### Title
ITOUGH2 Software Qualification




### Authors
Finsterle, S
Pruess, K
Fraser, P



### Publication Date
1996-10-01

LBNL-39489
UC-800

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

# ITOUGH2 Software Qualification

S. Finsterle, K. Pruess, and P. Fraser
**Earth Sciences Division**

October 1996

**DISCLAIMER**

LBNL-39489
UC-800

ITOUGH2

ITOUGH2

ITOUGH2

ITOUGH2

ITOUGH2

# ITOUGH2 SOFTWARE QUALIFICATION

S. FINSTERLE
K. PRUESS
P. FRASER

OCTOBER 1996

BERKELEY LAB

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY
EARTH SCIENCES DIVISION
ONE CYCLOTRON ROAD
BERKELEY, CA 94720

THIS WORK WAS PREPARED UNDER THE OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT, U.S. DEPARTMENT OF ENERGY CONTRACT NO. DE-AC03-76SF00098, AND DE-A108-78ET44802 ADMINISTERED BY THE NEVADA OPERATIONS OFFICE IN COOPERATION WITH THE U.S. GEOLOGICAL SURVEY, DENVER

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 1. INTRODUCTION

## 1.1 Software Qualification

The purpose of this report is to provide all software baseline documents necessary for the software qualification of ITOUGH2. ITOUGH2 is a computer program providing inverse modeling capabilities for TOUGH2. TOUGH2 is a numerical simulation code for multi-dimensional coupled fluid and heat flow of multiphase, multicomponent fluid mixtures in porous and fractured media. Software qualification of the TOUGH2 simulation program is described in Pruess et al. (1996).

In accordance with Yucca Mountain Project (YMP) / Lawrence Berkeley National Laboratory (LBNL) Quality Implementing Procedure YMP-LBNL-QIP-SI.0, *Software Qualification*, all components of the specified baseline documentation are provided. This report cinatins the following sections: Requirements Specification, Design Description, Software Validation Test Plan and Report, Software User Documentation, and Appedices. These sections comprise sequential parts of the Software Life Cycle. They are not intended to stand alone but should be used in conjunction with the *TOUGH User's Guide* (Pruess, 1987), *TOUGH2 - A General Purpose Numerical Simulator for Multiphase Fluid and Heat Flow* (Pruess, 1991), *ITOUGH2 User's Guide, Version 2.2* (Finsterle, 1993), and the above-referenced TOUGH2 software qualification document. The qualification package is complete with the attached Software Identification Form and source code for ITOUGH2.

The version of TOUGH2 used with the software being qualified herein is the November 1994 Standard Version 1.11, as qualified in Pruess et al. (1996) and housed at the Department of Energy's Energy Science and Technology Software Center (ESTSC) in Oak Ridge, Tennessee. The qualification status of TOUGH2-related software is shown in Table 1.

***Table 1.*** *Qualification status of TOUGH2-related software.*

| Previously Qualified | | Qualified | To Be |
|---|---|---|---|
| (Pruess et al., 1996) | (Wu et al., 1996) | Herein | Qualified |
| *TOUGH2 Standard Version 1.11* | *Additional TOUGH2 Modules* | *ITOUGH2* | *Additional TOUGH2 Modules* |
| Water, Water with Tracer (EOS1) | Single-Phase Gas (EOS1G) | EOS1 | Hysteresis (T2HYST) |
| | | EOS2 | |
| Water, Carbon Dioxide (EOS2) | Effective Continuum Method (ECM) | EOS3 | Semianalytical Dual-Porosity Model |
| Water, Air (EOS3) | Saturated/Unsaturated Flow (EOS9) | EOS4 | |
| | | EOS5 | |
| Water, Air, Vapor Pressure Lowering (EOS4) | Radionuclide Transport (T2R3D) | EOS9 | |
| | | ECM | |
| Water, Hydrogen (EOS5) | | | |

## 1.2 ITOUGH2 and its Relationship to the TOUGH2 Family of Codes

Figure 1 shows the relationship of ITOUGH2 to the modules of the TOUGH2 family of codes. The TOUGH2 simulator for non-isothermal flows of multicomponent, multiphase fluids in porous and fractured media (Pruess, 1987, 1991; Pruess et al. 1996) and its associated modules are used in ITOUGH2 as subroutines. They are repeatedly called during the optimization process. In this sense, ITOUGH2 is wrapped around the TOUGH2 family of codes which is sometimes referred to as MULKOM. Minor modifications have to be made in some of these modules to make them compatible with ITOUGH2. These modifications include adding a COMMON block with an identifier of the module being used. Furthermore, COMMON blocks holding input parameters and output variables have to be made available in ITOUGH2. Figure 1 is an adjusted version of Figure 1 in Pruess et al. (1996). It contains only those modules that are currently linked to the ITOUGH2 code. Additional TOUGH2 modules can be made available for use within ITOUGH2.

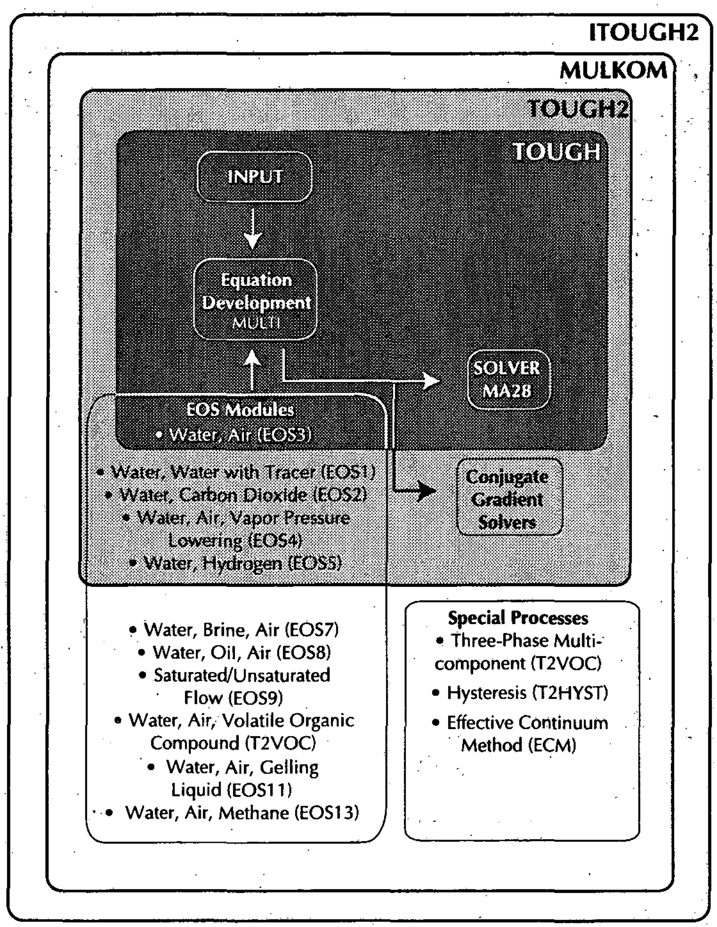


***Figure 1.*** *Relationship of ITOUGH2 to the TOUGH2 family of codes.*

## 1.3 ITOUGH2 Description

ITOUGH2 is a computer program that provides inverse modeling capabilities for the TOUGH2 code. While the main purpose of ITOUGH2 is to estimate model-related hydraulic properties by calibrating TOUGH2 models to laboratory or field data, the information obtained by evaluating parameter sensitivities can be used to optimize the design of an experiment and to analyze the uncertainty of model predictions.

ITOUGH2 solves the inverse problem by automatic model calibration based on the maximum likelihood approach. All TOUGH2 input parameters can be considered unknown or uncertain. The parameters are estimated based on any type of observations for which a corresponding TOUGH2 output is available, including prior information about the parameters to be estimated. A number of different objective functions and minimization algorithms are available. One of the key features of ITOUGH2 is its extensive error analysis which provides statistical information about residuals, estimation uncertainties, and the ability to discriminate among model alternatives. The impact of parameter uncertainties on model predictions can be studied by means of linear error propagation analysis or Monte Carlo simulations.

The procedure of inverse modeling is summarized in Table 2 and visualized in Figure 2, where the process of iteratively updating model parameters to improve the match of observed data is described. The key elements of an inverse modeling code are (1) a simulation program to model the flow and transport in the hydrogeologic system ("forward modeling"), (2) the objective function which measures the misfit between the model output and the data, (3) the minimization algorithm which reduces the objective function by automatically updating parameter values, and (4) the error analysis which allows one to judge the quality of the estimates (see the *ITOUGH2 User's Guide, Version 2.2* (Finsterle, 1993) for more information).

***Table 2.*** *Stepwise Procedure of Parameter Estimation by Inverse Modeling*

| Step | Element | Comment |
|---|---|---|
| 1 | model conceptualization | Develop a TOUGH2 model that simulates the experiment. |
| 2 | prior information | Select reasonable initial parameter values. |
| 3 | simulation | Calculate system response. |
| 4 | objective function | Compare observed and calculated system state. |
| 5 | minimization algorithm | Update parameters so that the value of the objective function decreases. |
| 6 | iteration / convergence | Repeat step 3-5 until no further improvement of the fit can be achieved. |
| 7 | error analysis | Perform sensitivity and residual analysis, calculate estimation uncertainty and parameter correlations, calculate model identification criteria, perform model test, calculate prediction error. |

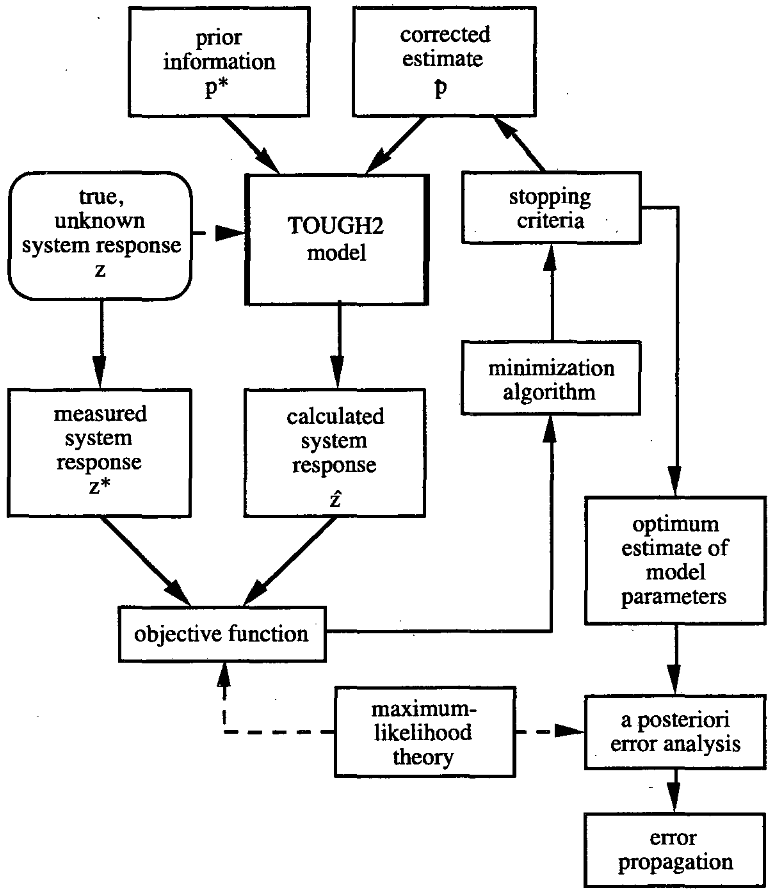


***Figure 2.*** *ITOUGH2 inverse modeling flow chart*

## 2. REQUIREMENTS SPECIFICATION

This section describes specific requirements for ITOUGH2 and should be considered in addition to those previously covered for TOUGH2 in Pruess et al. (1996).

Despite extensive use of numerical modeling techniques with TOUGH2 in the core of the program, ITOUGH2 is not a mere simulator for non-isothermal flow and transport in the subsurface. Its main purpose is to provide reasonable, model-related parameter values based on information contained in observations of the system response and some direct measurements of the parameters of interest. The parameter set estimated by means of an ITOUGH2 inversion should be interpreted in light of the uncertainty measures given by the error analysis, as well as external information. The main sources of parameter estimation errors are (1) systematic errors in the TOUGH2 model developed to simulate the experiment, (2) the data used for calibration, (3) an ill-posed inverse problem, and (4) misinterpretation of ITOUGH2 results. All these types of errors are different in nature from the ones made by the simulation program itself. The latter occur, for example, as a result of a loss of precision due to an inaccurate solution technique, whereas wrong parameter values arise if the concept of inverse modeling is applied in an inappropriate way. The software qualification program for ITOUGH2 therefore focuses on a few elements of the code that require and are accessible to technical verification.

## 2.1 Requirements

A. Perform a standard TOUGH2 simulation in forward mode, i.e., correctly simulate the processes described in Pruess et al. (1996). Since parameters estimated by ITOUGH2 are always model-related and will be used in subsequent prediction runs conducted with the standard TOUGH2 code, the equivalence of the model structure for inversion and prediction is important.

B. Estimate model-related parameters by automatically improving the fit between the calculated system response and observed data. The minimization algorithm has to be efficient and robust for a variety of well-posed inverse problem.

C. Perform a linear error analysis to assess estimation uncertainty and the correlation structure of the estimated parameters.

## 2.2 Prior Examples of Requirements Verification

ITOUGH2 has been applied to a variety of problems related to nuclear waste isolation, geothermal reservoir engineering, and environmental remediation. Early developments of ITOUGH2 were done in collaboration with the Swiss National Cooperative for the Disposal of Radioactive Wastes (Nagra, Wettingen, Switzerland). ITOUGH2 has been used to design and analyze a number of experiments conducted at the Grimsel Test Site (Finsterle and Pruess, 1993; Eugster and Finsterle, 1995; Finsterle and Pruess, 1995a), and is regularly used for welltest analysis at a potential site for the storage of low- and intermediate level radioactive wastes (Finsterle, 1994; Finsterle and Pruess, 1994; Finsterle, 1995a; Domski et al., 1995). ITOUGH2 has also been used for the calibration of a large-scale model for aquifer gas storage (Witherspoon et al., 1995), and geothermal reservoir applications (Finsterle and Pruess, 1995b; White, 1995). The optimization of cleanup operations using ITOUGH2 has been demonstrated in Finsterle et al. (1995a), Finsterle and Pruess (1995c), and Finsterle (1995b). Preliminary examples of ITOUGH2 applications in the Yucca Mountain project have been presented by Ahlers et al. (1995), and Finsterle et al. (1995a). Finally, the calibration of the unsaturated zone site-scale model was performed using ITOUGH2 (Bodvarsson and Bandurraga, eds., 1996).

# 3. DESIGN DESCRIPTION

This section describes the formal structure and architecture of ITOUGH2. It details the functional requirements of the software and how they are implemented. It provides a description of numerical models and methods, describes the software structure, logic, and data structure and flow, and includes performance requirements and design constraints.

