

# The TOUGH Codes - A Family of Simulation Tools for Multiphase Flow and Transport Processes in Permeable Media

*Karsten Pruess*

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

K\_Pruess@lbl.gov

## Abstract

Numerical simulation has become a widely practiced and accepted technique for studying flow and transport processes in the vadose zone and other subsurface flow systems. This article discusses a suite of codes developed primarily at Lawrence Berkeley National Laboratory with a capability to model multiphase flows with phase change. We summarize history and goals in the development of the TOUGH codes, and present the governing equations for multiphase, multicomponent flow. Special emphasis is given to space discretization by means of integral finite differences (IFD). Issues of code implementation and architecture are addressed, as are code applications, maintenance and future developments.

## 1. Introduction

TOUGH2 is a numerical simulation program for non-isothermal flows of multiphase, multi-component fluids in permeable (porous or fractured) media (Pruess, 1991a; Pruess et al., 1999). Developed primarily for applications to geothermal reservoir engineering, nuclear waste disposal, environmental remediation problems, and vadose zone hydrology, TOUGH2 and related codes (Table 1) are currently in use in approximately 300 installations in over 30 countries. This article gives a brief summary of the scope, design, and methods used in TOUGH2. It serves as an introduction to this Special Issue of *Vadose Zone Journal*, which features expanded versions of selected papers focusing on the vadose zone that were originally presented at the TOUGH Symposium 2003 (Finsterle et al., 2003; <http://esd.lbl.gov/TOUGHsymposium/>).

We begin with some brief remarks about the history of TOUGH2 and then present the physical and mathematical model implemented in the code. Special emphasis is given to the "integral finite difference" (IFD) method used for space discretization and to general issues of code architecture and maintenance. The paper concludes with some comments on current and future code developments.

Table 1. Development of the TOUGH codes.

Simulator	Application	Phases/ <i>Components</i>	Comments
MULKOM	geothermal, nuclear waste, oil and gas	multi/ <i>multi</i>	research code; operational 1981; no public release
TOUGH	geothermal, nuclear waste	aqueous, gas <i>/water, air</i>	released 1987
TOUGH2	general purpose	aqueous, gas/ <i>water, non-condensable gas</i>	released 1991
T2VOC	environmental	aqueous, gas, NAPL <i>/water, air, VOC</i>	released 1995
iTOUGH2	inverse modeling; sensitivity analysis, uncertainty propagation	multi/ <i>multi</i>	released 1999
TOUGH2 V 2.0	general purpose	multi/ <i>multi</i>	released 1999
TMVOC	environmental	aqueous, gas, NAPL <i>/water, air, multiple VOCs and NCGs</i>	released 2002
TOUGHREACT	reactive chemistry	aqueous, gas, solid <i>/multi</i>	release 2003 (expected)
TOUGH-FLAC	geomechanics	aqueous, gas <i>/water, CO2</i>	research code

## 2. Historical Remarks

The ancestor to the current TOUGH codes is a simulation program known as MULKOM, which was developed at Lawrence Berkeley National Laboratory (LBNL) in the early 1980s (see Table 1). The architecture and methodology of MULKOM was based on the recognition that the governing equations for non-isothermal flows of multicomponent, multiphase fluids have the same mathematical form, regardless of the nature and number of fluid components and phases. MULKOM was a research code that served as a test bed for developing much of the approaches and methodology that were subsequently implemented in TOUGH and TOUGH2. A stripped-down version of MULKOM for two-phase flow of water-air mixtures was released to the public in 1987 under the name TOUGH (Pruess, 1987). The acronym "TOUGH" stands for "transport of unsaturated groundwater and heat," and is also an allusion to the tuff formations at Yucca

Mountain, which represented one of the chief application areas of the code at the time. A more comprehensive subset of MULKOM modules was later released under the name TOUGH2 (Pruess, 1991a) through the Energy Science and Technology Software Center (ESTSC; <http://www.osti.gov/estsc/>) of the U.S. Department of Energy, and was more recently updated to TOUGH2 V 2.0 (Pruess et al., 1999). T2VOC (Falta et al., 1995) and TMVOC (Pruess and Battistelli, 2002) represent offshoots of TOUGH2 that are focused on environmental contamination problems with non-aqueous phase liquids (NAPLs). Other related codes include iTOUGH2, which provides capabilities for inverse modeling, optimization and sensitivity and uncertainty analysis (Finsterle, 1999), TOUGHREACT, which couples TOUGH2 with a general chemical speciation and reaction progress package (Xu and Pruess, 2001), and TOUGH-FLAC, which couples TOUGH2 with a commercial rock mechanics code FLAC3D (Rutqvist and Tsang, 2002).

### 3. Governing Equations

The basic mass conservation equations governing the flow of multiphase, multicomponent fluids in permeable media can be written in the following 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  $\Gamma_n$ . The quantity  $M$  appearing in the accumulation term (left hand side) represents mass per volume, with  $\kappa = 1, \dots, NK$  labeling the components (water, air,  $\text{CO}_2$ , other non-condensable gases, solutes, volatile organic chemicals, ...).  $\mathbf{F}$  denotes mass flux (see below), and  $q$  denotes sinks and sources.  $\mathbf{n}$  is a normal vector on surface element  $d\Gamma_n$ , pointing inward into  $V_n$ . Eq. (1) expresses the fact that the rate of change of fluid mass in  $V_n$  is equal to the net inflow across the surface of  $V_n$ , plus net gain from fluid sources.

