LBL-29400
UC-251

Lawrence Berkeley Laboratory

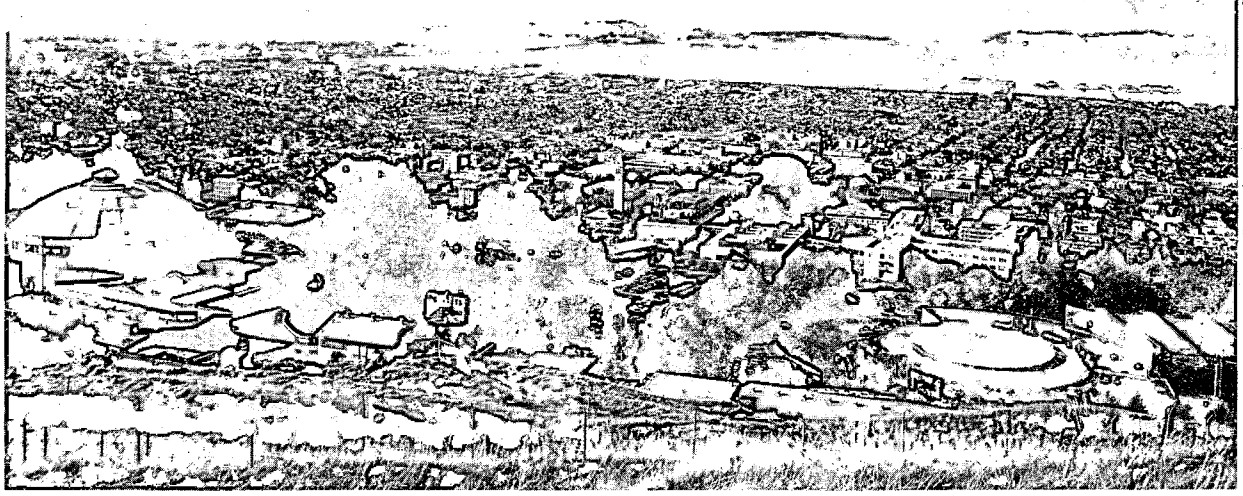
UNIVERSITY OF CALIFORNIA

EARTH SCIENCES DIVISION

TOUGH2—A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow

K. Pruess

May 1991



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

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.

LBL-29400

TOUGH2—A General-Purpose Numerical Simulator for Multiphase Fluid and Heat Flow

Karsten Pruess

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

May 1991

This work was supported in large part by the Assistant Secretary for Conservation and Renewable Energy, Office of Renewable Energy, Geothermal Technology Division, with additional support from the Director, Office of Energy Research, Office of Basic Energy Sciences, Engineering and Geosciences Division; the Yucca Mountain Project of Sandia National Laboratories; and the Manager, Repository Technology Program, Repository Technology and Transportation Division, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Abstract

TOUGH2 is a numerical simulation program for nonisothermal flows of multicomponent, multiphase fluids in porous and fractured media. The chief applications for which TOUGH2 is designed are in geothermal reservoir engineering, nuclear waste disposal, and unsaturated zone hydrology. A successor to the TOUGH program, TOUGH2 offers added capabilities and user features, including the flexibility to handle different fluid mixtures, facilities for processing of geometric data (computational grids), and an internal version control system to ensure referenceability of code applications.

This report includes a detailed description of governing equations, program architecture, and user features. Enhancements in data inputs relative to TOUGH are described, and a number of sample problems are given to illustrate code applications.

Table of Contents

List of Figures vii

List of Tables ix

1. Introduction 1

2. Methodology and Architecture of MULKOM and TOUGH2 5

   2.1 Scope and Methodology 5
   2.2 Program Architecture 6

3. TOUGH2 Array Structure and Handling 9

   3.1 Primary Thermodynamic Variables 9
   3.2 Thermophysical Property Arrays 11
   3.3 Linear Equation Setup 14
   3.4 Dimensioning of Major Arrays 15

4. Equation-of-State Modules 17

   4.1 EOS1 (Water, Water with Tracer) 18
   4.2 EOS2 (Water, CO_2) 20
   4.3 EOS3 (Water, Air) 21
   4.4 EOS4 (Water, Air) 23
   4.5 EOS5 (Water, Hydrogen) 25

5. User Features 27

   5.1 Specification of Flow Geometry 28
   5.2 Initial Conditions and Restarting 32
   5.3 Boundary Conditions and ''Inactive'' Elements 33
   5.4 Heat Exchange with Confining Beds 35
   5.5 Future Code Developments 37

6. Preparation of Input Data 41

   6.1 Enhancements in TOUGH Data Blocks 41
   6.2 New TOUGH2 Data Blocks 44
   6.3 Input Formats for MESHMAKER 46

7. Sample Problems 53

7.1 Problem No. 1—Code Demonstration and Comparison with TOUGH 53

7.2 Problem No. 2—Heat Pipe in Cylindrical Geometry 56

7.3 Problem No. 3—Heat Sweep in a Vertical Fracture 64

7.4 Problem No. 4—Five-spot Geothermal Production/Injection 71

8. Concluding Remarks 79

Acknowledgements 82

Nomenclature 83

References 87

Appendix A. Mass and Energy Balances 91

Appendix B. Space and Time Discretization 95

Appendix C. Description of Flow in Fractured Media 99

List of Figures

		page
Figure 1.	Modular architecture of MULKOM and TOUGH2.	2
Figure 2.	Structure of thermophysical property arrays in MULKOM and TOUGH2.	11
Figure 3.	Linear equation structure.	12
Figure 4.	User options for supplying geometry data.	29
Figure 5.	TOUGH input formats with TOUGH2 extensions.	42
Figure 6.	Input formats for new TOUGH2 data blocks.	43
Figure 7.	Input formats for MESHMAKER module.	47
Figure 8.	TOUGH2 input file for sample problem 1 – code demonstration.	54
Figure 9.	Selected output for problem 1.	55
Figure 10.	Input file for problem 2 – heat pipe in cylindrical geometry	57
Figure 11.	Profiles of temperature, pressure, liquid saturation, and air mass fraction for problem 2.	58
Figure 12.	MESHMAKER input for finer gridding in problem 2.	59
Figure 13.	Comparison of TOUGH2 results with similarity solution.	60
Figure 14.	Calling sequence of program units in problem 2.	61
Figure 15.	Selected output for problem 2, run with EOS4.	62–63
Figure 16.	Schematic diagram of injection-production system in vertical fracture. (I and I′ are injection points, P and P′ production points.)	64
Figure 17.	Input file for problem 3 – heat sweep in a vertical fracture.	66
Figure 18.	TOUGH2 printout of mesh pattern in problem 3.	67
Figure 19.	Produced fluid temperature versus time for vertical fracture problem.	69
Figure 20.	Five-spot well pattern with grid for modeling a 1/8 symmetry domain.	71
Figure 21.	Input file for problem 4 – five-spot production/injection. (Only part of ELEME and CONNE data blocks are shown.)	73
Figure 22.	Output from MINC processing of problem 4.	74
Figure 23.	Selected output from problem 4 flow simulation.	76
Figure 24.	Temperature profiles for problem 4 along a line from injection to production well after 36.5 yrs.	77

page

Figure 25. Weighting procedures for absolute (k) and relative permeability (k_r) at grid block interfaces. 80

Figure 26. Space discretization and geometry data in the integral finite difference method. 96

Figure 27. Idealized ''double-porosity'' model of a fractured porous medium. 99

Figure 28. Subgridding in the method of ''multiple interacting continua'' (MINC). 100

Figure 29. Flow connections in the ''dual permeability'' model. 102

List of Tables

		page
Table 1.	TOUGH2 Fluid Property Modules	1
Table 2.	Summary of Major Common Blocks	16
Table 3.	Summary of EOS1	20
Table 4.	Summary of EOS2	21
Table 5.	Summary of EOS3	23
Table 6.	Summary of EOS4	25
Table 7.	Thermophysical Properties of Hydrogen	26
Table 8.	TOUGH2 Disk Files	27
Table 9.	New Data Blocks in TOUGH2 (all optional)	41
Table 10.	Specifications for Generation Types	44
Table 11.	Parameters for Fracture Flow Problem	65
Table 12.	Parameters for Five-spot Problem	72

1. Introduction

TOUGH2 is a numerical simulation program for multi-dimensional coupled fluid and heat flows of multiphase multicomponent fluid mixtures in porous and fractured media. It belongs to the MULKOM family of codes (Pruess, 1983b, 1988) and is a more general version of the TOUGH simulator (Pruess, 1987) to which it is closely related in methodology, architecture, and input/output formats. TOUGH2 includes a number of fluid property modules (also referred to as "equation-of-state" or "EOS" modules), which make the code applicable to a variety of subsurface flow systems, including groundwater aquifers, unsaturated zones, and geothermal reservoirs (see Table 1).

Table 1. TOUGH2 Fluid Property Modules

Module	Capabilities
EOS1	water, water with tracer*
EOS2	water, CO_2
EOS3	water, air*†
EOS4	water, air, with vapor pressure lowering
EOS5	water, hydrogen*

*optional constant-temperature capability
†similar to the EOS-module of TOUGH

Additional fluid property modules that have been developed for MULKOM are being adapted for future inclusion in the TOUGH2 program package. Applications of the simulator are facilitated by a number of user features. These include flexible dimensioning of major arrays, capabilities for internal processing of flow geometry data (mesh generation), and enhanced facilities for specifying initial and boundary conditions. TOUGH2 input formats are upward compatible with those of TOUGH. Thus, TOUGH input files can be run with TOUGH2, so that existing TOUGH applications can be maintained. Note, however, that some default parameter settings are different in TOUGH2, so

that minor adjustments in TOUGH input files may be advisable.

TOUGH2 implements a flexible general-purpose architecture (see Fig. 1) for simulating fluid and heat flow in systems in which any number of components or species can be distributed among several coexisting phases. In this report we provide information on architecture and user features of TOUGH2, and we describe the various fluid property (EOS) modules included in the present TOUGH2 package. A key feature of the code architecture is an array structure that allows for flexible interfacing between the module that sets up and solves the fluid flow equations and the EOS modules, which represent fluid mixtures with different numbers of components and phases. A basic understanding of this structure and some familiarity with the source code is necessary for successful applications.

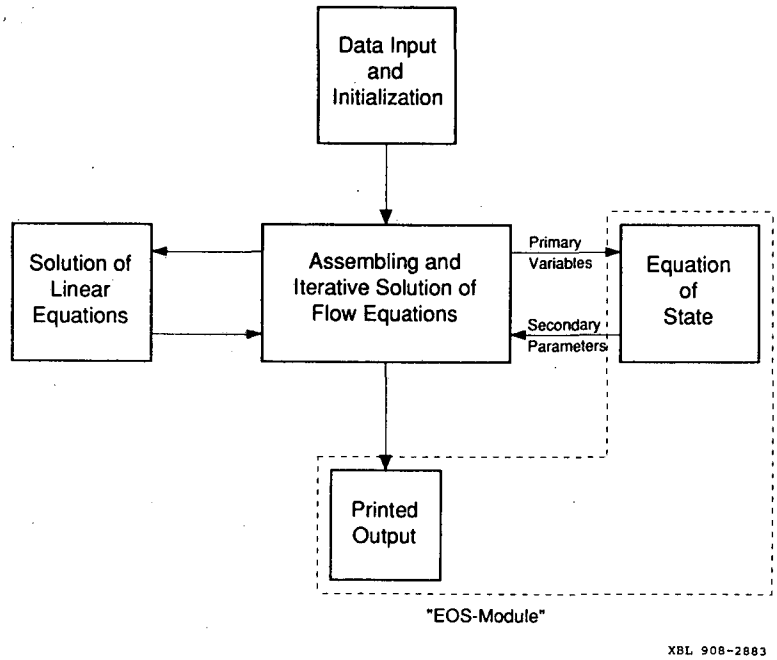


Figure 1. Modular architecture of MULKOM and TOUGH2.

The TOUGH2 program consists of a number of functional units with flexible and transparent interfaces. Much of what program units do is spelled out in internal

comments and in printed output. It is hoped that this "open" architecture will facilitate applications, and will encourage TOUGH2 users to further develop and enhance the code. At the same time TOUGH2 provides mechanisms, by means of a tight and visible "version control" system, for meeting stringent demands on reliability and referenceability of code applications. Each program unit, when first called during a TOUGH2 simulation run, writes a one-line message specifying its name, version number and date, and purpose. All version messages are optionally printed to OUTPUT at the end of a simulation run. (See the example in Fig. 14). Users who wish to modify the code can maintain a referenceable record of code changes and applications by appropriately updating the version messages.

The development of TOUGH2 was carried out on CDC-7600 and Cray X-MP computers. The coding complies with the ANSI X3.9-1978 (FORTRAN 77) standard.* 64-bit word length is required for successful execution. The present document provides essential information needed for TOUGH2 applications. It is not intended as a "stand alone" report, but should be used in conjunction with the TOUGH User's Guide (Pruess, 1987). The source code is being distributed together with several INPUT files for sample problems (see section 7). Besides providing a check on proper code installation, the sample problems illustrate code capabilities and serve as a brief tutorial for applications.

*For linking with default input and output files "INPUT" and "OUTPUT," the code has a "CALL LINK (...)" statement in the main (TOUGH2) program. This is peculiar to the Cray computer at the National Energy Research Supercomputer Center, Lawrence Livermore National Laboratory, and should be removed for installation at other computers.

that the mass- and energy-balance equations governing multiphase flow have the same mathematical structure, regardless of the number and nature of fluid components and phases present in a flow system. It is this modular architecture which gives MULKOM the flexibility to handle a wide variety of multicomponent multiphase flow systems. The nature and properties of specific fluid mixtures enter into the governing equations only through thermophysical parameters, such as fluid density, viscosity, enthalpy, etc. Different fluid mixtures can therefore be simulated with the same flow module, the thermophysical properties (or "PVT properties") of the specific fluid mixture of interest being provided by an appropriate "EOS" (equation-of-state) module.

Although the basic concepts used in the design of MULKOM are simple and straightforward, the code has never been easy to use because various research applications have led to a proliferation of specialized program modules and options. This situation led to the development and release of a specialized version of MULKOM for non-isothermal flow of water and air, named TOUGH (*). TOUGH is an acronym for "**t**ran-sport **o**f **u**nsaturated **g**roundwater and **h**eat," and is also an allusion to the tuff formations at Yucca Mountain, Nevada, which are presently being evaluated by the U.S. Department of Energy for their suitability as a host medium for a high-level nuclear waste repository.

The TOUGH User's Guide (Pruess, 1987) includes a technical description of the code and its architecture. It also provides complete documentation for preparing input files, and includes a set of sample problems which illustrate code applications. TOUGH can perform "conventional" (water only) geothermal reservoir simulations simply by setting air mass fraction equal to zero in the input file.

With TOUGH2 we are releasing a considerably more general subset of MULKOM-modules. TOUGH2 can interface with different EOS modules and thereby model different fluid mixtures that consist of a variable number of NK components distri-

(*) The TOUGH code and associated documentation is available from the National Energy Software Center, c/o Argonne National Laboratory, 9700 S. Cass Ave., Argonne, Ill. 60439.

buted among NPH phases. It also contains facilities for mesh generation and internal processing of geometric data. Except for this added flexibility the code is subroutine-for-subroutine actually very similar to TOUGH. The TOUGH2 input structure is upwardly compatible with that of TOUGH itself (see below); users should refer to the TOUGH User's Guide for the applicable input formats. A cautionary remark is in order here: even though TOUGH input decks will run with TOUGH2, they will not necessarily produce the exact same results even with the water-air EOS because a number of default parameter settings are different.

2. Methodology and Architecture of MULKOM and TOUGH2

Numerical simulators for nonisothermal multiphase flows have been under development at Lawrence Berkeley Laboratory for more than ten years. This work was focussed primarily on geothermal reservoir simulation. Additional incentives were provided by flow problems arising in the context of high-level nuclear waste isolation, oil and gas recovery and storage, and the protection of groundwater resources. The desire to model systems containing different fluid mixtures led to the development of a flexible general-purpose simulator MULKOM. TOUGH2 is essentially a subset of MULKOM, consisting of a selection of the better tested and documented MULKOM program modules.

2.1 Scope and Methodology

MULKOM and TOUGH2 solve mass and energy balance equations that describe fluid and heat flow in general multiphase multicomponent systems (Appendix A). Fluid flow is described with a multiphase extension of Darcy's law; in addition there is diffusive mass transport in the gas phase. Heat flow occurs by conduction and convection, the latter including sensible as well as latent heat effects. The description of thermodynamic conditions is usually based on the assumption of local equilibrium of all phases (liquid, gaseous, and solid). (MULKOM has also been used to model non-equilibrium conditions, such as chemical reactions proceeding with finite rates.) All fluid and formation parameters can be arbitrary non-linear functions of the primary thermodynamic variables.

For numerical simulation the continuous space and time variables must be discretized. In all members of the MULKOM family of codes, space discretization is made directly from the integral form of the basic conservation equations, without converting them into partial differential equations (Appendix B). This ''integral finite difference''

method (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, or three dimensions. The method also makes it possible, by means of simple preprocessing of geometric data, to implement double- and multiple-porosity methods for fractured media, as well as higher-order differencing methods (Pruess and Narasimhan, 1982, 1985; Pruess, 1983a; Pruess and Bodvarsson, 1983). For a system of regular grid blocks referred to a global coordinate system the integral finite difference method is completely equivalent to conventional finite differences. Time is discretized fully implicitly as a first-order backward finite difference. This together with 100% upstream weighting of flux terms at interfaces is necessary to avoid impractical time step limitations in flow problems involving phase (dis-) appearances, and to achieve unconditional stability (Peaceman, 1977).

The discretization results in a set of strongly coupled non-linear algebraic equations (Appendix B). These are solved completely simultaneously, using Newton-Raphson iteration. Time steps can be automatically adjusted (increased or reduced) during a simulation run, depending on the convergence of the iteration process. The linear equations arising at each iteration step are solved with the MA28 package from the Harwell program library, which implements a sparse version of LU-decomposition and backsubstitution (Duff, 1977).* The accuracy of MULKOM has been tested by comparison with many different analytical and numerical solutions, and with results from laboratory experiments (Pruess and Bodvarsson, 1984; Pruess and Narasimhan, 1985; Verma, 1986; Pruess, 1987; Pruess *et al.*, 1987; Lam *et al.*, 1988; Doughty and Pruess, 1990; Pruess, 1990b; Doughty and Pruess, 1991).

2.2 Program Architecture

MULKOM has a modular architecture (Fig. 1), which was built on the recognition

*Other solvers may be used.

3. TOUGH2 Array Structure and Handling

3.1 Primary Thermodynamic Variables

MULKOM has been used for modeling processes in which all (fluid and solid) phases are in local thermodynamic equilibrium, as well as for processes involving non-equilibrium conditions with kinetic rate expressions. The version released as TOUGH2 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' phase rule it follows that the number of thermodynamic degrees of freedom in such a system is

$$f = NK + 2 - NPH \tag{1}$$

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

$$\sum_{\beta=1}^{NPH} S_\beta = 1 \tag{2}$$

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

$$\begin{aligned} NK1 &= f + NPH - 1 \\ &= NK + 1 \end{aligned} \tag{3}$$

which is equal to the total number of balance equations per grid block, namely, NK mass balance and one energy balance equation. The thermodynamic state of a discretized flow system consisting of NEL volume elements, or grid blocks, is then completely specified by a set of NEL*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.

An important consideration in the modeling of fluid and heat flow processes with phase change is the choice of 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. A.9 applies.)

There are two alternative ways for dealing with this problem. One possibility is to use a set of ''persistent'' variables such as (pressure, enthalpy) or (density, internal energy), which remain independent even as phase conditions change, so that they can be used throughout the single- and two-phase regions. This approach has been successfully implemented in a number of multiphase codes (Pritchett, 1975; Faust and Mercer, 1975; Pruess *et al.*, 1979; Pruess and Schroeder, 1980). A drawback of this approach is that parametric relationships for thermophysical properties are usually formulated in terms of the ''natural'' variables pressure and temperature, so that their computation as functions of ''persistent'' variables becomes either more difficult (requiring solution of implicit equations) or entails some sacrifice in accuracy. The other possibility is to use the variables (pressure, temperature) only for single-phase conditions, and to ''switch'' to variables (pressure, saturation) when a transition to two-phase conditions occurs. Experience has proven this variable-switching approach to be a very robust method for treating multiphase systems, and it has been implemented in the MULKOM, TOUGH, and TOUGH2 codes.

The choice of primary variables and the switching procedures for phase transitions are different in different equation-of-state modules (see below).

3.2 Thermophysical Property Arrays

In TOUGH2 all of the NEL*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 Fig. 2). The starting location for primary variables for grid block N is NLOC + 1, where NLOC = (N – 1)*NK1.