## 3.1 Design Background of ITOUGH2

In principle, the core of ITOUGH2 can be any of the modules of the TOUGH2 family of codes (see Figure 1). The first version, named ITOUGH (Finsterle, 1992), was based on the TOUGH code (Pruess, 1987). It was then adapted to the architecture of TOUGH2 (Pruess, 1991) with its flexible array structure. This version is documented in Finsterle (1993). ITOUGH2 has been continuously updated to accommodate new TOUGH2 modules such as T2CG1 (Moridis and Pruess, 1995) and T2VOC (Falta et al., 1995). Furthermore, new parameter and observation types as well as program options have been added as needed. They are documented in the ITOUGH2 help module.

## 3.2 Structure and Architecture

The program architecture of ITOUGH2 allows for a flexible adaptation of the inverse code to any modification of the simulation program TOUGH2 and its derivatives. ITOUGH2 also provides an interface for specifying problem-dependent parameters and observations. This flexibility greatly increases the applicability of the code to a variety of problems, for which an inverse solution is sought. In this document, however, we only verify a base version of the code. A key feature of ITOUGH2 is the possibility to incorporate user-specified functions; these functions have to be verified individually.

The coding of ITOUGH2 complies with the FORTRAN 77 standard, using the following features to minimize programming errors and enable easy, flexible, and safe modification of the code:

(1) Version control system. Each program unit writes a one-line message specifying its name, version number and date, and purpose. An example is shown in Figure 3.

(2) An "IMPLICIT NONE" statement is introduced in each program unit, i.e., all variables have to be declared explicitly by specifying the variable type. This reduces the risk that a typing error remains undetected.

(3) COMMON blocks are merged in include files, ensuring consistent use of variable names in each subroutine. Changes in a COMMON block are consistently updated throughout the program.

(4) All "PARAMETER" statements for dimensioning of global arrays are centralized in the file *maxsize.inc*, included in this document as Appendix A, allowing for flexible redimensioning and ensuring consistency.

(5) Compiling ITOUGH2 by means of a MAKEFILE ensures that all modified modules are updated and properly linked together.

(6) The optimization routines of ITOUGH2 are separated from the simulation modules of TOUGH2, as shown in Figure 4. Both modules share the COMMON blocks containing parameter values and calculated output. The user has to provide two input files. The first is a standard TOUGH2 input file. The second file is the ITOUGH2 input file, in which the user defines the parameters to be estimated, the data used for model calibration, and various program options (see Section 5.3 and Appendix D). This program architecture allows for great flexibility in including modifications of TOUGH2 into the inverse code.

(7) A log-file is maintained holding a list of all past ITOUGH2 runs with date and file information. ITOUGH2 version number, starting and ending date and time, input file names, user login name, machine type, and host name is written to the ITOUGH2 output file.

| PROGRAM | VERSION | DATE | COMMENT |
|---|---|---|---|
| ITOUGH2 | | Current version | V3.0 (JULY 12, 1996) |
| ITOUGH2 | 1.0 | 1 SEPTEMBER 1992 | First version is an adaptation of ITOUGH V1.0 |
| ITOUGH2 | 1.1 | 1 JANUARY 1993 | INOBSDAT,INPAIRED: Flexible time specification |
| | | 1 JANUARY 1993 | INERROR,MTCARLO: number of classes can be specified |
| | | 1 FEBRUARY 1993 | INUSER,USERPAR: user specified parameter |
| | | 1 FEBRUARY 1993 | INUSROBS,USEROBS: user specified observations |
| | | 15 FEBRUARY 1993 | PLOTFI,PLOTCHAR: Reformat plotfiles, plot characteristic curves |
| | | 15 FEBRUARY 1993 | PLOTCHAR: Plots characteristic curves |
| ITOUGH2 | 1.2 | 1 APRIL 1993 | Add version for IBM RS/6000 |
| | | 26 MAY 1993 | INANNEAL,ANNEAL: Simulated Annealing minimization |
| ITOUGH2 | 2.0 | 12 AUGUST 1993 | File t2cg1.f: Conjugate gradient solvers added |
| ITOUGH2 | 2.1 | 23 SEPTEMBER 1993 | Rearrange parameter vector |
| | | 29 SEPTEMBER 1993 | Rearrange observation vector |
| | | 15 FEBRUARY 1994 | Add new observation types |
| ITOUGH2 | 2.2 | 1 FEBRUARY 1994 | Steady-state data points allowed<br>This version is documented in<br>LBL-34581, ITOUGH2 User's Guide, Version 2.2 |
| ITOUGH2 | 2.3 | 10 MAY 1994 | Correlation chart, performance comparison, stopping and restarting |
| | | 16 JUNE 1994 | New parameter types added |
| ITOUGH2 | 2.4 | 15 DECEMBER 1994 | Sensitivity analysis |
| ITOUGH2 | 2.5 | 10 JANUARY 1995 | Automatic parameter selection |
| | | 13 JANUARY 1996 | Aligned common blocks in include files |
| ITOUGH2 | 3.0 | 12 JULY 1996 | YMP Software Qualification |
| WHATCOM | 1.0 | 10 AUGUST 1993 | #35: Q: WHAT COMPUTER IS USED? A: IBM |
| CALLSIG | 2.5 | 20 MAY 1996 | #112: SIGNAL HANDLER |
| CPUSEC | 1.0 | 10 AUGUST 1993 | #--: RETURNS CPU-TIME (VERSION IBM) |
| OPENFILE | 2.5 | 4 JUNE 1996 | #31: OPENS MOST OF THE FILES |
| LENOS | 1.0 | 1 MARCH 1992 | #28: RETURNS LENGTH OF LINE |
| PREC | 1.0 | 1 AUGUST 1992 | #86: CALCULATE MACHINE DEPENDENT CONSTANTS |
| ITHEADER | 1.0 | 1 AUGUST 1992 | #29: PRINTS ITOUGH2 HEADER |
| DAYTIM | 1.0 | 10 AUGUST 1993 | #32: RETURNS DATE AND TIME (VERSION IBM) |
| THEADER | 1.1 | 27 MAY 1993 | #30: PRINTS TOUGH2 HEADER |
| INPUT | 2.5 | 15 MARCH 1996 | READ ALL DATA PROVIDED THROUGH FILE *INPUT*, + SECONDARY MESH + USERX |
| MESHM | 1.0 | 24 MAY 1990 | EXECUTIVE ROUTINE FOR INTERNAL MESH GENERATION |
| MINC | 1.0 | 22 JANUARY 1990 | EXECUTIVE ROUTINE FOR MAKING A "SECONDARY" FRACT.-POROUS MEDIUM MESH |
| PART | 1.0 | 22 JANUARY 1990 | READ SPECIFICATIONS OF MINC-PARTITIONING FROM FILE *INPUT* |
| GEOM | 1.0 | 1 MAY 1991 | CALCULATE GEOMETRY PARAMETERS OF SECONDARY (FRACTURED-POROUS) MESH |
| PROX | 1.0 | 22 JANUARY 1990 | CALCULATE PROXIMITY FUNCTIONS FOR DIFFERENT MATRIX BLOCK SHAPES |
| INVER | 1.0 | 22 JANUARY 1990 | INVERT A MONOTONIC FUNCTION THROUGH NESTED BISECTIONS |
| MINCME | 1.0 | 14 FEBRUARY 1990 | PROCESS PRIMARY MESH FROM FILE *MESH*, AND WRITE SECONDARY MESH FILE |
| CHECKMAX | 1.0 | 11 MAY 1996 | #41: CHECK KEY DIMENSIONS |
| FLOPP | 1.0 | 11 APRIL 1991 | CALCULATE NUMBER OF SIGNIFICANT DIGITS FOR FLOATING POINT ARITHMETIC |
| RFILE | 2.5 | 11 APRIL 1996 | INITIALIZE DATA FROM FILES *MESH* OR *MINC*, *GENER*, AND *INCON* |
| ITINPUT | 1.0 | 1 AUGUST 1992 | # 2: READS COMMANDS OF COMMAND LEVEL 1 |
| READCOMM | 2.5 | 14 JUNE 1996 | #24: READS A COMMAND |
| FINDKEY | 1.1 | 4 AUGUST 1993 | #25: READS A KEYWORD |
| LTU | 1.0 | 1 AUGUST 1992 | #26: CONVERTS LOWER TO UPPER CASE |
| INPARAME | 2.5 | 11 MAY 1996 | # 3: READS PARAMETERS TO BE ESTIMATED |
| INPAR | 2.5 | 14 JUNE 1996 | # 4: READS PARAMETER VALUES, WEIGHTS, ETC. |
| INELEM | 2.2 | 11 MARCH 1994 | #23: READS GRID BLOCK NAME AFTER A COLON |
| NEXTWORD | 2.5 | 9 FEBRUARY 1996 | #27: EXTRACTS NEXT WORD ON A LINE |
| INWBP | 2.5 | 14 JUNE 1996 | #11: READS WEIGHT, BOUNDS, ANNOTATION, AND PARAMETERS |
| READREAL | 1.0 | 1 AUGUST 1992 | #22: READS A REAL AFTER A COLON |
| READINT | 1.0 | 1 AUGUST 1992 | #21: READS AN INTEGER AFTER A COLON |
| INOBSERV | 2.5 | 13 JUNE 1996 | #12: READS TYPE OF OBSERVATION |
| INTIMES | 2.5 | 13 JANUARY 1996 | #13: READS TIMES AT WHICH OBSERVATIONS ARE AVAILABLE |
| INOBS | 2.5 | 13 DECEMBER 1995 | #15: READS OBSERVATION INFOS |
| INOBSDAT | 2.5 | 13 JANUARY 1996 | #17: READS ALL OBSERVED DATA |
| INPAIRED | 2.5 | 3 JULY 1996 | #19: READS PAIRED DATA SET |
| INWEIGHT | 1.1 | 5 AUGUST 1993 | #20: READS WEIGHTS |
| INCOMPUT | 1.0 | 1 AUGUST 1992 | #16: READS VARIOUS COMPUTATIONAL PARAMETERS |
| INTOLER | 2.5 | 11 JUNE 1996 | #83: READS TOLERANCE/STOPPING CRITERIA |
| INJACOB | 1.0 | 1 AUGUST 1992 | #84: READS PARAMETERS FOR COMPUTING JACOBIAN |
| INOPTION | 2.5 | 6 MARCH 1996 | #85: READS PROGRAM OPTIONS |
| INPRINT | 2.5 | 13 JANUARY 1996 | #80: READS OUTPUT OPTIONS |
| GETINDEX | 2.2 | 11 MARCH 1994 | #45: GETS INDEX OF ELEMENTS, CONNECTIONS, AND SOURCES |
| INIGUESS | 2.5 | 12 JUNE 1996 | #38: INITIAL GUESS OF PARAMETERS (XGUESS) |
| GETNMAT | 2.1 | 21 SEPTEMBER 1993 | #44: IDENTIFIES MATERIAL NUMBER |
| IXLBXUB | 2.1 | 21 SEPTEMBER 1993 | #43: INITIALIZES ARRAY XLB AND XUB |
| SETWSCAL | 2.5 | 22 MAY 1996 | #39: INITIALIZES ARRAY WSCALE |
| OBSMEAN | 1.0 | 1 AUGUST 1992 | #40: CALCULATES MEAN OF OBSERVATIONS |
| SETXSCAL | 1.0 | 1 AUGUST 1992 | #42: INITIALIZES ARRAY XSCALE |
| IN_OUT | 2.5 | 8 JUNE 1996 | #35: PRINTS A SUMMARY OF INPUT DATA |
| TIMEWIND | 2.5 | 30 NOVEMBER 1995 | #53: SETS TIME WINDOW |
| PRSTATUS | 2.5 | 5 JULY 1996 | #91: PRINTS STATUS MESSAGES |
| ERROR | 2.5 | 21 MARCH 1996 | #34: PRINTS ERROR MESSAGES |
| LEVMAR | 2.5 | 26 MARCH 1996 | #99: LEVENBERG-MARQUARDT OPTIMIZATION ALGORITHM |
| FCNLEV | 2.3 | 10 JANUARY 1995 | #50: RETURNS WEIGHTED RESIDUAL VECTOR |

***Figure 3.*** *Version control of ITOUGH2 run*

```
UPDATE   2.5      11 MAY       1996       #37: UPDATES PARAMETERS
PRIORINF 2.1      21 SEPTEMBER 1993       #48: PRIOR INFORMATION
OBSERVAT 2.5      13 JUNE      1996       #62: COMPARES MEASURED AND CALCULATED QUANTITIES
GETMESH  1.1      15 APRIL     1993       #47: READS FILE MESH, MINC, GENER, AND INCON
GETINCON 2.5      13 JANUARY   1996       #46: READS FILE INCON
INITTOUG 2.5      18 APRIL     1996       #54: INITIALIZES TOUGH2 RUN (REPLACES PART OF CYCIT)
EOS      1.0      15 AUGUST    1990       *EOS1* ... THERMOPHYSICAL PROPERTIES MODULE FOR WATER
SAT      1.0      22 JANUARY   1990       STEAM TABLE EQUATION: SATURATION PRESSURE AS FUNCTION OF
TEMPERATURE
RELP     2.5      13 JANUARY   1996       RELATIVE PERMEABILITIES
TSAT     1.0      14 MARCH     1991       SATURATION TEMPERATURE AS FUNCTION OF PRESSURE
PCAP     2.5      18 JUNE      1996       CAPILLARY PRESSURE
COWAT    1.0      22 JANUARY   1990       LIQUID WATER DENSITY AND INT. ENERGY AS FUNCTION OF TEMP. AND
PRESS.
SUPST    1.0      29 JANUARY   1990       VAPOR DENSITY AND INTERNAL ENERGY AS FUNCTION OF TEMP. AND PRESSURE
VIS      1.0      22 JANUARY   1990       VISCOSITY OF LIQUID WATER AND VAPOR AS FUNCTION OF TEMP. AND PRES.
BALLA    1.0      22 JANUARY   1990       PERFORM SUMMARY BALANCES FOR VOLUME, MASS, AND ENERGY
CALLTOUG 2.5      19 JUNE      1996       #55: CALLS TOUGH2 FOR ONE TIME STEP
TSTEP    1.0       4 MARCH     1991       ADJUST TIME STEPS TO COINCIDE WITH USER-DEFINED TARGET TIMES
MULTI    2.3      10 JULY      1994       ASSEMBLE ALL ACCUMULATION AND FLOW TERMS
QU       1.02     18 FEBRUARY  1993       ASSEMBLE ALL SOURCE AND SINK TERMS
                                          "RIGOROUS" STEP RATE CAPABILITY FOR MOP(12) = 2
LINEQ    0.91 CG  31 JANUARY   1994       INTERFACE FOR LINEAR EQUATION SOLVERS
                                          CAN CALL MA28 OR A PACKAGE OF CONJUGATE GRADIENT SOLVERS
MC19A                                     HARWELL SUBROUTINE FOR SCALING MATRIX
VISW     1.0      22 JANUARY   1990       VISCOSITY OF LIQUID WATER AS FUNCTION OF TEMPERATURE AND PRESSURE
CONVER   2.5      13 JUNE      1996       UPDATE PRIMARY VARIABLES AFTER CONVERGENCE IS ACHIEVED
OUT      1.0      15 AUGUST    1990       PRINT RESULTS FOR ELEMENTS, CONNECTIONS, AND SINKS/SOURCES
OBSERVED 2.1      15 NOVEMBER  1993       #78: RETURNS OBSERVED DATA AS A FUNCTION OF TIME
OBJFUN   2.5      21 MARCH     1996       #49: COMPUTE OBJECTIVE FUNCTION
WRITEPAR 1.0      17 JUNE      1996       #56: WRITE BEST FIT PARAMETER SET AND BLOCK ROCKS
PLOTFILE 2.5      13 JANUARY   1996       #58: WRITES PLOTFILE IN PLOPO-FORMAT
RESIDUAL 2.1      29 SEPTEMBER 1993       #90: BACKCALCULATES RESIDUALS
RANDOM   1.0       1 AUGUST    1992       #71: RANDOM NUMBER GENERATOR
JAC      2.5      14 JUNE      1996       #51: CALCULATES FINITE DIFFERENCE JACOBIAN
TERMINAT 2.5       5 JULY      1996       61: PERFORM ERROR ANALYSIS AND TERMINATE ITOUGH2
WRIFI    2.5      13 JANUARY   1996       AT THE COMPLETION OF A TOUGH2 RUN, WRITE PRIM. VAR. ON FILE *SAVE*
QFISHER  2.2      16 FEBRUARY  1994       #77: RETURNS QUANTILE OF F-DISTRIBUTION
QCHI     1.0       1 AUGUST    1992       #88: RETURNS CHI-SQUARE QUANTILE
POLYNOM  1.0       1 AUGUST    1992       #89: EVALUATES POLYNOM
EIGEN    2.5      16 MAY       1996       #59: PERFORMS EIGENANALYSIS
LOGLIKE  2.1      29 SEPTEMBER 1993       #68: COMPUTE LOG-LIKELIHOOD
MLLAMBDA 2.2      14 FEBRUARY  1994       #67: ESTIMATES NEW LAMBDAS
QNORMAL  2.5      13 JANUARY   1996       #87: RETURNS QUANTILE OF NORMAL DISTRIBUTION
PLOTIF   1.0      15 FEBRUARY  1993       #96: PLOT INTERFACE
REFORMAT 1.1      15 APRIL     1993       #97: REFORMATS PLOT FILES
QUOTES   1.0      15 FEBRUARY  1993       #98: RETURNS TEXT BETWEEN QUOTES
----------------------------------------------------------------------------------------------------
```