The general form of the accumulation term is

$$M^\kappa = \phi \sum_{\beta} S_{\beta} \rho_{\beta} X_{\beta}^{\kappa} \quad (2)$$

In Eq. (2), the total mass of component  $\kappa$  is obtained by summing over the fluid phases  $\beta$  (= liquid, gas, NAPL).  $\phi$  is porosity,  $S_{\beta}$  is the saturation of phase  $\beta$  (i.e., the fraction of pore volume occupied by phase  $\beta$ ),  $\rho_{\beta}$  is the density of phase  $\beta$ , and  $X_{\beta}^{\kappa}$  is the mass fraction of component  $\kappa$  present in phase  $\beta$ . TOUGH2 V 2.0 allows for a more general form of the mass accumulation term that

includes equilibrium sorption on the solid grains (Pruess et al., 1999). The EWASG module for water-air-salt (NaCl) includes solid salt as an active phase (Battistelli et al., 1997).

Advective mass flux is a sum over phases,

$$\mathbf{F}^{\kappa} \Big|_{\text{adv}} = \sum_{\beta} X_{\beta}^{\kappa} \mathbf{F}_{\beta} \quad (3)$$

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

$$\mathbf{F}_{\beta} = \rho_{\beta} \mathbf{u}_{\beta} = -k \frac{k_{r\beta} \rho_{\beta}}{\mu_{\beta}} (\nabla P_{\beta} - \rho_{\beta} \mathbf{g}) \quad (4)$$

Here  $\mathbf{u}_{\beta}$  is the Darcy velocity (volume flux) in phase  $\beta$ ,  $k$  is absolute permeability,  $k_{r\beta}$  is relative permeability to phase  $\beta$ ,  $\mu_{\beta}$  is viscosity, and

$$P_{\beta} = P + P_{c\beta} \quad (5)$$

is the fluid pressure in phase  $\beta$ , which is the sum of the pressure  $P$  of a reference phase (usually taken to be the gas phase), and the capillary pressure  $P_{c\beta}$  ( $\leq 0$ ).  $\mathbf{g}$  is the vector of gravitational acceleration. TOUGH2 V 2.0 also considers diffusive fluxes in all phases, and includes coupling between diffusion and phase partitioning that can be very important for volatile solutes in multiphase conditions (Pruess, 2002). Special TOUGH2 versions that include a conventional Fickian model for hydrodynamic dispersion have also been developed (Oldenburg and Pruess, 1993, 1995; Wu and Pruess, 2000).

Multiphase flows often involve strong heat transfer effects, requiring consideration of an energy balance along with mass balances. Equations similar to the foregoing can be formulated for energy conservation and are given in the TOUGH2 user's guide (Pruess et al., 1999).

By applying Gauss' divergence theorem, Eq. (1) can be converted into the following partial differential equation (PDE)

$$\frac{\partial M^{\kappa}}{\partial t} = -\text{div} \mathbf{F}^{\kappa} + q^{\kappa} \quad (6)$$

which is the form commonly used as the starting point for deriving finite difference or finite element discretization approaches. Of special interest is a simplified version of Eq. (6) for an approximate description of water seepage in the unsaturated zone. Using Eqs. (2) and (4) in (6), neglecting phase change effects, and assuming that the gas phase acts as a “passive bystander” with negligible gas pressure gradients, the following equation for liquid phase flow is obtained

$$\frac{\partial}{\partial t} \phi S_1 \rho_1 = \text{div} \left[ k \frac{k_{rl}}{\mu_1} \rho_1 \nabla (P_1 + \rho_1 g z) \right] \quad (7)$$

Neglecting variations in liquid phase density and viscosity, as is appropriate for (nearly) isothermal conditions, Eq. (7) simplifies to Richards’ equation (1931)

$$\frac{\partial}{\partial t} \theta = \text{div} [K \nabla h] \quad (8)$$

where  $\theta = \phi S_1$  is specific volumetric moisture content,  $K = k k_{rl} \rho_1 g / \mu_1$  is effective hydraulic conductivity, and  $h = z + P_1 / \rho_1 g$  is the hydraulic head. Eq. (8) is far easier to solve numerically than a full multiphase treatment. Because of its simplicity and importance for modeling saturated-unsaturated flows, a special fluid property module has been developed for TOUGH2 that implements Eq. (8).

#### 4. Setting up and Solving Discretized Equations

In order to discretize the continuous space variables, we introduce volume and area averages, as follows.

$$\int_{V_n} M \, dV = V_n M_n \quad (9)$$

$$\int_{\Gamma_n} \mathbf{F}^k \cdot \mathbf{n} \, d\Gamma = \sum_m \int_{A_{nm}} \mathbf{F}^k \cdot \mathbf{n} \, d\Gamma = \sum_m A_{nm} F_{nm} \quad (10)$$

Here we have written the surface integral as a sum of integrals over surface segments  $A_{nm}$  between volume elements (or grid blocks)  $V_n$  and  $V_m$ , with  $F_{nm}$  the average flux over a surface segment. The discretized flux corresponding to the basic Darcy flux term, Eq. (4), is expressed in terms of parameters for volume elements  $V_n$  and  $V_m$  as follows,

$$F_{\beta, nm} = -k_{nm} \left[ \frac{k_{r\beta} \rho_{\beta}}{\mu_{\beta}} \right]_{nm} \left[ \frac{P_{\beta, n} - P_{\beta, m}}{D_{nm}} - \rho_{\beta, nm} g_{nm} \right] \quad (11)$$

where the subscripts (nm) denote a suitable averaging at the interface between grid blocks n and m (interpolation, harmonic weighting, upstream weighting).  $D_{nm} = D_n + D_m$  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. The basic geometric parameters used in space discretization are illustrated in Fig. 1.