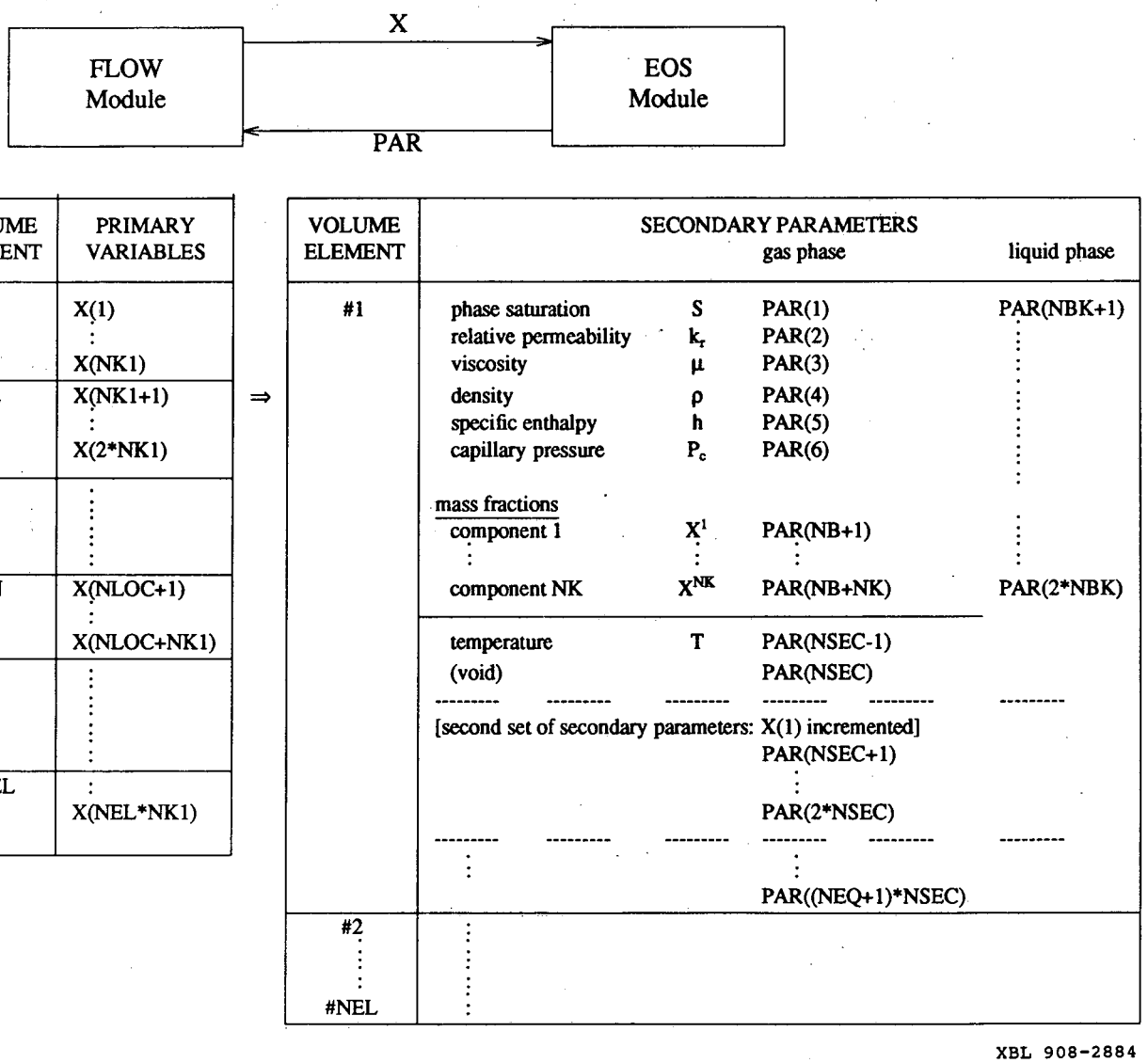


Figure 2. Structure of thermophysical property arrays in MULKOM and TOUGH2.

There are two additional arrays DX and DELX with structure identical to X. While X holds the primary variables corresponding to the last successful (converged) time step,

DX holds the latest increments calculated during the Newton-Raphson iteration process. Thus the latest updated primary variables are the quantities X + DX. The array DELX holds small increments of the X themselves (typically of order 10^{-7} * X), which are used to calculate incremented parameters needed for the numerical calculation of the derivatives in the Jacobian matrix $\mathbf{J} = -\partial R_n/\partial x_i$ (see Fig. 3; x_i denotes the collection of all primary independent thermodynamic variables). At the conclusion of a converged time step, the primary variables X are updated, $X \rightarrow X + DX$.

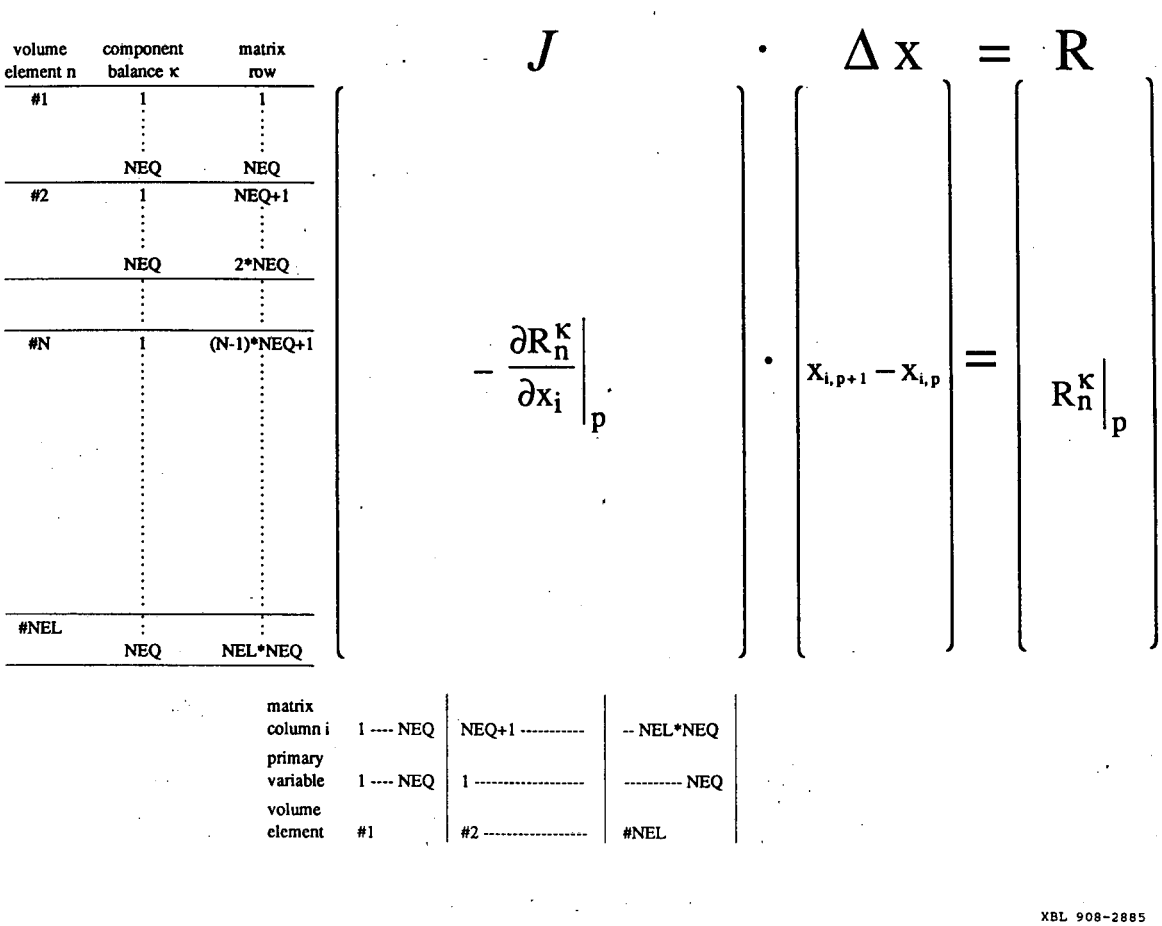


Figure 3. Linear equation structure.

It was stated above that the number of mass- and energy-balance equations per grid block is the same as the number NK1 of primary thermodynamic variables. In many

applications, however, the heat effects may be so small that temperature changes would be insignificant, and it would be sufficient to consider just the mass balance equations. The simplest way of forcing temperature changes to zero would be to 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 TOUGH2, 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 $\partial T/\partial t = 0$. For certain EOS modules, TOUGH2 offers a more elegant way of running problems without temperature changes, at a considerable savings of computing time. By convention we always take the first NK equations per grid block to represent mass balances, while the energy balance equation comes last as # NK1. TOUGH2 uses a parameter NEQ, distinct from NK, to number the balance equations per grid block, and normally we would have NEQ = NK + 1. However, the user 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, or a total of NEL. For the smaller set of NK equations we only solve for NK primary variables; thus the option NEQ = NK can only be used with those EOS modules that have temperature as primary variable # NK1 (see Table 1).

The EOS module calculates all thermophysical properties (''secondary parameters'') needed to assemble the mass- and energy-balance equations for the latest updated primary variables X + DX. These parameters are then stored sequentially in a large array ''PAR'' (see Fig. 2). The number of secondary parameters other than component mass fractions is NB (usually NB = 6); in addition there are NK mass fractions so that the total number of secondary parameters per fluid phase is NBK = NB + NK. The PAR array structure is shown in Fig. 2 for the case of two fluid phases; however, the coding permits any number of phases, as specified by the parameter NPH. The NPH*NBK phase-specific parameters are followed by temperature T and a void (unused) array member, so

that the total number of secondary parameters is NSEC = NPH*NBK + 2.

Note that the thermophysical properties are needed not only for calculating the residuals of the mass- and energy-balance equations (B.6), but also for calculating their derivatives in the Jacobian matrix (Eqs. B.7 and B.8). Thus, we require secondary parameters not only at the ''state point'' (latest X + DX), but also for the NEQ additional sets of primary variables in which one of the primary variables at a time is incremented by DELX. Therefore, the total number of secondary parameters per grid block is (NEQ + 1)*NSEC. Secondary parameters for grid block #N start after location #NLOC2 = (N – 1)*(NEQ + 1)*NSEC of the PAR array.

3.3 Linear Equation Setup

The data provided by the PAR-array are used in the flow module of TOUGH2 to assemble the linear equations (B.8) that are solved at each step of the Newton-Raphson iteration procedure. These equations are arranged and numbered sequentially, as shown in Fig. 3, 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. If the option NEQ = NK is chosen, only NK mass balance equations will be set up 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.

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*NEQ submatrix that is located on the diagonal of the Jacobian **J**. The flow terms, being dependent on primary variables of two grid blocks, will generate two non-

zero NEQ*NEQ submatrices of derivatives, which are located in the off-diagonal matrix locations corresponding to the two grid blocks.

In TOUGH2 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*NEQ. These are stored sequentially in a one-dimensional array CO; matrix elements for grid block N begin after location (N – 1)*NEQ*NEQ in CO. The corresponding row and column indices are stored separately in arrays IRN and ICN, respectively. 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 **J** may arise from sink and source terms if present; these are assigned in subroutine QU called from MULTI. Subsequently all flux terms are evaluated. These depend in general on the 2*NEQ primary variables of the two connected grid blocks, so that a total of 2*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 subroutine package MA28 (Duff, 1977) is called to solve the linear equations (B.8). 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. If convergence is not achieved within a specified maximum number of iterations (usually 8) the time step is repeated with reduced time increment.

3.4 Dimensioning of Major Arrays

The major problem-size dependent arrays reside in COMMON blocks, which are

dimensioned by means of a PARAMETER statement in the main (TOUGH2) program. An informational statement on permissible problem size (number of grid blocks, etc.) is provided in the printed output of a TOUGH2 run. When problem specifications exceed array dimensions the execution stops with a diagnostic printout. The user must then increase PARAMETER assignments accordingly, recompile the main program and relink. A list of major arrays used in TOUGH2 and their dimensions is given in Table 2 below.

Table 2. Summary of Major Common Blocks

Reference to	Common blocks	Length
Elements	E1 – E6 VINWES, AHTRAN	NEL (= number of elements)
Primary variables	P1 – P7	NEL*NK1†
Connections (interfaces)	C1 – C11 COMPO, PORVEL	NCON (= number of connections) NCON*NPH
Linear equations	L1 L2, L3 L4 L5 L6 L7 L8	$\gtrsim$ NZ = NEL*NEQ**2 + 2*NCON*NEQ**2 $\approx$ (2–4) *NZ NEL*NEQ NEL*5*NEQ NEL*8*NEQ NZ NEL*5*NEQ
Secondary parameters	SECPAR	NEL*(NEQ + 1) *NSEC

†NK1 ≡ NK + 1

4. Equation-of-State Modules

The thermophysical properties of fluid mixtures needed in assembling the governing mass- and energy-balance equations are provided by "equation-of-state" (EOS) modules. There is nothing in the MULKOM formulation to restrict the number of fluid components and phases that may be present. The flow module of TOUGH2 is coded in general fashion for calculating mass balances of an arbitrary number of NK components that are distributed among NPH phases. The EOS-modules included in the present release of TOUGH2, however, are all limited to flow systems with at most two components and two phases. More complex modules for three or more components and phases are in use at Lawrence Berkeley Laboratory, and source codes and documentation for these are expected to become available in the future.

Besides providing values for all secondary (thermophysical) parameters as functions of the primary variables, the EOS module must fulfill three additional important functions:

(i) the phase conditions pertaining to a given set of primary variables must be recognized (element-for-element),

(ii) the appearance or disappearance of phases must be diagnosed as primary variables change during the Newton-Raphson iteration process, and

(iii) primary variables must be switched in response to a change of phase.

The primary variables/secondary parameters concept as implemented in MULKOM and TOUGH2 essentially eliminates any direct connection between the choice of primary variables, and the secondary parameters that are used to set up the flow equations. This provides maximum flexibility and convenience in the choice of primary variables, because only secondary parameters are used in the flow equations. There is one single exception to this separation, namely, pressure (of a reference phase) is by convention

always the first primary variable, and it is used directly in the flow equations. The choice of all other primary variables is completely free. Note that of the several EOS modules described below, only one at a time should be linked with the other TOUGH2 modules.

4.1 EOS1 (Water, Water with Tracer)

This is the first, most basic EOS-module developed for MULKOM. It provides a description of pure water in its liquid, vapor, and two-phase states suitable for geothermal reservoir studies, and has a capability of representing "two waters" of identical physical properties, which contain different trace constituents. The default parameter settings for a single water component are (NK, NEQ, NPH, NB) = (1, 2, 2, 6). The option NEQ = 1 is available for running problems that involve only liquid water, or only superheated steam, under constant temperature conditions. The primary variables are (P, T) for single-phase points, (P_g, S_g) for two-phase points. For the convenience of the user it is possible to initialize two-phase points as (T, S_g); a numerical value of the first primary variable less than 374.15 will automatically be taken to mean temperature (in °C) instead of gas pressure, and will cause variables to be internally converted from (T, S_g) to (P_{sat}(T), S_g) prior to execution.

The two-waters capability can be invoked by specifying (NK, NEQ, NPH, NB) = (2, 3, 2, 6) in data block "MULTI" (see below). With this option, two water mass balances will be set up, allowing separate tracking of the two components. For example, one could specify the water initially present in a flow system as "water 1," while water being injected is specified as "water 2." The primary variables in this case are (P, T, X) for single-phase points, and (P_g, S_g, X) for two-phase points, where X is the mass fraction of "water 2" present. All thermophysical properties (density, specific enthalpy, viscosity) are assumed independent of the component mixture; i.e., independent of the mass fraction X. This approximation is applicable for problems in which the identity of different waters is distinguished by the presence of different trace constituents, which

occur in concentrations low enough to not appreciably affect the thermophysical properties.

All water properties (density, specific enthalpy, viscosity, saturated vapor pressure) are calculated from the steam table equations as given by the International Formulation Committee (1967). The formulation includes subregion 1 (subcooled water below T = 350°C), subregion 2 (superheated steam), and subregion 6 (saturation line up to T = 350°C). Within these regions, density and internal energy are represented within experimental accuracy. Viscosity of liquid water and steam are represented to within 2.5% by correlations given in the same reference. For details of the formulation, its accuracy and range of validity, refer to the original publication.

The phase diagnostic procedures are as follows. When initializing a problem, each grid block has two primary variables, (X1, X2). Whether X2 means gas saturation (two-phase) or temperature (single phase) is decided from the numerical value: For X2 > 1.5, X2 is taken to be temperature in °C, otherwise it is gas saturation. (Although physically saturation is restricted to the range $0 < S < 1$, it is necessary to allow saturations to exceed 1 during the Newton-Raphson iteration). If X2 is temperature, we have single phase conditions; specifically, for P (= X1) > $P_{sat}(T)$ we have single phase liquid, otherwise we have single phase steam. Subsequent to initialization, the phase condition is identified simply based on the value for S_g, as stored in the array PAR. $S_g = 0$: single phase liquid, $S_g = 1$: single phase vapor, $0 < S_g < 1$: two-phase.

Phase change is recognized as follows. For single phase points the temperature (second primary variable) is monitored, and the corresponding saturation pressure is compared with P. For a vapor (liquid) point to remain vapor (liquid), we require that $P < P_{sat}$ ($P > P_{sat}$); if this requirement is violated, a transition to two-phase conditions takes place. The primary variables are then ''switched'' to (P_g, S_g), and these are initialized as $P_g = P_{sat}(T)$, $S_g = 0.999999$ if the point was in the vapor region, and $S_g = 0.000001$ if it was in the liquid region. For two-phase points S_g is monitored; we require that $0 < S_g < 1$

for a point to remain two-phase. If $S_g < 0$ this indicates disappearance of the gas phase; the primary variables are then switched to (P, T), and the point is initialized as single phase liquid, with T taken from the last Newton/Raphson iteration, and P = 1.000001 * $P_{sat}(T)$. For $S_g > 1$ the liquid phase disappears; again the primary variables are switched to (P, T), and the point is initialized as single phase vapor, with T taken from the last Newton/Raphson iteration, and P = 0.999999 * P_{sat} (T). Note that in these transitions we preserve temperature rather than pressure from the last iteration. This is preferable because in most flow problems temperature tends to be more slowly varying than pressure. A summary of EOS1 specifications and parameters is given in Table 3 below.

Table 3. Summary of EOS1

Components	# 1: water # 2: "water 2" (optional)
Parameter Choices	
(NK, NEQ, NPH, NB) =	(1, 2, 2, 6) one water component, non-isothermal (default) (1, 1, 2, 6) only liquid, or only vapor, isothermal (2, 3, 2, 6) two waters, non-isothermal*
Primary Variables	
single phase conditions (P, T, [X]) – (pressure, temperature, [mass fraction of water 2]†)	
two-phase conditions (P_g, S_g, [X]) – (gas phase pressure, gas saturation, [mass fraction of water 2]†)	

*two waters cannot be run in isothermal mode, because in this case temperature is not the last primary variable

†optional, for NK = 2 only

4.2 EOS2 (Water, CO_2)

This fluid property module was developed by O'Sullivan *et al.* (1985) for describing fluids in gas-rich geothermal reservoirs, which often contain CO_2 mass fractions from a few percent to occasionally 80% or more (Atkinson *et al.*, 1980). It accounts for non-

ideal behavior of gaseous CO_2, and dissolution of CO_2 in the aqueous phase according to Henry's law with heat-of-solution effects. The thermophysical property correlations are based on the model of Sutton and McNabb (1977); a formulation due to Pritchett *et al.* (1981) is used for the viscosity of vapor-CO_2 mixtures.

Specifications and parameters of EOS2 are summarized in Table 4. A more detailed description and applications to geothermal reservoir problems are given in the paper by O'Sullivan *et al.* (1985).

Table 4. Summary of EOS2

Components	# 1: water # 2: CO_2
Parameter Choices	
(NK, NEQ, NPH, NB) =	(2, 3, 2, 6) no other options are available
Primary Variables	
single phase conditions (P, T, PCO_2) – (pressure, temperature, CO_2 partial pressure)	
two-phase conditions (P_g, S_g, PCO_2) – (gas pressure, gas saturation, CO_2 partial pressure)	

4.3 EOS3 (Water, Air)

This module is an adaptation of the EOS module of TOUGH for the TOUGH2 program, and implements the same thermophysical properties model (see Pruess, 1987). Thus, all water properties are represented by the steam table equations as given by the International Formulation Committee (1967). Air is approximated as an ideal gas, and additivity is assumed for air and vapor partial pressures in the gas phase, $P_g = P_a + P_v$. The viscosity of air-vapor mixtures is computed from a formulation given by Hirschfelder *et al.* (1954). The solubility of air in liquid water is represented by Henry's law; i.e., dissolved air mole fraction $x_l^{(air)}$ is proportional to air partial pressure in the gas

phase.

$$P_a = K_H \cdot x_l^{(air)} \tag{4}$$

Here K_H is Henry's constant, which is a slowly varying function of temperature, varying from 6.7×10^9 Pa at 20°C to 1.0×10^{10} Pa at 60°C and 1.1×10^{10} Pa at 100°C (Loomis, 1928). Because air solubility is small, this kind of variation is not expected to cause significant effects, and a constant $K_H = 1.0 \times 10^{10}$ Pa was adopted.

EOS3 differs from the EOS module of TOUGH in one important respect, namely, the choice of primary thermodynamic variables. In TOUGH we have (P,T,X) for single phase conditions, (P_g,S_g,T) for two-phase conditions. The choice made in EOS3 is (P,X,T) for single phase, (P_g,10.+S_g,T) for two-phase. The rationale for the seemingly ''bizarre'' choice of 10.+S_g as a primary variable is as follows. As an option, we wish to be able to run isothermal two-phase flow problems with the specification NEQ = NK, so that the then superfluous heat balance equation needs not be engaged. This requires that temperature T be the third primary variable. The logical choice of primary variables would then appear to be (P,X,T) for single phase and (P_g,S_g,T) for two-phase conditions. However, both X and S_g vary over the range (0,1), so that this would not allow a distinction of single phase from two-phase conditions solely from the numerical range of primary variables. By taking the second primary variable for two-phase conditions to be X2 = 10.+S_g, the range of that variable is shifted to the interval (10,11), and a distinction between single and two-phase conditions can be easily made. As a convenience to TOUGH users, primary variables can optionally be initialized identical to TOUGH specifications by setting MOP(19) = 1. A summary of EOS3 specifications is given in Table 5.

