transfer Conference, Salt Lake City, Utah, August 1977.
- Hadgu, T. and D.H. Freeston, A Multi-Purpose Wellbore Simulator. *Geothermal Resources Council Transactions*, Vol. 14, pp. 1279 - 1286, 1990.
- Hadgu, T., R.W. Zimmerman and G.S. Bodvarsson, Coupled Reservoir-Wellbore Simulation of Geothermal Reservoir Behavior. *Geothermics*, Vol. 24, No. 2, pp. 145-166, 1995.
- International Formulation Committee, *A Formulation of the Thermodynamic Properties of Ordinary Water Substance*, IFC Secretariat, Düsseldorf, Germany, 1967.

- Jung, Y. and A. Battistelli, User's guide for biodegradation reactions in TMVOCBio. Report LBNL-1005179, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2016.
- Klinkenberg, L.J., The Permeability of Porous Media to Liquids and Gases, in *API Drilling and Production Practice*, pp. 200–213, 1941.
- Kruger, P. and H.J. Ramey, Jr., Stimulation and Reservoir Engineering of Geothermal Resources, Stanford Geothermal Program Report SGR-TR-1, Stanford University, Stanford, CA, 1974.
- Lam, S.T., A. Hunsbedt, P. Kruger and K. Pruess, Analysis of the Stanford Geothermal Reservoir Model Experiments Using the LBL Reservoir Simulator, *Geothermics*, 17 (4), 595 - 605, LBL-25957, 1988.
- Land, C.S., Calculation of imbibition relative permeability for two- and three-phase flow from rock properties, paper SPE-1942, *SPE Journal*, Vol. 8, no. 2, 149–156, 1968.
- Lenhard, R.J. and J.C. Parker, A model for hysteretic constitutive relations governing multiphase flow, 2. Permeability-saturation relations, *Water Resour. Res.* Vol. 23, No. 12, pp. 2197-205, 1987
- Leverett, M.C., Capillary Behavior in Porous Solids, *Trans. Soc. Pet. Eng. AIME*, 142, 152-169, 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.*, Vol. 34, No. 10, pp. 2633-2646, 1998.
- Luckner, L., M. Th. van Genuchten, and D. Nielsen, A consistent set of parametric models for the two-phase flow of immiscible fluids in the subsurface, *Water Resour. Res.*, Vol. 25, No. 10, pp. 2187–2193, 1989.
- Mason, E.A. and A.P. Malinauskas, *Gas Transport in Porous Media: The Dusty Gas Model*, Elsevier, Amsterdam, The Netherlands, 1983.
- Millington, R.J. and J.P. Quirk, Permeability of Porous Solids, *Trans. Faraday Soc.*, 57, 1200-1207, 1961.
- Milly, P.C.D., Moisture and Heat Transport in Hysteretic, Inhomogeneous Porous Media: A Matric-Head Based Formulation and a Numerical Model, *Water Resour. Res.*, Vol. 18, No. 3, pp. 489 - 498, 1982.
- Moridis, G. and K. Pruess, TOUGH Simulations of Updegraff's Set of Fluid and Heat Flow Problems, Lawrence Berkeley Laboratory Report LBL-32611, Berkeley, CA, November 1992.

- Moridis, G. and K. Pruess, Flow and Transport Simulations Using T2CG1, a Package of Conjugate Gradient Solvers for the TOUGH2 Family of Codes, Lawrence Berkeley Laboratory Report LBL-36235, Berkeley, CA, 1995.
- Moridis, G. and K. Pruess, T2SOLV: An Enhanced Package of Solvers for the TOUGH2 Family of Reservoir Simulation Codes, *Geothermics*, Vol. 27, No. 4, pp. 415 - 444, 1998.
- Mualem, Y., A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media, *Water Resour. Res.*, Vol. 12, No. 3, pp. 513 - 522, 1976.
- Murray, L. and C. Gunn, Toward Integrating Geothermal Reservoir and Wellbore Simulation: TETRAD and WELLSIM, Proc. 15th NZ Geothermal Workshop, Auckland, New Zealand, 1993.
- Narasimhan, T.N. and P.A. Witherspoon, An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media, *Water Resour. Res.*, Vol. 12, No. 1, pp. 57 - 64, 1976.
- Narasimhan, T.N., P.A. Witherspoon and A.L. Edwards, Numerical Model for Saturated-Unsaturated Flow in Deformable Porous Media, Part 2: The Algorithm, *Water Resour. Res.*, Vol. 14, No. 2, pp. 255-261, 1978.
- Navarro, M, private communication, 2015.
- Oldenburg, C.M., G.J. Moridis, N. Spycher, and K. Pruess, EOS7C Version 1.0: TOUGH2 Module for Carbon Dioxide and Nitrogen in Natural Gas (Methane) Reservoirs, Report LBNL-56589, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2004.
- Oldenburg, C.M., EOS7CA Version 1.0: TOUGH2 Module for Gas Migration in Shallow Subsurface Porous Media Systems, Report LBNL-175204, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015.
- Pan, L., N. Spycher, C. Doughty, and K. Pruess, ECO2N V2.0: A TOUGH2 Fluid Property Module for Mixtures of Water, NaCl, and CO₂, Report LBNL-6930E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2015.
- Parker, J.C. and R.J. Lenhard, A model for hysteretic constitutive relations governing multiphase flow, 1. Saturation-pressure relations, *Water Resour. Res.*, Vol. 23, No. 12, pp. 2187-96, 1987.
- Parker, J.C., R.J. Lenhard and T. Kuppusamy. A parametric model for constitutive properties governing multiphase flow in porous media, *Water Resour. Res.*, Vol. 23, No. 4, pp. 618 - 624, 1987.
- Patterson, C.G. and R.W. Falta, A Simple History-Dependent Nonwetting-Phase Trapping