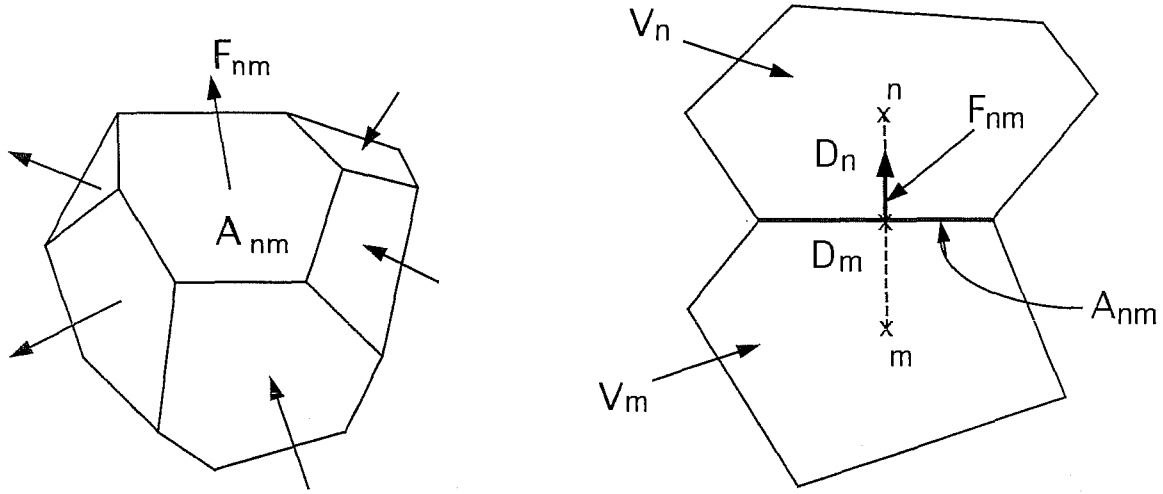


Figure 1. Space discretization and geometry data in the integral finite difference method.

Substituting Eqs. (9, 10) into (1), we obtain a set of first-order ordinary differential equations in time.

$$\frac{dM_n^K}{dt} = \frac{1}{V_n} \sum_m A_{nm} F_{nm}^K + q_n^K \quad (12)$$

Time is discretized as a first-order finite difference, and the flux and sink and source terms on the right-hand side of Eq. (12) are evaluated at the new time level,  $t^{k+1} = t^k + \Delta t$ , 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  $t^{k+1}$ , 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 non-linear, algebraic equations

$$\begin{aligned}
R_n^{\kappa,k+1} &= M_n^{\kappa,k+1} - M_n^{\kappa,k} - \frac{\Delta t}{V_n} \left\{ \sum_m A_{nm} F_{nm}^{\kappa,k+1} + V_n q_n^{\kappa,k+1} \right\} \\
&= 0
\end{aligned} \tag{13}$$

where we have introduced residuals  $R_n^{\kappa,k+1}$ . For each volume element (grid block)  $V_n$ , there are NEQ equations ( $\kappa = 1, 2, \dots, \text{NEQ}$ ), where usually  $\text{NEQ} = \text{NK} + 1$ , with NK the number of mass components, and an additional equation set up for energy balance. For a flow system with NEL grid blocks, Eqs. 13 represent a total of  $\text{NEL} \times \text{NEQ}$  coupled non-linear equations. The unknowns are the  $\text{NEL} \times \text{NEQ}$  independent primary variables  $\{x_i; i = 1, \dots, \text{NEL} \times \text{NEQ}\}$  which completely define the thermodynamic state of the flow system at time level  $t^{k+1}$ .

In order to apply Eqs. (13) to a specific flow system, it is necessary to express all parameters appearing in these equations as functions of the primary thermodynamic variables. This involves (1) PVT relations for fluid properties, and (2) constitutive relations for the interaction between fluids and the permeable medium. Different equation-of-state (EOS) formulations have been developed for different fluid mixtures. Table 2 lists the fluid property modules that are presently included in TOUGH2 V 2.0. Additional modules have been developed and may be made available to the public in the future. The most important fluid in the subsurface is of course water, and its properties are calculated in the TOUGH codes within experimental accuracy from the steam table equations as given by the International Formulation Committee (IFC, 1967). A library of user-selectable relative permeability and capillary pressure formulations is available in the code.