Table 5. Summary of EOS3

Components	# 1: water # 2: air
Parameter Choices	
(NK, NEQ, NPH, NB) =	(2,3,2,6) water and air, noniso-thermal (default) (2,2,2,6) water and air, isothermal
Primary Variables*	
single phase conditions (P, X, T) – (pressure, air mass fraction, temperature)	
two-phase conditions (P_g,S_g+10,T) – (gas phase pressure, gas saturation plus 10, temperature)	

*By setting MOP(19) = 1, initialization can be made with TOUGH-style variables (P,T,X) for single phase, (P_g,S_g,T) for two-phase.

4.4 EOS4 (Water, Air, with Vapor Pressure Lowering Capability)

The specification of thermophysical properties in this EOS differs from EOS3 in that provision is made for vapor pressure lowering effects. Vapor pressure is expressed by Kelvin's equation (Eq. A.9); it is a function not only of temperature, but depends also on capillary pressure, which in turn is a function of saturation. The primary variables are (P,T,P_a) for single phase conditions and (P_g,S_g,P_a) for two-phase conditions. Note that in two-phase conditions temperature is not among the primary variables. It is implicitly determined from the relationship $P_g - P_a = P_v$, with $P_v = P_v(T,S_l)$ as given in Eq. A.9.

It would be possible to use other sets of primary variables, in particular, temperature could be used also in two-phase conditions. Our test calculations for a number of examples indicated, however, that the choice (P_g,S_g,P_a) usually led to better convergence behavior than the choice (P_g,S_g,T). The reason for the numerically inferior behavior of the latter set is in the air mass balance. With the variables (P_g,S_g,T), the amount of air present in a grid block becomes controlled by the difference between total gas pressure

P_g and effective vapor pressure $P_v = P_{sat}(T) \cdot f_{VPL}(T,S_l)$, which can be subject to very severe numerical cancellation. From the applications viewpoint, however, initialization of a flow problem with the set (P_g,S_g,T) may be much more physical and convenient. EOS4 allows to initialize two-phase points as (P_g,S_g,T); this capability can be selected by specifying MOP(19) = 1 in the INPUT file. The default option for MOP(19) = 0 is (P_g,S_g,P_a).

As a further convenience to users, the choice MOP(19) = 2 allows EOS4 to be initialized with EOS3 variables of (P, X, T) for single phase, $(P_g, S_g + 10, T)$ for two-phase. This way continuation runs with EOS4 can be made from EOS3-generated conditions.

Note that, when using MOP(19) $\neq$ 0 options, data block or file INCON *must* terminate on a blank record (' '). If '+++' is encountered in INCON, it is assumed that primary variables are provided in agreement with internal usage; MOP(19) is then reset to zero and an informative message is printed.

Vapor pressure lowering effects raise new issues because it is now possible for a liquid phase to be present under conditions where vapor partial pressure and gas phase total pressure are less than the saturation pressure. What is the appropriate pressure at which liquid phase density, enthalpy and viscosity are to be evaluated? We believe that a physically plausible choice is to take $P_l = \max(P_g, P_{sat})$, and this has been implemented in EOS4. The implementation faces a difficulty, however, because temperature is not among the primary variables in two-phase conditions, so that P_{sat} is only implicitly known; moreover, vapor pressure lowering effects are functionally dependent on liquid phase density, which is also a function of temperature. This leads to a potentially unstable situation with regard to the choice of liquid phase pressure under conditions where $P_g \approx P_{sat}$, which happens to be a common occurrence in boiling regions. In order to avoid this problem we evaluate liquid water density in the Kelvin equation for vapor pressure lowering (Eq. A.9) always at $P_l = P_{sat}$, which will be an excellent approximation due to the small compressibility of liquid water. In all accumulation and flow terms, the

density of liquid water is evaluated at $P_l = \max(P_g, P_{sat})$. Vapor pressure lowering can be optionally suppressed by setting MOP(20) = 1. A summary of EOS4 specifications is given in Table 6.

Table 6. Summary of EOS4

Components	# 1: water # 2: air
Parameter Choices	
(NK, NEQ, NPH, NB) =	(2, 3, 2, 6) water and air, non-isothermal (no other choices available)
MOP (20) = 1: optionally suppress vapor pressure lowering effects	
Primary Variables*†	
single phase conditions (P, T, P_a) – (pressure, temperature, air partial pressure)	
two-phase conditions (P_g, S_g, P_a) – (gas phase pressure, gas saturation, air partial pressure)	

*By setting MOP (19) = 1, initialization of two-phase conditions can be made with (P_g, S_g, T).
†By setting MOP(19) = 2, initialization can be made with EOS3-style variables of (P,X,T) for single phase, $(P_g, S_g + 10, T)$ for two-phase.

4.5 EOS5 (Water, Hydrogen)

In a number of waste disposal projects, corrosive metals are to be emplaced in geologic formations beneath the water table. These will evolve a mixture of gases, with hydrogen being the chief constituent. The EOS5 fluid property module was developed to study the behavior of groundwater systems in which hydrogen release is taking place. It is a close ''cousin'' of EOS3, the main difference being that the air component is replaced by hydrogen, with considerably different thermophysical properties (see Table 7). The assignment and handling of primary thermodynamic variables in EOS5 is identi-

cal to EOS3 (see Table 5). The main differences in the assignment of secondary parameters are as follows. Density of gaseous hydrogen is computed from the ideal gas law. Viscosity and water solubility of hydrogen are interpolated from the data given in Table 7. For temperatures in excess of 25°C, the solubility at 25°C is used. The parameter specifications of EOS5 are identical to those of EOS3 as given in Table 5, with "air" replaced by "hydrogen."

Table 7. Thermophysical Properties of Hydrogen

Density at P = 1 bar	Experimental*	Ideal Gas Law†
T = 280 K	.086546 kg/m^3	.08660 kg/m^3
T = 300 K	.080776 kg/m^3	.08082 kg/m^3
Viscosity*		
	T = 0°C	T = 100°C
P = 1 bar	8.40×10^{-6} Pa·s	10.33×10^{-6} Pa·s
P = 100 bar	8.57×10^{-6} Pa·s	10.44×10^{-6} Pa·s
Solubility in water at P = 1 bar§		
T = 0°C	1.92×10^{-6} g H_2/g H_2O	
T = 25°C	1.54×10^{-6} g H_2/g H_2O	

*from Vargaftik (1975), p. 39.

†universal gas constant R = 8314.56 J/mol/°C; molecular weight of hydrogen 2.0160.

§after Dean (1985).
Solubility at different pressures is computed from Henry's law.

5. User Features

Much of the data handling in TOUGH2 is accomplished by means of a number of disk files, which are written in a format of 80 characters per record, so that code users can edit and modify them with any normal text editor. Table 8 summarizes the disk files other than (default) INPUT and OUTPUT used in TOUGH2. Most of these are also used in TOUGH, and files with the same names in both codes have identical formats. The use and function of these files is described in the following sections as well as in the sample problems; further information is available in the TOUGH User's Guide.

Table 8. TOUGH2 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 thermodynamic conditions
SAVE	written in subroutine WRIFI to record thermodynamic conditions at the end of a TOUGH2 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, etc.) read (optionally) in subroutine RFILE to initialize geometry data for a fractured-porous system
LINEQ	written in linear equation solver ''MA28'', to provide informative messages on linear equation solution
TABLE	(optional; available only with certain EOS modules) written in subroutine QLOSS to record data on heat exchange with impermeable confining layers, or heat and fluid exchange with embedded matrix blocks in a fractured-porous medium read in QLOSS in a continuation run

VERS	written in all TOUGH2 program units with informational message on version number, date, and function read in main program "TOUGH2" and printed to default OUTPUT at the conclusion of a TOUGH2 simulation run; printing of version information is suppressed when keyword 'NOVER' is present in INPUT file

5.1 Specification of Flow Geometry

Handling of flow geometry data in TOUGH2 is upward compatible with TOUGH input formats and data handling. As in other "integral finite difference" codes (Edwards, 1972; Narasimhan and Witherspoon, 1976), flow geometry is defined by means of a list of volume elements ("grid blocks"), and a list of flow connections between them. This formulation can cope with 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).

Volume elements in TOUGH2 are identified by five-character names, such as "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 the corresponding geometric data (see Fig. 4). As in TOUGH, 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. The data formats on file MESH are identical with the format specifications for data blocks ELEME and CONNE.

TOUGH2 offers additional avenues for defining flow system geometry. By means of the keyword 'MESHMAKER' in the INPUT file, a special program module can be

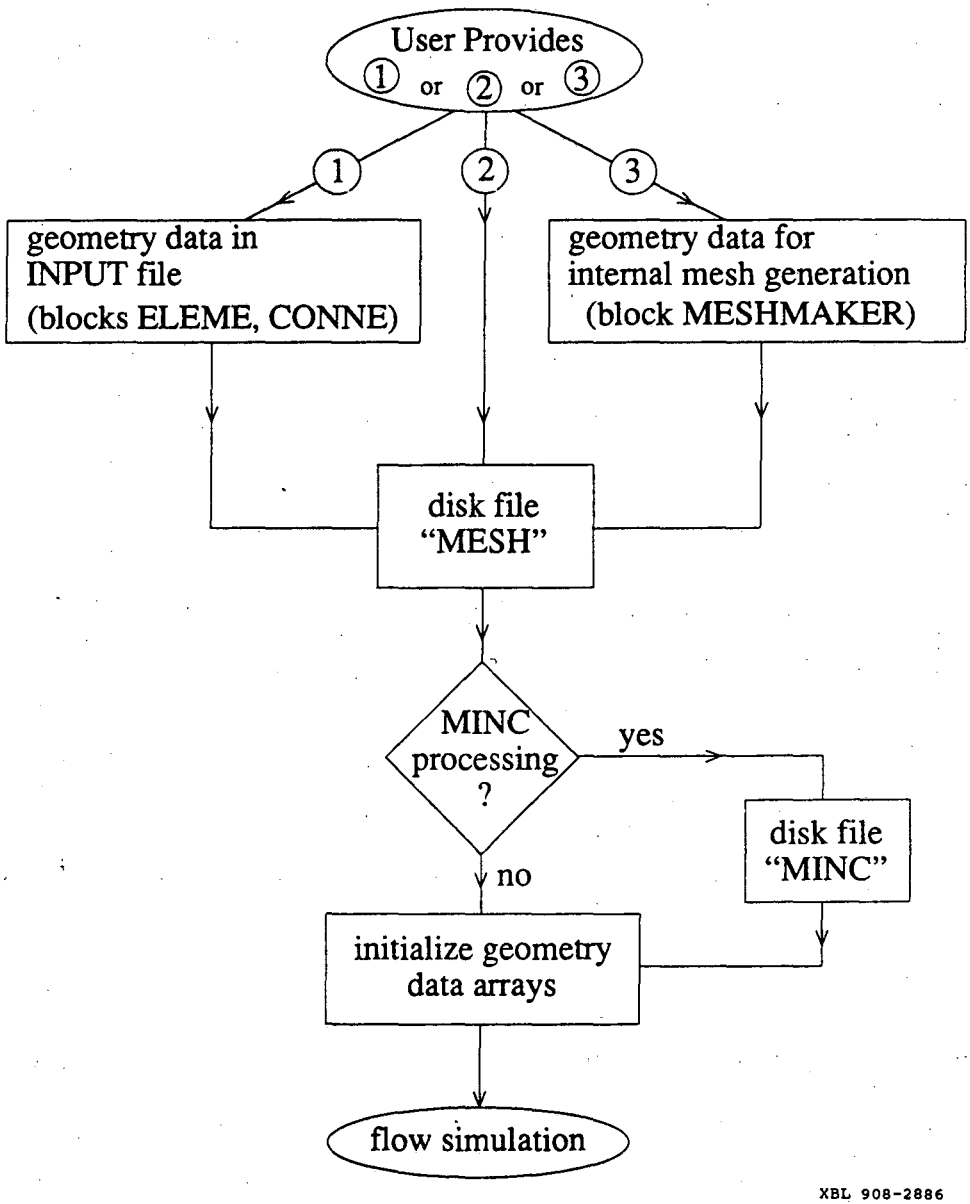


Figure 4. User options for supplying geometry data.

invoked to perform a number of mesh generation and processing operations. The MESH-MAKER module itself has a modular structure; present sub-modules include "RZ2D" for generating two-dimensional radially symmetric (R-Z) meshes, and "XYZ" for one-, two-, and three-dimensional rectilinear (Cartesian) grids. Multiple-porosity processing for simulation of flow in naturally fractured reservoirs can be invoked by means of a key-word 'MINC,' which stands for "multiple interacting continua" (Pruess and Narasimhan, 1982, 1985; Pruess, 1983a; see Appendix C). 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." (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; see Fig. 4.) As a convenience for users desiring graphical display of data, the internal mesh generation process will also write nodal point coordinates on file MESH. These data are written in 3E10.4 format into columns 51–80 of each grid block entry in data block ELEME. At the present time, no internal use whatsoever is made of nodal point coordinates in TOUGH2.

In TOUGH2 elements are referenced by names consisting of a string of five characters, '12345.' These are arbitrary, except that the last two characters (#4 and 5) must be numbers. Specific naming conventions have been adopted in the internal mesh generation process. For RZ2D, the last two characters directly number the radial grid blocks, from 1 through 99. Character #3 is blank for the first 99 radial blocks, and then runs through the sequence 1, 2, ..., 9, A, B, ..., Z for a maximum total of 3599 radial blocks. The second character counts up to 35 grid layers as 1, 2, ..., 9, A, B, ..., Z. The first character is 'A' for the first 35 layers, and is incremented to B, C, ..., Z, 1, 2, ..., 9 for subsequent groups of 35 layers.

For rectilinear meshes generated by XYZ, characters 4 and 5 together number the

grid blocks in X-direction, while character #3 = 1, 2, ..., 9, A, B, ..., Z numbers Y-direction grid blocks, and character #2, running through the same sequence as #3, numbers grid blocks in Z direction. ''Overflows'' in any of these (more than 99 X-blocks, more than 35 Y- or Z-blocks) advance character #1 through the sequence A, B, C, ..., Z. 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, or by appropriate source code changes in subroutines WRZ2D and GXYZ.

TOUGH2 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 Fig. 18.

The naming conventions for the MINC process are somewhat different from those originally adopted in the GMINC program (Pruess, 1983a), and 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 (POMED), then follows the effective fracture continuum (FRACT), and then the rock matrix (MATRX). An example is given in sample problem 4.

Users should beware that the MINC process may lead to ambiguous element names when the ''inactive'' element device (see below) is used to keep a portion of the primary mesh as unprocessed porous medium.

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 a subsequent flow simulation. MESHMAKER input formats are described in Section 6, and examples of practical applications are given in the sample problems. Execution of MESHMAKER produces printed output which is self-explanatory.

5.2 Initial Conditions and Restarting

As in the TOUGH code, initial thermodynamic conditions for the volume elements in the flow domain can be assigned to identical default values for all elements, or they can be prescribed for each element individually by means of a data block ''INCON.'' A file ''INCON,'' written to the same specifications as data block ''INCON,'' can also be used for initialization.

A simulation problem can be conveniently run in several segments. At the end of a simulation run TOUGH2 writes the primary thermodynamic variables of all elements on a disk file ''SAVE'' with format specifications identical to ''INCON.'' For a subsequent continuation run, file ''SAVE'' can be merged into the INPUT file as data block ''INCON,'' or it can be renamed as file ''INCON.'' In the latter case no data block INCON can be present in the INPUT file, as this would cause the INCON file to be overwritten.

TOUGH2 offers the additional facility of assigning initial conditions uniformly throughout selected zones of the simulation grid. This is invoked by means of a data block ''INDOM,'' which provides information on the thermodynamic conditions in user-defined domains. The format specifications for block ''INDOM'' are similar to those used in ''INCON'' (see Section 6). Thermodynamic conditions given in block ''INDOM'' take precedence over default assignments for the entire flow domain; specifications for individual grid blocks in ''INCON'' supercede all other assignments.

The "normal" way of defining initial conditions is by directly providing the primary thermodynamic variables. Note that these variables are generally different for different EOS modules. The thermodynamic state variables that are used internally in TOUGH2 as primary dependent variables may not always be the most convenient variables for a user to initialize a flow problem. The parameter MOP(19) offers a variety of choices, which permit initialization with variables different from the internally used primary variables. These choices are different for different EOS modules, and are documented in the one-page informational printout produced by each EOS.

At the end of a simulation run, file "SAVE" will always be written with the internal primary variables of the EOS module used. When modifying an INPUT file for a continuation run, MOP(19) might therefore have to be changed to its default value MOP(19) = 0 for proper initialization. To minimize the possibility of user error, an automatic "switch" has been implemented in TOUGH2, as follows. The file "SAVE" as internally written by TOUGH2 terminates on a record with '+++' in the first three columns, followed by one record with restart information. When the data block "INCON" or file "INCON" terminates on '+++' rather than on a blank line, it is assumed that this "INCON" was internally generated in a previous TOUGH2 run and that, therefore, it is written with the internally used set of primary variables. Accordingly, when '+++' is encountered in INCON the switch MOP(19) is reset to zero, and an informative message to this effect is printed.

5.3 Boundary Conditions and "Inactive" Elements

Boundary conditions can be of two basic types. Dirichlet conditions prescribe thermodynamic conditions, such as pressure, temperature, etc. on the boundary, while 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 with simplicity itself, namely, by not specifying any

flow connections across the boundary. More general flux conditions are prescribed, as in TOUGH, by introducing sinks or sources of appropriate strengths into the elements adjacent to the boundary.

TOUGH did not offer any special means of prescribing Dirichlet boundary conditions. Such conditions were implemented simply by assigning very large volumes, 10^{50} m^3, say, to certain grid blocks, so that their thermodynamic conditions would remain practically unaltered during a simulation run. TOUGH2 offers an alternative means for implementing Dirichlet conditions, which provides savings in computational work along with added user conveniences in running simulation problems. This is accomplished with the simple device of ''active'' and ''inactive'' elements.

By convention, elements encountered in data block ''ELEME'' (or files ''MESH'' or ''MINC'') are taken to be ''active'' until the first element entry with a zero or negative volume is encountered. The first element with volume less than or equal to zero, and all subsequent elements, are taken to be ''inactive.'' For the inactive elements no mass or energy balance equations are set up, and their primary thermodynamic variables are not included in the list of unknowns. Otherwise, however, inactive elements can appear in flow connections and initial condition specifications like all other elements. This feature can be conveniently used to specify Dirichlet boundary conditions, by gathering all elements beyond the desired flow domain boundary at the end of the ELEME-block, and inserting a ''dummy'' volume element of zero volume in front of them. Thermodynamic conditions for the inactive elements will be rigorously maintained during a simulation run. Their computational overhead is moderate because they do not increase the dimensionality of the flow problem (number of equations and unknowns; see Fig. 3).

The inactive-element concept can be conveniently used for simulating flow problems that evolve through different process segments. For example, in many reservoir simulation problems it is desired to first calculate a ''natural'' steady state, corresponding to gravitational (or gravity-capillary) equilibrium, or to steady mass and heat flows

produced by certain pressure and temperature conditions at the flow system boundaries. Subsequently one wishes to simulate the transient changes in response to man-made perturbations or to naturally occurring changes, while maintaining gravity-equilibrated pressure conditions at the boundary. Such boundary conditions can be implemented by including elements beyond the desired boundary in the gravity equilibration run, and making these elements inactive in a subsequent continuation run involving production or injection operations.

The specification of inactive elements can also be used in the MESHMAKER module to steer the MINC-process of subgridding volume elements. By convention, inactive elements will remain unpartitioned, i.e., they will be treated as a single porous medium.

5.4 Heat Exchange with Confining Beds

One possibility for modeling heat exchange with confining beds is to simply extend the computational grid into the cap- and base-rock, which would be given small or vanishing permeability. In this approach heat exchange would be treated no different than flow in the reservoir. A drawback of this approach is that even for modest accuracy requirements, the number of grid blocks in the heat flow domain could easily become larger than the number of grid blocks in the reservoir, leading to a very inefficient calculation. A much more efficient alternative is application of a semi-analytical method, which requires no grid blocks outside of the fluid flow domain, and permits better accuracy for short- and long-term heat exchange. TOUGH2 provides an option to use the method of Vinsome and Westerveld (1980), which gives excellent accuracy for heat exchange between reservoir fluids and confining beds 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 cap- and base-rock 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:

$$T(x,t) - T_i = (T_f - T_i + px + qx^2) \exp(-x/d) \tag{5}$$

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

$$d = \frac{(\theta t)^{1/2}}{2} \tag{6}$$

where $\theta = K/\rho C$ is the thermal diffusivity, K 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. (5). 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.

There is no separate input data block for specifying a semianalytical heat exchange calculation. Instead, a number of parameters have to be specified in different blocks to engage this option, as follows. A semi-analytical heat loss calculation will be performed only when the parameter MOP(15) in record PARAM.1 is set equal to 1. Initial temperature as well as heat capacity and conductivity of the confining beds is specified by means of data provided for the very last volume element in data block ELEME. The initial temperature is taken as the temperature with which the last element is initialized. Heat capacity and conductivity are taken from data provided in block ROCKS for the particular

domain to which the last element belongs. Thus, if a semianalytical heat exchange calculation is desired, the user would append an additional inactive element in block ELEME, and provide the desired parameters as initial conditions and domain data, respectively, for this element. Finally, 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 grid block data in block ELEME. Volume elements for which a zero-interface area is specified will not be subject to heat exchange.