- Model for the Tough Simulators, In: Finsterle et al. (Eds.), Proceedings of the TOUGH Symposium 2012. Report LBNL-5808E. Lawrence Berkeley National Laboratory, Berkeley, Calif., 2012.
- Peaceman, D.W., *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam, The Netherlands, 1977.
- Peaceman, D.W., Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Nonsquare Grid Blocks and Anisotropic Permeability, paper SPE-10528, presented at the Sixth SPE Symposium on Reservoir Simulation of the Society of Petroleum Engineers, New Orleans, LA, January 1982.
- Pickens, J.F., R.W. Gillham and D.R. Cameron, Finite Element Analysis of the Transport of Water and Solutes in Tile-Drained Soils, *J. of Hydrology*, 40, 243-264, 1979.
- Pruess, K., GMINC - A Mesh Generator for Flow Simulations in Fractured Reservoirs, Lawrence Berkeley Laboratory Report LBL-15227, Berkeley, CA, March 1983.
- Pruess, K., TOUGH User's Guide, Nuclear Regulatory Commission Report NUREG/CR-4645; also Lawrence Berkeley Laboratory Report LBL-20700, 1987.
- Pruess, K., C. Oldenburg, and G. Moridis, TOUGH2 User's Guide, Version 2.1, Report LBNL-43134, Lawrence Berkeley Laboratory, Berkeley, Calif., 2012.
- Pruess, K., ECO2N: A TOUGH2 Fluid Property Module for Mixtures of Water, NaCl, and CO₂, Lawrence Berkeley National Laboratory Report LBNL-57952, Berkeley, Calif., 2005.
- Pruess, K., ECO2M: A TOUGH2 Fluid Property Module for Mixtures of Water, NaCl, and CO₂, Including Super- and Sub-Critical Conditions, and Phase Change Between Liquid and Gaseous CO₂, Lawrence Berkeley National Laboratory Report LBNL-4590E, Berkeley, Calif., 2011.
- Pruess, K., and T.N. Narasimhan, On Fluid Reserves and the Production of Superheated Steam from Fractured, Vapor-Dominated Geothermal Reservoirs, *J. Geophys. Res.*, Vol. 87, No. B11, pp. 9329 - 9339, 1982.
- Pruess, K. and G.S. Bodvarsson, A Seven-Point Finite Difference Method for Improved Grid Orientation Performance in Pattern Steam Floods, *Proceedings, Seventh Society of Petroleum Engineers Symposium on Reservoir Simulation*, Paper SPE-12252, pp. 175 - 184, San Francisco, CA, 1983.
- Pruess, K. and T.N. Narasimhan, A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media, *Soc. Pet. Eng. J.*, Vol. 25, No. 1, pp. 14 - 26, February 1985.

- 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.
- Richards, L.A., Capillary Conduction of Liquids Through Porous Mediums, *Physics*, Vol. 1, pp. 318 - 333, 1931.
- Sleijpen, G.L.G. and D. Fokkema, BiCGSTAB(m) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum, *Electronic Transactions on Numerical Analysis*, Vol. 1, pp. 11 - 32, 1993.
- 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.
- Stone, H.L. Probability model for estimating three-phase relative permeability, *Trans. SPE of AIME*, 249, 214-218, 1970.
- Thomas, G.W., *Principles of Hydrocarbon Reservoir Simulation*. International Human Resources Development Corporation, Boston, 1982.
- Tsang, Y.W. and K. Pruess, Further Modeling Studies of Gas Movement and Moisture Migration at Yucca Mountain, Nevada, Lawrence Berkeley Laboratory Report LBL-29127, Berkeley, CA, 1990.
- Tuminaro R. S., M. Heroux, S. A. Hutchinson, and J. N. Shadid, Official Aztec User's Guide: Version 2.1, December, 1999.
- Udell, K.S. and J.S. Fitch, Heat and Mass Transfer in Capillary Porous Media Considering Evaporation, Condensation, and Non-Condensable Gas Effects, paper presented at 23rd ASME/AIChE National Heat Transfer Conference, Denver, CO, 1985.
- van der Vorst, H.A., Bi-CGSTAB: A Fast and Smoothly Converging Variant of BiCG in the Presence of Rounding Errors, *SIAM J. Sci. Statist. Comput.*, Vol. 13, pp. 631 - 644, 1992.
- van Genuchten, M.Th., A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, *Soil Sci. Soc.* , Vol. 44, pp. 892 - 898, 1980.
- Vargaftik, N.B., *Tables on the Thermophysical Properties of Liquids and Gases*, 2nd Ed., John Wiley & Sons, New York, NY, 1975.

- Verma, A.K., K. Pruess, C.F. Tsang and P.A. Witherspoon, A Study of Two-Phase Concurrent Flow of Steam and Water in an Unconsolidated Porous Medium, Proc. 23rd National Heat Transfer Conference, Am. Society of Mechanical Engineers, Denver, CO, 135–143, 1985.
- Vinsome, P.K.W. and J. Westerveld, A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators, *J. Canadian Pet. Tech.*, Vol. 19, No. 3, pp. 87 – 90, July-September 1980.
- Walker, W.R., J.D. Sabey, and D.R. Hampton, Studies of Heat Transfer and Water Migration in Soils, Final Report, Department of Agricultural and Chemical Engineering, Colorado State University, Fort Collins, CO, 80523, April 1981.
- Warren, J.E. and P.J. Root, The Behavior of Naturally Fractured Reservoirs, *Soc. Pet. Eng. J., Transactions, AIME*, 228, 245-255, September 1963.
- Webb, S.W., Gas-Phase Diffusion in Porous Media - Evaluation of an Advective-Dispersive Formulation and the Dusty Gas Model for Binary Mixtures, *J. Por. Media*, Vol. 1, No. 2, pp. 187 - 199, 1998.
- Webb, S.W. and C.K. Ho, Review of Enhanced Vapor Diffusion in Porous Media, in: K. Pruess (ed.), *Proceedings of the TOUGH Workshop '98*, Lawrence Berkeley National Laboratory Report LBNL-41995, pp. 257 - 262, Berkeley, CA, 1998a.
- Webb, S.W. and C.K. Ho, Enhanced Vapor Diffusion in Porous Media - LDRD Final Report, Sandia National Laboratories Report SAND98-2772, Albuquerque, NM, 1998b.
- Wu, Y.S., K. Pruess and Z.X. Chen, Buckley-Leverett Flow in Composite Media, *SPE Advanced Technology Series*, Vol. 1, No. 2, pp. 36 - 42, 1993.
- Zhang, K., Y.-S. Wu, and K. Pruess, User's Guide for TOUGH2-MP—A Massively Parallel Version of the TOUGH2 Code. Report LBNL-315E, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2008.
- 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.

Appendix A. Relative Permeability Functions

TOUGH3 provides several relative permeability functions for two-phase or three-phase flow problems. The original TOUGH2 two-phase functions have been retained (from $IRP = 1$ to $IRP = 8$). The modified Brooks-Corey and van Genuchten models and two versions of the hysteresis model (regular and simple) are implemented ($IRP = 10, 11, 12,$ and 13). The three-phase functions from TMVOC and ECO2M start at $IRP = 31$. For TMVOC, if one of the two-phase functions is chosen, the NAPL relative permeability is assumed to be zero. For ECO2M, the two-phase functions are only applicable when an aqueous phase and a single CO_2 -rich phase are present. If one of the two-phase functions is chosen, the relative permeability of the CO_2 -rich phase will be the same function of saturation, regardless whether the CO_2 -rich phase is liquid or gas. The notation used below is: k_{rl} – aqueous phase relative permeability; k_{rg} – gas (for all two-phase EOSs and TMVOC) or gaseous CO_2 (for ECO2M) phase relative permeability; k_{rn} – NAPL (for TMVOC) or liquid CO_2 (for ECO2M) phase relative permeability. S_l , S_g , and S_n are the saturations of aqueous, gas (or gaseous CO_2) and NAPL (or liquid CO_2) phases, respectively.