The nature of the primary thermodynamic variables depends on the phase state of the fluid system. For single-phase, single-component water, natural primary variables are pressure P and temperature T. In two-phase conditions, however, P and T are no longer independent, because P is equal to the saturated vapor pressure,  $P = P_{\text{sat}}(T)$ . Two-phase water-steam mixtures may be described with (P, S) as primary variables, where S is vapor saturation. In the TOUGH codes we employ "variable switching" to treat phase transitions. For a single-component water system, if a grid block is in single-phase liquid conditions, we monitor fluid pressure P and compare it with the saturated vapor pressure. If  $P \geq P_{\text{sat}}$  single-phase liquid conditions are maintained, whereas when  $P < P_{\text{sat}}$  we make a transition to two-phase conditions, switching primary variables to (P, S), and initializing S with a small non-zero number,  $S = 10^{-6}$ . In a two-component water-air system as in the unsaturated zone, another primary variable is needed in addition to (P, T). In single-phase liquid conditions the additional primary variable may be chosen as air mass fraction X. To test for the possibility that a gas phase may evolve we compute the "bubble pressure"  $P_{\text{bub}} = P_{\text{sat}} + P_{\text{air}}$  and

Table 2. Fluid property modules in TOUGH2, Version 2.0.

MODULE	PURPOSE
EOS1	basic fluid property module for water (liquid, vapor, two-phase), including "two waters" for tracing fluid movement
EOS2	water-CO <sub>2</sub> mixtures; originally developed by O'Sullivan et al. (1985)
EOS3	water-air mixtures; an adaptation of the EOS module of the TOUGH simulator (Pruess, 1987)
EOS4	water-air mixtures, including vapor pressure lowering according to Kelvin's equation (Edlefsen and Anderson, 1943)
EOS5	water-hydrogen mixtures
EOS7	mixtures of water-brine-air
EOS7R	water-brine-air, plus two volatile and water-soluble radionuclides
EOS8	fluid property module for three-phase flow of water, non-condensable gas, and black oil
EOS9	fluid property module for saturated/unsaturated flow according to Richards' equation (gas phase a passive bystander)
EWASG	fluid property module for three-component two-phase mixtures of water, water-soluble salt, and non-condensable gas; includes salt dissolution and precipitation, and associated porosity and permeability change
T2VOC	an adaptation of the T2VOC code (Falta et al., 1995) to the TOUGH2 Version 2.0 environment
TMVOC	fluid property module for three-phase flow of water, non-condensable gas, and non-aqueous phase liquid (NAPL), where the NAPL may consist of a multicomponent mixture of volatile organic chemicals

compare it with total fluid pressure  $P$ . When  $P_{\text{bub}} > P$  a gas phase will evolve, and primary variable  $X$  is replaced by gas saturation  $S$ . In flow systems with spatially variable phase composition, we will in general have different kinds of primary variables in different grid blocks. Our experience has shown variable switching to be a very robust and efficient technique for treating phase appearance and disappearance.

Eqs. (13) are solved by Newton/Raphson iteration, which is implemented as follows. We introduce an iteration index  $p$  and expand the residuals  $R_n^{k,k+1}$  in Eq. (13) at iteration step  $p + 1$  in a Taylor series in terms of those at index  $p$ .



$$R_n^{\kappa, k+1}(x_{i, p+1}) = R_n^{\kappa, k+1}(x_{i, p}) + \sum_i \left. \frac{\partial R_n^{\kappa, k+1}}{\partial x_i} \right|_p (x_{i, p+1} - x_{i, p}) + \dots = 0 \quad (14)$$

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

$$- \sum_i \left. \frac{\partial R_n^{\kappa, k+1}}{\partial x_i} \right|_p (x_{i, p+1} - x_{i, p}) = R_n^{\kappa, k+1}(x_{i, p}) \quad (15)$$

All terms  $\partial R_n / \partial x_i$  in the Jacobian matrix are evaluated by numerical differentiation, to achieve maximum flexibility in the manner in which various terms in the governing equations may depend on the primary thermodynamic variables. Eq. (15) is solved by sparse direct matrix methods (Duff, 1977) or iteratively by means of preconditioned conjugate gradients (Moridis and Pruess, 1998). Iteration is continued until all residuals are reduced below a preset convergence tolerance,  $|R_n^{\kappa, k+1} / M_n^{\kappa, k+1}| \leq \epsilon$ , typically chosen as  $\epsilon = 10^{-5}$ .

## 5. Integral Finite Differences

The approach used in the TOUGH codes of directly discretizing the integrals, without going through partial differential equations (PDEs), is termed "integral finite differences" (IFD). The practical application of this technique to problems of fluid flow and heat transfer was pioneered by Edwards (1972) and had been referred to as "integrated finite differences" in the earlier literature (Narasimhan and Witherspoon, 1976). Use of the term "integrated" implies that one starts from PDEs which are integrated prior to discretization. However, from a physics viewpoint we consider the integral statement Eq. (1) to be the more fundamental mass conservation equation, with Eq. (6) a derived form, so that "integral" rather than "integrated" finite differences seems a more appropriate descriptive term.

IFD encourages a "physical" view of model building, analogous to assembling a laboratory experiment. It provides a very simple conceptual basis for assigning boundary conditions. A flow system can be viewed as a network of boxes that exchange mass and energy. If temperature is to be held constant at a bounding surface, all that is required is defining a grid block on the "other" side of the surface, with a "small" nodal distance, giving it a very large heat capacity (e.g. by assigning a very large rock grain density, or a very large volume), and assigning it the desired boundary temperature. Such a grid block would be treated like any other block during a flow simulation. Heat

and mass fluxes entering or leaving that block will not be able to effect a noticeable change in temperature, due to the very large heat capacity. Boundary conditions of constant pressure (or moisture tension, or solute concentration) can be realized in analogous fashion. No-flow boundaries are modeled in IFD with simplicity itself, by simply not including any grid blocks beyond the no-flow boundary. More general and time-dependent boundary conditions can also be modeled with TOUGH2.