***Figure 3 (cont.).** Version control of ITOUGH2 run*

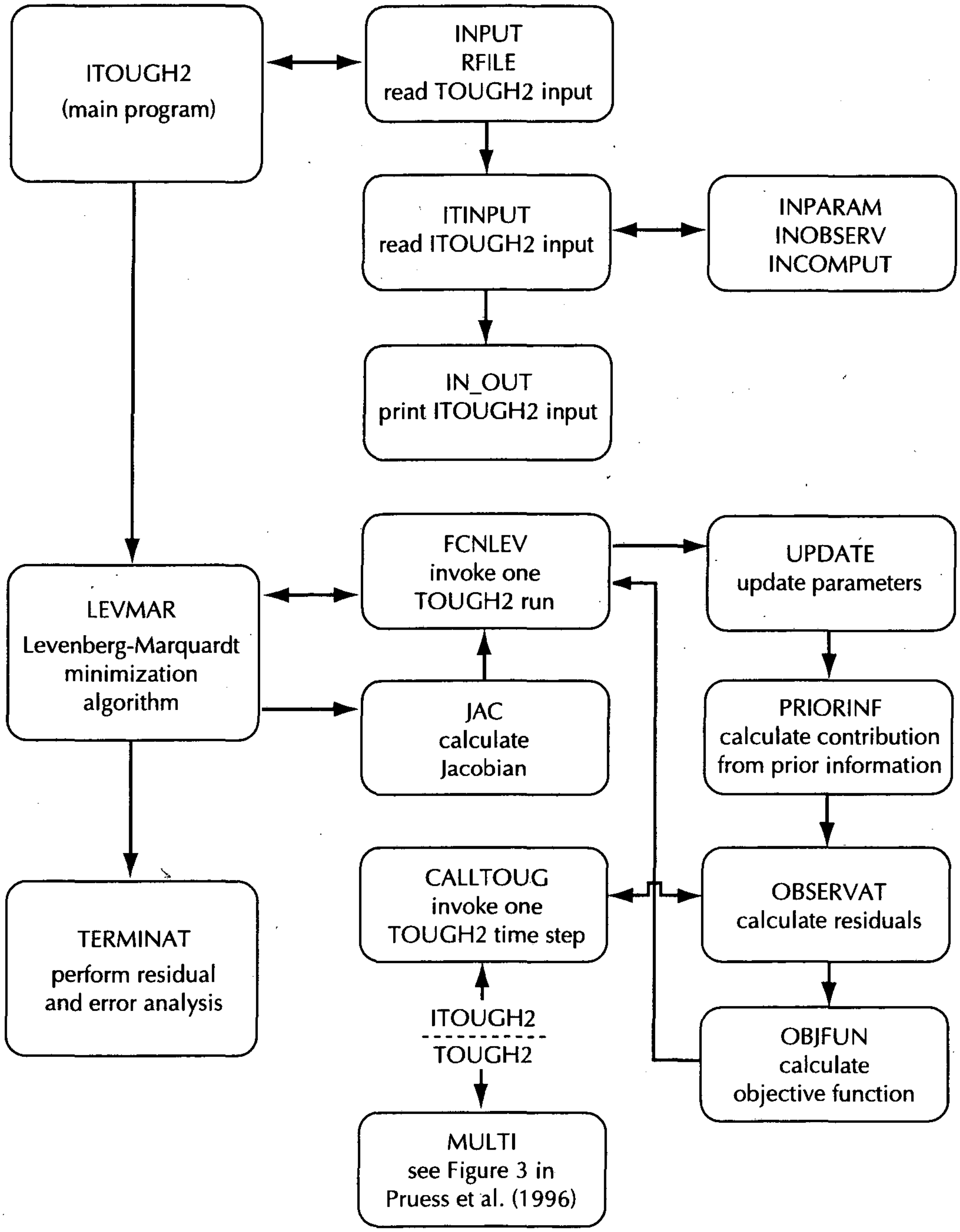


***Figure 4.*** *Simplified ITOUGH2 flow chart showing connection between TOUGH2 and ITOUGH2*

### 3.3 Physical Processes and Approximations

The physical processes and approximations considered in ITOUGH2 are the same as for the TOUGH2 simulator summarized in Pruess et al. (1996).

### 3.4 Mathematical and Numerical Methods

ITOUGH2 estimates elements of a parameter vector **p** based on observations summarized in vector **z** by minimizing a weighted least-square objective function S using the Levenberg-Marquardt minimization algorithm (refer to Table 3, following).

Vector **p** of length *n* contains the parameters to be estimated by inverse modeling. The parameters have to be TOUGH2 input parameters or a (user-specified) function thereof. Elements of vector **p** may include hydrogeologic properties such as absolute permeability or a parameter of the capillary pressure function, initial and boundary conditions, or geometrical features such as fracture spacing which is an input parameter of TOUGH2's MINC processor, a part of the MESHMAKER utility (see Pruess et al., 1996).

Vector **z** of length *m* contains dependent, observable variables at discrete points in space and time (= calibration points). Elements of **z** refer to both measured quantities (data) and simulation results. The most commonly used observations for calibration are pressure, flow rate, temperature and concentration measurements. The vector of observable variables may also contain measured parameter values. For example, if permeability has been measured on cores in the laboratory, this information can be considered as an additional data point, and treated along with the direct observations of the system response. Such measured parameter values are referred to as "prior information".

The residual vector **r** contains the differences between the measured and calculated system response:

Residual vector: $$\mathbf{r} = \mathbf{z}^* - \hat{\mathbf{z}} \qquad (3.4.1)$$

For example, $r_i$ is the difference between the measured and calculated pressure at a certain point in space and time. A special type of residual is the difference between the measured parameters (prior information) and the estimated parameter values. This difference, appropriately weighted, can be used as a plausibility criterion to constrain the inversion.

The unexplained part (measurement errors) of the system response cannot be described individually, but must be explained by means of a stochastic model. A reasonable assumption about the measurement errors is that they are a result of many individual error sources, and are thus uncorrelated, normally distributed random variables with zero mean (the *a posteriori* residual analysis can be used to test whether the normality assumption is justified). The distributional assumption can therefore be summarized in a covariance matrix $\mathbf{C}_{zz}$. $\mathbf{C}_{zz}$ is an *m*×*m* diagonal matrix. The i-th diagonal element of matrix $\mathbf{C}_{zz}$ is the variance $\sigma_i^2$ representing the measurement error of observation $z_i$.

$$\mathbf{C}_{zz} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_i^2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_n^2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sigma_j^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sigma_m^2 \end{bmatrix} \tag{3.4.2}$$

The objective function S is a measure of the misfit between the data and the model calculation. If the measurement errors ($\mathbf{z}^*$-$\mathbf{z}$) are normally distributed with mean $E([\mathbf{z}^*-\mathbf{z})]=0$ and covariance matrix $E[(\mathbf{z}^*-\mathbf{z})(\mathbf{z}^*-\mathbf{z})^T]=C_{ZZ}$, maximum-likelihood theory yields the following objective function to be minimized:

$$S = (\mathbf{z}^* - \mathbf{z}(\mathbf{p}))\mathbf{C}_{zz}^{-1}(\mathbf{z}^* - \mathbf{z}(\mathbf{p}))^T = \mathbf{r}^T\mathbf{C}_{zz}^{-1}\mathbf{r} \tag{3.4.3}$$

Minimizing the objective function (3.4.3) can be performed by local linearization and setting the derivative of S with respect to **p** to zero. This leads to the Jacobian matrix **J** of dimensions *m*×*n*, the elements of which are defined as follows:

$$J_{ij} = \frac{\partial z_i}{\partial p_j} \tag{3.4.4}$$

ITOUGH2 uses either forward or centered finite differences to calculate the elements $J_{ij}$:

forward:
$$J_{ij} \sim \frac{z_i(p_j + \delta p_j) - z_i(p_j)}{\delta p_j} \tag{3.4.5a}$$

centered:
$$J_{ij} \sim \frac{z_i(p_j + \delta p_j) - z_i(p_j - \delta p_j)}{2\delta p_j} \tag{3.4.5b}$$

where each parameter is perturbed by a small percentage of its value:

$$\delta p_j = \alpha \cdot p_j \tag{3.4.6}$$

Note that calculating a forward finite difference approximation of the Jacobian matrix (3.4.5a) requires *n*+1 solutions of the forward problem, i.e., *n*+1 transient TOUGH2 simulations from time zero to the time of the last calibration point. Using a centered finite difference quotient (3.4.5b) for calculating the Jacobian is more accurate, but requires 2*n*+1 runs.

The Levenberg-Marquardt modification of the Gauss-Newton algorithm is a rather general and robust procedure for iteratively updating the parameter vector **p** (Levenberg, 1944; Marquardt, 1963). It can be described as shown in Table 3.

*Table 3. Levenberg-Marquardt Algorithm*

| | |
|---|---|
| $\Delta \mathbf{p} = \left(\mathbf{J}^T \mathbf{C}_{zz}^{-1} \mathbf{J} + \lambda \mathbf{D}\right)^{-1} \mathbf{J}^T \mathbf{C}_{zz}^{-1} \mathbf{r}$ | (3.4.7) |
| $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + \Delta \mathbf{p}$ | (3.4.8) |
| decrease $\lambda$ by factor $\nu$ if $S(\mathbf{p}^{(k+1)}) < S(\mathbf{p}^{(k)})$ | |
| increase $\lambda$ by factor $\nu$ if $S(\mathbf{p}^{(k+1)}) > S(\mathbf{p}^{(k)})$ | |
| $\lambda$ = Levenberg parameter ($\lambda^{(k=0)} \approx 0.001$) | |
| $\nu$ = Marquardt parameter ($\nu \approx 10$) | |
| $\mathbf{D}$ = Diagonal matrix with $\mathbf{D}_{ii} = (\mathbf{J}^T \mathbf{C}_{zz}^{-1} \mathbf{J})_{ii}$ | |

If $\lambda$ is large, the first term on the right-hand side of (3.4.7) becomes a matrix with a dominant diagonal. This leads to a small step along the gradient of S. Stepping along the steepest descent direction is robust, but inefficient. With decreasing Levenberg parameter, (3.4.7) converges to the Gauss-Newton algorithm with its quadratic convergence rate.

One of the key advantages of a formalized approach to parameter estimation is the possibility to perform an extensive *a posteriori* error analysis. First, the residual analysis provides some measure of the overall goodness-of-fit and allows identification of systematic errors or flaws in the stochastic model. Next we can determine the uncertainty of the estimated parameters. Note that a good match does not necessarily mean that the estimates are reasonable. They may be highly uncertain due to the lack of sensitivity or high parameter correlations which is an indication that the invser problem is over-parameterized. The covariance matrix of the estimated parameters can be further analyzed to obtain correlation coefficients, indicating parameter combinations that lead to similar matches, etc. Finally, we can calculate the uncertainty of the simulation results, which also allows us to identify potential outliers in the data.

The estimated error variance represents the variance of the mean weighted residual and is thus a measure of goodness-of-fit:

$$s_0^2 = \frac{\mathbf{r}^T \mathbf{C}_{zz}^{-1} \mathbf{r}}{m-n} \qquad (3.4.9)$$

Note that if the residuals are consistent with the distributional assumption about the measurement errors (i.e., matrix $\mathbf{C}_{zz}$), then the estimated error variance assumes a value close to one. Next we calculate the expected value and the covariance matrix $\mathbf{C}_{pp}$ of the estimated parameters. Based on the linearity assumption it is easy to show that the expected value of the estimated parameter is equal to the parameter itself. The definition of the covariance matrix yields:

$$\mathbf{C}_{pp} = s_0^2 \cdot \left(\mathbf{J}^T \mathbf{C}_{zz}^{-1} \mathbf{J}\right)^{-1} \qquad (3.4.10)$$

The Jacobian $\mathbf{J}$ is evaluated at the optimum parameter set. The interpretation of the covariance matrix $\mathbf{C}_{pp}$ provides the key criteria to evaluate the inverse modeling results.

## 3.5 Array Structure and Handling

The main connection between ITOUGH2 and the simulation module TOUGH2 is maintained through COMMON blocks with major arrays that hold TOUGH2 input parameters and TOUGH2 output variables. These arrays are the ones defined in the original TOUGH2 code; they are described in detail in Pruess et al. (1996). In addition to those arrays, a number of ITOUGH2 vectors and matrices are also stored in COMMON blocks. The key arrays are discussed in this section. They are grouped according to their main dimensions, `MAXN` and `MAXM`, where `MAXN` corresponds to $n$, the length of the parameter vector **p**, and `MAXM` corresponds to $m$, which is the length of observation vector **z**. Table 4 gives a summary of selected arrays, each of which is described in detail in the appropriate include file. Recall that the dimension of each array is set in include file *maxsize.inc* (see Appendix A). Apart from the arrays shown in Table 4, there are additional vectors holding more detailed information about a certain parameter or observation, such as the rock type or element it is referring to, its type, an annotation, etc.