$IRP = 1$ **Linear functions**
 k_{rl} increases linearly from 0 to 1 in the range
 $RP(1) \leq S_l \leq RP(3)$;
 k_{rg} increases linearly from 0 to 1 in the range
 $RP(2) \leq S_g \leq RP(4)$
Restrictions: $RP(3) > RP(1)$; $RP(4) > RP(2)$.
 If $RP(5) > 0$,
 k_{rn} increases linearly from 0 to 1 in the range
 $RP(1) \leq S_n \leq RP(3)$;
 If $RP(5) > 0$ and $RP(6) > 0$,
 k_{rn} increases linearly from 0 to 1 in the range
 $RP(5) \leq S_n \leq RP(6)$
Restrictions: $RP(6) > RP(5)$.

$IRP = 2$ $k_{rl} = S_l^{**} RP(1)$
 $k_{rg} = 1$.

$IRP = 3$ **Corey's curves (1954)**

$$k_{rl} = \hat{S}^4$$

$$k_{rg} = (1 - \hat{S})^2 (1 - \hat{S}^2)$$

where $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$

with $S_{lr} = RP(1)$; $S_{gr} = RP(2)$

Restrictions: $RP(1) + RP(2) < 1$

IRP = 4 **Grant's curves** (Grant, 1977)

$$k_{rl} = \hat{S}^4$$

$$k_{rg} = 1 - k_{rl}$$

where $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$

with $S_{lr} = RP(1)$; $S_{gr} = RP(2)$

Restrictions: $RP(1) + RP(2) < 1$

IRP = 5 **All phases perfectly mobile**
 $k_{rg} = k_{rl} = 1$ for all saturations; no parameters

IRP = 6 **Functions of Fatt and Klikoff** (1959)

$$k_{rl} = (S^*)^3$$

$$k_{rg} = (1 - S^*)^3$$

where $S^* = (S_l - S_{lr}) / (1 - S_{lr})$

with $S_{lr} = RP(1)$.

Restriction: $RP(1) < 1$.

IRP = 7 **van Genuchten-Mualem Model** (Mualem, 1976; van Genuchten, 1980)

$$k_{rl} = \begin{cases} \sqrt{S^*} \left\{ 1 - (1 - [S^*]^{1/\lambda})^\lambda \right\}^2 & \text{if } S_l < S_{ls} \\ 1 & \text{if } S_l \geq S_{ls} \end{cases}$$

Gas relative permeability can be chosen as one of the following two forms, the second of which is due to Corey (1954)

$$k_{rg} = \begin{cases} 1 - k_{rl} & \text{if } S_{gr} = 0 \\ (1 - \hat{S})^2 (1 - \hat{S}^2) & \text{if } S_{gr} > 0 \end{cases}$$

subject to the restriction $0 \leq k_{rl}, k_{rg} \leq 1$

Here, $S^* = (S_l - S_{lr}) / (S_{ls} - S_{lr})$, $\hat{S} = (S_l - S_{lr}) / (1 - S_{lr} - S_{gr})$

Parameters: $RP(1) = \lambda$
 $RP(2) = S_{lr}$
 $RP(3) = S_{ls}$
 $RP(4) = S_{gr}$

Notation: Parameter λ is m in van Genuchten's notation, with $m = 1 - 1/n$; parameter n is often written as β .

IRP = 8 **Functions of Verma et al. (1985)**

$$k_{rl} = \hat{S}^3$$

$$k_{rg} = A + B\hat{S} + C\hat{S}^2$$

where $\hat{S} = (S_l - S_{lr}) / (S_{ls} - S_{lr})$

Parameters as measured by Verma et al. (1985) for steam-water flow in an unconsolidated sand:

$$S_{lr} = RP(1) = 0.2$$

$$\begin{aligned}
S_{ls} &= RP(2) = 0.895 \\
A &= RP(3) = 1.259 \\
B &= RP(4) = -1.7615 \\
C &= RP(5) = 0.5089
\end{aligned}$$

$IRP = 10$

Modified Brooks-Corey Model

A modified version of the Brooks-Corey model (Luckner et al., 1989) has been implemented to prevent the capillary pressure from decreasing towards negative infinity as the effective saturation approaches zero. The modified Brooks-Corey model is invoked by setting both IRP and ICP to 10.

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

$$k_{rg} = \begin{cases} (1 - S_{ek})^2 \left(1 - S_{ek}^{\frac{2+\lambda}{\lambda}}\right) & \text{if } RP(3) = 0 \\ 1 - k_{rl} & \text{if } RP(3) \neq 0 \end{cases}$$

where

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

Parameters: $RP(1) = S_{lrk}$

$$RP(2) = S_{gr}$$

$RP(3)$ = flag to indicate which equation is used for k_{rg}

$IRP = 11$

Modified van Genuchten Model

A modified version of the van Genuchten model (Luckner et al., 1989) has been implemented to prevent the capillary pressure from decreasing towards negative infinity as the effective saturation approaches zero. The modified van Genuchten model is invoked by setting both IRP and ICP to 11.

$$k_{rl} = S_{ekl}^g \times S_{ekl}^{(1-g)/h} \times \frac{\hat{e}}{\hat{e}} \left[1 - \left(1 - S_{ekl}^{(1-g)/m} \right)^m \right]^2$$

$$k_{rg} = \begin{cases} \left(1 - S_{ekg} \right)^\zeta \left[1 - S_{ekg}^{1/m} \right]^{2m} & \text{if } RP(3) = 0 \\ 1 - k_{rl} & \text{if } RP(3) \neq 0 \end{cases}$$

where

$$S_{ekl} = \frac{S_l - S_{lrk}}{1 - S_{lrk}}, \quad S_{ekg} = \frac{S_l}{1 - S_{gr}}$$

Parameters: $RP(1) = S_{lrk}$

if negative, $S_{lrk} = 0$ for calculating k_{rg} , and absolute value is used for calculating k_{rl}

$RP(2) = S_{gr}$

if negative, $S_{gr} = 0$ for calculating k_{rl} , and absolute value is used for calculating k_{rg}

$RP(3) =$ flag to indicate which equation is used for k_{rg}

$RP(4) = \eta$ (default = 1/2)

$RP(5) = \varepsilon_k$

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 Appendix C)

$RP(7) = \zeta$ (default = 1/3)

$IRP = 12$

Regular hysteresis

The hysteretic form of the van Genuchten model (Parker and Lenhard, 1987; Lenhard and Parker, 1987) has been implemented. Details of the implementation are described in Doughty (2013). The regular hysteresis model is invoked by setting both IRP and ICP to 12.

$$k_{rl} = \sqrt{\bar{S}_l} \left[1 - \left(1 - \frac{\bar{S}_{gt}}{1 - \bar{S}_l^\Delta} \right) \left(1 - (\bar{S}_l + \bar{S}_{gt})^{1/m} \right)^m - \left(\frac{\bar{S}_{gt}}{1 - \bar{S}_l^\Delta} \right) \left(1 - (\bar{S}_l)^{1/m} \right)^m \right]^2$$

$$k_{rg} = k_{rg\max} (1 - (\bar{S}_l + \bar{S}_{gt}))^\gamma (1 - (\bar{S}_l + \bar{S}_{gt})^{1/m})^{2m},$$

where

$$\bar{S}_l = \frac{S_l - S_{lr}}{1 - S_{lr}}, \quad \bar{S}_l^\Delta = \frac{S_l^\Delta - S_{lr}}{1 - S_{lr}}, \quad \bar{S}_{gt} = \frac{S_{gr}^\Delta (S_l - S_l^\Delta)}{(1 - S_{lr})(1 - S_l^\Delta - S_{gr}^\Delta)}$$

$$S_{gr}^\Delta = \frac{1}{1/(1 - S_l^\Delta) + 1/S_{gr\max} - 1/(1 - S_{lr})}$$

S_l^Δ is the turning-point saturation, and S_{gr}^Δ is the residual gas saturation.