The IFD technique for space discretization has a number of advantages over conventional finite difference discretizations. All geometric quantities are defined locally, and the entire geometric information for the flow problem is provided through a list of volumes  $V_n$ , interface areas  $A_{nm}$ , nodal distances  $D_{nm}$ , and components  $g_{nm}$  of gravity acceleration in the direction of the line connecting the nodal points. There is no need to make reference to a global system of coordinates, and unstructured grids may be used. This provides great flexibility when dealing with spatially irregular features. 1-D, 2-D, and 3-D flow systems with regular or irregular grids can all be dealt with on the same footing. More advanced discretization approaches for fractured and highly heterogeneous media, such as double-porosity (Barenblatt et al., 1960), multiple interacting continua (MINC; Pruess and Narasimhan, 1985) and multi-region models (Gwo et al., 1996) can all be implemented simply by appropriate geometric preprocessing. Higher-order discretization methods, such as nine-point differencing (Yanosik and McCracken, 1979), can also be realized by means of appropriate geometric preprocessing, without any changes in the simulation code (Pruess, 1991b). This flexibility comes at no cost to the user, because for regular grids referred to a global coordinate system, IFD is mathematically equivalent to conventional finite differences (Moridis and Pruess, 1992).

As introduced in the previous section, the IFD technique for space discretization is very general but also entirely formal, in the sense that it does not provide specific guidance as to how discrete subdomains (grid blocks) should be defined. The fundamental IFD Eqs. (12, 13) are always valid, regardless of space discretizations; however, these equations will be practically useful only for such space discretizations for which an approximate evaluation of interface fluxes can be made from volume averages in the grid blocks involved (Eq. 11). The IFD technique itself does not offer "intrinsic" means to estimate and minimize space discretization errors. For regular grid systems, such errors can be estimated from Taylor series expansions as used in conventional finite differences. In more general cases, physical arguments can be used to judge how well the pressure gradient driving flux across an interface can be approximated in terms of average pressures in the participating grid blocks. In isotropic media, it is the normal component of pressure gradient at an interface that drives flow between the two grid blocks. In order to be able to obtain this normal

component from information for just two grid blocks, it is necessary that the interface be perpendicular to the nodal line connecting grid block centers. An important type of discretization that meets this constraint is (generally unstructured) Voronoi grids, where interfaces are constructed as the perpendicular bisectors of nodal lines.

As currently implemented in the TOUGH codes, a "flow connection" is an ordered pair of two volume elements. This is very simple and convenient from a user's viewpoint, but it provides only a single component of pressure (or other) gradients, along the nodal line. There is nothing intrinsic to IFD that would preclude providing more geometric information locally, so that all components of gradients at the interface could be obtained. For example, the Fickian model for hydrodynamic dispersion is anisotropic, and requires the full concentration gradient vector at the interface, not just the normal component. For regular grid systems the full gradient can be calculated in a straightforward manner by using information from additional neighboring grid blocks (Oldenburg and Pruess, 1993, 1995). Similarly, higher-order techniques for more accurate representation of advective terms can be easily implemented in structured grids (Oldenburg and Pruess, 2000). Although ordered-pair connections allow the use of unstructured Voronoi grids, we believe that the IFD is far more powerful and flexible than has been realized in implementations to date. It should be possible to define "multiblock connections" that, in addition to geometry data of the two grid blocks between which flow is occurring, include data of additional neighboring grid blocks, to obtain an accurate estimation of the full pressure gradient (and other driving forces) at the interface. Such more general connections should be able to deal with irregular geometric features, such as fault offsets, interfaces that are not perpendicular to nodal lines, and locally refined grids.

## **6. Code Architecture**

As had been mentioned before, the governing equations for multiphase, multicomponent flow have the same form, regardless of the nature and number of fluid components and phases. This suggests to set up a modular architecture in which the main flow and transport module can interface with different fluid property (or EOS, equation-of-state) modules to be able to model different fluid mixtures (Fig. 2).

The architecture of TOUGH2 stands in stark contrast to current ideas of "object-oriented" programming (Cao, 2002). The code is built around two large arrays, the array X holding all primary thermodynamic variables of all grid blocks at the latest completed time step (or at initialization), and the array PAR of "secondary parameters," which provides all other quantities appearing in the governing Eqs. (13) that are functions of the primary variables. A TOUGH2 simulation can be viewed as a process of sequential operation and updating of arrays X and PAR.