***Table 4.*** *Summary of Major ITOUGH2 Arrays*

| Array | Dimension | description | include file |
|---|---|---|---|
| XGUESSR | MAXN,MAXR, MAXXGR | parameter value from TOUGH2 input file | *guess.inc* |
| XIGUESS | MAXN | parameter value from ITOUGH2 input file | *guess.inc* |
| XGUESS | MAXN | initial guess | *guess.inc* |
| XPAR | MAXN | current estimate | *guess.inc* |
| XLB,XUB | MAXN | lower/upper bound | *guess.inc* |
| XIPAR | MAXN | prior information | *estim.inc* |
| WPAR | MAXN | weight of prior information | *estim.inc* |
| SPAR | MAXN | maximum step size | *estim.inc* |
| XBEST | MAXN | best estimate | *best.inc* |
| GRADIENT | MAXN | gradient of objective function | *gradient.inc* |
| QXX | MAXN,MAXN | covariance matrix of estimated parameters | *covar.inc* |
| FSCALE | MAXM | diagonal elements of prior covariance matrix | *maxm.inc* |
| FVEC | MAXM | residuals | *maxm.inc* |
| F0 | MAXM | residuals after completion of iteration | *fnull.inc* |
| FJACOBI | MAXM,MAXN | Jacobian | *jacobi.inc* |

## 3.6 Software and Hardware Considerations

ITOUGH2 is portable to platforms that have a FORTRAN 77 compiler. In order to make portability easy, all machine-dependent functions are assembled in a special source file named *mdep???.f*, where "*???*" indicates the machine type such as "IBM", "SUN", "DEC", "HP", and "STAR", for IBM RISC 6000, SUN, DEC alpha, HP, and Stardent workstations, respectively. There is also a version for PC using the Lahey compiler; the corresponding file is named *mdeplah.f*. The appropriate machine-dependent module is selected in the Makefile.

Execution of ITOUGH2 is performed under UNIX by means of a script file, *itough2*. The script file has several mandatory arguments (such as the TOUGH2 and ITOUGH2 input file names and the number of the EOS module being used) as well as optional arguments (such as output file names, return of auxiliary files, etc.). The script file ensures that multiple inversions can be performed simultaneously without generating conflicting file names. It also supports the use of two script files, *prista* and *kit*, for displaying the current status of an ITOUGH2 run and its termination, respectively. ITOUGH2 can also be run conventionally by typing the name of the executable. Figure 5 shows the file structure of ITOUGH2 on a UNIX workstation. It includes directory *itough2* with the ITOUGH2 source code and executable, a directory *bin* with the script files, and the working directories with the TOUGH2 and ITOUGH2 input and output files. During execution, a temporary directory is created with a unique name *it2_PID*, where *PID* is the process identification number.

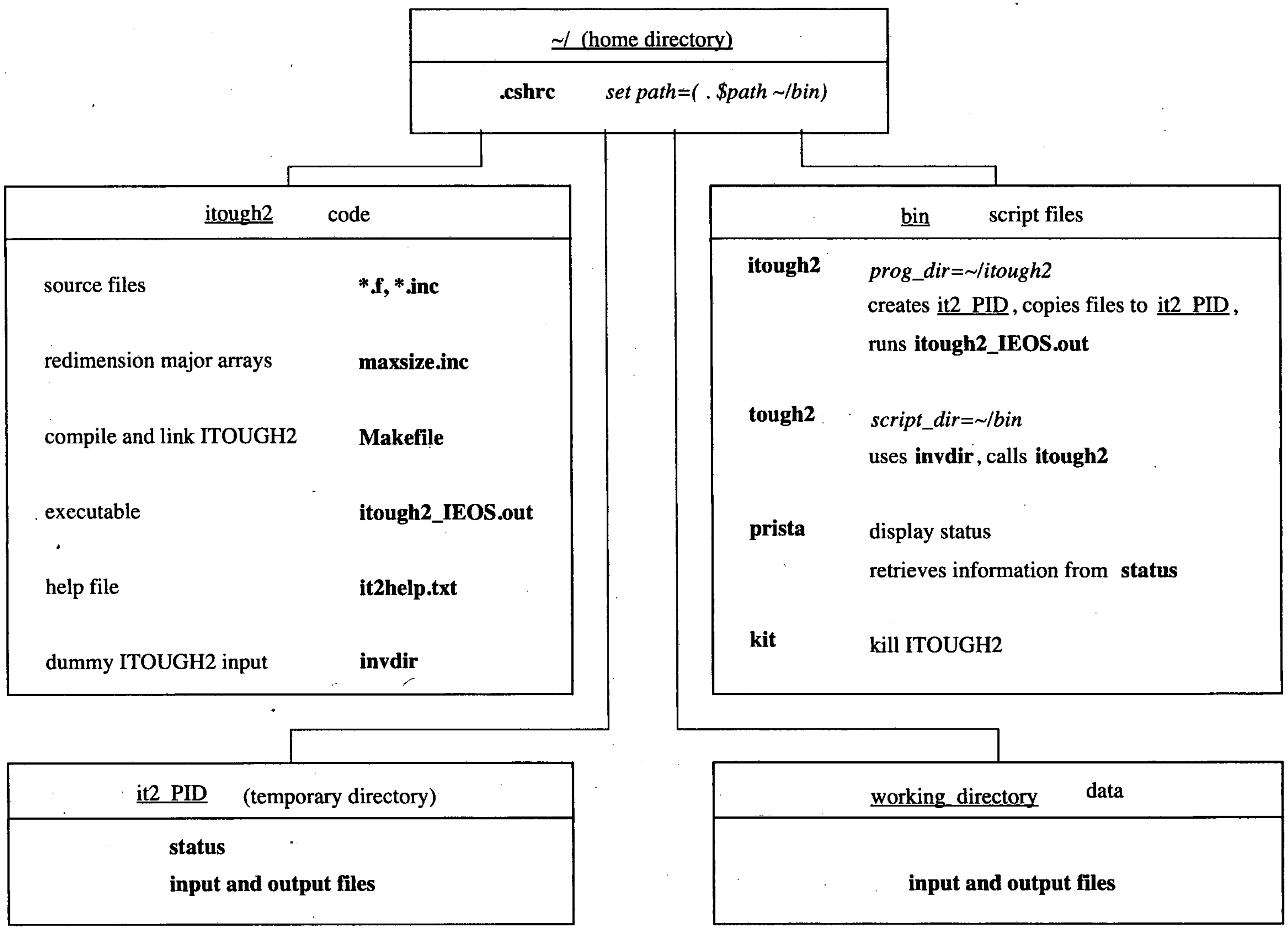


*Figure 5. ITOUGH2 file structure on UNIX platform*

## 3.7 Performance Requirements and Design Constraints

ITOUGH2 uses no pre- or post-processors and requires no interfaces with external hardware. Source files with machine-dependent subroutines for a number of UNIX workstations are provided (see Section 3.6). The memory requirements highly depend on the problem size and the dimensioning of the major arrays which can easily be adjusted (see Section 3.2). As pointed out before, the key module of ITOUGH2 is the TOUGH2 simulator which solves the forward problem. Proper use of the TOUGH2 code and correct set-up of the inverse problem is essential for running successful ITOUGH2 inversions. Since formulation of an ill-posed inverse problem as the main difficulty depends on the available data, the flow problem, and the parameters to be estimated, no general recommendations can be made. Recall, however, that ITOUGH2 provides a number of measures to assess whether the inverse problem is reasonably well posed.

## 3.8 Future Developments

ITOUGH2 developments will mainly consist of adapting modifications made on the TOUGH2 module and its derivatives. Furthermore, new parameter and observation types will be introduced.

## 3.9 Verification of Design Description

Verification that ITOUGH2 meets the design description was conducted during testing and debugging of the code. The applicability of the code to a variety of problems related to nuclear waste isolation, geothermal energy, gas storage, and environmental remediation has been demonstrated (see also Section 2.2 of this report).

# 4. SOFTWARE VALIDATION TEST PLAN AND REPORT

## 4.1 Validation Test Plan and Acceptance Criteria

The Validation Test Plan for ITOUGH2 consists of the TOUGH2 Validation Test Plan described in Pruess et al. (1996), and the following validation test runs conducted to check the performance specifications listed in Section 2.1.

In the terminology of software engineering, "software validation" tests are those that ensure that the software meets the requirements specified for it. The "software verification" tests also fulfill the definition of "software validation", because in order to obtain good results when compared to analytical solutions, the software must be able to model the processes of interest in the problem.

The acceptance criteria for ITOUGH2 are related to each of the performance requirements specified in Section 2.1.

Requirement A, Test 1: While small differences are expected between a standard TOUGH2 simulation and an ITOUGH2 run in forward mode, the deviations have to be small (less than 0.1%) to be considered acceptable.

Requirement B, Test 2 and 3: The minimization algorithm is considered successfully tested if the true parameter set is identified within 1% in a synthetic inversion. Note that the 1% limit is somewhat arbitrary. Less tolerance could be requested if more data were added, or fewer parameters were simultaneously estimated. If noisy data are used, the difference between the true and estimated parameter value should be less than two times the standard deviation of the estimate.

Requirement C, Test 4: The results of the error analysis are confirmed if the actual confidence region is reasonably well represented by its linear approximation. The agreement is considered reasonable if the orientation of the error ellipse coincides with the orientation of the ellipsoidal region described by the corresponding contour line of the objective function. Furthermore, the size of the ellipse expressed by the eigenvalues of the covariance matrix should be comparable to the size of the actual error region.

***Table 5.*** *Summary of Validation Problems for ITOUGH2*

| # | Problem Title | Dimension | Features | Issues | References |
|---|---|---|---|---|---|
| 1 | forward problem | 2-D, MINC | non-isothermal, production / injection, phase transition, MINC | verification of forward run with ITOUGH2 against verified TOUGH2 solution | #4 in Pruess (1991); Pruess and Wu (1993); Finsterle and Pruess (1995b) |
| 2 | inversion using exact data | 2-D, MINC | estimation of six parameters based on four observation types, correct data | verification of minimization algorithm | this report |
| 3 | inversion using noisy data | 2-D, MINC | estimation of six parameters based on four observation types, noisy data | validation of minimization algorithm, sensitivity coefficients, error analysis | Finsterle and Pruess (1995b) |
| 4 | error analysis | 1-D, linear | estimation of two parameters based on water potential and flow rate data | verification of covariance matrix of estimated parameters | Finsterle (1993); Finsterle and Pruess (1995a) |

## 4.2 Validation Tests

### 4.2.1 Test Case 1: Solving the Forward Problem

The purpose of this test is to confirm that the forward problem is correctly solved in ITOUGH2. Comparison of the output from ITOUGH2 with standard TOUGH2 results is sufficient to test the coupling of the simulation module (TOUGH2) with the optimization part of ITOUGH2 (see also Figure 4).

As a test case, we consider a two-dimensional five-spot production-injection problem previously studied by Pruess (1991) and Pruess and Wu (1993). Due to the symmetry of the five-spot configuration, only 1/8 of the basic pattern needs to be modeled (Figure 6). The problem specifications correspond to conditions typically encountered in deeper zones of hot two-phase flow geothermal reservoirs. The medium is assumed to be fractured with embedded impermeable matrix blocks in the shape of cubes with side lengths of 50 m. The permeable volume fraction is 2% with a porosity of the fracture domain of 50%. Reservoir thickness is 305m. Cold water with an enthalpy of 500kJ/kg is injected into the initially hot reservoir (300 °C) at a rate of 30kg/s. Production rate is also 30kg/s. Five years of injection and production are simulated. The resulting liquid and vapor flow rates in the production well are shown in Figure 7. The calculated flow rates using TOUGH2 (symbols) and ITOUGH2 (solid line) are identical, verifying the correct solution of the forward problem in the ITOUGH2 code and thereby meeting the stated acceptance criteria.

Note that this validation checks many options of the EOS1 module. Similar comparisons were done for all the other EOS modules, confirming that TOUGH2 and ITOUGH2 yield identical solutions to the forward problem involving complex flow processes. It should be realized that in many ITOUGH2 runs the time stepping is adapted to provide output at the times where calibration data are available. This may result in minor differences due to differences in the time discretization error. However, whenever the exact same time stepping is enforced in a standard TOUGH2 run, identical results are obtained.

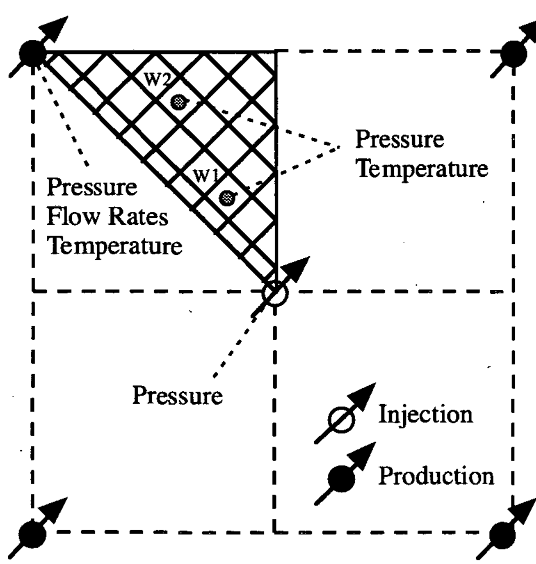


***Figure 6.*** *Five-spot well pattern with grid for modeling 1/8 symmetric domain. Observation points and type of data are also indicated (see Test Case 2).*

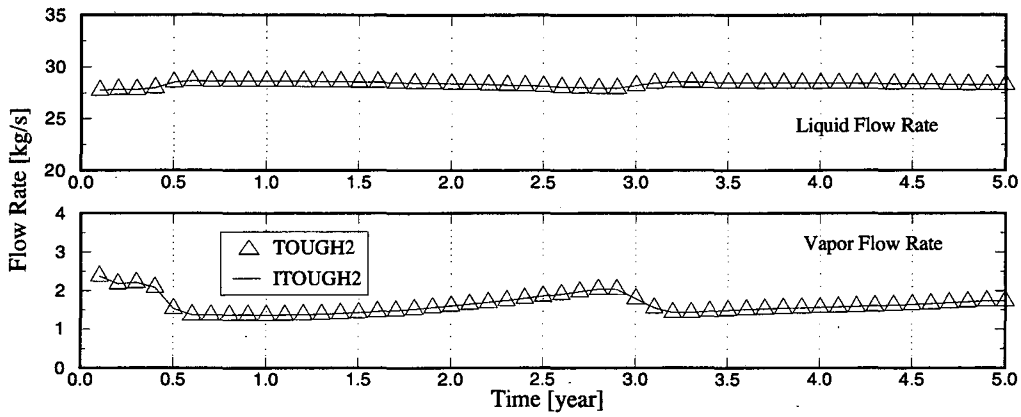


*Figure 7.* *Solution of the forward problem with TOUGH2 and ITOUGH2.*

### 4.2.2 Test Case 2: Inversion Using Exact Data

The purpose of this test is to demonstrate that ITOUGH2 is able to identify the correct parameter values if the model structure is correct and exact data are available. The two conditions can easily be met by matching synthetically generated data.