$RP(1) = m$; van Genuchten m for liquid relative permeability (need not equal $CP(1)$ or $CP(6)$); k_{rl} uses the same m for drainage and imbibition.

$RP(2) = S_{lr}$: $k_{rl}(S_{lr}) = 0$, $k_{rg}(S_{lr}) = k_{rg\max}$. Must have $S_{lr} > S_{lmin}$ in capillary pressure ($CP(2)$). S_{lr} is minimum saturation for transition to imbibition branch. For $S_l < S_{lr}$, curve stays on primary drainage branch even if S_l increases.

$RP(3) = S_{gr\max}$; maximum possible value of S_{gr}^Δ . Note that the present version of the code requires that $S_{lr} + S_{gr\max} < 1$, otherwise there will be saturations for which neither fluid phase is mobile, which the code cannot handle. Setting $S_{gr\max} = 0$ effectively turns off hysteresis. As a special option, a constant, non-zero value of S_{gr} may be employed by setting $CP(10) > 1$ and making $RP(3)$ negative. The code will set $S_{gr}^\Delta = -RP(3)$ for all grid blocks at all times.

$RP(4) = \gamma$; typical values 0.33 – 0.50.

$RP(5) = k_{rg\max}$

$RP(6)$ = fitting parameter for k_{rg} extension for $S_l < S_{lr}$ (only used when $k_{rg\max} < 1$); determines type of function for extension and slope of k_{rg} at $S_l = 0$.
 ≤ 0 use cubic spline for $0 < S_l < S_{lr}$, with slope at $S_l = 0$ of $RP(6)$
 > 0 use linear segment for $0 < S_l < RP(8) S_{lr}$ and cubic spline for $RP(8) S_{lr} < S_l < S_{lr}$, with slope at $S_l = 0$ of $-RP(6)$.

$RP(7)$ = numerical factor used for k_{rl} extension to $S_l > S_l^*$. $RP(7)$ is the fraction of S_l^* at which k_{rl} curve departs from the original van Genuchten function. Recommended range of values: 0.95–0.97. For $RP(7) = 0$, $k_{rl} = 1$ for $S_l > S_l^*$ (not recommended).

$RP(8)$ = numerical factor used for linear k_{rg} extension to $S_l < S_{lr}$ (only used when $k_{rg\max} < 1$). $RP(8)$ is the fraction of S_{lr} at which the linear and cubic parts of the extensions are joined.

$RP(9)$ = flag to turn off hysteresis for k_{rl} (no effect on P_c and k_{rg} ; to turn off hysteresis entirely, set $S_{gr\max} = 0$ in $RP(3)$).
 $= 0$ hysteresis is on for k_{rl}
 $= 1$ hysteresis is off for k_{rl} (force k_{rl} to stay on primary drainage branch (k_{rl}^d) at all times)

$RP(10) = m_{gas}$; van Genuchten m for gas relative permeability (need not equal $CP(1)$ or $CP(6)$); k_{rg} uses same m_{gas} for drainage and imbibition. If zero or blank, use $RP(1)$ so that $m_{gas} = m$.

IRP = 13 **Simple hysteresis**

The regular hysteresis option (*IRP* = *ICP* = 12) provides a rigorous representation of hysteretic relative permeability and capillary pressure curves. However, it can significantly slow down TOUGH3 simulations, because small time steps are often required at turning points, when a grid block switches between drainage and imbibition, because the slopes of the characteristic curves are discontinuous. Moreover, several control parameters are needed, which generally must be determined by trial and error, for the code to run smoothly. An alternative means of capturing the essence of hysteresis, while maintaining continuous slopes and requiring no additional control parameters, is the simple hysteresis algorithm of Patterson and Falta (2012), which is invoked with *IRP* = *ICP* = 13. Presently this option is only available when ECO2N is being used.

The Mualem (1976) relative permeability model is used for the non-wetting phase:

$$k_m = \sqrt{1 - \bar{S}_{wn}} (1 - \bar{S}_{wn}^{1/m})^{2m},$$

where

$$\bar{S}_{wn} = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}},$$

and S_{wr} and S_{nr} are residual saturations of the wetting and non-wetting phases, respectively. Hysteresis is implemented by considering S_{nr} to be a variable, which is calculated from the maximum historical non-wetting phase saturation in a grid block, S_{nmax} . The user has the option to specify S_{nr} as a linear function of the historical S_{nmax} :

$$S_{nr} = f_{snr} S_{nmax},$$

or S_{nr} can be calculated using a modified form of the Land (1968) relationship

$$S_{nr} = \frac{S_{nmax}}{1 + CS_{nmax}},$$

with

$$C = \frac{1}{S_{nrmax}} - \frac{1}{1 - S_{wr}},$$

where f_{snr} and $S_{nr \max}$, the maximum residual non-wetting phase saturation, are user-specified material properties. S_{nr} is calculated during every Newton-Raphson iteration. If S_n drops below S_{nr} by dissolution or compression, $S_{n\max}$ is recalculated as

$$S_{n\max} = \frac{S_n}{f_{snr}} \text{ or } S_{n\max} = \frac{S_n}{1 - CS_n}.$$

Wetting-phase relative permeability (non-hysteretic) is from van Genuchten (1980)

$$k_{rw} = \sqrt{\bar{S}_w} \left[1 - (1 - \bar{S}_w^{1/m})^m \right]^2,$$

where

$$\bar{S}_w = \frac{S_w - S_{wr}}{S_{ws} - S_{wr}},$$

- Parameters:
- $RP(1) = m$ to use in k_{rw}
 - $RP(2) = S_{wr}$
 - $RP(3) = S_{ws}$ (recommend 1)
 - $RP(4)$ If $<0 = -f_{snr}$ in linear trapping model
If $>0 = S_{nr \max}$ in Land trapping model
 - $RP(5) = m_{gas}$, m to use in k_m ; if zero or blank, use $RP(1)$
 - $RP(6) =$ power to use in first term in k_m (default $1/2$)
 - $RP(7)$ If $= 0$ Use $(1 - \bar{S}_{wn})$ in first term in k_m (Muallem, 1976)
If > 0 Use S_g in first term in k_m (Charbeneau, 2007), so that k_m does not go to 1 when immobile liquid phase is present

$IRP = 31$ **All phases perfectly mobile**

$$k_{rg} = 1$$

$$k_{rl} = 1$$

$$k_{rn} = 1$$

no parameters.