In the present version of TOUGH2, a semi-analytical heat exchange calculation can be performed only when the ''geothermal'' EOS modules EOS1 or EOS2 are used. An implementation for other EOS modules, and an extension to heat exchange with finite-size impermeable rock matrix blocks embedded in the flow domain, will be included in a future release (Pruess and Wu, 1989).

At the termination of a run the data necessary for continuing the heat exchange calculation in a TOUGH2 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, TOUGH2 assumes that no prior heat exchange with confining beds has taken place, and takes their temperatures to be uniform and equal to the temperature of the very last volume element in block ELEME.

5.5 Future Code Developments

It had been mentioned above that the present release of TOUGH2 as described in this report includes a selection of some of the better tested and useable MULKOM program modules. The LBL group has developed a large number of additional modules and descendants of MULKOM and TOUGH which are expected to be made available in future releases. These include additional EOS modules for multiphase fluid mixtures containing hydrocarbon and non-Newtonian fluids, capabilities for modeling rock-fluid

interactions with dissolution and precipitation processes and associated porosity and permeability change, modules for production and injection well operations and scheduling, preprocessor programs for higher-order differencing schemes to diminish space discretization errors, and specialized routines for the study of multiphase flow processes in fractured media, including hysteresis effects. Many developments in the TOUGH code have also been made by users outside LBL to enhance process descriptions, improve execution speed, and add user conveniences (Pruess, 1990b).

Further improvements in TOUGH2 are desirable to represent certain processes in a more accurate and complete fashion. For example, the present coding represents only advection and does not include any diffusive or dispersive processes in the liquid phase. Appropriate flux terms could be added in a continuum-based formulation; alternatively, marker particles could be employed to model species transport. The description of gas flow processes includes Knudsen (slip flow) effects only in a rough approximation due to Klinkenberg (1941); a more unified treatment of Darcy, Knudsen, and diffusive effects is desirable. Such future enhancements in process description are expected to be readily feasible by adding appropriate modules to the existing TOUGH2 structure. Also, many user features such as a more interactive and graphically oriented interface could be added to facilitate code applications (see the discussion in Pruess, 1990a).

When making changes in the code, it is essential to preserve a continuous dependence of all secondary parameters on the primary thermodynamic variables. True numerical discontinuities, such as a non-zero capillary air entry pressure, are inadmissible. They may lead to an unstable situation in which the residuals in the governing equations (B.6) become discontinuous functions of the primary variables, so that it may be impossible to reduce them to small values. A finite transition region for continuous variation of parameters must be provided. For example, if a user wishes to define a capillary pressure function with air entry effects, (s)he must provide a finite interval, from $S_l = 1$ to $S_l = 0.99$, say, over which capillary pressure changes from 0 to $P_{a.e.}$. For ease of conver-

gence during the non-linear Newton/Raphson process, it is actually desirable that the derivatives of secondary parameters with respect to primary variables also be continuous.

Future releases of TOUGH2 will remain upward compatible with the present version. We will strive to maintain a transparent and accessible source code that will facilitate applications to complex flow problems, and will be amenable to further enhancements. Users making code modifications are urged to utilize the simple version control system provided in TOUGH2. Any time changes in the code are made, the dates in the WRITE(11,899) statement appearing at the top of the affected program units should be updated, so that a traceable and referenceable record of source code developments and applications may be maintained (see the discussion in the Introduction, and the example given in Fig. 14).

6. Preparation of Input Data

TOUGH2 input is to be provided through a file INPUT, organized into data blocks which are labeled by five-character keywords. Apart from a few exceptions, discussed below, the order of the data blocks in the INPUT file is arbitrary. As has been mentioned before, the TOUGH2 input formats are upward compatible with those of TOUGH. Figure 5 gives a listing of TOUGH input formats, and indicates a small number of optional additional parameters that in TOUGH2 are provided through the same data blocks. TOUGH2 has a number of new, optional data blocks. These are listed in Table 9, and the corresponding input formats are shown in Fig. 6.

Table 9. New Data Blocks in TOUGH2 (all optional)

keyword	function
MESHM	invokes internal mesh generation and processing facilities
MULTI	allows to select number of fluid components and balance equations per grid block; applicable only with certain EOS modules that offer different options
INDOM	permits domain-specific initialization of thermodynamic conditions
NOVER	if present, optionally suppresses a printout of versions and dates of the program units executed in a TOUGH2 run
ENDFI	alternative to "ENDCY" for closing a TOUGH2 input file; will cause flow simulation to be skipped; useful if only mesh generation is desired

We shall first summarize the new parameters and options of TOUGH2 in the "old" (TOUGH) data blocks. Subsequently, we shall discuss the input formats and choices available through the new data blocks. For a detailed discussion of those input variables that are identical to the ones used in TOUGH, refer to the TOUGH User's Guide.

6.1 Enhancements in TOUGH Data Blocks

As a convenience to the user, comments or text can be inserted between data blocks anywhere in a TOUGH2 input file. Such records will generate a one-line printed output

TOUGH-Input with TOUGH2 Extensions

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

TITLE

ROCKS

MAT	NAD	DROK	POR	PER(1)	PER(2)	PER(3)	CWET	SPHT
COM		EXPAN	CDRY	TORTX	GK			
IRP		RP(1)	RP(2)	RP(3)	RP(4)	RP(5)	RP(6)	RP(7)
ICP		CP(1)	CP(2)	CP(3)	CP(4)	CP(5)	CP(6)	CP(7)

PARAM

NO TE KDATA	MCYC	MSEC	MCYPR	MOP(I), I=1,24	DIFFO	TEXP	BE	
TSTART	TIMAX	DELTEN or -NDLT	DELTMX	ELST		GF	REDLT	SCALE
DLT(1)	DLT(2)	DLT(3)						
						DLT(M) (M≤8*NDLT)		
RE1	RE2	U	WUP	WNR	DFAC			
DEP(1)		DEP(2)		DEP(3)				

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

(optional)

RIPCAP

IRP		RP(1)	RP(2)	RP(3)	RP(4)	RP(5)	RP(6)	RP(7)
ICP		CP(1)	CP(2)	CP(3)	CP(4)	CP(5)	CP(6)	CP(7)

(optional)

TIMES

ITI	ITE	DELAF	TINTER					
TIS(1)		TIS(2)	TIS(3)			TIS(ITI)		

ELEME

EL	NE	NSEQ	NADD	MA1	MA2	VOLX	AHTX		X	Y	Z

CONNE

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

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

ITAB

GENER

EL	NE	SL	NS	NSEQ	NADD	NADS	LTAB		TYPE	GX	EX	HG
F1(1)					F1(2)			F1(LTAB)				
F2(1)					F2(2)			F2(LTAB)				
F3(1)					F3(2)			F3(LTAB)				

INCON

EL	NE	NSEQ	NADD	PORX		
X1			X2		X3	

(optional)

START

ENDCY

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

Figure 5. TOUGH Input Formats with TOUGH2 Extensions.

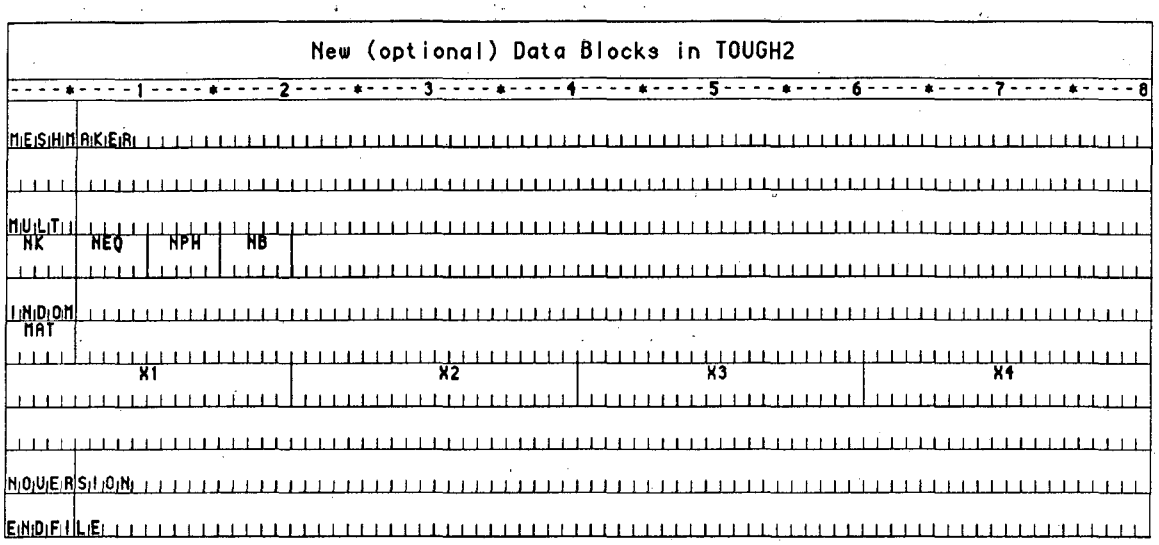


Figure 6. Input Formats for new TOUGH2 data blocks.

"Have read unknown block label '[first five characters]' – ignore this, and continue reading input data," but will otherwise be ignored. (In TOUGH, execution simply stopped when an unknown block label was encountered.)

Several of the "MOP" parameters (first record in block 'PARAM'), that control optional printout and some calculational choices, have different options and settings than in TOUGH. Each TOUGH2 run will produce a one-page informative printout of available selections and options chosen. Additional parameters provided through TOUGH data blocks are as follows (see Fig. 5). The second (optional) record in block 'ROCKS' has a parameter GK which is the Klinkenberg parameter b in the gas phase permeability relationship $k = k_0(1 + b/P)$. In partially saturated media, vapor diffusion can be considerably enhanced in comparison to the expression given in Eq. (A.7), due to phase change effects (condensation/evaporation) at the pore level. Such enhanced diffusion can be modeled by specifying a suitable value, typically of order 1, for the parameter $B = \phi S_g \tau$. This is to be entered as parameter BE in the first record in block 'PARAM.' In block 'ELEME,' AHTX is the contact area of a grid block with the top or bottom boundary of

the flow system. This can be used with certain EOS modules for a heat exchange calculation with semi-infinite half spaces that represent the confining beds of a flow system (see example problem 3, below). The X,Y,Z-data in element records are nodal point coordinates. These are not used at all in TOUGH2, but can be optionally provided in the 'ELEME' block to facilitate plotting.

When working with different EOS modules, there is a need to be able to specify injection of different fluid components (or heat). Table 10 lists the TYPE specifications that can be used in data block GENER in the input file.

Table 10. Specifications for Generation Types

Component Index	Code Words (variable "TYPE" in block GENER)	Component				
		EOS1	EOS2	EOS3	EOS4	EOS5
#1	COM1, MASS, WATE	water1	water	water	water	water
#2	COM2, AIR, WATR	water2	CO_2	air	air	H_2
#3*	COM3	–	–	–	–	–
#4*	COM4	–	–	–	–	–
#NK1†	HEAT	heat	heat	heat	heat	heat

*not used in EOS1 through EOS5

†NK1 = NK + 1

Thus, a user working with the "two waters" option of EOS module EOS1 would specify TYPE = COM1 (or MASS, or WATE) to inject "water 1," while specification of TYPE = COM2 (or AIR, or WATR) would allow injection of "water 2."

6.2 New TOUGH2 Data Blocks (see Fig. 6)

MESHMAKER introduces parameters for internal mesh generation and processing. The MESHMAKER input has a modular structure, which is explained in section 6.3. The MESHMAKER data block has a variable number of records; its end is indicated by a blank record.

MULTI — permits the user to select options as to the number and nature of balance equations solved. Available options are different for different EOS modules (see Tables 3–6). The keyword 'MULTI' is followed by a single data record.

Record MULTI.1

Format (4I5)
NK, NEQ, NPH, NB

NK	number of mass components
NEQ	number of balance equations per grid block (usually NEQ = NK + 1, for an energy equation in addition to NK mass balance equations)
NPH	number of phases that can be present
NB	number of secondary parameters in PAR-array other than component mass fractions (NB = 6 for all presently available EOS modules)

INDOM — introduces domain-specific initial 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, X4

A set of primary variables assigned to all grid blocks in the domain specified in record INDOM.1

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'.

NOVERSION — (optional)
One record with 'NOVER' typed in columns 1–5 will suppress printing of a summary of versions and dates of the program units used in a TOUGH2 run.

ENDFILE — (optional)
One record with 'ENDFI' typed in columns 1–5 will terminate reading of the INPUT file, and will cause the flow simulation to be skipped. This option can be used when only mesh processing is desired.

6.3 Input Formats for MESHMAKER

At the present time there are three sub-modules available in MESHMAKER (see Fig. 7): 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, 1983a) 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). The meshes generated under keyword 'RZ2D' or 'XYZ' are internally written to file MESH. The 'MINC' processing operates on the data in file MESH, so that invoking the 'RZ2D' or 'XYZ' options, or assignment of 'ELEME' and 'CONNE' blocks in the INPUT file must precede the MESHMAKER/MINC data. We shall now separately describe the preparation of input data for the three MESHMAKER sub-modules.

Generation of radially symmetric grids

RZ2D
(or RZ2DL) — invokes generation of a radially symmetric mesh.

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 5.3). Assignment of inactive elements would be made by using a text editor on the RZ2D-generated ''MESH'' file, and moving groups of elements towards the end of the 'ELEME' block, past a ''dummy'' element with zero volume. '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.

MESHMAKER - Two-dimensional R-Z Grids

MESHMAKER							
RZ2D							
RADII NRAD							
RC(1)	RC(2)	RC(3)			RC(NRAD)		
EQUID NEQU		DR					
LOGAR NLOG		RLOG	DR				
LAYER NLAY							
H(1)	H(2)	H(3)			H(NLAY)		

MESHMAKER-Rectilinear Grids

MESHMAKER							
XYZ DEG							
NX NO1	DEL						
DX(1)	DX(2)	DX(3)			DX(NO1)		
NX NO2	DEL						
DX(NO1+1)	DX(NO1+2)					DX(NO1+NO2)	
NY NO	DEL						
DY(1)	DY(2)	DY(3)			DY(NO)		
NZ NO	DEL						
DZ(1)	DZ(2)	DZ(3)			DZ(NO)		

MESHMAKER - MINC Processing for Fractured Media

MESHMAKER								
MINC PART	TYPE		DUAL					
J NVOL	WHERE	PAR(1)	PAR(2)	PAR(3)	PAR(4)	PAR(5)	PAR(6)	PAR(7)
VOL(1)		VOL(2)	VOL(3)					
						VOL(NVOL)		

Figure 7. Input formats for MESHMAKER module.

Record RADII.1

Format(A5)
NRAD
NRAD number of radius data that will be read.

Record RADII.2

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(A5, 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. (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).

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° < DEG < 90° 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 a 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, 1983a), 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. See Appendix C for further discussion.

PART is the first keyword following 'MINC'; it introduces 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.

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) 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, 1983a).

DUAL 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 ($J \leq 36$).

NVOL total number of explicitly provided volume fractions ($NVOL \leq J$). If $NVOL < J$, the volume fractions with indices NVOL+1, ..., 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.

7. Sample Problems

7.1 Problem No. 1 – Code Demonstration and Comparison with TOUGH

This problem is identical to sample problem 1 from the TOUGH User's Guide, and can serve as a check on proper installation of TOUGH2 as well as for cross-referencing to TOUGH. It involves a number of one- and two-element subproblems, which are entirely independent of each other (no flow connections between subproblems), except that being run together they all must go through the same sequence of time steps. The sub-problems perform flow and/or injection and withdrawal of water, air, and heat, with highly non-linear phase and component (dis-)appearances that engage some subtle numerical procedures. A more detailed description is available in the TOUGH User's Guide. The input file for running with the EOS3 fluid properties module is almost completely identical to that of TOUGH sample problem 1, with a few MOP-parameters (first record in data block PARAM) set differently because of different defaults in TOUGH2 (see Fig. 8). The differences are: MOP(14) = 2, to suppress new matrix decomposition when the linear equation solver MA28 encounters a small pivot; MOP(17) = 7, to apply a scaling to the linear equation matrix; MOP(18) = 1, to compute fluid densities at grid block interfaces by averaging; and MOP(19) = 1, to permit initialization of the EOS3 module with TOUGH-style primary variables of (P,T,X) for single phase, (P,S,T) for two-phase.

Figure 9 gives some printed output. The results are virtually identical to TOUGH, as they should be. Minor differences occur in the maximum residuals during the iteration process. These residuals, being computed as {left hand side} - {right hand side} of the governing balance equations (see Eq. B.6), involve severe numerical cancellations and therefore constitute a very sensitive check on the numerics. Time step 2 also produces results closely identical to those of TOUGH (not shown), but subsequently TOUGH2 takes different time steps because of different default settings.

```
*SAM1* CODE DEMONSTRATION: PHASE TRANSITIONS, COMPONENT (DIS-)APPEARANCES
     THIS INPUT FILE IS SLIGHTLY DIFFERENT FROM THAT FOR SAMPLE PROBLEM 1
     OF TOUGH. THESE COMMENTS ARE INSERTED TO NOTE THE DIFFERENCES. KEYWORDS
     ARE FOLLOWED BY A COLUMN COUNTER TO FACILITATE PROPER ALIGNMENT OF DATA.
     A FEW MOP-OPTIONS ARE SET DIFFERENTLY (SEE SECTION 7.1 IN TOUGH2 GUIDE).
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
TRANS          2650.       .50    1.E-14                          2.10     1000.
SHOME    2     2650.       .50    1.E-14                          2.10     1000.
                           1.8        .1
    3            .40       .10
    1           1.E5        .2        1.

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   2   4       110003010000000200711
                   -1.                    F   1
      1.E2      5.E3

                45.E5                   .5                 250.
RPCAP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    3     .30       .05
    1              1.
TIMES----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    1    3                 2.E3
      1.E3
ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
F   1    9    1TRANS       10.
SHO 1    9    1SHOME       10.
SHO11    1    1SHOME      1.E4

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

INCON----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
F   1
                1.E5                20.                  1.
F   2
                1.E6               170.                  0.
F   3
                1.E5               .001                99.5
F   4
               99.E5               .999                310.
F   5
                1.E6               100.                  0.
F   6
               10.E6               100.                  1.
F   7
                1.E5                20.                  0.
F   8
                1.E7               300.                  0.
F   9
                1.E5                .99                 90.
F  10
               40.E5               280.                  0.
SHO11
               50.E5               240.
SHO12
               40.E5               100.

GENER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
F   7AIR                         AIR       5.E-3   9.882E4
F   8WEL                         MASS     -1.5E-2
F   9HOT                         HEAT       2.E6
F  10COL                         HEAT      -5.E5
SHO 1P   1                       MASS1       -1.      1.E6
SHO 2P   2                       FUNY        -1.
SHO 3P   3    2    1    1    4   MASS
            0.          1.E2          2.E2          4.E3
          -0.1          -0.2          -0.3          -1.1
SHO 6P   6    2    1    0    4   MASS1
            0.          1.E2          2.E2          4.E3
          -0.1          -0.2          -0.3          -1.1
          1.E6          2.E6          3.E6         1.1E7
SHO 9P   9                   1    DELV     1.E-12       1.E6
SHO10P  10                   4    WATE1
            0.          1.E2          2.E2          3.E3
          1.1           1.0           0.9           0.1
          1.E6         1.2E6         1.4E6         3.0E6
SHO11WEL00                   2    DELV     1.E-12                1.E2
SHO12WEL00                        DELV     2.E-12       1.E6     1.E2

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

Figure 8. TOUGH2 input file for sample problem 1 – code demonstration.