We assume that temperature and pressure measurements are taken in the injection (Inj) and production well (Pro) as well as in two observation wells (W1, W2; see Figure 6). Furthermore, liquid and vapor flow rates are measured in the production well. Note that temperature and pressure measurements are redundant as long as two-phase conditions prevail. TOUGH2 is run in forward mode to generate data for five years of field performance history.

Six model parameters are subjected to the estimation process. They are: (1) the absolute permeability of the fracture system, (2) fracture zone porosity, (3) heat conductivity, (4) specific heat of the rock grains, (5) fracture spacing $a$, which is a parameter of the MINC processor, and (6) the initial reservoir temperature $T_i$. These six parameters are perturbed from their true values to the ones shown in column 3 of Table 6. Subsequently, the TOUGH2 model is automatically calibrated against the observations. As shown in Table 6, the value of the objective function is reduced from initially 1.41E5 to 2.39E-4. The true parameter values are accurately identified, i.e., within the 1 % acceptance criterion. Small deviations can be attributed to an insufficient curvature of the objective function at the minimum. They are a result of a (very mildly) ill-posed inverse problem, due to small sensitivity and high correlation of fracture porosity, heat conductivity, and fracture spacing (see also discussion of Test Case 3 below).

*Table 6.* *True, Initial, and Estimated Parameter Set*

| Parameter | True Value | Initial Guess | Best Estimate |
|---|---|---|---|
| log (permeability [m$^2$]) | -14.22 | -13.00 | -14.22 |
| fracture zone porosity [-] | 0.50 | 0.30 | 0.50 |
| specific heat [J/kg°C] | 1000.00 | 800.00 | 999.79 |
| heat conduct. [W/m°C] | 2.10 | 1.50 | 2.09 |
| fracture spacing [m] | 50.00 | 20.00 | 49.84 |
| temperature [°C] | 300.00 | 250.00 | 300.00 |
| objective function | 0.00 | 1.41E5 | 2.39E-4 |

### 4.2.3 Test Case 3: Inversion Using Noisy Data

The purpose of this case is to validate the performance of ITOUGH2 in a more realistic situation involving measurement errors. Furthermore, we demonstrate that even though some of the parameters, i.e., the ones that are not very sensitive, are not exactly identified, the predictions made with such a model are still very good due to the estimation of model-related parameters during the calibration process. Using the same model as discussed in the previous validation problems, the synthetic observations were perturbed by a random noise using the standard deviations shown in Table 7. The best estimate parameter set identified after fifteen Levenberg-Marquardt iterations is shown in Table 8. The covariance and correlation matrices of the estimated parameters are summarized in Table 9; their correctness will be verified in Test Case 4.

***Table 7.*** *Observations Used for Model Calibration*

| Data Type | Location | Standard Deviation |
|---|---|---|
| Pressure | Inj/Pro/W1/W2 | 2.00 bar |
| Temperature | Pro/W1/W2 | 5.00 °C |
| Liquid flow rate | Pro | 1.60 kg/s (~5 %) |
| Vapor flow rate | Pro | 0.08 kg/s (~5 %) |

***Table 8.*** *True, Initial, and Estimated Parameter Set*

| Parameter | True Value | Initial Guess | Best Estimate |
|---|---|---|---|
| log (permeability [m$^2$]) | -14.22 | -13.00 | -14.22 |
| fracture zone porosity [-] | 0.50 | 0.30 | 0.56 |
| specific heat [J/kg°C] | 1000.00 | 800.00 | 971.00 |
| heat conduct. [W/m°C] | 2.10 | 2.50 | 2.25 |
| fracture spacing [m] | 50.00 | 20.00 | 50.50 |
| temperature [°C] | 300.00 | 250.00 | 300.10 |

***Table 9.*** *Variance-Covariance Matrix (Main Diagonal and Lower Triangle) and Correlation Matrix (Upper Triangle)*

| | log($k$) | $\phi$ | $C_R$ | $K$ | $a$ | $T_i$ |
|---|---|---|---|---|---|---|
| log($k$) | 5E-6 | 0.21 | 0.17 | -0.25 | -0.21 | -0.18 |
| $\phi$ | 2E-5 | 2E-3 | 0.17 | -0.23 | -0.18 | -0.04 |
| $C_R$ | 0.01 | 0.23 | 845 | 0.39 | 0.53 | -0.07 |
| $K$ | -5E-4 | -0.01 | 10.15 | 0.79 | 0.98 | 0.29 |
| $a$ | -4E-3 | -0.08 | 14.52 | 8.27 | 89.22 | 0.19 |
| $T_i$ | -4E-5 | -2E-4 | 0.24 | 0.02 | 0.20 | 0.01 |

Note that the standard deviation, (the square-root of the diagonal elements of Table 9) refers to the joint probability density function, i.e., it takes into account the uncertainty of the parameter itself and the influence from correlated parameters. The parameters with less agreement between the true and the estimated value are those exhibiting a large standard deviation, mainly due to low sensitivity and high correlation with the other parameters. In conclusion, ITOUGH2 provides the measures for a correct interpretation of the inverse modeling results.

The system response as observed in the injection, production and observation wells is shown in Figure 8. The squares are the synthetically generated and perturbed data points used to calibrate the model. The triangles represent the future system response for the true parameter set. The solid lines are the pressures, temperatures, water and vapor flow rates simulated using the estimated parameter set. For the first 5 years, the deviations between the solid lines and the squares minimize the objective function. Beyond 5 years, the solid lines are predictions, i.e., an extrapolation of the system response matched during the calibration period. The model predictions are uncertain due to uncertainties in the estimated parameters. The standard deviation of the calculated system response, e.g., the uncertainty of the predicted temperature in the production well at a certain point in time, is the square root of the corresponding diagonal element of matrix $\mathbf{C}_{zz}$ which is calculated using first-order error propagation analysis:

$$\mathbf{C}_{zz} = \mathbf{J}\,\mathbf{C}_{pp}\,\mathbf{J}^T \qquad (4.2.3.1)$$

Here, matrix $\mathbf{J}$ is the sensitivity matrix for the predicted system response, and $\mathbf{C}_{pp}$ is the covariance matrix of the estimated parameters (Table 9). The resulting error bands on the model predictions are shown as dash-dotted lines in Figure 8. The true system response (triangles) lies within the estimated error band despite the fact that the parameter set used for the prediction does not exactly correspond to the true one. The error band shown in Figure 8 approximates the 95 % confidence region. The demonstrated performance thus meets the acceptance criterion.

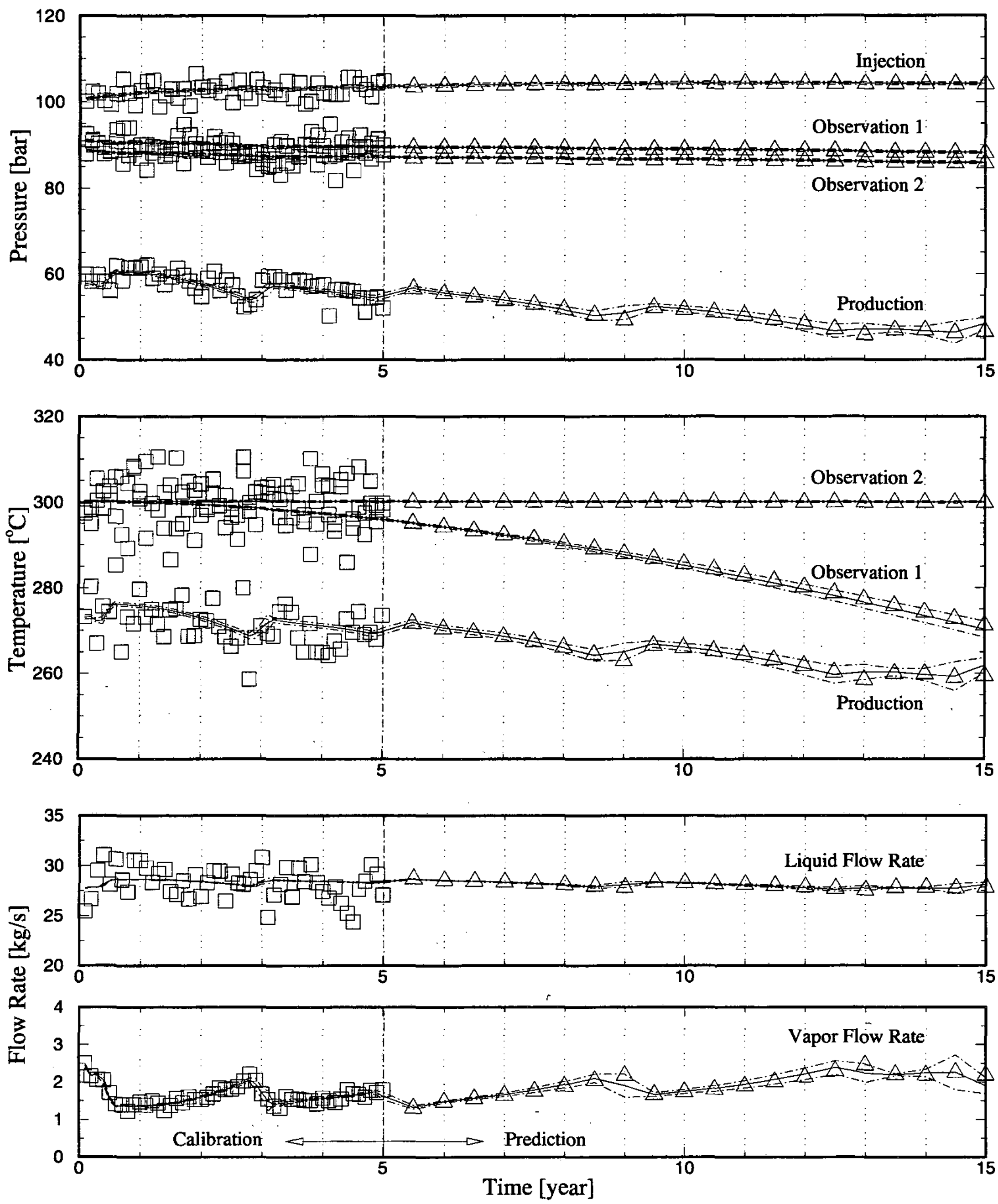


***Figure 8.*** *Calibration and prediction of pressures, temperatures, water and vapor flow rates. Squares are synthetic data points used for calibration. Triangles represent the true system response. Simulation results based on the estimated parameter set are shown as solid lines. Error bands on the 95 % confidence level (dash-dotted lines) are calculated using linear error propagation analysis.*

### 4.2.4 Test Case 4: Error Analysis

The purpose of this problem is to verify the calculation of the covariance matrix of the estimated parameters. Two parameters, the van Genuchten parameters $n$ and $1/\alpha$, are jointly estimated in an inversion of synthetically generated matric potential and flow rate data. The problem set up is described in Finsterle (1993). After optimization, the covariance matrix of the estimated parameters, $\mathbf{C_{pp}}$, is calculated. The actual confidence region on a given significance level is bounded by the contour of the objective function on level $S(\hat{\mathbf{p}}) + s_0^2 \cdot n \cdot F_{n,n-m,1-\alpha}$ (Finsterle and Pruess, 1995a). The linear approximation of this confidence region is given by $\mathbf{C_{pp}}$. For $n$=2, the approximated confidence region is elliptical. The orientation of the axis is given by the eigenvectors of $\mathbf{C_{pp}}$, and their lengths are proportional to the square-root of the corresponding eigenvalues. Figure 9 shows a contour plot of the objective function, and the ellipse from the linear approximation of the confidence region, accurately matching the contour on the corresponding level. This verifies the calculation of $\mathbf{C_{pp}}$ in ITOUGH2, indicating that the acceptance criteria are met. Note that if the model is highly non-linear in the parameters, the linearity assumption may be violated, leading to too optimistic estimates of parameter uncertainty. ITOUGH2 provides a correction procedure to account for non-linearity, based on a proposal by Carrera (1984). This approach is described and verified in Finsterle and Pruess (1995a).

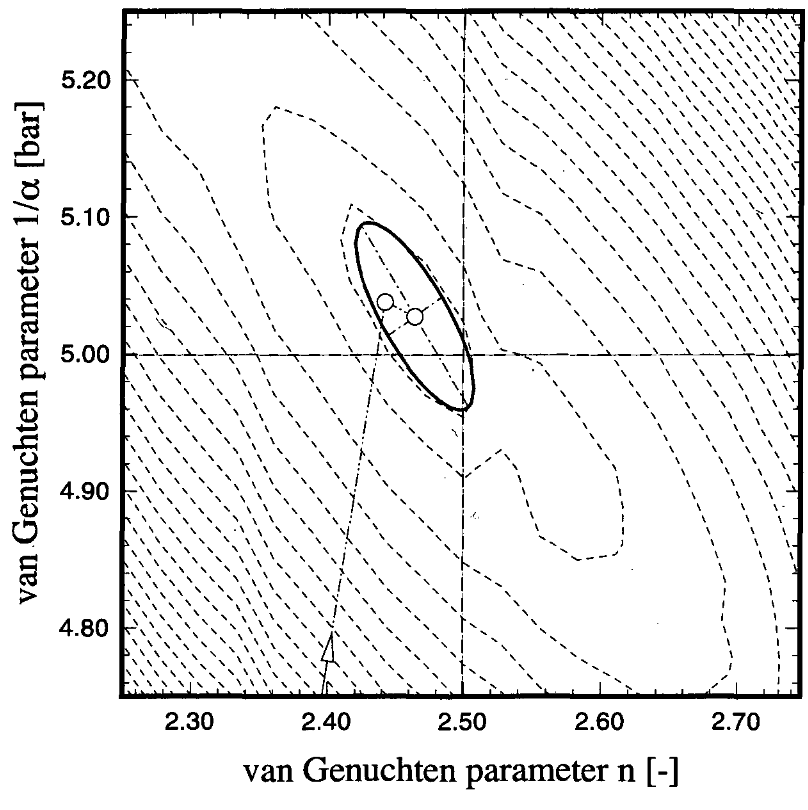


*Figure 9. Objective function, solution path, and 95 % confidence region.*

### 4.3 Summary

Software validation of ITOUGH2 was accomplished by running four test cases described above. A comparison of a standard TOUGH2 simulation and an ITOUGH2 run in forward mode yielded the exact same results, proving the connection between TOUGH2 and the optimization routine to be correct. The minimization algorithm is able to identify the true parameter set describing a variety of hydrogeological properties, initial conditions, and geometrical features. The inversion was performed based on synthetically generated temperature, pressure, liquid and vapor flow rate data, and the model structure was perfectly known. More important is the fact that reasonable estimates were obtained using noisy data, i.e., the estimated parameter set was consistent with the true one if the calculated standard deviations were taken into account. The error analysis performed by ITOUGH2 was further scrutinized by comparing the confidence region from the calculated covariance matrix with the actual confidence region, which can be depicted as a certain contour line of the objective function. The two regions compare very well, validating the calculation of the covariance matrix, and rendering the linearity and normality assumption a reasonable approximation for this Test Case

Table 10 provides a cross-check of the requirements against the validation problems described above that satisfy those requirements.