IRP = 32

Modified version of Stone's first three-phase method (Stone, 1970).

$$k_{rg} = \left[\frac{S_g - S_{gr}}{1 - S_{ar}} \right]^n$$

$$k_{rl} = \left[\frac{S_l - S_{lr}}{1 - S_{lr}} \right]^n$$

$$k_m = \left[\frac{1 - S_g - S_l - S_{nr}}{1 - S_g - S_{lr} - S_{nr}} \right] \left[\frac{1 - S_{lr} - S_{nr}}{1 - S_l - S_{nr}} \right] \left[\frac{(1 - S_g - S_{lr} - S_{nr})(1 - S_l)}{(1 - S_{nr})} \right]^n$$

When $S_n = 1 - S_l - S_g - S_s$ is near irreducible liquid saturation, $S_{nr} \leq S_n \leq S_{nr} + .005$, liquid relative permeability is taken to be

$$k'_m = k_m \cdot \frac{S_n - S_{nr}}{.005}$$

Parameters are $S_{lr} = RP(1)$, $S_{nr} = RP(2)$, $S_{gr} = RP(3)$, $n = RP(4)$.

IRP = 33

Three-phase functions of Parker et al. (1987).

$$m = 1 - 1/n$$

$$\bar{S}_g = S_g / (1 - S_m)$$

$$\bar{S}_l = (S_l - S_m) / (1 - S_m)$$

$$\bar{S}_n = (S_l + S_n - S_m) / (1 - S_m)$$

$$k_{rg} = \sqrt{\bar{S}_g} [1 - (\bar{S}_n)^{1/m}]^{2m}$$

$$k_{rl} = \sqrt{\bar{S}_l} \left\{ 1 - [1 - (\bar{S}_l)^{1/m}]^m \right\}^2$$

$$k_m = \sqrt{\bar{S}_n - \bar{S}_l} \left\{ [1 - (\bar{S}_l)^{1/m}]^m - [1 - (\bar{S}_n)^{1/m}]^m \right\}^2$$

where k_{rg} , k_{rl} , and k_{rn} are limited to values between 0 and 1, with $S_m = RP(1)$, and $n = RP(2)$.

$IRP = 34$ same as $IRP = 32$, except that

$$k_{rg} = 1 - \left[\frac{S_n + S_l - S_{lr}}{1 - S_{lr}} \right]^n$$

$IRP = 35$ **Power-law relative permeabilities for all phases $\beta = l, n, g$:**

$$k_{r\beta} = \left[\frac{S_\beta - S_{\beta r}}{1 - S_{\beta r}} \right]^n$$

Parameters are $S_{lr} = RP(1)$, $S_{nr} = RP(2)$, $S_{gr} = RP(3)$, $n = RP(4)$.

$IRP = 36$ **Functions used by Faust (1985) for two-phase Buckley-Leverett problem**

$$k_{rl} = (S_l - 0.16)^2 / 0.64$$

$$k_{rg} = 0$$

$$k_{rn} = (0.8 - S_l)^2 / 0.64$$

where k_{rl} and k_{rn} are limited to values between 0 and 1, no parameters.

$IRP = 37$ same as $IRP = 32$, except a correction factor is applied to k_{rn} such as to make k_{rn} equal to k_{rg} for two-phase conditions with the same aqueous phase saturation.

If the user wishes to employ other relative permeability relationships, these need to be programmed into subroutine RELP in module `Utility.f90`. The routine has the following structure:

```
SUBROUTINE RELP (SATU, RELPERM, NNPH, NMAT, USRX)
    ...
    RELP_FUNCTION: SELECT CASE (IRP (NMAT))
        CASE (1)
            CALL RELP_LINEAR (...)
        CASE (2)
            ...
        ...
    END SELECT RELP_FUNCTION

END
```

To code an additional relative permeability function, the user needs to insert a code segment analogous to that shown above, beginning with a CASE option which would be identical to *IRP* and calls a subroutine for the additional relative permeability function.

Appendix B. Capillary Pressure Functions

TOUGH3 provides several capillary pressure functions for two-phase or three-phase flow problems. The original TOUGH2 two-phase functions have been retained (from $ICP = 1$ to $ICP = 8$). The modified Brooks-Corey and van Genuchten models and two versions of the hysteresis model are implemented ($ICP = 10, 11, 12,$ and 13). The three-phase functions from TMVOC and ECO2M start at $IRP = 31$. If one of the two-phase functions is chosen, the gas-NAPL (for TMVOC) or gaseous CO_2 -liquid CO_2 (for ECO2M) capillary pressure will be assumed to be zero. The notation used below is: $P_{cgn} = P_n - P_g =$ gas-NAPL capillary pressure for TMVOC, which is equivalent to $P_{cgl} = P_l - P_g =$ gaseous CO_2 -liquid CO_2 capillary pressure for ECO2M; $P_{cgl} = P_l - P_g =$ gas-aqueous capillary pressure for TMVOC, which is equivalent to $P_{cga} = P_a - P_g =$ gaseous CO_2 -aqueous capillary pressure for ECO2M. It should be noted that P_{cni} is given by $P_{cni} = P_{cgl} - P_{cgn}$ for TMVOC, and similarly $P_{cni} = P_{cga} - P_{cgl}$ for ECO2M. $S_l, S_g,$ and S_n are the saturations of aqueous, gas (or gaseous CO_2) and NAPL (or liquid CO_2) phases, respectively.

$ICP = 1$ **Linear function**

$$P_{cap} = \begin{cases} -CP(1) & \text{for } S_l \leq CP(2) \\ 0 & \text{for } S_l \geq CP(3) \\ -CP(1) \frac{CP(3) - S_l}{CP(3) - CP(2)} & \text{for } CP(2) < S_l < CP(3) \end{cases}$$

Restriction: $CP(3) > CP(2)$.

If $CP(4) \neq 0$,

$$P_{cgn} = P_{cap} (= P_{cgl})$$

$$P_{cap}(P_{cgl}) = 0$$

$ICP = 2$ **Function of Pickens et al. (1979)**

$$P_{cap} = -P_0 \left\{ \ln \left[\frac{A}{B} \left(1 + \sqrt{1 - B^2/A^2} \right) \right] \right\}^{1/x}$$

with

$$A = (1 + S_l/S_{l0})(S_{l0} - S_{lr})/(S_{l0} + S_{lr})$$

$$B = 1 - S_l/S_{l0}$$

where

$$P_0 = CP(1) \quad S_{lr} = CP(2) \quad S_{l0} = CP(3) \quad x = CP(4)$$

Restrictions: $0 < CP(2) < 1 \leq CP(3)$; $CP(4) \neq 0$

ICP = 3 **TRUST capillary pressure** (Narasimhan et al., 1978)

$$P_{cap} = \begin{cases} -P_e - P_0 \left[\frac{1 - S_l}{S_l - S_{lr}} \right]^{1/\eta} & \text{for } S_l < 1 \\ 0 & \text{for } S_l = 1 \end{cases}$$

where

$$P_0 = CP(1) \quad S_{lr} = CP(2) \quad \eta = CP(3) \quad P_e = CP(4)$$

Restrictions: $CP(2) \geq 0$; $CP(3) \neq 0$

ICP = 4 **Milly's function** (Milly, 1982)

$$P_{cap} = -97.783 \times 10^A$$

with

$$A = 2.26 \left(\frac{0.371}{S_l - S_{lr}} - 1 \right)^{1/4}$$

where $S_{lr} = CP(1)$

Restriction: $CP(1) \geq 0$.