```
1********** VOLUME- AND MASS-BALANCES ************************************************************************************

********** [KCYC,ITER] = [   0,  0] *****                                THE TIME IS  0.          SECONDS, OR  0.          DAYS

PHASE VOLUMES IN PLACE
GAS  0.49950e+02 M**3;   LIQUID  0.10050e+05 M**3

MASS IN PLACE
GAS  0.14263e+04 KG;     LIQUID  0.89195e+07 KG;    AIR  0.56359e+03 KG;    VAPOR  0.86436e+03 KG;   LIQUID WATER  0.89195e+07 KG

*****************************************************************************************************************************

...ITERATING...  AT [   1,  1] --- DELTEX = 0.100000e+03   MAX. RES. = 0.393471e+01  AT ELEMENT SHO 9  EQUATION   2
$$$$$$$$$$$$$ LIQUID PHASE EVOLVES AT ELEMENT *F   1*  $$$$$      PS = 0.120559e+05  PSAT = 0.233989e+04
$$$$$$$$$$$$$$$$ GAS PHASE EVOLVES AT ELEMENT *F   5*  $$$$$    XAIR = 0.679569e-03    PX = 0.240509e+07   PG = 0.433013e+07
$$$$$$$$$$$$$$$$ GAS PHASE EVOLVES AT ELEMENT *F   7*  $$$$$    XAIR = 0.100168e-03    PX = 0.320895e+06   PG = 0.625505e+06
...ITERATING...  AT [   1,  2] --- DELTEX = 0.100000e+03   MAX. RES. = 0.863043e+00  AT ELEMENT SHO 9  EQUATION   2
...ITERATING...  AT [   1,  3] --- DELTEX = 0.100000e+03   MAX. RES. = 0.803526e-02  AT ELEMENT SHO 9  EQUATION   2
F   1(   1,  4) ST = 0.100000e+03 DT = 0.100000e+03 DX1= 0.234670e+04 DX2= 0.999993e+01 T =  20.006 P =   102347. S = 0.999928e+00

1*SAM1* CODE DEMONSTRATION: PHASE TRANSITIONS, COMPONENT (DIS-)APPEARANCES

        OUTPUT DATA AFTER (   1,  4)-2-TIME STEPS                                                THE TIME IS  0.11574e-02 DAYS

0@@@
```

7.2 Problem No. 2 – Heat Pipe in Cylindrical Geometry

Heat pipes are systems in which an efficient heat transfer takes place by means of a liquid-vapor counterflow process, with vaporization and condensation occurring at the hot and cold ends, respectively. Heat pipe processes occur naturally on a large scale (kilometers) in two-phase geothermal reservoirs, and they may be induced artificially if heat-generating nuclear waste packages are emplaced above the water table in partially saturated geologic formations.

The present problem models such high-level nuclear waste emplacement in an approximate way. The TOUGH2 input file for use with the EOS3 fluid property module is shown in Fig. 10. It specifies a cylindrical heater of 0.3 m radius and 4.5 m height, that provides a constant output of 3 kW into a porous medium with uniform initial conditions of 18°C temperature, 1 bar pressure, and 20% gas saturation. The MESHMAKER module is used to generate a one-dimensional radial grid of 120 active elements extending to a radius of 10,000 m (practically infinite for the time scales of interest here), with an additional inactive element of zero volume representing constant boundary conditions. Properly speaking, the problem represents one unit of an infinite linear string of identical heaters; if a single heater were to be modeled, important end effects would occur at the top and bottom, and a two-dimensional R-Z grid would have to be used.

Most of the formation parameters are identical to data used in previous modeling studies of high-level nuclear waste emplacement at Yucca Mountain (Pruess *et al.*, 1990). As we do not include fracture effects in the present simulation, heat pipe effects would be very weak at the low rock matrix permeabilities (of order 1 microdarcy) encountered at Yucca Mountain. To get a more interesting behavior, we have arbitrarily increased absolute permeability by something like a factor 10,000, to 20 millidarcy, and for consistency have reduced capillary pressures by a factor $(10{,}000)^{1/2} = 100$ in comparison to typical Yucca Mountain data.

```
*RHP* 1-D RADIAL HEAT PIPE
MESHMAKER1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
RZ2D
RADII
    1
        0.
EQUID
    1              .3
LOGAR
   99            1.E2
LOGAR
   20            1.E4
EQUID
    1             0.0
LAYER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    1
       4.5

ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
POMED           2550.        .10   20.E-15   20.E-15   20.E-15        2.0     800.0

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   2 250     2500000300000002 47 1
           3.15576E8       -1.
      1.E3      9.E3      9.E4      4.E5
     1.E-5     1.E00                                  1.E-7
               1.E5                 0.20                 18.
RPCAP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    7        0.45000    9.6E-4        1.
    7        0.45000    1.0E-3   8.0E-05      5.E8        1.
TIMES----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    3
 3.15576E7  1.2559E8 3.15576E8
INCON----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8

GENER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A1  1HTR 1                          HEAT       3.E3

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

Figure 10. Input file for problem 2 – heat pipe in cylindrical geometry.

The reason for choosing a constant rate of heat generation rather than accounting for the natural decline of high-level wastes is that this way the heat pipe problem admits a semi-analytical solution in terms of the similarity variable $r/(t)^{1/2}$ (O'Sullivan, 1981; Doughty and Pruess, 1990): Under the stated conditions, the partial differential equations for this complex transient two-phase flow problem can be transformed into a set of ordinary differential equations in the variable $r/(t)^{1/2}$, which can be easily solved to any degree of accuracy desired by means of one-dimensional numerical integration. Comparison with the similarity solution affords a rather comprehensive code verification, as all of the non-linearities of two-phase flow behavior (relative permeability and capillary pressure) and of fluid and heat flow coupling are rigorously described by the similarity solution.

Figure 11 shows profiles of temperature, gas phase pressure, liquid saturation, and air mass fraction as a function of $z = \log[r/(t)^{1/2}]$. The dotted line labeled "coarse mesh" represents the conditions obtained from TOUGH2 with the input file of Fig. 10 after ten years (3.15576×10^8 sec). In order to examine space discretization effects, two additional runs were made, labeled "medium mesh" and "fine mesh," respectively. The MESH-MAKER input data for these runs are shown in Fig. 12; otherwise the input file was the same as in Fig. 10 (except for unimportant differences in printout times). Figure 13 compares the fine mesh results (points) with the exact similarity solution as calculated by C. Doughty (Doughty and Pruess, 1991). The agreement is excellent.

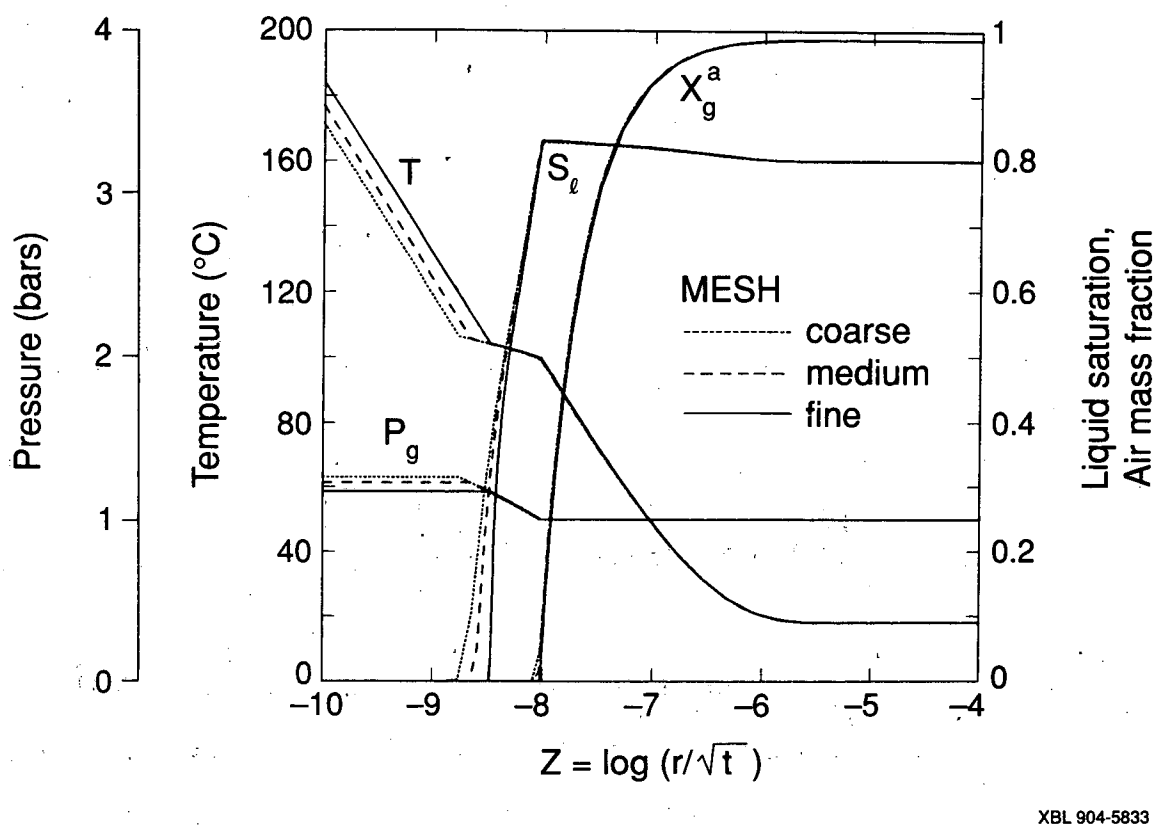


Figure 11. Profiles of temperature, pressure, liquid saturation, and air mass fraction for problem 2.

```
MESHMAKER1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
RZ2D
RADII
    3
        Ø.          .3          .4
LOGAR
   99             1.E2
LOGAR
   2Ø             1.E4
EQUID
    1              Ø.Ø
LAYER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    1
       4.5
     ----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
```

```
MESHMAKER1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
RZ2D
RADII
    2
        Ø.          .3
EQUID
    6               .2
EQUID
   7Ø              .Ø3
LOGAR
   1Ø              1Ø.
LOGAR
   1Ø             1.E2
LOGAR
   1Ø             1.E4
EQUID
    1              Ø.Ø
LAYER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    1
       4.5
     ----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
```

Figure 12. MESHMAKER input for finer gridding in problem 2.

Figure 14 reproduces the summary of program units used in sample problem 2, as printed at the end of the output file. Besides documenting the versions and dates of subroutines, this listing is instructive as a record of the calling sequence of program units during execution.

The input file as shown in Fig. 10 can also be executed with the EOS4 fluid property module, which includes vapor pressure lowering effects. Part of the output generated

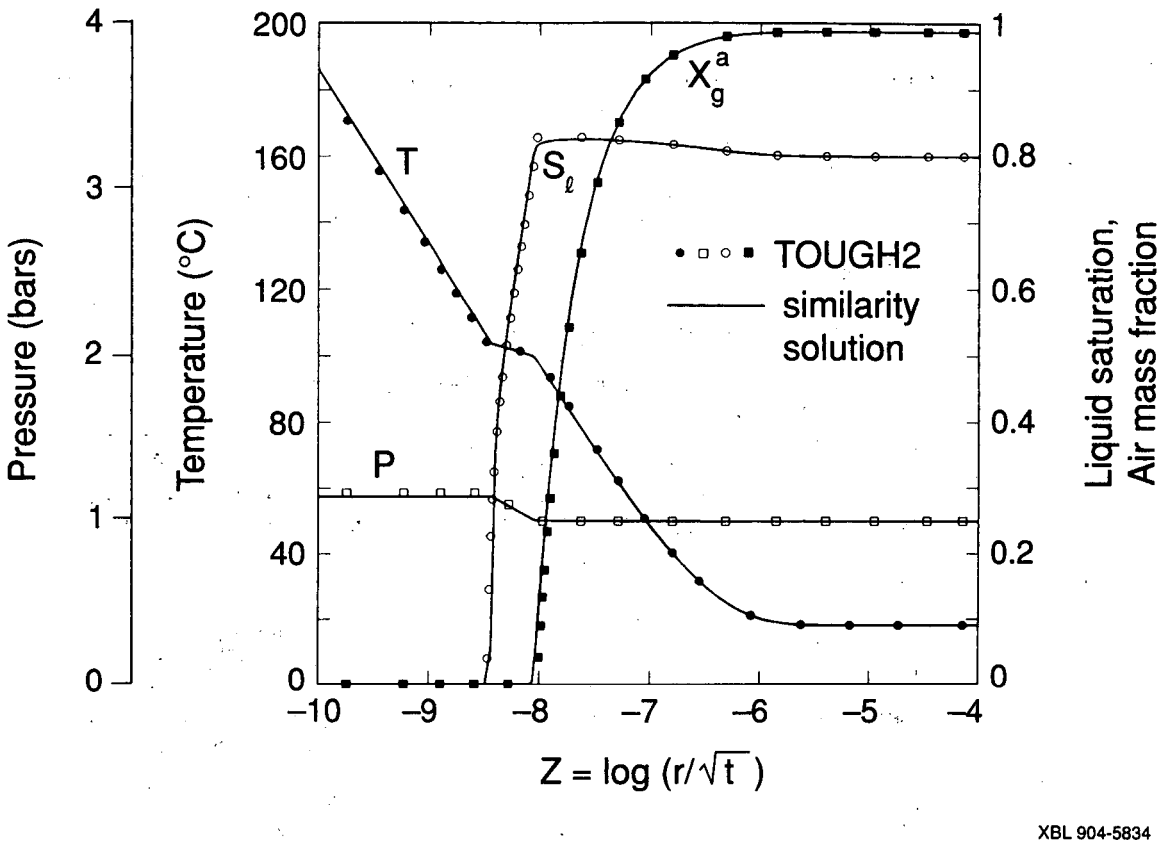


Figure 13. Comparison of TOUGH2 results with similarity solution.

when running with EOS4 is shown in Fig. 15; this can serve as a benchmarking reference for the EOS4 module. The results are actually quite similar to those obtained with EOS3, except that because of very strong vapor pressure lowering effects, drying-out near the heater is slowed.

```
!*****************************************************************************************************************
*                                                                                                                *
*                                          SUMMARY OF PROGRAM UNITS USED                                         *
*                                                                                                                *
*****************************************************************************************************************
```

UNIT	VERSION	DATE	COMMENTS
IO	1.0	15 APRIL 1991	OPEN FILES *VERS*, *MESH*, *INCON*, *GENER*, *SAVE*, *LINEQ*, AND *TABLE*
TOUGH2	1.0	29 MARCH 1991	MAIN PROGRAM
INPUT	1.0	11 APRIL 1991	READ ALL DATA PROVIDED THROUGH FILE *INPUT*
MESHM	1.0	24 MAY 1990	EXECUTIVE ROUTINE FOR INTERNAL MESH GENERATION
RZ2D	1.0	9 APRIL 1991	CALCULATE 2-D R-Z MESH FROM INPUT DATA
WRZ2D	1.0	26 MARCH 1991	WRITE DATA FOR 2-D R-Z MESH ON FILE *MESH*
PRZ2D	1.0	27 MARCH 1991	MAKE STRUCTURED PRINTOUT OF 2-D R-Z MESH
FLOP	1.0	11 APRIL 1991	CALCULATE NUMBER OF SIGNIFICANT DIGITS FOR FLOATING POINT ARITHMETIC
RFILE	1.0	23 APRIL 1991	INITIALIZE DATA FROM FILES *MESH* OR *MINC*, *GENER*, AND *INCON*
CYCIT	1.0	5 MARCH 1991	EXECUTIVE ROUTINE FOR MARCHING IN TIME
EOS	1.0	28 MARCH 1991	*EOS3* ... THERMOPHYSICAL PROPERTIES MODULE FOR WATER/AIR
SAT	1.0	22 JANUARY 1990	STEAM TABLE EQUATION: SATURATION PRESSURE AS FUNCTION OF TEMPERATURE
COWAT	1.0	22 JANUARY 1990	LIQUID WATER DENSITY AND INT. ENERGY AS FUNCTION OF TEMPERATURE AND PRESSURE
SUPST	1.0	29 JANUARY 1990	VAPOR DENSITY AND INTERNAL ENERGY AS FUNCTION OF TEMPERATURE AND PRESSURE
VISW	1.0	22 JANUARY 1990	VISCOSITY OF LIQUID WATER AS FUNCTION OF TEMPERATURE AND PRESSURE
VISCO	1.0	1 FEBRUARY 1990	CALCULATE VISCOSITY OF VAPOR-AIR MIXTURES
COVIS	1.0	1 FEBRUARY 1990	COEFFICIENT FOR GAS PHASE VISCOSITY CALCULATION
VISS	1.0	22 JANUARY 1990	VISCOSITY OF VAPOR AS FUNCTION OF TEMPERATURE AND PRESSURE
RELP	1.0	25 JANUARY 1990	LIQUID AND GAS PHASE RELATIVE PERMEABILITIES AS FUNCTIONS OF SATURATION
PCAP	1.0	4 MARCH 1991	CAPILLARY PRESSURE AS FUNCTION OF SATURATION
BALLA	1.0	5 MARCH 1991	PERFORM SUMMARY BALANCES FOR VOLUME, MASS, AND ENERGY
TSTEP	1.0	4 MARCH 1991	ADJUST TIME STEPS TO COINCIDE WITH USER-DEFINED TARGET TIMES
MULTI	1.0	30 MARCH 1991	ASSEMBLE ALL ACCUMULATION AND FLOW TERMS
QU	1.0	22 JANUARY 1990	ASSEMBLE ALL SOURCE AND SINK TERMS
LINEQ	1.0	22 JANUARY 1990	INTERFACE FOR THE LINEAR EQUATION SOLVER MA28
MC19A			HARWELL SUBROUTINE FOR SCALING MATRIX
CONVER	1.0	4 MARCH 1991	UPDATE PRIMARY VARIABLES AFTER CONVERGENCE IS ACHIEVED
PP	1.0	1 FEBRUARY 1990	CALCULATE VAPOR PRESSURE, DENSITY, INT. ENERGY AS F(P,T,X)
OUT	1.0	5 MARCH 1991	PRINT RESULTS FOR ELEMENTS, CONNECTIONS, AND SINKS/SOURCES
WRIFI	1.0	22 JANUARY 1990	AT THE COMPLETION OF A TOUGH2 RUN, WRITE PRIMARY VARIABLES ON FILE *SAVE*

```
*****************************************************************************************************************
```

Figure 14. Calling sequence of program units in problem 2.

```
1********** VOLUME- AND MASS-BALANCES ************************************************************************************

 ********** [KCYC,ITER] = [   0,  0] *****                                THE TIME IS  0.         SECONDS, OR  0.         DAYS

 PHASE VOLUMES IN PLACE
 GAS  0.28274e+08;  LIQUID  0.11310e+09

 MASS IN PLACE
 GAS  0.33561e+08;  LIQUID  0.11295e+12;  AIR  0.34905e+08;  VAPOR  0.43437e+06; LIQUID WATER  0.11295e+12

*****************************************************************************************************************************

 A1  1(   1,  3) ST = 0.100000e+04 DT = 0.100000e+04 DX1= 0.146786e+02 DX2= -.156853e-03 T =  19.065 P =   100015. S = 0.199843e+00
 A1  1(   2,  4) ST = 0.100000e+05 DT = 0.900000e+04 DX1= 0.744626e+01 DX2= -.128979e-02 T =  27.246 P =   100022. S = 0.198553e+00
 A1  1(   3,  6) ST = 0.100000e+06 DT = 0.900000e+05 DX1= 0.980597e+01 DX2= -.352032e-02 T =  63.142 P =   100032. S = 0.195033e+00
 MATRIX NUMERICALLY SINGULAR --------------------------  PERFORM NEW DECOMPOSITION
 A1  1(   4,  8) ST = 0.500000e+06 DT = 0.400000e+06 DX1= 0.455759e+04 DX2= 0.117922e+00 T = 100.885 P =   104590. S = 0.312955e+00
 A1  1(   5,  6) ST = 0.900000e+06 DT = 0.400000e+06 DX1= 0.130323e+05 DX2= 0.228709e+00 T = 104.237 P =   117622. S = 0.541664e+00
 A1  1(   6,  5) ST = 0.130000e+07 DT = 0.400000e+06 DX1= 0.612647e+04 DX2= 0.670902e-01 T = 105.704 P =   123748. S = 0.608755e+00
 A1  1(   7,  4) ST = 0.170000e+07 DT = 0.400000e+06 DX1= 0.427324e+04 DX2= 0.345346e-01 T = 106.692 P =   128022. S = 0.643289e+00
 A1  2(   8,  7) ST = 0.250000e+07 DT = 0.800000e+06 DX1= 0.210099e+04 DX2= 0.573465e-01 T = 100.223 P =   102141. S = 0.242792e+00
 A1  1(   9,  5) ST = 0.330000e+07 DT = 0.800000e+06 DX1= 0.504930e+04 DX2= 0.720171e-01 T = 109.160 P =   139207. S = 0.766330e+00
 A1  1(  10,  5) ST = 0.410000e+07 DT = 0.800000e+06 DX1= 0.335713e+04 DX2= 0.523432e-01 T = 109.872 P =   142564. S = 0.818673e+00
 A1  2(  11,  4) ST = 0.490000e+07 DT = 0.800000e+06 DX1= 0.297170e+04 DX2= 0.479475e-01 T = 103.475 P =   114541. S = 0.498440e+00
 A1  3(  12,  7) ST = 0.650000e+07 DT = 0.160000e+07 DX1= 0.173144e+04 DX2= 0.488149e-01 T = 100.123 P =   101772. S = 0.229192e+00
 A1  1(  13,  6) ST = 0.810000e+07 DT = 0.160000e+07 DX1= 0.357581e+04 DX2= 0.430044e-01 T = 112.063 P =   153039. S = 0.949414e+00
 A1  1(  14,  6) ST = 0.970000e+07 DT = 0.160000e+07 DX1= 0.259115e+04 DX2= 0.232400e-01 T = 112.672 P =   155630. S = 0.972654e+00
 A1  1(  15,  6) ST = 0.113000e+08 DT = 0.160000e+07 DX1= 0.208588e+04 DX2= 0.120255e-01 T = 113.286 P =   157716. S = 0.984679e+00
 A1  1(  16,  6) ST = 0.129000e+08 DT = 0.160000e+07 DX1= 0.195398e+04 DX2= 0.805740e-02 T = 114.363 P =   159670. S = 0.992737e+00
 A1  1(  17,  8) ST = 0.145000e+08 DT = 0.160000e+07 DX1= 0.920061e+03 DX2= 0.375059e-02 T = 116.820 P =   160590. S = 0.996487e+00
 A1  1(  18,  8) ST = 0.161000e+08 DT = 0.160000e+07 DX1= -.107512e+04 DX2= 0.143879e-02 T = 122.868 P =   159515. S = 0.997926e+00
 A1  1(  19,  8) ST = 0.177000e+08 DT = 0.160000e+07 DX1= -.443019e+04 DX2= 0.522534e-03 T = 134.409 P =   155085. S = 0.998449e+00
 A1  1(  20,  7) ST = 0.193000e+08 DT = 0.160000e+07 DX1= -.711043e+04 DX2= 0.188214e-03 T = 148.170 P =   147974. S = 0.998637e+00
 A1  1(  21,  6) ST = 0.209000e+08 DT = 0.160000e+07 DX1= -.659648e+04 DX2= 0.737170e-04 T = 159.229 P =   141378. S = 0.998711e+00
 A1  1(  22,  5) ST = 0.225000e+08 DT = 0.160000e+07 DX1= -.998681e+03 DX2= 0.150070e-04 T = 162.564 P =   140379. S = 0.998726e+00
 A1  2(  23,  5) ST = 0.241000e+08 DT = 0.160000e+07 DX1= 0.112066e+04 DX2= 0.133116e-01 T = 109.770 P =   141136. S = 0.975460e+00
 A1  2(  24,  5) ST = 0.257000e+08 DT = 0.160000e+07 DX1= 0.924703e+03 DX2= 0.865090e-02 T = 110.127 P =   142061. S = 0.984111e+00
 A1  1(  25,  6) ST = 0.273000e+08 DT = 0.160000e+07 DX1= 0.761429e+03 DX2= 0.965991e-06 T = 164.185 P =   142826. S = 0.998729e+00
 A1  2(  26,  6) ST = 0.289000e+08 DT = 0.160000e+07 DX1= 0.578302e+03 DX2= 0.350226e-02 T = 111.219 P =   143404. S = 0.993169e+00
 A1  2(  27,  6) ST = 0.305000e+08 DT = 0.160000e+07 DX1= 0.318358e+03 DX2= 0.255778e-02 T = 112.510 P =   143722. S = 0.995726e+00
 A1  2(  28,  5) ST = 0.315576e+08 DT = 0.105760e+07 DX1= -.231627e+03 DX2= 0.105436e-02 T = 113.937 P =   143491. S = 0.996781e+00

*****************************************************************************************************************************
```

Figure 15. Selected output for problem 2 run with EOS4.

```
1          *RHP* 1-D RADIAL HEAT PIPE

           OUTPUT DATA AFTER (  28,  5)-2-TIME STEPS                                                    THE TIME IS  0.36525e+03 DAYS

0@@@
```

7.3 Problem No. 3 - Heat Sweep in a Vertical Fracture

In many geothermal fields there is evidence of rapid migration of injected fluids along "preferential flow paths," presumably along fractures. The present problem is designed to study thermal interference along such paths, by modeling nonisothermal injection into and production from a single vertical fracture, as illustrated in Fig. 16 (from Pruess and Bodvarsson, 1984). The fracture is bounded by semi-infinite half-spaces of impermeable rock, which provide a conductive heat supply. Initial temperature is 300°C throughout. Water at 100°C temperature is injected at one side of the fracture at a constant rate of 4 kg/s. Production against a specified wellbore pressure occurs at the other side, at a distance of 240 m from the injection point. Problem parameters are given in Table 11, and the TOUGH2 input file for injecting at point I′ and producing at point P is shown in Fig. 17.