***Table 10.*** *Requirements Validation Cross-Check*

| Requirement | Test Case | Criteria Met? | Reference |
|---|---|---|---|
| A. Perform a standard TOUGH2 simulation in forward mode, simulating the processes described in Pruess et al. (1996). | 1 | met | Pruess, 1991; Pruess and Wu, 1993; Finsterle and Pruess, 1995b |
| B. Estimate model-related parameters by automatically improving the fit between the calculated system response and observed data. The minimization algorithm has to be efficient and robust for any well-posed inverse problem. | 2, 3 | met | this report; Finsterle and Pruess, 1995b |
| C. Perform a linear error analysis to assess estimation uncertainty and the correlation structure of the estimated parameters. | 4 | met | Finsterle, 1993 Finsterle and Pruess, 1995b |

In conclusion, all acceptance criteria are met, suggesting that the requirement specifications for ITOUGH2 are satisfied. Moreover, ITOUGH2 has been successfully applied to a number of inverse problems on different scales, and involving a variety of physical processes.

# 5. SOFTWARE USER DOCUMENTATION

## 5.1 Installation Procedures

Instructions for installation, and execution of the ITOUGH2 code is provided in the ITOUGH2 *read.me* file which is included in Appendix B.

## 5.2 Hardware and Software Operating Environments

Machine-dependent subroutines are provided for IBM RISC, SUN, HP, Stardent, SGI, and DEC alpha workstations. If using the Lahey compiler, ITOUGH2 can also be installed on a PC. Adaptation of the code to other operating environments can easily be achieved by providing the appropriate system routines in file *mdep???.f*, in analogy to the above mentioned examples. ITOUGH2 is developed and most conveniently run under the UNIX operating system.

## 5.3 Input and Output

ITOUGH2 requires two input files: (1) a standard TOUGH2 input file (Pruess, 1987, 1991), and (2) an ITOUGH2 input file (Finsterle, 1993). ITOUGH2 provides a structured high-level command input language in free format. An example is given in Figure 10. The complete ITOUGH2 command index can be found in Appendix C.

```
> PARAMETER

  >> estimate ABSOLUTE permeability
     >>> ROCK: ATMOS BOUND SOI_1 +3   (one value for 6 rocks)
         >>>> print LIST of all possible commands
         >>>> ANNOTATION          : permeability
         >>>> estimate LOGARITHM
         >>>> INDEX               : 1 2 (horizontal perm. only)
         >>>> initial GUESS is : -14.0
         >>>> RANGE : -18.0  -12.0    (upper and lower bounds)
         >>>> maximum STEP        : 1.0 (please HELP!)
         >>>> don't WEIGHT p.i.: 0.0
         <<<<
     <<<

> OBSERVATION

  >> calibration TIMES: 20 LOGARITHMICALLY spaced [MINUTES]
     1.0  180.0

  >> CONCENTRATION
     >>> ELEMENT: A1__1  A1__2
         >>>> ANNOTATION    :  TCE conc. in gas phase
         >>>> LOGARITHM
         >>>> PHASE         :  1
         >>>> COMPONENT     :  3
         >>>> COLUMN        : 1  3
         >>>> DATA on FILE : test.dat [SECONDS]
         >>>> DEVIATION     : 0.5
         <<<<
     <<<
  <<

> COMPUTATION

  >> CONVERGENCE
     >>> number of ITERATIONS: 5
     <<<
  <<
```

***Figure 10.*** *Excerpt of ITOUGH2 input file.*

## 5.4 User Features

User features are described in Finsterle (1993). An ITOUGH2 log book ensures tractability of all inverse modeling runs.

## 5.5 Summary

Software verification of this phase of the Software Life Cycle, which relates to installation, user documentation, and provision of code demonstration cases, is completed by providing with this package the sample problems described in Appendix D. The problems discussed in Appendix D demonstrate how the user sets up input files to address a number of issues solvable by ITOUGH2. In addition, an updated ITOUGH2 Users' Guide will be available from LBNL upon request when published.

## 5.6 Version History

ITOUGH2 Version 3.0 is herein qualified for use, and will be released to ESTSC upon completion of the Users' Guide.

## ACKNOWLEDGMENTS

The authors are grateful to Ardyth Simmons, Christine Doughty, and Don Mangold who provided thoughtful reviews. We would like to thank Jackie Gamble for processing the document. This work was supported by the Director, Office of Civilian Radioactive Waste Management, U.S. Department of Energy, through Memorandum Purchase Order EA9013MC5X between TRW Environmental Safety Systems, Inc. and the Ernest Orlando Lawrence Berkeley National Laboratory, under contract No. DE-AC03-765F00098.

## REFERENCES


Ahlers, C.F., S. Finsterle, Y.S. Wu, and G.S. Bodvarsson, Determination of Pneumatic Permeability of a Multi-Layered System by Inversion of Pneumatic Pressure Data, paper presented at the AGU Fall Meeting, December 11-15, San Francisco, Calif., 1995.

Bodvarsson, G.S., and T. M. Bandurraga, Eds., Development and Calibration of the Three-Dimensional Site-Scale Unsaturated Zone Model of Yucca Mountain, Nevada, Lawrence Berkeley Laboratory Report, LBNL-39315, Berkeley, CA, August 1996.

Carrera, J., Estimation of Aquifer Parameters Under Transient and Steady State Conditions, Ph.D. dissertation, Dep. of Hydrology and Water Resources, University of Arizona, Tucson, 1984.

Domski, P., C. Enhachescu, W. Harding, R. Johns, J. M. Lavanchy, R. Senger, and J. Wozniewicz, WLB: Hydraulic Testing SB4a/v, Nagra Internal Report, Nagra, Wettingen, Switzerland, December 1995.

Eugster, S., and S. Finsterle, GTS-TPF: Design Calculations for an Extended Gas Threshold Pressure Test at Grimsel Test Site, Nagra Internal Report, Nagra, Wettingen, Switzerland, September, 1995.

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

Finsterle, S., ITOUGH User's Guide, Version 1.0, Nagra Internal Report, Nagra, Wettingen, Switzerland, August, 1992.

Finsterle, S., Wellenberg: One-Dimensional Stress Release Model Assuming Two-Phase Flow Conditions, Nagra Internal Report, Wettingen, Switzerland, September, 1992.

Finsterle, S., ITOUGH2 User's Guide, Version 2.2, Lawrence Berkeley National Laboratory Report, LBL-34581, Berkeley, CA, August 1993.

Finsterle, S., Inverse Modeling of Test SB4-VM2/216.7 at Wellenberg, Lawrence Berkeley National Laboratory Report, LBL-35454, Berkeley, CA, March 1994.

Finsterle, S., Design of a Welltest for Determining Two-Phase Hydraulic Properties, Lawrence Berkeley National Laboratory Report, LBL-37448, Berkeley, CA, January 1995a.

Finsterle, S., Forward and Inverse Modeling of Multiphase Flow Systems, paper presented at the 1st GAMM Conference on Modelling and Computation in Environmental Sciences, Stuttgart, Germany, October 12/13, 1995b.

Finsterle, S., and K. Pruess, FLG: Design Calculations for a Combined Ventilation and Brine Injection Experiment at the Grimsel Rock Laboratory, Lawrence Berkeley National Laboratory Report, LBL-34460, Berkeley, CA, July 1993.

Finsterle, S., and K. Pruess, Estimating Two-Phase Hydraulic Properties by Inverse Modeling, paper presented at the International High Level Radioactive Waste Management Conference, Las Vegas, Nevada, May 22-26, 1994.

Finsterle, S., and K. Pruess, Solving the Estimation-Identification Problem in Two-Phase Flow Modeling, *Water Resour. Res., 31* (4), 913-924, April, 1995a.

Finsterle, S., and K. Pruess, Automatic History Matching of Geothermal Field Performance, Proceedings of the 17th New Zealand Geothermal Workshop, 193-198, Auckland, New Zealand, November 8-10, 1995b.

Finsterle, S., and K. Pruess, Using Simulation-Optimization Techniques to Improve Multiphase Aquifer Remediation, Proceedings of the TOUGH Workshop '95, Lawrence Berkeley National Laboratory Report, LBL-37200, 181-186, Berkeley, CA, March 20-22, 1995c.

Finsterle, S., G. S. Bodvarsson, and G. Chen, Inverse Modeling as a Step in the Calibration of the LBL-USGS Site-Scale Model of Yucca Mountain, paper presented at the International High Level Radioactive Waste Management Conference, Las Vegas, Nevada, May 1-5, 1995a.

Finsterle, S., C.F. Tsang, S.M. Benson, and G.J. Moridis, Use of Blocking Agent to Improve the Effectiveness of Soil Vapor Extraction, paper presented at the ACE, I&EC Symposium Emerging Technologies in Hazardous Waste Management VII, Atlanta, Georgia, September 17-20, 1995b.

Finsterle, S., and K. Pruess, ITOUGH2: Solving TOUGH Inverse Problems, Proceedings of the TOUGH Workshop '95, March 20-22, , Lawrence Berkeley National Laboratory Report, LBL-37200, 287-292, Berkeley, CA, 1995c.

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

Levenberg, K., A Method for the Solution of Certain Nonlinear Problems in Least Squares, *Quart. Appl. Math., 2*, 164-168, 1944.

Marquardt, D.W., An Algorithm for Least Squares Estimation of Nonlinear Parameters, SIAM J. Appl. Math., 11, 431-441, 1963.

Moridis, G., and Pruess, K., Flow and Transport Simulations Using T2CG1, a Package of Conjugate Gradient Solvers for the TOUGH2 Family of Codes, Report LBL-36235, Lawrence Berkeley National Laboratory, Berkeley, California, April 1995.

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

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

Pruess, K., A. Simmons, Y.S. Wu, and G. Moridis, TOUGH2 Software Qualification, Lawrence Berkeley National Laboratory Report, LBL-38383, Berkeley, CA, 1996.

Pruess, K., and Wu, Y.S., A New Semianalytical Method for Numerical Simulation of Fluid and Heat Flow in Fractured Reservoirs, *Soc. Pet. Engineers Adv. Tech. Ser., 1,* 63-72, 1993.

Witherspoon, P. A., P. Fuller, and S. Finsterle, Three-Dimensional Multiphase Effects in Aquifer Gas Storage, Proceedings of the TOUGH Workshop '95, Lawrence Berkeley National Laboratory, Report, LBL-37200, 137-156, Berkeley, CA, March 20-22, 1995.

White, S. P., Inverse modelling of the Kawerau geothermal reservoir, NZ, Proceedings of the 17th New Zealand Geothermal Workshop, 211-216, Auckland, New Zealand, November 8-10, 1995.

Wu, Y. S, R. Ahlers, P. Frazer, and A. Simmons, Software Qualification of Selected TOUGH2 Modules, Lawrence Berkeley National Laboratory Report, LBL-39292, Berkeley, CA, 1996.

## APPENDIX A. ITOUGH2 PARAMETER STATEMENTS, FILE *MAXSIZE.INC*

```
C$$$$$$$$$ PARAMETERS FOR SPECIFYING THE MAXIMUM PROBLEM SIZE $$$$$$$$$$
C
C ********************************************************************
C ITOUGH2 and TOUGH2 parameter statements
C ********************************************************************
C
C --- MAXTIM    : Maximum number of specified calibration times
      INTEGER MAXTIM
      PARAMETER (MAXTIM=250)
C
C ********************************************************************
C TOUGH2 parameter statements
C ********************************************************************
C
C --- Maximum number of elements/blocks
      INTEGER MAXEL
      PARAMETER (MAXEL=10000)
C
C --- Maximum number of connections
      INTEGER MAXCON
      PARAMETER (MAXCON=20000)
C
C --- Maximum number of equations per block
      INTEGER MAXEQ
      PARAMETER (MAXEQ=3)
C
C --- Maximum number of phases
      INTEGER MAXPH
      PARAMETER (MAXPH=2)
C
C --- Maximum number of components/species
      INTEGER MAXK
      PARAMETER (MAXK=2)
C
C --- Maximum number of phase-dependent secondary variables other than
C     component mass fractions
      INTEGER MAXB
      PARAMETER (MAXB=8)
C
C --- Maximum number of sources/sinks
      INTEGER MAXSS
      PARAMETER (MAXSS=200)
C
C --- Maximum average number of table entries per sink/source
      INTEGER MAVTAB
      PARAMETER (MAVTAB=20)
C
C --- Maximum number of rock types
      INTEGER MAXROC
      PARAMETER (MAXROC=200)
C
C --- Maximum number of specified time steps divided by 8
      INTEGER MAXTSP
      PARAMETER (MAXTSP=5)
```

```
C
C --- Maximum number of reservoir layers for deliverability
      INTEGER MAXLAY
      PARAMETER (MAXLAY=1)
C
C --- Maximum number of parameters for a relative permeability or a
C     capillary pressure function (to get more than 7 more input lines
C     may be needed!).
      INTEGER MXRPCP
      PARAMETER (MXRPCP=7)
C
C --- Maximum points in table of ECM capillary pressure vs. saturation
      INTEGER MXPCTB
      PARAMETER (MXPCTB=30)
C
C --- Maximum number of elements with time vs. pressure boundary
C     condition, and maximum number of time vs. pressure data
      INTEGER MXTBC,MXTBPT
      PARAMETER (MXTBC=2)
      PARAMETER (MXTBPT=4500)
C
C --- Storage for MA28.  LIRN is the size of IRN and needs to be larger
C     than the number of non-zeros NZ=(NEL+2*NCON)*NEQ*NEQ.  LIRN is
C     the length of ICN and CO.
      INTEGER LICN,LIRN
      PARAMETER (LIRN=2*(MAXEL+2*MAXCON)*MAXEQ*MAXEQ)
      PARAMETER (LICN=4*(MAXEL+2*MAXCON)*MAXEQ*MAXEQ)
C
C --- Parameters for conjugate gradient package t2cg1
      INTEGER NREDM,MNZ,NRWORK,NIWORK
      PARAMETER (NREDM=MAXEQ*MAXEL)
      PARAMETER (MNZ=(MAXEL+2*MAXCON)*MAXEQ*MAXEQ)
      PARAMETER (NRWORK=1000+MNZ+38*NREDM)
      PARAMETER (NIWORK=32+MNZ+5*NREDM)

C ***********************************************************************
C ITOUGH2 parameter statements
C ***********************************************************************
C --- MAXN     : Maximum number of parameters to be estimated
      INTEGER MAXN
      PARAMETER (MAXN=20)
C
C --- MAXO     : Maximum number of datasets
      INTEGER MAXO,MAXOTWO
      PARAMETER (MAXO=40)
      PARAMETER (MAXOTWO=2*MAXO)
C
C --- MAXM     : Maximum number of calibration points
C     (approx. number of datasets times number of calibration times)
      INTEGER MAXM
      PARAMETER (MAXM=750)
C
C --- MAXPD    : Max number of paired data
      INTEGER MAXPD
      PARAMETER (MAXPD=2500)
```

```
C
C --- MAXR      : Dimension of array RPAR and IPAR, ROBS and IOBS
      INTEGER MAXR
      PARAMETER (MAXR=40)
C
C --- MAXBRK    : Max number of points in time at which SAVE file is
C                 written an reread
      INTEGER MAXBRK
      PARAMETER (MAXBRK=25)
C
C --- MAXEBRK   : Max number of elements with new initial conditions
C                 after break
      INTEGER MAXEBRK
      PARAMETER (MAXEBRK=10)
C
C --- MAXCOEFF  : Max number of coefficients for interpolation functions
      INTEGER MAXCOEFF
      PARAMETER (MAXCOEFF=5)
C
C --- MAXMCS    : Max number of Monte Carlo simulations
      INTEGER MAXMCS
      PARAMETER (MAXMCS=1000)
C
C --- MAXCURVE  : Max number of curves to be plotted
      INTEGER MAXCURVE
      PARAMETER (MAXCURVE=500)
C
C --- MAXXGR    : Max dimension of third index of array XGUESSR
      INTEGER MAXXGR
      PARAMETER (MAXXGR=3)
C
C --- MTYPE     : Max number of observation types
      INTEGER MTYPE
      PARAMETER (MTYPE=14)
C
C --- MPFMT     : Number of plot file formats
      INTEGER MPFMT
      PARAMETER (MPFMT=6)
```