ICP = 6 **Leverett's function** (Leverett, 1941; Udell and Fitch, 1985)

$$P_{cap} = -P_0 \cdot \sigma(T) \cdot f(S_l)$$

with

$\sigma(T)$ - surface tension of water (supplied internally in TOUGH3)

$$f(S_l) = 1.417 (1 - S^*) - 2.120 (1 - S^*)^2 + 1.263 (1 - S^*)^3$$

where

$$S^* = (S_l - S_{lr}) / (1 - S_{lr})$$

Parameters: $P_0 = CP(1)$ $S_{lr} = CP(2)$

Restriction: $0 \leq CP(2) < 1$

If $CP(3) \neq 0$,

$$P_{cgn} = P_{cap} (= P_{cgl})$$

$$P_{cap}(P_{cgl}) = 0$$

$ICP = 7$

van Genuchten function (van Genuchten, 1980)

$$P_{cap} = -P_0 \left([S^*]^{-1/\lambda} - 1 \right)^{1-\lambda}$$

subject to the restriction

$$-P_{max} \leq P_{cap} \leq 0$$

Here,

$$S^* = (S_l - S_{lr}) / (S_{ls} - S_{lr})$$

Parameters: $CP(1) = \lambda = 1 - 1/n$

$CP(2) = S_{lr}$ (should be chosen smaller than the corresponding parameter in the relative permeability function; see note below.)

$CP(3) = 1/P_0 = \alpha/\rho_w g$ (proportional to \sqrt{k})

$CP(4) = P_{max}$

$CP(5) = S_{ls}$

Notation: Parameter λ is m in van Genuchten's notation, with $m = 1 - 1/n$; parameter n is often written as β .

Note on parameter choices: In van Genuchten's derivation (1980), the parameter S_{lr} for irreducible water saturation is the same in the relative permeability and capillary pressure functions. As a consequence, for $S_l \leq S_{lr}$ we have $k_{rl} \leq 0$ and $P_{cap} \leq -\infty$, which is unphysical because it implies that the radii of capillary menisci go to zero as liquid phase is becoming immobile (discontinuous). In reality, no special capillary pressure effects are expected when liquid phase becomes discontinuous. Accordingly, we recommend to always choose a smaller S_{lr} for the capillary pressure as compared to the relative permeability function.

$ICP = 8$ **No capillary pressure**

$$P_{cap} \equiv 0 \quad \text{for all saturations; no parameters}$$

$ICP = 10$ **Modified Brooks-Corey Model**

A modified version of the Brooks-Corey model (Brooks and Corey, 1964) 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 ($S_{lrc} + \varepsilon$), where ε is a small number. The slope of the linear extrapolation is identical with the slope of the capillary pressure curve at $S_l = S_{lrc} + \varepsilon$. 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.

$$P_c = \begin{cases} -p_e (S_{ec})^{-1/\lambda} & \text{for } S_l > (S_{lrc} + \varepsilon) \\ -p_e \left(\frac{\varepsilon}{1 - S_{lrc}} \right)^{-1/\lambda} + \frac{p_e}{\lambda} \frac{1}{1 - S_{lrc}} \left(\frac{\varepsilon}{1 - S_{lrc}} \right)^{\frac{1+\lambda}{\lambda}} (S_l - S_{lrc} - \varepsilon) & \text{for } S_l < (S_{lrc} + \varepsilon) \end{cases}$$

$$P_c \geq -p_{c,max}$$

where

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

Parameters: $CP(1)$: λ (pore size distribution index)

$CP(2)$: P_e (gas entry pressure [Pa])

if $CP(2)$ is negative and $USERX(1, N)$ is non-zero, apply

Leverett's rule: $P_e = -CP(2) \sqrt{USERX(1, N) / PER(NMAT)}$

if $USERX(2, N)$ is positive, $P_e = USERX(2, N)$

if $USERX(2, N)$ is negative, $P_e = -USERX(2, N) \cdot CP(2)$

$CP(3)$: ε or $P_{c,max}$

if $CP(3) = 0$, $p_{c,max} = 10^{50}$, $\varepsilon = -1$

if $0 < CP(3) < 1$, use linear model for $S_l < S_{lrc} + \varepsilon$

if $CP(3) \geq 1$, $p_{c,max} = CP(3)$, $\varepsilon = -1$

$CP(6)$: S_{lrc}

$ICP = 11$ **Modified van Genuchten Model**

The van Genuchten model (Luckner et al., 1989) has been modified to prevent the capillary pressure from decreasing towards negative infinity as the effective saturation approaches zero. The approach is identical to that in $ICP=10$, except that two extensions (linear and log-linear) are available. The modified van Genuchten model is invoked by setting both IRP and ICP to 11.

$$p_c = -\frac{1}{a} \frac{\hat{e}(S_{ec})^{(g-1)/m} - 1}{\hat{e}}^{1/n} \quad \text{for } S_l \geq (S_{lrc} + \varepsilon)$$

$$p_c = -\frac{1}{a} \frac{\hat{e} S_{ec}^{(g-1)/m} - 1}{\hat{e}}^{1/n} - b \times (S_l - S_{lrc} - \varepsilon) \quad \text{for } S_l < (S_{lrc} + \varepsilon)$$

with

$$\text{linear extension: } b = -\frac{(1-g)}{anm} \times \frac{1}{(1-S_{lrc})} \times \left(S_{ec}^{(g-1)/m} - 1 \right)^{\frac{1}{n}-1} S_{ec}^{\frac{2g-1-m}{m}} \frac{0}{0}$$

$$p_c = -\frac{1}{a} \left(S_{ec}^{(g-1)/m} - 1 \right)^{1/n} \times 10^{b(S_l - S_{lrc} - \varepsilon)} \quad \text{for } S_l < (S_{lrc} + \varepsilon)$$

with

$$\text{log-linear extension: } b = -\log_{10}(e) \times \frac{1-m}{m} \times \frac{g-1}{e} \times \frac{1}{S_{ec}^{(1-g)/m} - 1}$$

$$p_c \geq -P_{c,\max}$$

where

$$S_{ec} = \frac{S_l - S_{lrc}}{1 - S_{lrc}}, \quad S_{ec}^* = \frac{\varepsilon}{1 - S_{lrc}}$$