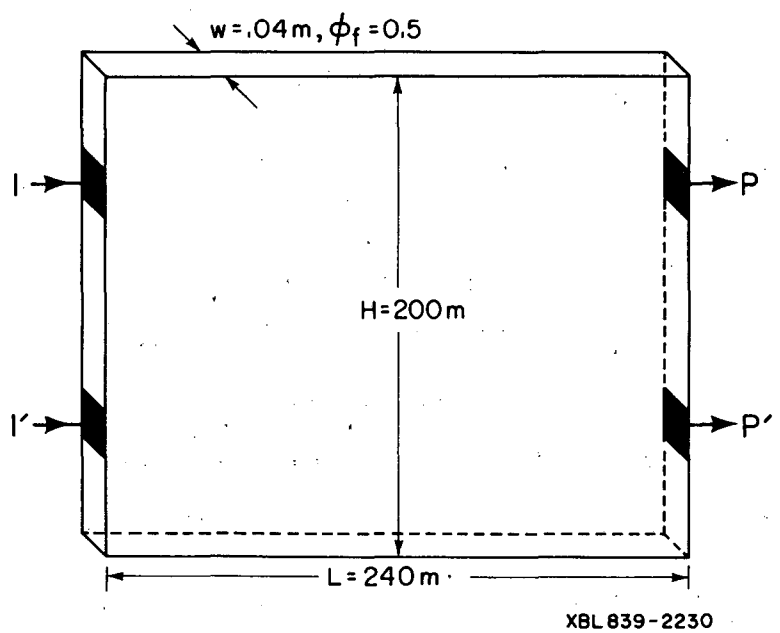


Figure 16. Schematic diagram of injection-production system in vertical fracture. I and I′ are injection points, P and P′ production points.

Table 11. Parameters for Fracture Flow Problem

Rock	
Thermal conductivity	2.1 W/m°C
Specific heat	1000 J/kg°C
Density	2650 kg/m^3
Permeability	0
Fracture	
Height	200 m
Length	240 m
Aperture	0.04 m
Permeability	200×10^{-12} m^2 ($\approx$ 200 darcy)
Porosity	50 %
Initial Conditions	
Temperature	300°C
Pressure	hydrostatic profile
Average pressure	100 bar
Injection	
Enthalpy	4.2×10^5 J/kg (appr. 100°C)
Rate	4 kg/s
Production	
Productivity index	4×10^{-12} m^3
Flowing pressure	96.5 bar

A special feature of the problem is that the semi-analytical method is used to describe heat conduction in the confining layers (see Section 5.4), reducing the dimensionality of the problem from 3-D to 2-D. Water remains in single-phase liquid conditions throughout, so that no data block 'RPCAP' for relative permeabilities and capillary pressures is needed.

The problem uses the EOS1 fluid property module, and is run in three separate segments. A first run performs mesh generation only, using the MESHMAKER/XYZ module. For this run, the data records from 'MESHM' through 'ENDFI' in the input file are inserted right behind the first record with the problem title. The mesh consists of 12 horizontal by 10 vertical blocks of 20 m $\times$ 20 m. Ordinarily, we would specify NX = 12

```
*RVF* - VERTICAL FRACTURE
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
FRACT          2650.        .50  200.E-12  200.E-12  200.E-12      0.00     1000.
CONBD          2650.        .00    0.E-12    0.E-12    0.E-12       2.1     1000.

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   1  40      991000000000000021
           1.57788E8       -1.          A1312            9.81
      1.E2       9.E2      9.E3       9.E4       9.E5      2.E6       5.E6
     1.E-5
                100.E5                 300.
GENER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
A18 1INJ 1                          MASS          4.      4.2E5
A1312PRO 1                          DELV      4.E-12    9.65E6

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

MULTI----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    1    1    2    6
ENDCY----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
MESHMAKER1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
XYZ
       90.
NX      12        20.
NY      10        20.
NZ       1        .04


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

Figure 17. Input file for problem 3 – heat sweep in a vertical fracture.

and NZ = 10 to make such a mesh; however, special considerations arise here because we desire appropriate surface areas for heat conduction to be placed in the MESH file. By default, in the MESHMAKER/XYZ module the interface areas with impermeable confining beds are always taken to be in the X-Y plane, so that in a mesh with vertical Z-axis the interface areas for conductive heat transfer will be assigned to the top and bottom boundaries. To properly assign the desired lateral heat transfer areas, the mesh is generated as an X-Y mesh (NX = 12, NY = 10, NZ = 1), and the Y-axis is specified to make an angle of 90° with the horizontal, i.e., to point in the vertical direction. The MESHMAKER input terminates on 'ENDFI', to bypass the flow simulation and to limit processing to mesh generation only. Figure 18 shows the mesh pattern printout generated by TOUGH2. The elements 'A13 1' and 'A18 1' correspond to the injection points I and I′, respectively, in Fig. 16, while 'A1312' and 'A1812' correspond to the production

points P and P′. The MESH file is then edited, and a "dummy" element of zero volume is appended at the end of the ELEME block, to provide the thermal data for the conductive boundaries. This element belongs to domain # 2 (specify MA2 = 2); and is initialized with the default conditions of 300°C temperature.

```
1*************************************************************************************************************
*                    CARTESIAN MESH WITH NX*NY*NZ =   12 *  10 *   1  GRID BLOCKS                                *
*************************************************************************************************************
*                                                                                                             *
*                    THE MESH WILL BE PRINTED AS SLICES FOR K = 1 TO K = NZ =   1                              *
*                                                                                                             *
*                    IN EACH MESH SLICE, ROWS WILL GO FROM  J = 1 TO J = NY =  10                              *
*                                                                                                             *
*                    IN EACH ROW, COLUMNS WILL GO FROM      I = 1 TO I = NX =  12                              *
*                                                                                                             *
*************************************************************************************************************

SLICE WITH K =   1

  COLUMN I =  1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
ROWS
 J =    1   A11 1 A11 2 A11 3 A11 4 A11 5 A11 6 A11 7 A11 8 A11 9 A1110 A1111 A1112
 J =    2   A12 1 A12 2 A12 3 A12 4 A12 5 A12 6 A12 7 A12 8 A12 9 A1210 A1211 A1212
 J =    3   A13 1 A13 2 A13 3 A13 4 A13 5 A13 6 A13 7 A13 8 A13 9 A1310 A1311 A1312
 J =    4   A14 1 A14 2 A14 3 A14 4 A14 5 A14 6 A14 7 A14 8 A14 9 A1410 A1411 A1412
 J =    5   A15 1 A15 2 A15 3 A15 4 A15 5 A15 6 A15 7 A15 8 A15 9 A1510 A1511 A1512
 J =    6   A16 1 A16 2 A16 3 A16 4 A16 5 A16 6 A16 7 A16 8 A16 9 A1610 A1611 A1612
 J =    7   A17 1 A17 2 A17 3 A17 4 A17 5 A17 6 A17 7 A17 8 A17 9 A1710 A1711 A1712
 J =    8   A18 1 A18 2 A18 3 A18 4 A18 5 A18 6 A18 7 A18 8 A18 9 A1810 A1811 A1812
 J =    9   A19 1 A19 2 A19 3 A19 4 A19 5 A19 6 A19 7 A19 8 A19 9 A1910 A1911 A1912
 J =   10   A1A 1 A1A 2 A1A 3 A1A 4 A1A 5 A1A 6 A1A 7 A1A 8 A1A 9 A1A10 A1A11 A1A12

*************************************************************************************************************

MESH GENERATION COMPLETE --- EXIT FROM MODULE *MESHMAKER*
```

Figure 18. TOUGH2 printout of mesh pattern in problem 3.

The next processing step calculates a hydrostatic pressure equilibrium in the fracture under isothermal conditions. This calculation uses the modified MESH file obtained above, and requires several small modifications in the input file of Fig. 17. The time step counter MCYC is changed from 40 to 4, as 4 time steps are sufficient to obtain an accu-

rate gravity equilibrium. The generation items need to be removed, which can be achieved by deleting them from the input file or, more simply, by changing the 'A' in the host element names to 'B', so that generation will then occur in elements B18 1 and B1312, which are "unknown" (not present in the MESH data) and hence will be ignored. The convergence tolerance RE1 is changed from 1×10^{-5} to 1×10^{-8}, to achieve a tighter control on gravity equilibrium. The 'ENDCY' statement preceding the 'INCON' record is removed, to enforce default initial conditions by way of an empty INCON data block, and to engage the MULTI data block with specifications of (NK,NEQ,NPH,NB) = (1,1,2,6) for an isothermal calculation (mass balance only). We also set MOP(15) = 0, to disengage the semi-analytical heat exchange calculation. Gravity equilibration results in a pressure trend ranging from 106.34175 bars in the bottom row of grid blocks (A1A 1, A1A 2, ..., A1A12), to 93.70950 bars in the top row (A11 1, ..., A1112).

The subsequent production/injection run uses the input file exactly as given in Fig. 17, with the MESH file as used in the gravity equilibration. The SAVE file produced by the gravity equilibration run is renamed file INCON, and used for initialization, after removing the last two records and replacing them with a blank record, to reset time step and simulation time counters to zero. The specified maximum time of 1.57788×10^8 seconds (5 years) is reached after 37 time steps; at this time production occurs with a rate of 3.9998 kg/s and an enthalpy of 0.87031 MJ/kg; temperature in the producing element is 203.25°C. A plot of the transient temperature changes at the producing element is given in Fig. 19.

The fracture production/injection problem lends itself to several interesting extensions and variations. These can be implemented by means of small modifications in the input file and are mentioned here without giving calculational results. For example, the problem could be restarted with a zero injection rate, to examine the rate of temperature recovery in the production block. Initial conditions could be chosen appropriate for

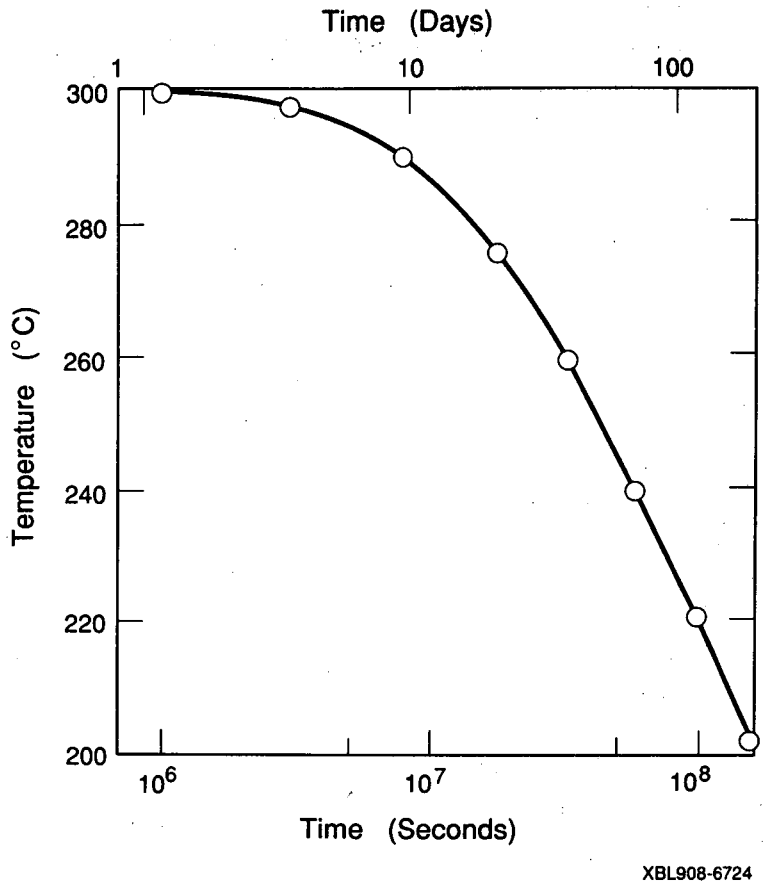


Figure 19. Produced fluid temperature versus time for vertical fracture problem.

depleted zones in vapor-dominated reservoirs, e.g., T = 240°C, P = 8 bars, to examine injection response with strong vaporization effects. Note that for injection into vapor-dominated systems, two-phase conditions will evolve and a data block 'RPCAP' with relative permeability and capillary pressure data will be required. We point out that strong grid orientation effects may arise when modeling water injection into vapor-dominated resevoirs and careful mesh design or inclusion of diffusive effects (capillary pressures) is required to obtain realistic results (Pruess, 1991). Parameters (NK,NEQ,NPH,NB) = (2,3,2,6) could be used with injection type 'COM2' instead of 'MASS', to inject ''water 2'' and thereby track the advance and arrival of injected water at the production point. The problem could also be run with the EOS2 fluid property

module, with some CO_2 initially present in the reservoir fluid, to study the changes in non-condensible gas content of produced fluids in response to injection.

7.4 Problem No. 4 - Five-Spot Geothermal Production/Injection

In geothermal reservoir development, production and injection wells are often sited in more or less regular geometric patterns. The present problem considers a ''large'' well field with wells arranged in a ''five-spot'' configuration (Fig. 20). Because of symmetry only 1/8 of the basic pattern needs to be modeled. The computational grid was generated by means of a separate preprocessor program which has not yet been integrated into the TOUGH2 package. The grid has six rows, each containing between one and eleven elements, for a total of thirty-six volume elements (see Fig. 20); for simplicity, only a single layer of 305 m thickness is modeled. The problem specifications as given in Table 12 correspond to conditions that may typically be encountered in deeper zones of hot and fairly tight fractured two-phase reservoirs (Pruess, 1983c; Pruess and Narasimhan, 1985).

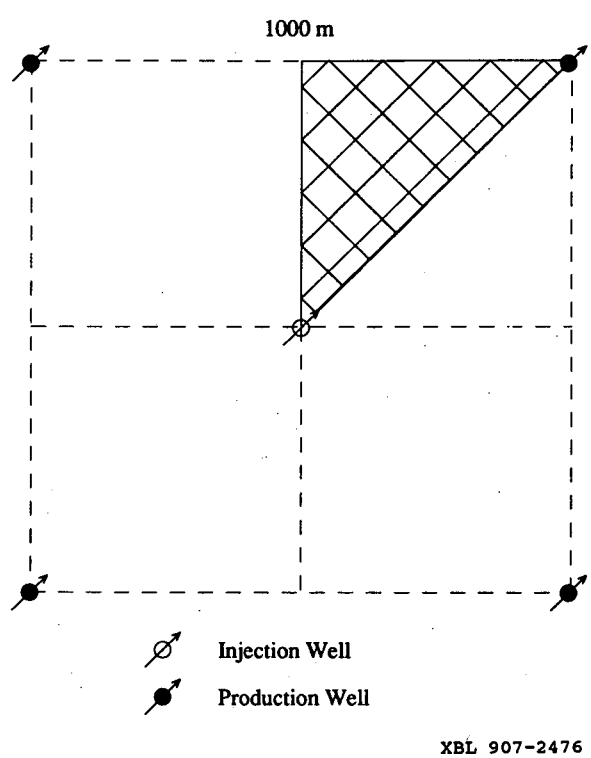


Figure 20. Five-spot well pattern with grid for modeling a 1/8 symmetry domain.

Table 12. Parameters for Five-spot Problem

Formation	
Rock grain density	2650 kg/m^3
Specific heat	1000 J/kg°C
Heat conductivity	2.1 W/m°C
Permeable volume fraction	2%
Porosity in permeable domain	50%
Impermeable blocks: cubes with side length	50 m, 250 m
Effective permeability	6.0×10^{-15} m^2
Thickness	305 m
Relative permeability: Corey curves with $S_{lr} = 0.30$, $S_{vr} = 0.05$	
Initial temperature	300°C
Initial liquid saturation	0.99
Initial pressure	85.93 bar
Production/Injection	
Pattern area	1 km^2
Distance between producers and injectors	707.1 m
Production rate*	30 kg/s
Injection rate*	30 kg/s
Injection enthalpy	500 kJ/kg

*Full well basis

The INPUT file for use with the EOS1 fluid property module (Fig. 21) models the system as a fractured medium with embedded impermeable matrix blocks in the shape of cubes (partition type 'THRED' with three equal fracture spacings). The matrix blocks were assigned a non-vanishing porosity of 10^{-10}, so that they will contain a small amount of water. This will have no noticeable impact on fluid and heat flows, but it prevents the

```
*RFP* - 36 BLOCKS PARALLEL FIVE-SPOT GRID (CF. SPE-18426)
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
POMED          2650.        .01    6.E-15    6.E-15    6.E-15        2.1     1000.
FRACT          2650.        .50    6.E-15    6.E-15    6.E-15        2.1     1000.
MATRX          2650.     1.E-10    0.E-15    0.E-15    0.E-15        2.1     1000.

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   1  99      990000000000000 470
          1.151852E9       -1. 3.15576E7 KA 1
      1.E5
      1.E-5                                                  1.E-8
                300.                  0.01                    5.E5
RPCAP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    3      .30        .05
    1                1.
TIMES----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    2    2
 1.57788E8 7.88940E8
GENER----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
 AA 1INJ 1                          MASS        3.75      5.0E5
 KA 1PRO 1                          MASS       -3.75

ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
 AA 1          POMED0.1906E+060.1250E+04          0.         0.         0.1525E+03
 BA 1          POMED0.7625E+060.5000E+04          0.7071E+020.          0.1525E+03
 CA 1          POMED0.7625E+060.5000E+04          0.1414E+030.          0.1525E+03
 DA 1          POMED0.7625E+060.5000E+04          0.2121E+030.          0.1525E+03
 EA 1          POMED0.7625E+060.5000E+04          0.2828E+030.          0.1525E+03
 FA 1          POMED0.7625E+060.5000E+04          0.3536E+030.          0.1525E+03
 GA 1          POMED0.7625E+060.5000E+04          0.4243E+030.          0.1525E+03
 HA 1          POMED0.7625E+060.5000E+04          0.4950E+030.          0.1525E+03
 IA 1          POMED0.7625E+060.5000E+04          0.5657E+030.          0.1525E+03
 JA 1          POMED0 7625E+060.5000E+04          0.6364E+030.          0.1525E+03
 KA 1          POMED0.1906E+060.1250E+04          0.7071E+030.          0.1525E+03
 BB 1          POMED0.7625E+060.5000E+04          0.7071E+020.7071E+020.1525E+03
 CB 1          POMED0.1525E+070.1000E+05          0.1414E+030.7071E+020.1525E+03
...
...
...
 GE 1          POMED0.7625E+060.5000E+04          0.4243E+030.2828E+030.1525E+03
 FF 1          POMED0.3812E+060.2500E+04          0.3536E+030.3536E+030.1525E+03
HTX00          POMED        0.