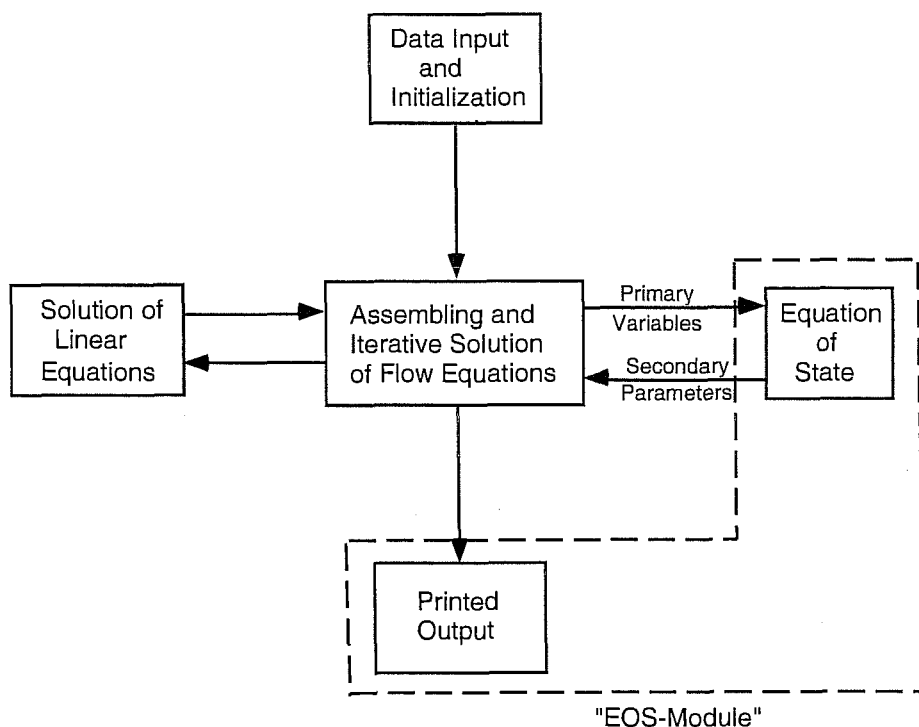


Figure 2. Modular architecture of TOUGH2.

Two ancillary arrays are in use, DX to hold latest (Newtonian iteration) increments to X, and an array DELX of "small" increments of X, which is used to calculate numerical derivatives of all terms in Eqs. (15) with respect to the primary variables. These arrays need to be available throughout many of the calculations performed in TOUGH2. They are placed in COMMON blocks, and the action of sub-programs is viewed as operating on these COMMON blocks.

The current version of TOUGH2 is written in FORTRAN 77, predating dynamic memory allocation. To achieve convenient scalability (flexibility in adjusting memory allocations to different problem sizes), the arrays X and PAR, as well as other problem-size dependent arrays, are each placed in a separate labeled COMMON block and are given actual dimension only in an easily modifiable PARAMETER statement in the main program, while they are dimensioned X(1) etc. in subroutines. The more recent TOUGH2 V 2.0 also uses INCLUDE files to facilitate flexible memory assignments.

An important design goal has been readability of the code. When reviewing a specific code section or subprogram, it should generally be transparent to the user what is being done, so that the code may be easily modified to suit particular application needs.

## 7. Applications

The basic mass-and energy-balance equations solved by TOUGH2 are very general and apply to a wide variety of subsurface flow systems, as well as to other natural or man-made permeable media. The original stimulus for the development of the TOUGH codes came from geothermal reservoir engineering, and geothermal applications remain an area of strong activity to this day (Pruess, 1985, 1998; Finsterle et al., 2003). TOUGH2 has also been extensively used for studies of nuclear waste isolation in geologic media (Bodvarsson and Tsang, 1999). Other areas where the TOUGH codes have been applied include environmental remediation problems, vadose zone hydrology, design and analysis of field and laboratory flow tests, oil and gas production and storage, mining engineering, chemical processing, and industrial drying of porous materials (Pruess, 1995, 1998; Finsterle et al., 2003).

Most applications of the TOUGH codes are currently being run on Unix workstations and on PCs. The spatial scale of flow systems simulated with TOUGH2 ranges from microscopic (Webb and Ho, 1998) to basin-scale. Time scales on which flow processes have been modeled range from a fraction of a second to geologic time. 3-D problems with a few thousand grid blocks are considered "modest" size with current computing platforms. For the nuclear waste storage investigations at Yucca Mountain, the LBNL group is routinely running 3-D problems with more than 100,000 grid blocks on PCs. A massively parallel version of TOUGH2 has also been developed and has been used for problems with over 2 million grid blocks (Wu et al., 2002; Zhang et al., 2003).

Special fluid property modules that are not currently part of the publicly distributed version of TOUGH2 include capabilities for supercritical temperatures (Brikowski, 2001), freezing water and hydrate formation (Moridis, 2002), multiple tracers and colloids (Moridis et al., 1999), and four-phase mixtures of water, liquid and gaseous CO<sub>2</sub>, and solid salt (Pruess, 2003). Modules for flow of foam and other non-Newtonian fluids have also been developed.

## 8. Concluding Remarks

Some of the basic features and methods of the TOUGH family of codes were designed 20 years ago and have stood the test of time. But there are also many recent developments that have greatly expanded the simulation capabilities. We can now treat a great variety of fluid mixtures, and we can model multiphase fluid flow and heat transfer processes more comprehensively, more accurately, and with greater efficiency. There are offshoots of TOUGH2 for reactive chemistry and for geomechanical effects. With the efficient solvers and preconditioners now available, 3-D

problems with tens of thousands of grid blocks can be run on PCs. Inverse modeling with iTOUGH2 has greatly enhanced the powers of the analyst.