# APPENDIX B. ITOUGH2 *READ.ME* FILE

```
******************************************************************************
******************    ITOUGH2  Version 3.0 October 1996  ********************
******************************************************************************


..........READ.ME.........READ.ME.........READ.ME.........READ.ME.........


   This flyer contains brief instructions for installing and running ITOUGH2
under the unix operating system. Machine-dependent routines are provided for
the following computer systems: STARDENT, IBM RS/6000, DEC ALPHA (UNIX),
HP 700 Series, Silicon Graphics, and SUN workstations.
Running ITOUGH2 on another computer system may require minor modifications
of the subroutines provided in file <mdep???.f>,

The distribution includes the source code, various utility script
files, and sample problems:


Utilities
---------

(1)  read.me        - the file you're reading.

(2)  Makefile       - UNIX makefile for compiling and linking (I)TOUGH2.

(3)  itough2        - UNIX script file for running (I)TOUGH2.
                      (in subdirectory ../bin).
                      See header of file for details.

(4)  tough2         - UNIX script file for running TOUGH2 as a dummy ITOUGH2 run.
                      (in subdirectory ../bin).
                      See header of file for details.

(5)  prista         - UNIX scrip file for displaying status of (I)TOUGH2 run.
                      (in subdirectory ../bin).
                      See header of file for details.

(6)  kit            - UNIX script file for sending signals to (I)TOUGH2.
                      (in subdirectory ../bin).
                      See header of file for details.

(7)  invdir         - dummy ITOUGH2 input file to solve direct problem only.

(8)  it2help.txt    - help file.


ITOUGH2 FORTRAN source files
----------------------------

(9)  *.inc          - include files containing COMMON blocks and PARAMETER
                      statements for dimensioning of major arrays
                      (see maxsize.inc).

(10) it2main.f      - ITOUGH2 main subroutines.
```

```
(11) it2input.f    - subroutines reading ITOUGH2 input file.

(12) it2user.f     - subroutines for user specified parameters, user
                     specified observations, user specified boundary
                     conditions, and user specified data functions.

(13) it2xxxx.f     - subroutines for minimization algorithm,
                     matrix operations, eigenanalysis, etc.

(14) mdep???.f     - machine dependent subroutines for ??? = ibm, dec, sun,
                     hp, sgi, or star.


TOUGH2 FORTRAN source files
---------------------------

(15) t2cg1.f       - contains conjugate gradient solvers.

(16) t2f.f         - core module of TOUGH2.

(17) meshm.f       - module with internal mesh generation facilities.

(18) eos#.f        - equation of state module No. #.

(19) ma28.f        - linear equation solver.


Sample problems (subdirectory <samples>)
--------------

(20) part?i        - ITOUGH2 input files  (? stands for 1, 2, 3, 4, 5)

(21) sample        - TOUGH2 input


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


   Files (15) - (19) are slightly modified versions of the standard TOUGH2
source files. All modifications are marked with a "CFi" comment character.
The main TOUGH2 program <t2m.f> is replaced by the ITOUGH2 main program.

INSTALLATION
++++++++++++

Installing ITOUGH2 requires basic knowledge about the UNIX operating system,
including shell programming, the makefile utility, changing permissions, and
adding a directory to the PATH shell variable. If ITOUGH2 is installed exactly
as recommended below, only very minor modifications have to be made to the
Makefile and the script files, if at all.

(1)  Create a new directory 'itough2' in your home directory. Type:
     cd ; mkdir itough2

(2)  Create a new directory 'bin' in your home directory (if not yet existent)
     Type:
     cd ; mkdir bin
```

(3) Copy all files of the distribution to directory ~/itough2

(4) Move the script files (tough2, itough2, prista, and kit) to subdirectory ~/bin. Type:
mv tough2 itough2 prista kit ~/bin

(5) If you want to change the dimensions of the major TOUGH2 and ITOUGH2 arrays, edit file <maxsize.inc>.

(6) Edit file <Makefile> to customize the following variables:

```
EOS = ? : Provide number of the EOS module being used.
COM = ? : Provide name of the computer system being used.
          Possibilities: ibm, sun, star, sgi, dec, and hp.
FOR = ? : name of FORtran compiler.
COO = ? : Provide COmpiler Options for compilation.
LIN = ? : Provide specific LINker options if required.
```

Compiler options are provided for IBM, SUN,STARDENT, DEC ALPHA, and HP workstations. Select the appropriate block by deleting the #-sign in the first column before COM, FOR, COO, (and LIN), and put #-signs elsewhere.

(7) If user-specified functions are required, they have to be programmed into the appropriate subroutine in file <it2user.f> (see examples therein and in the ITOUGH2 User's Guide).

(8) Customize ITOUGH2 (see ITOUGH2 User's Guide, Appendix A4), in particular: Set default plotting interface, variable IPLOTFMT in BLOCK DATA IT, file <it2main.f> (default: TECPLOT).

(9) Type "make" to run the Makefile. This compiles and links ITOUGH2. The name of the executable is <itough2_IEOS.out>, where IEOS is an integer indicating which EOS module is being used.

(10) On SUN and DEC ALPHA workstations, you may run into a severe linking error due to multiply defined subroutines. However, these compilers nevertheless create a file <itough2_IEOS.out>. This file is not executable. Type "make x" to make it executable.

(11) Add subdirectory ~/bin to the command search path (if not yet defined)
Add the following line to your ~/.cshrc file:
set path =($PATH ~/bin).
Type:
source ~/.cshrc

(12) Make sure the four script files <tough2>, <itough2>, <prista>, and <kit> in directory ../bin are executable. If not, go to directory ~/itough2 and type:
make x

(13) You may have to customize script files <prista> and <kit>.
See instructions therein.

(14) Check appropriate installation of script files:
Go to directory ~/itough2/samples, and type "prista" or "kit".
The following message should appear:
No (I)TOUGH2 simulation running!
Type "tough2" or "itough2" without any arguments. The command usage should be printed.

(15) The executable <itough2_IEOS.out> can also be used to run TOUGH2, i.e. to solve the forward problem without optimization. Running TOUGH2 as a dummy ITOUGH2 simulation assures that the same version is used to solve both the direct and the inverse problem. Furthermore, disk space can be saved since no separate TOUGH2 executable is needed.
A dummy ITOUGH2 input file <invdir> is provided, as well as a UNIX script file <tough2>.
Customize script file <tough2>, if needed:
script_dir = ? : Provide path to script file <itough2>.
Default: ~/bin

RUNNING ITOUGH2
+++++++++++++++

(1) Prepare a TOUGH2 and an ITOUGH2 input deck according to the user's manuals. For running ITOUGH2 type:

```
itough2 inv_file dir_file IEOS &
```

where:
- itough2 is the command name of the script file (or alias)
- inv_file is the file name of the ITOUGH2 input file
- dir_file is the file name of the TOUGH2 input file
- IEOS is the number of the EOS module being used

Additional options are available; type "itough2" without any arguments for a list. In order to run the sample problem, type:

```
itough2 part1i sample 3 &
```

It is important to add "&" at the end of the command line. This sends the execution of the script file to the background, which allows you to use prista, kit, and to continue your work.

The <itough2> script file generates a temporary directory ~/it2_PID. All files are then copied into this temporary directory. ITOUGH2 is executed, and the result files are copied back to your working directory. This allows one to run multiple inversions at the same time without generating conflicting file names.

(2) During execution, the status of the inverse modeling run can be displayed by running the <prista> script file. Follow instructions.

(3) If you wish to prematurely terminate an ITOUGH2 simulation or to send a signal which triggers a specific action (e.g. provides printout), use the <kit> script file and follow instructions.

Running TOUGH2
++++++++++++++

(1) Prepare a TOUGH2 input deck.

(2) Type "tough2 dir_file IEOS &" for execution, where:
- tough2 is the command name of the script file (or alias)
- dir_file is the file name of the TOUGH2 input deck
- IEOS is the number of the EOS module being used

Additional options are available; type "tough2" without any arguments for a list.

Note: Do not run TOUGH2 in the directory where file <invdir> is installed in!

Debugging
+++++++++

Run the sample problem to check the proper installation of the code. Type:

```
itough2 part1i sample 3 &
```

If no result is obtained, check:

(1) whether the script file <itough2> is executable and accessible from your working directory;

(2) whether the ITOUGH2 executable <itough2_3.out> exists;

(3) whether the path name to the ITOUGH2 executable is correct (see shell variable prog_dir in script file <itough2>);

(4) error messages in the ITOUGH2 output file (e.g. <part1i.out>);

(5) error messages in the TOUGH2 output file (e.g. <sample.out>);

(6) for error messages from the shell script (see file <part1i.msg> or <t2.msg>);

You may also rerun the sample problem using the -no_delete option, and examine all the files in the temporary directory ~/it2_PID.

SUGGESTIONS
+++++++++++

The following procedure is suggested:

(1) Use option ">>> stop after INPUT" to check ITOUGH2 input without starting the optimization; check printout of input data; resolve errors and warnings.

(2) Use option ">>> solve DIRECT problem" to run one forward calculation; check whether the TOUGH2 simulation was terminated normally; draw curves of measured and computed output (see plotfile <*i.tec>); check whether

```
    the initial guess was reasonable and whether the units and signs of your
    data were correct; check CPU time needed for one forward calculation.

(3) Perform one (1) iteration (">>> number of ITERATIONS: 1") and check the
    sensitivity coefficients; if certain parameters are not sensitive or highly
    correlated with other parameters, try to define new lumped parameters,
    or exclude the parameter from the optimization.
    Use option ">>> automatic parameter SELECTION" for a faster and more stable
    optimization.

(4) Perform optimization; set maximum number of iterations between 5 and 15.

(5) Carefully read warning and error messages in the ITOUGH2 output file.

(6) Please report code errors to the code developers.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


TOUGH2 is documented in:

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

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


ITOUGH2 is documented in:

   S. Finsterle, ITOUGH2 User's Guide, Version 2.2,
      Lawrence Berkeley Laboratory Report LBL-34581, August 1993.


Mailing address:

   Stefan Finsterle and Karsten Pruess
   Lawrence Berkeley National Laboratory
   Earth Sciences Division, MS 90-1116
   Berkeley, CA 94720
   U.S.A.

   e-mail: finster@lbl.gov
   phone:  (510) 486-5205
   fax:    (510) 486-5686

...........READ.ME..........READ.ME..........READ.ME..........READ.ME..........
```