CONNE----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
 AA 1 BA 1                   10.3536E+020.3536E+020.1078E+05
 BA 1 CA 1                   10.3536E+020.3536E+020.1078E+05
 BA 1 BB 1                   20.3536E+020.3536E+020.2157E+05
 CA 1 DA 1                   10.3536E+020.3536E+020.1078E+05
 CA 1 CB 1                   20.3536E+020.3536E+020.2157E+05
...
...
...
 FE 1 GE 1                   10.3536E+020.3536E+020.2157E+05
 FE 1 FF 1                   20.3536E+020.3536E+020.2157E+05

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

MESHMAKER1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
MINC
PART THRED      DFLT
  5  40UT        50.
       .02       .08        .20        .35

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

Figure 21. Input file for problem 4 – five-spot production/injection. (Only part of ELEME and CONNE data blocks are shown.)

water mass balance equation from degenerating into the singular form 0 = 0. The MESHMAKER module is used to perform MINC-partitioning of the primary grid. The first MINC continuum, corresponding to the fracture domain, occupies a volume fraction of 0.02 and has an intrinsic porosity of 50%, for an effective fracture porosity of 1%. By inserting an 'ENDCY' record in front of the MESHMAKER data block, the MINC process can be disabled and the problem run as an effective porous medium. Figures 22 and

```
1*****************************************************************************************************************
*                MESHMAKER - MINC: GENERATE MULTIPLE INTERACTING CONTINUA MESH FOR FRACTURED MEDIUM                *
*****************************************************************************************************************

FILE *MINC* EXISTS --- OPEN AS AN OLD FILE

CHOICE OF MATRIX-MATRIX FLOW HANDLING: "DFLT "

                    THE OPTIONS ARE: "     " (DEFAULT), NO GLOBAL MATRIX-MATRIX FLOW; GLOBAL FLOW ONLY THROUGH FRACTURES
                                     "MMVER", GLOBAL MATRIX-MATRIX FLOW IN VERTICAL DIRECTION ONLY
                                     "MMALL", GLOBAL MATRIX-MATRIX FLOW IN ALL DIRECTIONS


=================== GEOMETRY DATA, NORMALIZED TO A DOMAIN OF UNIT VOLUME ========================

 CONTINUUM     IDENTIFIER      VOLUME      NODAL DISTANCE     INTERFACE AREA   INTERFACE DISTANCE
                                                                                  FROM FRACTURES

 1-FRACTURES     * *         0.20000e-01     0.
                                                                0.11760e+00        0.
 2-MATRIX        *2*         0.80000e-01     0.34984e+00
                                                                0.11111e+00        0.69967e+00
 3-MATRIX        *3*         0.20000e+00     0.97637e+00
                                                                0.93970e-01        0.26524e+01
 4-MATRIX        *4*         0.35000e+00     0.23051e+01
                                                                0.59197e-01        0.72627e+01
 5-MATRIX        *5*         0.35000e+00     0.35475e+01

=================================================================================================

READ PRIMARY MESH FROM FILE *MESH*
          THE PRIMARY MESH HAS   37 ELEMENTS (   36 ACTIVE) AND   55 CONNECTIONS (INTERFACES) BETWEEN THEM

WRITE SECONDARY MESH ON FILE *MINC*
        THE SECONDARY MESH HAS  181 ELEMENTS (  180 ACTIVE) AND  199 CONNECTIONS (INTERFACES) BETWEEN THEM

*****************************************************************************************************************

MESH GENERATION COMPLETE --- EXIT FROM MODULE *MESHMAKER*
```

Figure 22. Output from MINC processing of problem 4.

23 show part of the printout of the MINC simulation run, and Fig. 24 gives temperature profiles along the line connecting the injection and production wells after 36.5 years. It is seen that the MINC results for 50 m fracture spacing are virtually identical to the porous medium results, while another MINC run for 250 m fracture spacing shows lower temperatures, indicating a less complete thermal sweep.

We mention again a number of problem variations, that can be easily realized with small modifications in the input file. For example, heat exchange with confining beds can be studied by setting MOP(15) = 1 (an appropriate inactive element to represent thermal parameters has already been included in the input file). This would be expected to be of minor significance for the porous medium and the D = 50 m fracture spacing cases, but could have major effects when fracture spacing is as large as 250 m. The problem could be run with permeable matrix blocks; typical matrix permeabilities in fractured geothermal reservoirs are of the order of 1 to 10 microdarcies (10^{-18} to 10^{-17} m^2). It would also be of interest to compare a ''production only'' case with various different injection scenarios. The input file as is can also be run with the EOS2 module; the third primary variable would then specify a CO_2 partial pressure of 5 bars.

```
KA 1(   1,  6) ST = 0.100000e+06 DT = 0.100000e+06 DX1= -.187193e+06 DX2= 0.199814e+00 T = 298.440 P =  8405499. S = 0.209814e+00
KA 1(   2,  4) ST = 0.200000e+06 DT = 0.100000e+06 DX1= -.299719e+06 DX2= 0.772233e-01 T = 295.886 P =  8105780. S = 0.287037e+00
KA 1(   3,  5) ST = 0.400000e+06 DT = 0.200000e+06 DX1= -.633811e+06 DX2= 0.393106e-01 T = 290.238 P =  7471969. S = 0.326348e+00
KA 1(   4,  4) ST = 0.600000e+06 DT = 0.200000e+06 DX1= -.748398e+06 DX2= 0.348865e-01 T = 283.077 P =  6723571. S = 0.361234e+00
KA 1(   5,  6) ST = 0.100000e+07 DT = 0.400000e+06 DX1= -.995672e+06 DX2= 0.279254e-01 T = 272.539 P =  5727899. S = 0.389160e+00
KA 1(   6,  4) ST = 0.140000e+07 DT = 0.400000e+06 DX1= -.198871e+06 DX2= -.139943e-01 T = 270.269 P =  5529028. S = 0.375165e+00
KA 1(   7,  4) ST = 0.220000e+07 DT = 0.800000e+06 DX1= 0.148974e+06 DX2= -.263612e-01 T = 271.975 P =  5678001. S = 0.348804e+00
KA 1(   8,  4) ST = 0.380000e+07 DT = 0.160000e+07 DX1= 0.104575e+06 DX2= -.140159e-01 T = 273.153 P =  5782577. S = 0.334788e+00
KA 1(   9,  4) ST = 0.700000e+07 DT = 0.320000e+07 DX1= -.380368e+05 DX2= -.576194e-02 T = 272.726 P =  5744540. S = 0.329026e+00
KA 1(  10,  5) ST = 0.134000e+08 DT = 0.640000e+07 DX1= -.264969e+05 DX2= -.714147e-02 T = 272.428 P =  5718043. S = 0.321885e+00
KA 1(  11,  4) ST = 0.198000e+08 DT = 0.640000e+07 DX1= 0.268126e+06 DX2= -.181076e-01 T = 275.399 P =  5986169. S = 0.303777e+00
KA 1(  12,  5) ST = 0.326000e+08 DT = 0.128000e+08 DX1= -.434483e+03 DX2= -.184620e-02 T = 275.394 P =  5985735. S = 0.301931e+00
KA 1(  13,  4) ST = 0.454000e+08 DT = 0.128000e+08 DX1= -.720791e+05 DX2= 0.817454e-03 T = 274.606 P =  5913656. S = 0.302749e+00
KA 1(  14,  5) ST = 0.710000e+08 DT = 0.256000e+08 DX1= -.277185e+06 DX2= 0.741485e-02 T = 271.503 P =  5636471. S = 0.310163e+00
KA 1(  15,  5) ST = 0.966000e+08 DT = 0.256000e+08 DX1= -.108444e+06 DX2= 0.274836e-03 T = 270.257 P =  5528026. S = 0.310438e+00
KA 1(  16,  5) ST = 0.122200e+09 DT = 0.256000e+08 DX1= 0.612782e+05 DX2= -.580427e-02 T = 270.964 P =  5589304. S = 0.304634e+00
KA 1(  17,  4) ST = 0.147800e+09 DT = 0.256000e+08 DX1= -.139462e+06 DX2= 0.430498e-02 T = 269.348 P =  5449842. S = 0.308939e+00
KA 1(  18,  4) ST = 0.157788e+09 DT = 0.998800e+07 DX1= 0.257186e+05 DX2= -.260333e-02 T = 269.648 P =  5475561. S = 0.306336e+00

1*RFP* - 36 BLOCKS PARALLEL FIVE-SPOT GRID (CF. SPE-18426)

        OUTPUT DATA AFTER (  18,  4)-2-TIME STEPS                                                    THE TIME IS  0.18262e+04 DAYS

0@@@
```

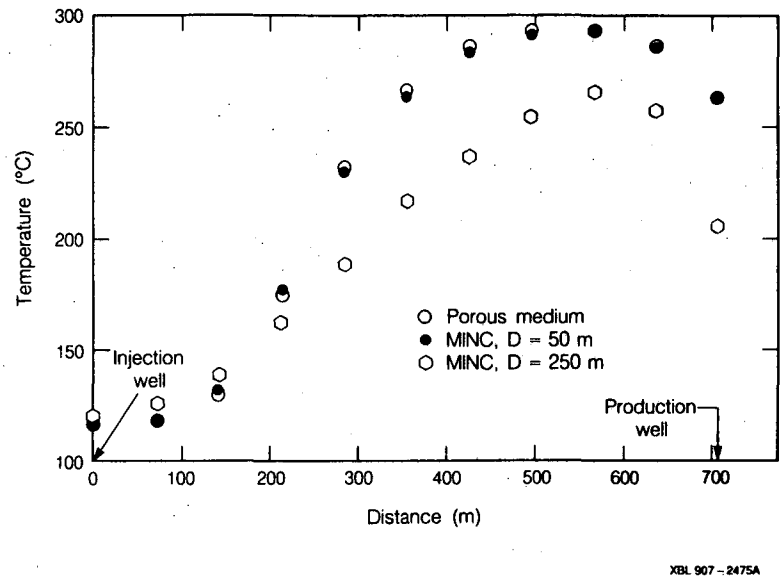


Figure 24. Temperature profiles for problem 4 along a line from injection to production well after 36.5 yrs.

8. Concluding Remarks

With TOUGH2 we are releasing a code that provides a flexible capability for simulating multicomponent multiphase fluid and heat flows in permeable media. TOUGH2 implements the general MULKOM architecture that separates and interfaces the flow and transport aspects of the problem (which do not depend on the nature and number of fluid components and phases) from the fluid property and phase composition aspects (which are specific to the particular fluid mixture under study). Another important aspect of TOUGH2 is the integral finite difference method used to discretize the flow system. This method provides a high degree of flexibility in the description of flow geometry. One-, two-, and three-dimensional flow problems with regular or irregular gridding can be treated on the same footing, and special discretization schemes for fractured media, or for higher-order differencing approximations, can be implemented through appropriate preprocessing of geometric data. For regular grid systems, the integral finite difference method is equivalent to conventional finite differences.

The emphasis in the development of the MULKOM concept, and its implementation in the TOUGH2 code, has been on flexibility and robustness. TOUGH2 is an adaptable research tool, that in the present form can handle a wide variety of flow problems in the fields of geothermal reservoir engineering, nuclear waste isolation, and hydrology. Fluid property modules to be included in future releases would allow applications to problems in petroleum engineering, natural gas recovery and storage, and environmental monitoring and remediation efforts.

TOUGH2 is intended to be a ''general purpose'' simulator. Applications to many different kinds of flow problems are possible, but should be made with caution. The diversity of multiphase fluid and heat flow problems is enormous, and careful consideration must be given to the peculiar features of any given problem if a reasonably accurate

and efficient solution is to be obtained. A case in point is multiphase flow in composite (layered) media, in which discontinuous permeability changes occur at the boundaries between different geologic units. 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 uniform media, relative permeability must be upstream weighted, or else phase fronts may be propagated with erroneous speed (Aziz and Settari, 1979). Recent studies at Lawrence Berkeley Laboratory have shown that for transient two-phase problems in composite media, both absolute and relative permeability must be fully upstream weighted to avoid the possibility of gross errors (Tsang and Pruess, 1990; Wu, Pruess, and Chen, 1990). The applicable weighting schemes for different flow problems are summarized in Fig. 25. Our somewhat disturbing conclusion

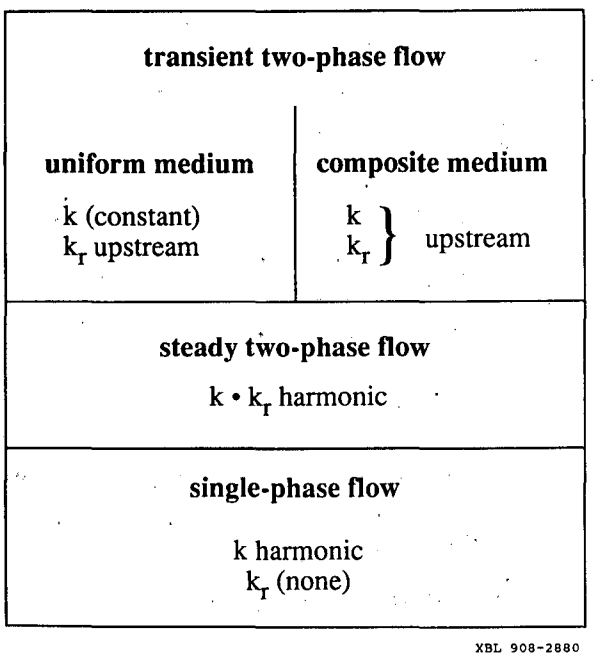


Figure 25. Weighting procedures for absolute (k) and relative permeability (k_r) at grid block interfaces.

is that 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. 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, as is often done within an integral finite difference approach because of the convenience and ease of implementation. 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, 1991; Brand *et al.*, 1991).

These examples are by no means exhaustive; they are simply intended to illustrate potential subtle and not so subtle pitfalls in the modeling of multiphase flows. Generally speaking, in the design and implementation of numerical schemes for such flows, there appears to be a trade-off between accuracy and efficiency on the one hand, and flexibility and robustness on the other. For any given problem, small modifications in the source code will often allow substantial gains in accuracy and efficiency. In many cases it may be advisable to use higher-order differencing schemes. The present version of TOUGH2

does not provide built-in capabilities for higher-order differencing; however, the integral finite difference methodology used in TOUGH2 makes it possible to implement such schemes through simple preprocessing of geometric data. Generally speaking, higher-order differencing schemes can be implemented by assigning additional flow connections, with appropriate weighting factors, between elements of the computational grid (Pruess and Bodvarsson, 1983).

Acknowledgements

The development and documentation of MULKOM and TOUGH2 was supported in large part by the Geothermal Technology Division, U.S. Department of Energy, under Contract No. DE-AC03-76SF00098, with additional funding from the DOE office of Basic Energy Science, the Yucca Mountain Project of Sandia National Laboratories, and the Repository Technology and Transportation Program, U.S. Department of Energy. During the long incubation period of TOUGH2, the author has benefited from many discussions and suggestions by his colleagues at Lawrence Berkeley Laboratory and elsewhere. T. N. Narasimhan of LBL and M. J. O'Sullivan of the University of Auckland made important contributions to the development of capabilities for simulating flow in fractured media and to the modeling of fluid mixtures. Y. Tsang and C. Doughty of LBL, M. Fewell and S. Webb of Sandia National Laboratories, M. J. O'Sullivan and D. Bullivant of the University of Auckland, and D. Holford of PNL assisted in testing the program, and made suggestions for improvements. C. Doughty provided the similarity solution for sample problem 2 (heat pipe). The documentation was reviewed by Y. Tsang, C. Doughty, C. Oldenburg, M. Fewell, S. Webb and A. H. Treadway. P. Fuller provided generous advice on matters of FORTRAN 77 programming. The manuscript was prepared by Mark Jasper. The author is grateful to all these individuals and organizations for their support.

Nomenclature

A	area, m^2
b	Klinkenberg parameter, Pa
B	effective vapor diffusion strength parameter, replaces the group $\phi S_g \tau$ in Eq. (A.7), dimensionless
C	specific heat, J/kg · °C
d	penetration depth for heat conduction, m
D	diffusion coefficient, m^2/s
D	distance, m
DELX	small increments of primary variables for computing numerical derivatives
DX	increments of primary variables during Newton-Raphson iteration
f	diffusive flux, $kg/m^2 \cdot s$
f_{VPL}	vapor pressure lowering factor, dimensionless (Eq. A.9)
F	mass or heat flux, $kg/m^2 \cdot s$ or W/m^2
g	gravity acceleration, m/s^2
h	specific enthalpy, J/kg
i	index of primary thermodynamic variable
J	Jacobian matrix
k	intrinsic permeability, m^2 ($10^{-12} m^2 \approx 1$ darcy)
k	time level index
k_r	relative permeability, dimensionless
K	thermal conductivity, W/m · °C
K_H	Henry's constant, Pa
m	index of volume element (grid block)
m_{air}	molecular weight of air
m_{H_20}	molecular weight of water
m_l	molecular weight of liquid
M	accumulation term in mass or energy balance equation, kg/m^3 or J/m^3

n	index of volume element (grid block)
N	index of volume element (grid block)
NB	number of secondary parameters other than mass fractions in PAR array (usually NB = 6)
NBK	NB + NK
NEL	number of volume elements (grid blocks) in flow domain
NEQ	number of balance equations per volume element; NEQ = NK1 or NEQ = NK
NK	number of mass components present
NK1	NK + 1
NLOC	storage location after which primary variables start for grid block N; NLOC = (N − 1)*NK1
NLOC2	storage location after which secondary parameters start for grid block N; NLOC2 = (N − 1)*(NEQ + 1)*NSEC
NPH	number of phases
NSEC	number of secondary parameters per volume element; NSEC = NPH*NBK + 2
p	index in Newton-Raphson iteration
P	pressure, Pa
$P_{a.e.}$	air entry pressure, Pa
P_c	capillary pressure, Pa
P_{sat}	saturated vapor pressure, Pa
q	volumetric sink or source rate, $kg/m^3 \cdot s$ or W/m^3
r	radius, m
R	residuals in mass or energy balance equations, kg/m^3 or J/m^3
R	universal gas constant, 8314 J/°C · mole
S	saturation (void fraction occupied by a fluid phase), dimensionless
t	time, s
T	temperature, °C
u	specific internal energy, J/kg
V	volume, m^3
x	distance, m

x, X	primary thermodynamic variable
$x_\beta^{(\kappa)}$	mol fraction of component κ in phase β
$X_\beta^{(\kappa)}$	mass fraction of component κ in phase β
$z = \log[r/(t)^{1/2}]$	similarity variable for cylindrical flow geometry

Greek

β	phase index (β = liquid, gas)
θ	thermal diffusivity, m^2/s
κ	component index
ρ	density, kg/m^3
Γ	area, m^2
ϕ	porosity, dimensionless
τ	tortuosity factor, dimensionless
μ	viscosity, $Pa \cdot s$

Subscripts

a	air
β	phase
c	capillary
f	fracture
g	gas
i	initial
l	liquid
r	relative
R	rock
v	vapor

References

Atkinson, P. G., R. Celati, R. Corsi and F. Kucuk, 1980. Behavior of the Bagnore Steam/CO_2 Geothermal Reservoir, Italy, *Soc. Pet. Eng. J.*, 228–238, August.

Aziz, K. and A. Settari, 1979. Petroleum Reservoir Simulation, Elsevier, London and New York.

Brand, C. W., J. E. Heinemann and K. Aziz, 1991. The Grid Orientation Effect in Reservoir Simulation, paper SPE-21228, presented at Society of Petroleum Engineers Eleventh Symposium on Reservoir Simulation, Anaheim, CA.

Dean, J. A. (ed.), 1985. *Lange's Handbook of Chemistry*, McGraw Hill, Thirteenth Edition, New York, NY.

Doughty, C. and K. Pruess, 1990. A Similarity Solution for Two-Phase Fluid and Heat Flow Near High-level Nuclear Waste Packages Emplaced in Porous Media, *Int. J. Heat and Mass Transfer, 33,* (6) 1205–1222.

Doughty, C. and K. Pruess, 1991. A Similarity Solution for Two-Phase Water, Air and Heat Flow Near a Linear Heat Source in a Porous Medium, Lawrence Berkeley Laboratory Report No. LBL-30051.

Duff, I. S., 1977. MA28 - A Set of FORTRAN Subroutines for Sparse Unsymmetric Linear Equations, AERE Harwell Report R 8730.

Edlefsen, N. E. and A. B. C. Anderson, 1943. Thermodynamics of Soil Moisture, *Hilgardia, 15*, (2), 31–298.

Edwards, A. L., 1972. TRUMP: A Computer Program for Transient and Steady State Temperature Distributions in Multidimensional Systems, National Technical Information Service, National Bureau of Standards, Springfield, VA.

Faust, C. R. and J. W. Mercer, 1975. Summary of Our Research in Geothermal Reservoir Simulation, Proc. Workshop on Geothermal Reservoir Engineering, report SGP-TR-12, Stanford, CA.

Hirschfelder, J. O., C. F. Curtiss and R. B. Bird, 1954. *Molecular Theory of Gases and Liquids*, John Wiley & Sons, New York, NY.

International Formulation Committee, 1967. A Formulation of the Thermodynamic Properties of Ordinary Water Substance, IFC Secretariat, Düsseldorf, Germany.

Klinkenberg, L. J., 1941. The Permeability of Porous Media to Liquids and Gases, *in:* API Drilling and Production Practice, pp. 200–213.

Lam, S. T., A. Hunsbedt, P. Kruger and K. Pruess, 1988. Analysis of the Stanford Geothermal Reservoir Model Experiments Using the LBL Reservoir Simulator, *Geothermics, 17*, (4), 595–605.

Loomis, A. G., 1928. Solubilities of Gases in Water, in: E. W. Washburn (ed.), International Critical Tables, Vol. III, pp. 255–257, McGraw-Hill, New York, NY.

Narasimhan, T. N. and P. A. Witherspoon, 1976. An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media, *Water Resour. Res., 12*, (1), 57–64.