Parameters: $CP(1)$: n (parameter related to pore size distribution index, see also $CP(4)$)

$CP(2)$: $1/\alpha$ (parameter related to gas entry pressure [Pa])

$$USERX(4, N) > 0: 1/\alpha_i = USERX(4, N)$$

$$USERX(4, N) < 0: 1/\alpha_i = USERX(4, N) \cdot CP(2)$$

if $CP(2)$ is negative, apply Leverett scaling rule:

$$1/\alpha_i = 1/\alpha_{ref} \cdot \sqrt{k_i/k_{ref}}$$

where:

$$1/\alpha_{ref} = |CP(2)|$$

$$k_{ref} = PER(NMAT)$$

$$USERX(1, N) > 0: k_i = USERX(1, N)$$

$$USERX(1, N) < 0: k_i = USERX(1, N) \cdot PER(NMAT)$$

$CP(3)$: ε or $P_{c,\max}$

$$\text{if } CP(3) = 0, p_{c,\max} = 10^{50}, \varepsilon = -1$$

if $0 < CP(3) < 1$, $\varepsilon = CP(3)$ and use linear extension

$$\text{if } CP(3) \geq 1, p_{c,\max} = CP(3), \varepsilon = -1$$

if $-1 < CP(3) < 0$, $\varepsilon = |CP(3)|$ and use log-linear extension

$CP(4)$: m

if zero then $m = 1 - 1/CP(1)$, else $m = CP(4)$ and $n = 1/(1-m)$

$CP(5)$: T_{ref}

if negative, $|CP(5)|$ is reference temperature to account for temperature dependence of capillary pressure due to changes in surface tension

$CP(6): \gamma$
parameter of Active Fracture Model (see Appendix C)

$CP(7): S_{lrc}$
if zero, then $S_{lrc} = RP(1) = S_{lrk}$

$ICP = 12$ **Regular hysteresis**

The hysteretic form of the van Genuchten model (Parker and Lenhard, 1987; Lenhard and Parker, 1987) has been implemented. Details of the implementation are described in Doughty (2013). The hysteretic model is invoked by setting both IRP and ICP to 12.

$$P_c = -P_0^p \left[\left(\frac{S_l - S_{l\min}}{1 - S_{gr}^\Delta - S_{l\min}} \right)^{-\left(\frac{1}{m^p}\right)} - 1 \right]^{(1-m^p)}$$

where

$$S_{gr}^\Delta = \frac{1}{1/(1 - S_l^\Delta) + 1/S_{gr\max} - 1/(1 - S_{lr})}$$

$CP(1) = m^d$; van Genuchten m for drainage branch $P_c^d(S_l)$.

$CP(2) = S_{l\min}$; saturation at which original van Genuchten P_c goes to infinity.
Must have $S_{l\min} < S_{lr}$, where S_{lr} is the relative permeability parameter $RP(2)$.

$CP(3) = P_0^d$; capillary strength parameter for drainage branch $P_c^d(S_l)$ [Pa].

$CP(4) = P_{c \max}$; maximum capillary pressure [Pa] obtained using original van Genuchten P_c . Inverting the original van Genuchten function for $P_{c \max}$ determines S_m , the transition point between the original van Genuchten function and an extension that stays finite as S_l goes to zero. For functional form of extension, see description of $CP(13)$ below.

$CP(5)$ = scale factor for pressures for unit conversion (1 for pressure in Pa).

$CP(6) = m^w$; van Genuchten m for imbibition branch $P_c^w(S_l)$. Default value is $CP(1)$ (recommended unless compelling reason otherwise).

$CP(7) = P_0^w$; capillary strength parameter for imbibition branch $P_c^w(S_l)$ [Pa]. Default value is $CP(3)$ (recommended unless compelling reason otherwise).

$CP(8)$ = parameter indicating whether to invoke non-zero P_c extension for $S_l > S_l^*$

$$S_l^* = 1 - S_{gr}^{\Delta}$$

=0 no extension; $P_c = 0$ for $S_l > S_l^*$
>0 power-law extension for $S_l^* < S_l < 1$, with $P_c = 0$ when $S_l = 1$. A non-zero $CP(8)$ is the fraction of S_l^* at which the P_c curve departs from the original van Genuchten function. Recommended range of values: 0.97–0.99.

$CP(9)$ = flag indicating how to treat negative radicand, which can arise for $S_l > S_l^{\Delta 23}$ for second-order scanning drainage curves ($ICURV = 3$), where $S_l^{\Delta 23}$ is the turning-point saturation between first-order scanning imbibition ($ICURV = 2$) and second-order scanning drainage. None of the options below have proved to be robust under all circumstances. If difficulties arise because $S_l > S_l^{\Delta 23}$ for $ICURV = 3$, also consider using $IEHYS(3) > 0$ or $CP(10) < 0$, which should minimize the occurrence of $S_l > S_l^{\Delta 23}$ for $ICURV = 3$.

=0 radicand = $\max(0, \text{radicand})$ regardless of S_l value

=1 if $S_l > S_l^{\Delta 23}$, radicand takes value of radicand at $S_l = S_l^{\Delta 23}$

=2 if $S_l > S_l^{\Delta 23}$, use first-order scanning imbibition curve ($ICURV = 2$)

$CP(10)$ = threshold value of $|\Delta S|$ (absolute value of saturation change since previous time step) for enabling a branch switch (default is 1E-6; set to any negative number to do a branch switch no matter how small $|\Delta S|$ is; set to a value greater than 1 to never do a branch switch). See also $IEHYS(3)$.

$CP(11)$ = threshold value of S_{gr}^{Δ} . If value of S_{gr}^{Δ} calculated from S_l^{Δ} (Equation (2)) is less than $CP(11)$, use $S_{gr}^{\Delta} = 0$. Recommended value 0.01–0.03; default is 0.02.

$CP(12)$ = flag to turn off hysteresis for P_c (no effect on k_{rl} and k_{rg} ; to turn off hysteresis entirely, set $S_{grmax} = 0$ in $RP(3)$).

=0 hysteresis is on for P_c

=1 hysteresis is off for P_c (switch branches of P_c as usual, but set $S_{gr} = 0$ in P_c calculation. Make sure other parameters of P_c^d and P_c^w are the same: $CP(1) = CP(6)$ and $CP(3) = CP(7)$)

$CP(13)$ = parameter to determine functional form of P_c extension for $S_l < S_{lmin}$ (i.e., $P_c > P_{cmax}$)

=0 exponential extension

>0 power-law extension with zero slope at $S_l = 0$ and $P_c(0) = CP(13)$. Recommended value: 2 to 5 times $CP(4) = P_{cmax}$. Should not be less than or equal to $CP(4)$.

$ICP = 13$ **Simple hysteresis**

An approximate hysteretic formulation based on the simple hysteresis theory of Patterson and Falta (2012) has been implemented. The simple hysteresis model is invoked by setting both IRP and ICP to 13. Currently, this option is only available when ECO2N is being used.