Most of the development of the TOUGH codes was carried out at LBNL. Important contributions were made by authors from outside Berkeley. The development of the various TOUGH codes at LBNL was generally not carried out as software development projects, but rather was driven by a desire to obtain an understanding of different kinds of flow systems. Whenever simulation capabilities evolved to the point where they were believed to be sufficiently accurate, efficient, and robust to be useful to others, then we would clean up and document the codes, and release them to the public.

There is no finite process by which all bugs that may be present in a large simulation program can be identified and corrected. Consequently, a user should expect that there may be bugs in his/her application. It is the user's responsibility to check the numbers. Multiphase flow systems come in an unending variety of features. It is not practically possible to exhaustively cross-check the mutual compatibility and proper performance of all simulator options. Fixing a bug may cause unanticipated problems elsewhere. Continuing vigilance and application testing are needed. From a practical viewpoint, the best approach for identifying and correcting coding errors is through extensive applications to a wide range of diverse problems, by a broad community of users/analysts. This has been one of the motivating forces behind our drive for public release and wide distribution of our simulation codes. The TOUGH2 homepage on the Internet (<http://www-esd.lbl.gov/TOUGH2/>) includes a section with reported bugs in the Version 2.0 code and suggested fixes, testifying to the value of an interactive user community for making a "better code."

The LBNL group has always offered free technical support to users of the TOUGH codes, but being an R&D organization, the time and resources we can devote to such support are limited. Public release of the TOUGH codes and their use by others are workable in practice only if users' needs for technical support are generally modest. We achieve this by releasing codes only after they are thoroughly tested, debugged, and documented. This involves a period of beta-testing by users outside the development team, so that errors or flaws in coding and documentation can be identified and corrected. Care is taken to maintain a continuous chain of code testing and verification, where newer codes are run and rerun on test problems used in older versions (Pruess et al., 1996). Adequate internal as well as external documentation is crucial for efficient code maintenance. Worked sample problems are an important aspect of this. They aid in code installation and debugging by providing benchmarks, and serve as templates to help jump-start new applications.

## Acknowledgement

The development of the TOUGH codes was primarily supported by the Office of Wind and Geothermal Technologies, and by the Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract DE-AC03-76SF00098. The author is indebted and deeply grateful to the many individuals, within and without LBNL, who have made important contributions to the codes. Thanks are due to Curt Oldenburg and Stefan Finsterle for their careful review of the manuscript and the suggestion of improvements.

## References

- Barenblatt, G.E., I.P. Zheltov and I.N. Kochina. Basic Concepts in the Theory of Seepage of Homogeneous Liquids in Fissured Rocks, *J. Appl. Math*, 24 (5), 1286-1303, 1960.
- 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.
- Bodvarsson, G.S. and Y. Tsang (eds.). Special Issue: Yucca Mountain Project, *J. Contam. Hydr.*, Vol. 38, Nos. 1-3, pp. 1 - 426, 1999.
- Brikowski, T.H. Modeling Supercritical Systems with TOUGH2: Preliminary Results Using the EOS1SC Equation of State Module, in *Proceedings*, Twenty-Sixth Workshop on Geothermal Reservoir Engineering, pp. 208 - 215, Stanford University, Stanford, CA, January 2001.
- Cao, H. Development of Techniques for General Purpose Simulators, PhD Thesis, Stanford University, Stanford, CA, 2002.
- Duff, I. S. MA28—A Set of FORTRAN Subroutines for Sparse Unsymmetric Linear Equations, AERE Harwell Report R 8730, 1977.
- 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.
- Falta, R.W., K. Pruess, S. Finsterle, and A. Battistelli. T2VOC User's Guide, Lawrence Berkeley Laboratory Report LBL-36400, March 1995.
- Finsterle, S. iTOUGH2 User's Guide, Lawrence Berkeley National Laboratory Report LBNL-40040, Berkeley, CA, January 1999.
- Finsterle, S., G. Moridis, C. Oldenburg and Y.S. Wu (eds.) *Proceedings, TOUGH Symposium 2003*, Lawrence Berkeley National Laboratory Report LBNL-52494, Berkeley, CA, May 2003.
- Gwo, J.P., P.M. Jardine, G.V. Wilson and G.T. Yeh. Using a Multiregion Model to Study the Effects of Advective and Diffusive Mass Transfer on Local Physical Nonequilibrium and Solute Mobility in a Structured Soil, *Water Resour. Res.*, Vol. 32, No. 3, pp. 561 - 570, 1996.