O'Sullivan, M. J., 1981. A Similarity Method for Geothermal Well Test Analysis, *Water Resour. Res., 17*, (2), 390–398, April 1981.

O'Sullivan, M. J., G. S. Bodvarsson, K. Pruess and M. R. Blakeley, 1985. Fluid and Heat Flow in Gas-rich Geothermal Reservoirs, *Soc. Pet. Eng. J., 25*, (2), 215–226, April 1985.

Peaceman, D. W., 1977. *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam.

Pritchett, J. W., 1975. Numerical Calculation of Multiphase Fluid and Heat Flow in Hydrothermal Reservoirs, Proc. Workshop on Geothermal Reservoir Engineering, report SGP-TR-12, pp. 201–205, Stanford, CA.

Pritchett, J. W., M. H. Rice and T. D. Riney, 1981. Equation-of-state for Water-carbon Dioxide Mixtures: Implications for Baca Reservoir, report DOE/ET/27163-8, Systems, Science and Software, La Jolla, CA, February 1981.

Pruess, K., 1983a. GMINC—A Mesh Generator for Flow Simulations in Fractured Reservoirs, Lawrence Berkeley Laboratory report LBL-15227, Berkeley, CA.

Pruess, K., 1983b. Development of the General Purpose Simulator MULKOM, Annual Report 1982, Earth Sciences Division, Lawrence Berkeley Laboratory report, LBL-15500.

Pruess, K., 1983c. Heat Transfer in Fractured Geothermal Reservoirs with Boiling, *Water Resour. Res., 19*, (1), 201–208.

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

Pruess, K., 1988. SHAFT, MULKOM, TOUGH: A Set of Numerical Simulators for Multiphase Fluid and Heat Flow, *Geotermia, Rev. Mex. Geoenergia, 4*, (1), 185–202 (also Lawrence Berkeley Laboratory report LBL-24430).

Pruess, K., 1990a. Modeling of Geothermal Reservoirs: Fundamental Processes, Computer Simulations, and Field Applications, *Geothermics, 19*, (1), 3–15.

Pruess, K. (ed.), 1990b. Proceedings of the TOUGH Workshop, Lawrence Berkeley Laboratory Report No. LBL-29710.

Pruess, K., 1991. Grid Orientation Effects in the Simulation of Cold Water Injection into Depleted Vapor Zones, presented at 16th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA, Lawrence Berkeley Laboratory Report No. LBL-30117.

Pruess, K. and G. S. Bodvarsson, 1983. A Seven-point Finite Difference Method for Improved Grid Orientation Performance in Pattern Steam Floods, paper SPE-12252, presented at the Seventh Society of Petroleum Engineers Symposium on Reservoir Simulation, San Francisco, CA.

Pruess, K. and G. S. Bodvarsson, 1984. Thermal Effects of Reinjection in Geothermal Reservoirs with Major Vertical Fractures, *J. Pet. Tech., 36*, (10), 1567–1578.

Pruess, K., C. Calore, R. Celati and Y. S. Wu, 1987. An Analytical Solution for Heat Transfer at a Boiling Front Moving Through a Porous Medium, *Int. J. Heat and Mass Trans., 30*, (12) 2595–2602.

Pruess, K. and T. N. Narasimhan, 1982. On Fluid Reserves and The Production of Superheated Steam from Fractured Vapor-dominated Geothermal Reservoirs, *J. Geophys. Res., 87*, (B11), 9329–9339.

Pruess, K. and T. N. Narasimhan, 1985. A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media, *Soc. Pet. Eng. J., 25*, 1, 14–26.

Pruess, K. and R. C. Schroeder, 1980. SHAFT79 User's Manual, Lawrence Berkeley Laboratory report LBL-10861, Berkeley, CA.

Pruess, K., R. C. Schroeder, P. A. Witherspoon and J. M. Zerzan, 1979. SHAFT78, A Two-phase Multidimensional Computer Program for Geothermal Reservoir Simulation, Lawrence Berkeley Laboratory report LBL-8264, Berkeley, CA.

Pruess, K., J. S. Y. Wang and Y. W. Tsang, 1990. On Thermohydrological Conditions Near High-level Nuclear Wastes Emplaced in Partially Saturated Fractured Tuff. Part 1: Simulation Studies with Explicit Consideration of Fracture Effects, *Water Resour. Res., 26*, (6), 1235–1248.

Pruess, K. and Y. S. Wu, 1989. A New Semianalytical Method for Numerical Simulation of Fluid and Heat Flow in Fractured Reservoirs, paper SPE-18426, presented at Tenth SPE Symposium on Reservoir Simulation, Houston, TX, February.

Sutton, F. M. and A. McNabb, 1977. Boiling Curves at Broadlands Field, New Zealand, *N. Z. J. Sci., 20*, 333–337.

Tsang, Y. W. and K. Pruess, 1990. Further Modeling Studies of Gas Movement and Moisture Migration at Yucca Mountain, Nevada, Lawrence Berkeley Laboratory, report LBL-29127, Berkeley, CA.

Vargaftik, N. B., 1975. *Tables on the Thermophysical Properties of Liquids and Gases*, 2nd Ed., John Wiley & Sons, New York, NY.

Verma, A. K., 1986. Effects of Phase Transformation on Steam-Water Relative Permeabilities, Doctoral Dissertation, University of California, Berkeley, CA, Lawrence Berkeley Laboratory Report No. LBL-20594.

Vinsome, P. K. W. and J. Westerveld, 1980. A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators, *J. Canadian Pet. Tech.*, pp. 87–90, July–September.

Yanosik, J. L. and T. A. McCracken, 1979. A Nine-Point, Finite Difference Reservoir Simulator for Realistic Prediction of Adverse Mobility Ratio Displacements, *Soc. Pet. Eng. J.*, 253–262.

Warren, J. E. and P. J. Root, 1963. The Behavior of Naturally Fractured Reservoirs, *Soc. Pet. Eng. J.*, 245–255, Transactions, AIME, 228.

Wu, Y. S., K. Pruess and Z. X. Chen, 1990. Buckley-Leverett Flow in Composite Media, Lawrence Berkeley Laboratory Report No. LBL-28937.

Appendix A. Mass and Energy Balances

The basic mass- and energy-balance equations solved by MULKOM, TOUGH and TOUGH2 can all be written in the following general form:

$$\frac{d}{dt} \int_{V_n} M^{(\kappa)} dV = \int_{\Gamma_n} \mathbf{F}^{(\kappa)} \cdot \mathbf{n} \, d\Gamma + \int_{V_n} q^{(\kappa)} \, dV \tag{A.1}$$

The integration here 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 denotes mass or energy per unit volume, with $\kappa = 1, \ldots, NK$ labeling the mass components, and $\kappa = NK + 1$ for the heat "component."

The general form of the mass accumulation term is

$$M^{(\kappa)} = \phi \sum_{\beta=1}^{NPH} S_\beta \, \rho_\beta \, X_\beta^{(\kappa)} \quad . \tag{A.2}$$

The total mass of component κ is obtained by summing over all fluid phases $\beta = 1, \ldots,$ NPH. S_β is the saturation (volume fraction) of phase β, ρ_β is density of phase β, and $X_\beta^{(\kappa)}$ is the mass fraction of component κ present in phase β. Similarly, the heat accumulation term in a multi-phase system is

$$M^{(NK+1)} = \phi \sum_{\beta=1}^{NPH} S_\beta \, \rho_\beta \, u_\beta + (1-\phi) \, \rho_R \, C_R \, T \quad , \tag{A.3}$$

where u_β denotes internal energy of fluid phase β.

The mass flux term is a sum over phases

$$\mathbf{F}^{(\kappa)} = \sum_{\beta=1}^{NPH} X_\beta^{(\kappa)} \, \mathbf{F}_\beta \tag{A.4}$$

for $\kappa = 1, \ldots, NK$. Individual phase fluxes are given by a multi-phase version of

Darcy's law:

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

Here k is absolute permeability, $k_{r\beta}$ is relative permeability of phase β, μ_β is viscosity, and

$$P_\beta = P + P_{c,\beta} \tag{A.6}$$

is the pressure in phase β, which is the sum of the pressure P of a reference phase, and the capillary pressure of phase β relative to the reference phase. **g** denotes the vector of gravitational acceleration. Gas phase permeability can be specified to depend on pressure, according to the Klinkenberg relationship $k = k_0(1 + b/P)$, where k_0 is absolute permeability at high pressure (Klinkenberg, 1941). In addition to Darcy flow, MULKOM and TOUGH also include binary diffusion in the gas phase for fluids with two gaseous (or volatile) components κ, κ'

$$\mathbf{f}_{\beta=\mathrm{gas}}^{(\kappa)} = -\phi\, S_g\, \tau\, D_{\kappa\kappa'}\, \rho_g\, \nabla X_g^{(\kappa)} \tag{A.7}$$

$D_{\kappa,\kappa'}$ is the coefficient of binary diffusion which depends on the nature of the gaseous components and on pressure and temperature. τ is a tortuosity factor. When binary diffusion is present the flux-term (A.7) simply gets added to that of (A.4).

Heat flux contains conductive and convective components (no dispersion)

$$\mathbf{F}^{(NK+1)} = -K\nabla T + \sum_\beta h_\beta \mathbf{F}_\beta \tag{A.8}$$

where K is thermal conductivity of the medium, and $h_\beta = u_\beta + P/\rho_\beta$ is the specific enthalpy of phase β.

MULKOM and TOUGH2 can model vapor pressure lowering due to capillary and phase adsorption effects. This is represented by Kelvin's equation (Edlefsen and Anderson, 1943):

$$P_v\,(T, S_l) = f_{VPL}\,(T, S_l) \cdot P_{sat}(T) \tag{A.9a}$$

where

$$f_{VPL} = \exp\left\{\frac{m_l \cdot P_c(S_l)}{\rho_l \, R\,(T + 273.15)}\right\} \tag{A.9b}$$

is the vapor pressure lowering factor. P_{sat} is saturated vapor pressure of bulk liquid, P_c is the difference between liquid and gas phase pressures, m_l is the molecular weight of the liquid, and R is the universal gas constant.

Appendix B. Space and Time Discretization

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

$$\int_{V_n} M dV = V_n M_n \tag{B.1}$$

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}:

$$\int_{\Gamma_n} \mathbf{F} \cdot \mathbf{n} \, d\Gamma = \sum_m A_{nm} F_{nm} \quad . \tag{B.2}$$

Here F_{nm} is the average value of the (inward) normal component of **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 Fig. 26. 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. (A.5), we have

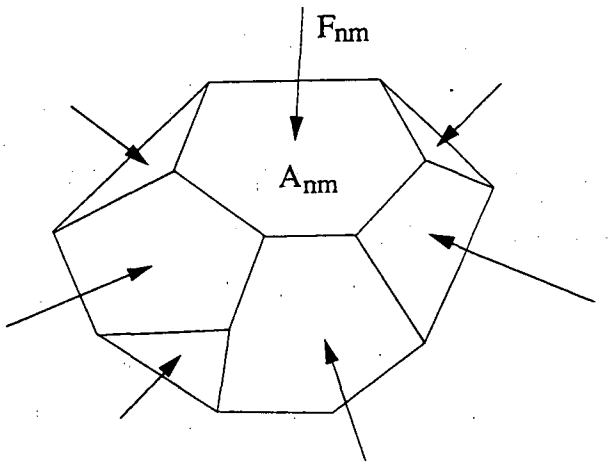
$$F_{\beta,nm} = - k_{nm} \left[\frac{k_{r\beta} \, \rho_\beta}{\mu_\beta} \right]_{nm} \left[\frac{P_{\beta,n} - P_{\beta,m}}{D_{nm}} - \rho_{\beta,nm} \, g_{nm} \right] \tag{B.3}$$

where the subscripts (nm) denote a suitable averaging (interpolation, harmonic weighting, upstream weighting). D_{nm} 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.

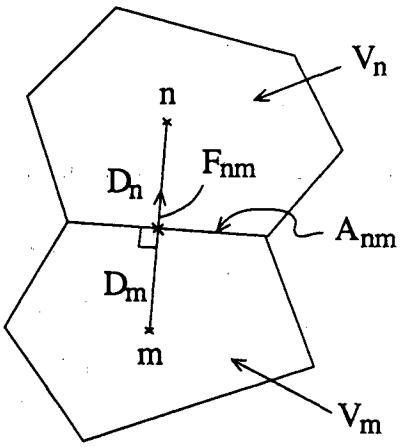
The discretized form of the binary diffusive flux in the gas phase is

$$f^{(\kappa)}_{\beta=gas,\,nm} = - \phi_{nm} \, S_{g,nm} \, \tau_{nm} \, (D_{\kappa\kappa'})_{nm} \, \rho_{g,nm} \, \frac{X_{g,n} - X_{g,m}}{D_{nm}} \quad . \tag{B.4}$$

Substituting Eqs. (B.1) and (B.2) into the governing Eq. (A.1) a set of first-order



XBL 908-2881



XBL 908-2882

Figure 26. Space discretization and geometry data in the integral finite difference method.

ordinary differential equations in time is obtained.

$$\frac{dM_n^{(\kappa)}}{dt} = \frac{1}{V_n} \sum_m A_{nm} F_{nm}^{(\kappa)} + q_n^{(\kappa)} \tag{B.5}$$

Time is discretized as a first order finite difference, and the flux and sink and source terms on the right hand side of Eq. (B.5) 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 multi-phase 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:

$$R_n^{(\kappa)k+1} \equiv M_n^{(\kappa)k+1} - M_n^{(\kappa)k} - \frac{\Delta t}{V_n} \left\{ \sum_m A_{nm} F_{nm}^{(\kappa)k+1} + V_n q_n^{(\kappa)k+1} \right\}$$

$$= 0 \tag{B.6}$$

The entire geometric information of the space discretization in Eq. (B.6) 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 (B.3, B.4) 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, x – y – z), Eq. (B.6) is identical to a conventional finite difference formulation (e.g. Peaceman, 1977).

For each volume element (grid block) V_n there are NEQ equations (κ = 1, 2, ..., NEQ; usually, NEQ = NK + 1), so that for a flow system with NEL grid blocks (B.6) represents a total of NEL · NEQ coupled non-linear equations. The unknowns are the NEL · NEQ independent primary variables $\{x_i;\ i = 1, ..., \text{NEL} \cdot \text{NEQ}\}$ which completely define the state of the flow system at time level t^{k+1}. 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. (B.6) at iteration step p + 1 in a Taylor series in terms of those at index p:

$$R_n^{(\kappa)k+1}(x_{i,p+1}) = R_n^{(\kappa)k+1}(x_{i,p}) + \sum_i \left.\frac{\partial R_n^{(\kappa)k+1}}{\partial x_i}\right|_p (x_{i,p+1} - x_{i,p}) + \ldots = 0 \tag{B.7}$$

Retaining only terms up to first order, we obtain a set of NEL · NEQ linear equations for

the increments $(x_{i,p+1} - x_{i,p})$:

$$-\sum_i \frac{\partial R_n^{(\kappa)k+1}}{\partial x_i} \bigg|_p (x_{i,p+1} - x_{i,p}) = R_n^{(\kappa)k+1}(x_{i,p}) \tag{B.8}$$

All terms $\partial R_n/\partial x_i$ in the Jacobian matrix are evaluated by numerical differentiation. Eq. (B.8) is solved with the Harwell subroutine package "MA28" (Duff, 1977). Iteration is continued until the residuals $R_n^{(\kappa)k+1}$ are reduced below a preset convergence tolerance (see Pruess, 1987).

Appendix C. Description of Flow in Fractured Media

Figure 27 illustrates the classical double-porosity concept for modeling flow in fractured-porous media as developed by Warren and Root (1963). Matrix blocks of low permeability are embedded in a network of inter-connected fractures. Global flow in the reservoir occurs only through the fracture system, which is described as an effective porous continuum. Rock matrix and fractures may exchange fluid (or heat) locally by means of "interporosity flow," which is driven by the difference in pressures (or temperatures) between matrix and fractures. Warren and Root approximated the interporosity flow as being "quasi-steady," with rate of matrix-fracture interflow proportional to the difference in (local) average pressures.

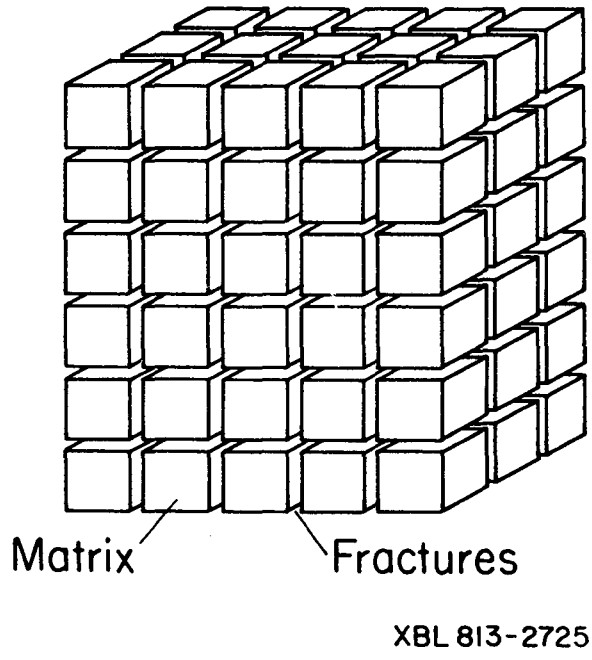


Figure 27. Idealized "double porosity" model of a fractured porous medium.

The quasisteady approximation is applicable to isothermal single phase flow of fluids with small compressibility, where pressure diffusivities are large, so that pressure

changes in the fractures penetrate quickly all the way into the matrix blocks. However, for multiphase flows or coupled fluid and heat flows, the transient periods for interporosity flow can be very long (tens of years). In order to accurately describe such flows it is necessary to resolve the driving pressure and temperature gradients at the matrix/fracture interface. In the method of "multiple interacting continua" (MINC; Pruess and Narasimhan, 1982, 1985), resolution of these gradients is achieved by appropriate subgridding of the matrix blocks, as shown in Fig. 28. 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 Fig. 28.

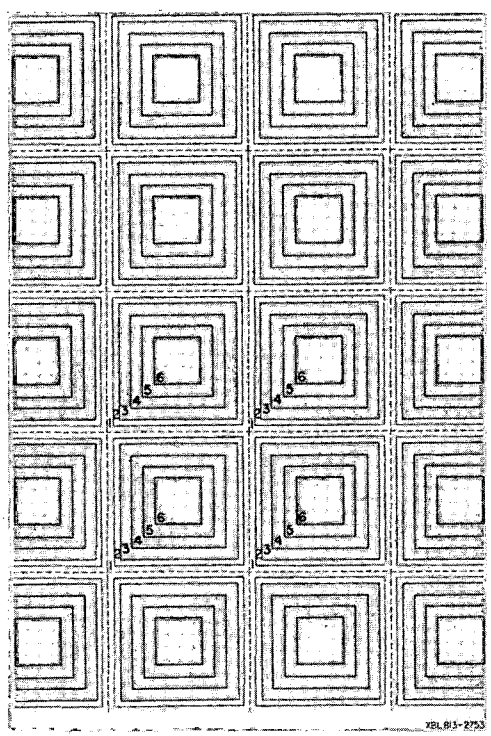


Figure 28. Subgridding in the method of "multiple interacting continua" (MINC).

In general it is not necessary to explicitly consider subgrids in all of the matrix blocks separately. Within a certain reservoir subdomain (corresponding to a finite-difference grid block), all fractures will be lumped into continuum # 1, all matrix material within a certain distance from the fractures will be lumped into continuum # 2, matrix material at larger distance becomes continuum # 3, and so on. Quantitatively, the subgridding is specified by means of a set of volume fractions VOL(j), j = 1, ..., J, 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_0, 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. Full details are given in a separate report (Pruess, 1983a).

The MINC-method as implemented in the MESHMAKER module can also describe global matrix-matrix flow. Figure 29 shows the most general approach, often referred to as ''dual permeability,'' in which global flow occurs in both fracture and matrix continua. It is also possible to permit matrix-matrix flow only in the vertical direction. 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.

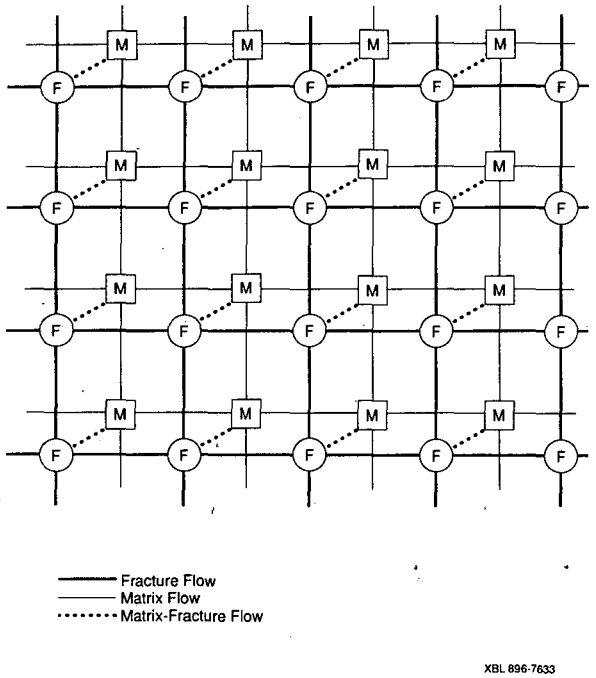


Figure 29. Flow connections in the ''dual permeability'' model.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
INFORMATION RESOURCES DEPARTMENT
BERKELEY, CALIFORNIA 94720

AAX574
LBL Libraries