The capillary pressure is the van Genuchten (1980) function

$$P_c = -P_0 (\bar{S}_{wn}^{-1/m} - 1)^{1-m},$$

where

$$\bar{S}_{wn} = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}},$$

and S_{wr} and S_{nr} are the residual saturations of the wetting phase and the non-wetting phase, respectively, and S_{nr} is a variable calculated as described in Appendix A for $IRP = 13$. If \bar{S}_{wn} is greater than or equal to one, then the capillary pressure is set to zero. For $S_w < S_{wr} + \varepsilon$, P_c is a linear extension that smoothly connects to the van Genuchten (1980) function and is capped by $P_{c \max}$.

Parameters:

CP (1)	m
CP (2)	S_{wr}
CP (3)	$1/P_0$ [Pa ⁻¹]
CP (4)	If $> 1 = P_{c \max}$ [Pa] ($\varepsilon = 1E-5$) If $< 1 = \varepsilon$ ($P_{c \max} = 1E50$ Pa) If $= 0$, $\varepsilon = 1E-5$, $P_{c \max} = 1E50$ Pa
CP (5)	S_{ls} (recommend 1)
CP (6)	0 unless Active Fracture Model is invoked (untested)
CP (7)	If $< 0 = -f_{snr}$ in linear trapping model If $> 0 = S_{nr \max}$ in Land trapping model

ICP=31

Three-phase capillary functions from Parker et al. (1987).

$$m = 1 - 1/n$$

$$\bar{S}_l = (S_l - S_m) / (1 - S_m)$$

$$\bar{S}_n = (S_l + S_n - S_m) / (1 - S_m)$$

$$P_{cgn} = -\frac{\rho_l g}{\alpha_{gn}} [(\bar{S}_n)^{-1/m} - 1]^{1/n}$$

$$P_{cgl} = -\frac{\rho_l g}{\alpha_{nl}} [(\bar{S}_l)^{-1/m} - 1]^{1/n} - \frac{\rho_l g}{\alpha_{gn}} [(\bar{S}_l)^{-1/m} - 1]^{1/n}$$

where $S_m = CP (1)$; $n = CP (2)$; $\alpha_{gn} = CP (3)$; $\alpha_{nl} = CP (4)$.

These functions have been modified so that the capillary pressures remain finite at low aqueous saturations. This is done by calculating the slope of the capillary pressure functions at \bar{S}_l and $\bar{S}_n = 0.1$. If \bar{S}_l or \bar{S}_n is less than 0.1, the capillary pressures are calculated as linear functions in this region with slopes equal to those calculated at scaled saturations of 0.1.

$ICP = 32$ as $ICP = 31$, except that the strength coefficients are directly provided as inputs, rather than being calculated from the parameters α_{gn} and α_{nl} . The capillary pressure functions are then

$$P_{cgn} = -P_{cgn,0} [(\bar{S}_n)^{-1/m} - 1]^{1/n}$$

$$P_{cgl} = -P_{cgl,0} [(\bar{S}_l)^{-1/m} - 1]^{1/n} - P_{cgn,0} [(\bar{S}_n)^{-1/m} - 1]^{1/n}$$

where $S_m = CP(1)$; $n = CP(2)$; $P_{cgn,0} = CP(3)$; $P_{cgl,0} = CP(4)$.

$ICP = 33$ as $ICP = 31$, except that the capillary pressures are modified for small gas saturations to reduce the derivative.

If $S_l + S_n > 0.99$,

$$P_{cgn} = P_{cgn} \frac{1 - S_l - S_n}{0.01}$$

If $S_l > 0.99$,

$$P_{cgl} = P_{cgl} \frac{1 - S_l}{0.01}$$

$ICP = 34$ as $ICP = 31$, except that the capillary pressures are smoothed out for small gas saturations.

If $S_l + S_n > 0.99$,

$$P_{cgn} = P_{cgn} [-10^6 (S_l + S_n - 0.99)^3 + 1]$$

If $S_l > 0.99$,

$$P_{cni} = P_{cni} [-10^6 (S_l - 0.99)^3 + 1]$$

Additional capillary pressure functions can be programmed into subroutine PCAP in a fashion completely analogous to that for relative permeabilities (see Appendix A).

Appendix C. Active Fracture Model

C.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:

$$f_a = S_e^\gamma$$

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

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

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:

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

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

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

and

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

The fracture-matrix interface area reduction factor is given by

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

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).

C.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 **CONN** 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

<i>ISOT</i>	<i>MOP</i> (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 Appendix C.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.

Appendix D. Input Format or Option Changes in TOUGH3

TOUGH3 is developed to be backward compatible with TOUGH2. However, the process of consolidating serial and parallel implementation and including stand-alone EOS modules inevitably introduces several exceptions. Here we summarize input format or option changes in TOUGH3.

- A consistent set of two- and three-phase relative permeability and capillary pressure functions is implemented across all EOS modules (see Appendices A and B). As a result, *IRP* and *ICP* options for the three-phase functions in TMVOC and ECO2M are different from those in the stand-alone versions. If *IRP* and *ICP* options in the stand-alone versions are used in the input file, they are automatically converted during the initialization phase.
- If two binary files MESH A and MESH B created from the disk file MESH during the previous simulation exist in the working folder, the code will ignore the MESH file *and the **ELEME** and **CONNE** blocks in the input file*, and read geometric information directly from these two files, which will reduce the memory requirement for the master processor and enhance I/O efficiency. If the mesh is changed, MESH A and MESH B *must* be deleted from the working folder to make the changes take effect.
- For the parallel solvers, only *MATSLV* in **SOLVR** is used and the other options should be specified through the PETSc (*.petscsrc*) and Aztec (*.aztecrc*) option files since it is difficult to encode the various options into the structure of the **SOLVR** block. In addition, the use of a separate option file ensures easy access to PETSc's latest linear solvers and its diagnostic tools.
- For **FOFT**, **COFT**, and **GOFT**, a flag to control the amount of printout is available. For **FOFT**, printout variables can also be controlled according to the input data in block **OUTPU** if *MOP2 (17) > 0*. The name of the sink/source is always required in **GOFT** even when a single sink/source is assigned to an element.
- *PMX* in **ELEME** can be active *without* a dummy domain 'SEED' in the **ROCKS** block. If a dummy domain 'SEED' is not specified in data block **ROCKS**, it can be used to specify grid block permeabilities or permeability modifiers. If a positive value less than 1e-4 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. Alternatively, the same information can be provided through *USRX* (columns 31–40) in block **INCON**. If *PMX* is blank for the first element, the element-by-element permeabilities are ignored. If a dummy domain 'SEED'

is not specified in data block **ROCKS**, strength of capillary pressure will not be automatically scaled. Leverett scaling of capillary pressure can be applied with $MOP2(6) > 0$ in data block **MOMOP**.

- A material name must be specified for every element. If a dummy element with no volume *and* no material name is used just to separate boundary elements, the code will generate an error message and stop the simulation.
- *ELST* in **PARAM** is no longer supported. For printout after each time step, use **FOFT** instead.