- International Formulation Committee. A Formulation of the Thermodynamic Properties of Ordinary Water Substance, IFC Secretariat, Düsseldorf, Germany, 1967.
- Moridis, G.J. Numerical Studies of Gas Production from Methane Hydrates, presented at SPE Gas Technology Symposium, Calgary, Alberta, Canada, April 2002 (also: Lawrence Berkeley National Laboratory Report LBNL-49765).
- 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. T2SOLV: An Enhanced Package of Solvers for the TOUGH2 Family of Reservoir Simulation Codes, *Geothermics*, Vol. 27, No. 4, pp. 415 - 444, 1998.
- Moridis, G.J., Y.S. Wu and K. Pruess. EOS9nT: A TOUGH2 Module for the Simulation of Water Flow and Solute/Colloid Transport in the Subsurface, Lawrence Berkeley National Laboratory Report LBNL-42351, Berkeley, CA, March 1999.
- 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.
- Oldenburg, C.M. and K. Pruess. A Two-Dimensional Dispersion Module for the TOUGH2 Simulator, Lawrence Berkeley Laboratory Report LBL-32505, Berkeley, CA, September 1993.
- Oldenburg, C.M. and K. Pruess. Dispersive Transport Dynamics in a Strongly Coupled Groundwater-Brine Flow System. *Water Resour. Res.*, Vol. 31, No. 2, pp. 289-302, 1995.
- Oldenburg, C.M. and K. Pruess. Simulation of Propagating Fronts in Geothermal Reservoirs with the Implicit Leonard Total Variation Diminishing Scheme, *Geothermics*, Vol. 29, pp. 1 - 25, 2000.
- Peaceman, D.W. *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam, The Netherlands, 1977.
- Pruess, K. TOUGH User's Guide, Nuclear Regulatory Commission Report NUREG/CR-4645; also Lawrence Berkeley Laboratory Report No. LBL-20700, 1987.
- Pruess, K. TOUGH2 - A General Purpose Numerical Simulator for Multiphase Fluid and Heat Flow, Lawrence Berkeley Laboratory Report LBL-29400, Lawrence Berkeley Laboratory, Berkeley, CA, May 1991a.
- Pruess, K. Grid Orientation and Capillary Pressure Effects in the Simulation of Water Injection into Depleted Vapor Zones, *Geothermics*, Vol. 20, No. 5/6, pp. 257 - 277, 1991b.
- Pruess, K. (ed.) Proceedings of the TOUGH Workshop '95, Lawrence Berkeley Laboratory Report LBL-37200, Berkeley, CA, March 1995.
- Pruess, K. (ed.) Proceedings of the TOUGH Workshop '98, Lawrence Berkeley National Laboratory Report LBNL-41995, Berkeley, CA, May 1998.
- Pruess, K. Numerical Simulation of Multiphase Tracer Transport in Fractured Geothermal Reservoirs, *Geothermics*, Vol. 31, pp. 475 - 499, 2002.



- Pruess, K. Numerical Simulation of Leakage from a Geologic Disposal Reservoir for CO<sub>2</sub>, with Transitions between Super- and Sub-Critical Conditions, in *Proceedings, TOUGH Symposium 2003*, Berkeley, CA 94720, May 2003.
- Pruess, K. and T.N. Narasimhan. A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media, *Soc. Pet. Eng. J.*, 25 (1), pp. 14 - 26, February 1985.
- Pruess, K., A. Simmons, Y.S. Wu and G. Moridis. TOUGH2 Software Qualification. Lawrence Berkeley National Laboratory Report LBL-38383, Berkeley, CA, February 1996.
- Pruess, K., C. Oldenburg and G. Moridis. TOUGH2 User's Guide, Version 2.0, Lawrence Berkeley National Laboratory Report LBNL-43134, Berkeley, CA, November 1999.
- Pruess, K. and A. Battistelli. TMVOC, a Numerical Simulator for Three-Phase Non-Isothermal Flows of Multicomponent Hydrocarbon Mixtures in Saturated-Unsaturated Heterogeneous Media, Lawrence Berkeley National Laboratory Report LBNL-49375, Berkeley, CA, April 2002.
- Richards, L.A. Capillary Conduction of Liquids Through Porous Mediums, *Physics*, 1, 318-333, 1931.
- Rutqvist J. and C.-F. Tsang. A Study of Caprock Hydromechanical Changes Associated with CO<sub>2</sub> Injection into a Brine Formation, *Environmental Geology*, Vol. 42, pp. 296 - 305, 2002.
- Webb, S.W. and C.K. Ho. Pore-Scale Modeling Using TOUGH2, in K. Pruess (ed), *Proceedings of the TOUGH Workshop '98*, Lawrence Berkeley National Laboratory Report LBNL-41995, pp. 288 - 293, 1998.
- Wu, Y.S. and K. Pruess. Numerical Simulation of Non-Isothermal Multiphase Tracer Transport in Heterogeneous Fractured Porous Media, *Adv. Wat. Resour.*, Vol. 23, pp. 699 - 723, 2000.
- Wu, Y.S., K. Zhang, C. Ding, K. Pruess, E. Elmroth and G.S. Bodvarsson. An Efficient Parallel-Computing Method for Modeling Nonisothermal Multiphase Flow and Multicomponent Transport in Porous and Fractured Media, *Adv. Wat. Resour.*, Vol. 25, pp. 243 - 261, 2002.
- Yanosik, J.L. and T.A. McCracken. A Nine-Point, Finite Difference Reservoir Simulator for Realistic Prediction of Adverse Mobility Ratio Displacements, *Soc. Pet. Eng. J.*, 253-262, 1979.
- Xu, T. and K. Pruess. Modeling Multiphase Non-isothermal Fluid Flow and Reactive Geochemical Transport in Variably Saturated Fractured Rocks: 1. Methodology, *American Journal of Science*, Vol. 301, pp. 16-33, 2001.
- Zhang, K., Y.S. Wu, C. Ding and K. Pruess. TOUGH2\_MP: A Parallel Version of TOUGH2, in *Proceedings, TOUGH Symposium 2003*, Berkeley, CA 94720, May 2003.