## APPENDIX C. ITOUGH2 COMMAND INDEX

```
> PARAMETER

  >> ABSOLUTE PERMEABILITY
  >> BOTTOMHOLE PRESSURE
  >> CAPACITY
  >> CAPILLARY PRESSURE FUNCTION
  >> COMPRESSIBILITY
  >> CONDUCTIVITY
  >> DRIFT
  >> ENTHALPY
  >> FACTOR
  >> GUESS (FILE: file_name)
  >> INITIAL (PRESSURE/: ipv)
  >> KLINKENBERG
  >> LAG
  >> MINC
  >> POROSITY
  >> PRODUCTIVITY INDEX
  >> PUMPING RATIO
  >> RATE
  >> RELATIVE PERMEABILITY FUNCTION
  >> SELEC
  >> SHIFT
  >> SKIN
  >> TIME
  >> USER (: anno)

     >>> DEFAULT
     >>> MATERIAL: mat_name (mat_name_i...) (+ iplus)
     >>> MODEL
     >>> NONE
     >>> ROCK: mat_name (mat_name_i...) (+ iplus)
     >>> SET: iset
     >>> SINK: sink_name (sink_name_i ...) (+ iplus)
     >>> SOURCE: source_name (source_name_i ...) (+ iplus)

         >>>> ANNOTATION: anno
         >>>> BOUND: lower  upper
         >>>> DEVIATION: sigma
         >>>> FACTOR
         >>>> GAUSS
         >>>> GUESS: guess
         >>>> INDEX: index (index_i ...)
         >>>> LIST
         >>>> LOGARITHM
         >>>> NORMAL
         >>>> PARAMETER: index (index_i ...)
         >>>> PERTURB: (-)alpha (%)
         >>>> PRIOR: guess
         >>>> RANGE: lower  upper
         >>>> STEP: max_step
         >>>> UNIFORM
         >>>> VALUE
         >>>> VARIANCE: sigma^2
         >>>> VARIATION: sigma
         >>>> WEIGHT: 1/sigma
```

```
> OBSERVATION

  >> CONCENTRATION (comp_name/COMPONENT: icomp)
                   (phase_name/PHASE: iphase)
  >> CONTENT (phase_name/PHASE: iphase)
  >> COVARIANCE (FILE: filename)
  >> CUMULATIVE (comp_name/COMPONENT: icomp)
                (phase_name/PHASE: iphase)
  >> ENTHALPY
  >> FLOW (phase_name/PHASE: iphase)
  >> GENERATION (phase_name/PHASE: iphase)
  >> MASS FRACTION (comp_name/COMPONENT: icomp)
                   (phase_name/PHASE: iphase)
  >> PRESSURE (CAPILLARY) (phase_name/PHASE: iphase)
  >> PRODUCTION (phase_name/PHASE: iphase)
  >> RESTART TIME: ntime (time_unit)
  >> SATURATION (phase_name/PHASE: iphase)
  >> TEMPERATURE
  >> TIME: ntime (EQUAL/LOGARITHMIC) (time_unit)
  >> TOTAL MASS (comp_name/COMPONENT: icomp)
                (phase_name/PHASE: iphase) (CHANGE)
  >> USER (: anno)
  >> VOLUME (phase_name/PHASE: iphase) (CHANGE)

     >>> CONNECTION:  elem1 elem2 (elem_i elem_j ...) (+ iplus)
     >>> ELEMENT: elem (elem_i ...) (+ iplus)
     >>> GRID BLOCK: elem (elem_i ...) (+ iplus)
     >>> INTERFACE:  elem1 elem2 (elem_i elem_j ...) (+ iplus)
     >>> MODEL
     >>> SINK: sink_name (sink_namei ...) (+ iplus)
     >>> SOURCE: source_name (source_namei ...) (+ iplus)

         >>>> ABSOLUTE
         >>>> ANNOTATION: anno
         >>>> AUTO
         >>>> AVERAGE (VOLUME)
         >>>> COLUMN: itime idata (istd_dev)
         >>>> COMPONENT comp_name/: icomp
         >>>> DATA (time_unit) (FILE: file_name)
         >>>> DEVIATION: sigma
         >>>> FACTOR: factor
         >>>> FORMAT: format
         >>>> HEADER: nskip
         >>>> INDEX: index (index_i ...)
         >>>> LOGARITHM
         >>>> MEAN (VOLUME)
         >>>> PHASE phase_name/: iphase
         >>>> PARAMETER: index (index_i ...)
         >>>> POLYNOM: idegree
         >>>> RELATIVE: rel_err (%)
         >>>> SET: iset
         >>>> SHIFT: shift (TIME)
         >>>> SKIP: nskip
         >>>> SUM
         >>>> USER
         >>>> VARIANCE: sigma^2
         >>>> WEIGHT: 1/sigma
         >>>> WINDOW: time_A time_B
```

```
> COMPUTATION

  >> CONVERGE/STOP/TOLERANCE

     >>> ADJUST TIMESTEP
     >>> CONSECUTIVE: max_iter1
     >>> FORWARD
     >>> INCOMPLETE: max_incomplete
     >>> INPUT
     >>> ITERATION: max_iter
     >>> LEVENBERG: lambda
     >>> MARQUARDT: mue
     >>> REDUCTION: max_red
     >>> SIMULATION: max_tough2
     >>> STEP: max_step
     >>> UPHILL: max_uphill
     >>> WARNING

  >> ERROR

     >>> ALPHA: alpha (%)
     >>> FISHER
     >>> FOSM (MATRIX: ndim) (ITOUGH2) (CORRELATION) (DIAGONAL)
     >>> HESSIAN
     >>> LINEARITY (: alpha (%))
     >>> MONTE CARLO (SEED: iseed) (GENERATE) (CLASS: iclass)
     >>> POSTERIORI
     >>> PRIORI
     >>> TAU: (-)niter

  >> JACOBI

     >>> CENTER
     >>> FORWARD (: iswitch)
     >>> HESSIAN
     >>> PERTURB : (-)perturb (%)

  >> OPTION

     >>> ANDREW: c
     >>> ANNEAL

         >>>> ITERATION: max_iter
         >>>> SCHEDULE: (-)beta
         >>>> STEP: max_step
         >>>> TEMPERATURE: (-)temp

     >>> CAUCHY
     >>> DIRECT
     >>> FORWARD
     >>> GRID-SEARCH: ninval1 (ninvali...)
     >>> GAUSS NEWTON
     >>> L1-ESTIMATOR
     >>> LEAST-SQUARE
     >>> LEVENBERG-MARQUARDT
     >>> OBJECTIVE: ninval1 (ninvali...)
     >>> QUADRATIC-LINEAR: c
```

```
      >>> SELECT

          >>>> CORRELATION: (-)rcorr
          >>>> ITERATION: niter
          >>>> SENSITIVITY: (-)rsens

      >>> SENSITIVITY
      >>> STEADY-STATE (SAVE) (: max_time_step)

   >> OUTPUT
      >>> CHARACTERISTIC
      >>> COVARIANCE
      >>> FORMAT: format (LIST)
      >>> INDEX
      >>> LIST
      >>> JACOBI
      >>> OBJECTIVE
      >>> PERFORMANCE
      >>> PLOTFILE: format (LIST)
      >>> PLOTTING: niter
      >>> SENSITIVITY
      >>> time_unit
      >>> UPDATE
      >>> RESIDUAL
      >>> VERSION
```

## APPENDIX D. ITOUGH2 SAMPLE PROBLEM

### D.1 Introduction

The purpose of this sample problem is to develop a sequence of simulations that demonstrate the three key applications of ITOUGH2, namely (i) sensitivity analysis for experimental design, (ii) parameter estimation by data inversion, and (iii) error propagation analysis. We focus here on a step-by-step description of the development of ITOUGH2 input files rather than on the interpretation of the inverse modeling results.

Before running these sample problems, ITOUGH2 must be properly installed according to the instructions provided in file *read.me*.

A synthetic laboratory experiment was chosen for its simplicity. A schematic of the experimental layout is shown in Figure D.1.1. Water is injected under constant pressure into a one-dimensional, horizontal column filled with uniform, partially saturated sand. This setup is similar to the one used for a standard steady-state Darcy experiment. However, there is a certain amount of free gas initially present in the column, and information about the transient behavior of pressures and flow rates are used to determine two-phase flow parameters.

The TOUGH2 input file for this sample problem is shown in Figure D.1.2.

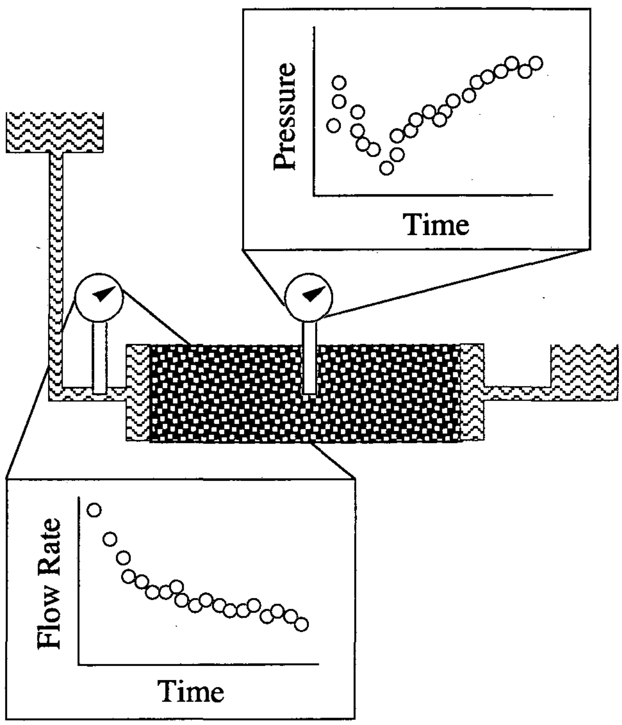


***Figure D.1.1.*** *Schematic of synthetic laboratory experiment*

```
TOUGH2 input file for simulating two-phase,transient "Darcy" experiment
ROCKS----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
SAND      1      2650.      .3500 2.000E-12                              2.51      920.
 1.000E-08
BOUND     0      2650.      .9900 2.000E-12                              2.51   100000.

RPCAP----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    3       0.200E+00 0.050E+00
    1       0.000E+00 0.000E+00 1.000E+00
PARAM----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
   2 500     99991000001000000000400003000
 0.000E+00 6.000E+02 2.500E+00 1.000E+01

 100000.000000000000 10.30000000000000  20.000000000000000
MULTI----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
    2    2    2    6
ELEME----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
IN              BOUND .1000E+50                     .0000E+00
A11 1    49     1SAND  .1000E-03                     .1000E-01
OUT             BOUND .1000E+50                     .5000E+00

CONNE----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
IN   A11 1                    1 .1000E-10 .5000E-02 .1000E-01
A11 1A11 2   48     1     1    1 .5000E-02 .5000E-02 .1000E-01
A1150OUT                      1 .5000E-02 .1000E-10 .1000E-01

START----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
INCON----1----*----2----*----3----*----4----*----5----*----6----*----7----*----8
IN
 110000.000000000000 0.00000000000000000  20.000000000000000
OUT
 100000.000000000000 0.00000000000000000  20.000000000000000

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

***Figure D.1.2.*** *TOUGH2 input file* sample

We assume that the objective of the experiment is to estimate the permeability and the porosity of the sand as well as the initial gas content. Furthermore, we assume that only one flow meter and one pressure transducer are available for data collection. The measurement uncertainty of the two instruments is 5 ml/min and 200Pa, respectively.

Step-by-step instructions will be given. In the first part (Section D.2), we use ITOUGH2 to simply solve the forward problem, producing a plot file, and at the same time generating the synthetic data base for the subsequent inversions. In Section D.3, a sensitivity analysis is performed to determine the optimum location of the flow meter and pressure transducer. This demonstrates how ITOUGH2 can be used to design an experiment. The estimation of the three parameters of interest based on synthetic flow rate and pressure data is discussed in Section D.4. Finally, we compare the uncertainy of the model predictions by using either linear error propagation analysis (FOSM), or Monte Carlo simulations (Section D.5). The specific ITOUGH2 feature addressed by each part of the sample problem is summarized in Table D.1.1. CPU time requirements for an IBM RS/6000 370 workstation are also indicated.

***Table D.1.1.*** *Summary of Issues Addressed by Sample Problem and CPU time Requirement for IBM RS/6000 370 Workstation.*

| Part | CPU time | Application/ITOUGH2 features |
|---|---|---|
| 1 | 6 sec | Solve forward problem using ITOUGH2 |
| 2 | 7 sec | Generate time series plots |
| 3 | 30 sec | Sensitivity analysis for test design |
| 4 | 2 min | Parameter estimation based on synthetic data |
| 5 | 45 sec | FOSM error propagation analysis |
| 6 | 10 min | Monte Carlo simulations |

## D.2 Part 1 and Part 2: Solving the Forward Problem with ITOUGH2

It is very important for any inverse modeling effort to solve the forward problem in a stable, robust, and efficient manner. It is therefore imperative to perform a standard TOUGH2 simulation and to check the appropriateness of the results, before more time-consuming inversions are initiated. Plotting the simulation results obtained with the initial parameter set against the data is also strongly recommended because frequent errors such as a wrong conversion factor, opposite signs of measured and calculated flow rates, time lags and data shifts, etc. can readily be detected. Furthermore, it provides a first assessment of the model, and allows one to estimate the CPU-time required to solve the inverse problem.

A single TOUGH2 simulation can be performed by typing the following command:

```
tough2 sample 3 &
```

This command makes use of the unix script file *tough2* which automatically takes file *~/itough2/invdir* as a dummy ITOUGH2 input file. If using this approach, potential error messages are printed to file *t2.msg*.

An alternative way is to write a short ITOUGH2 input file (*part1i*) as shown in Figure D.2.1:

```
> COMPUTATION
  >> OPTION
     >>> solve FORWARD problem only
     <<<
  <<
<
```

***Figure D.2.1.*** *Part 1: ITOUGH2 input file* part1i. *Solves direct problem.*

Use the following command to solve all parts of the sample problem:

```
itough2 part#i sample 3 &
```

where "#" indicates the part number.

Since no parameters or observations are provided, the ITOUGH2 output file *part1i.out* contains warning messages which can be ignored in this specific case. In general, however, one should always consult the following files to check for potential errors:

(i) the ITOUGH2 output file (e.g., *part1i.out*)
(ii) the TOUGH2 output file (e.g., *sample.out*)
(iii) the ITOUGH2 message file (e.g., *part1i.msg*)

In Part 2, we use ITOUGH2 to generate a time plot of pressure and flow rates at two selected points within the column. The plotfile will later be used as the synthetic data base for the inversion. ITOUGH2 provides convenient options in block `> OBSERVATIONS` to select points in space and time at which the value of certain TOUGH2 output variables can be examined. Here, we are interested in the pressure [Pa] in the center of the column, and the flow rate [ml/min] at the inlet. Sixty equally spaced points in time between 10 and 600 seconds are selected to generate the time series. Columns of time versus calculated flow rates and pressures are written to file *part2i.col* which can be processed by most visualization packages for plotting. The corresponding ITOUGH2 input file is shown in Figure D.2.2.

Note that the location for the pressure variable is identified by the grid block name at the center of the column. For flow rates, the two grid block names defining the connection at the inlet have to be specified. Since no measured data are available, command `>>>> NO DATA` is added which automatically generates dummy data points of value $10^{-50}$. A multiplication factor of -1.6667E-5 is specified, converting positive flow rates in units of [ml/min] to the flow rates calculated by TOUGH2 which are negative and in [kg/sec]. Note that the sign of the calculated flow rate is the result of a convention, i.e., it is arbitrarily defined by the ordering of elements in a connection. This convention has to be accounted for by choosing the correct sign for the conversion factor.

By default, the format of the plotfile concurs with TECPLOT. In order to arrange the plotfile in columns, command `>>> PLOTFILE: COLUMN` is used. The first column in file *part2i.col* (see Figure D.2.3) holds the times in the specified time units; the second and third column contain the observed data (or $10^{-50}$ if no data are provided) and the corresponding TOUGH2 output, respectively, for the first data set (e.g., pressures at element A1125); columns four and five contain the measured and calculated system response for the second data set (e.g., liquid flow rate across interface IN__0 A11_1), and so on. File *part2i.col* will be used in Part 4 as the file from which the synthetic data are extracted.

This simulation produces a relatively large TOUGH2 output file *sample.out* even for this small problem because full results are printed at 60 printout times. Specifying a negative number for parameter KDATA in TOUGH2 block PARAM.1 suppresses the printout of state variables in the TOUGH2 output file, saving disk space and making the inverse runs faster.

```
> OBSERVATION

  >> select : 60 points in TIME, EQUALLY spaced between
     10.0 600.0 seconds

  >> PRESSURE
     >>> ELEMENT          : A1125
         >>>> NO DATA needed
         <<<<
     <<<

  >> LIQUID FLOW RATE
     >>> CONNECTION       : IN__0  A11_1
         >>>> FACTOR      :-1.666667E-05 (ml/min - kg/sec)
         >>>> NO DATA needed
         <<<<
      <<<
  <<

> COMPUTATION

  >> OPTION
     >>> solve FORWARD problem only
     <<<

  >> OUTPUT
     >>> LIST all available plot file FORMATS and select: COLUMNS
     >>> output in MINUTES
     <<<
  <<
<
```

***Figure D.2.2**. Part 2: ITOUGH2 input file* part2i. *Solves forward problem and generates column file* part2i.col *with flow rates and pressures as a function of time.*

| TIME [min] | DATA P(GAS) A1125 | SIM0 P(GAS) A1125 | DATA F-L IN 0 A11 1 | SIM0 F-L IN 0 A11 1 |
|---|---|---|---|---|
| .16666667E+00 | .10000000E-49 | .10302017E+06 | .10000000E-49 | .85780174E+02 |
| .33333333E+00 | .10000000E-49 | .10351308E+06 | .10000000E-49 | .65323231E+02 |
| .50000000E+00 | .10000000E-49 | .10353113E+06 | .10000000E-49 | .57637722E+02 |
| .66666667E+00 | .10000000E-49 | .10340909E+06 | .10000000E-49 | .53653780E+02 |
| .83333333E+00 | .10000000E-49 | .10325474E+06 | .10000000E-49 | .50982470E+02 |
| .10000000E+01 | .10000000E-49 | .10310265E+06 | .10000000E-49 | .48876966E+02 |
| .11666667E+01 | .10000000E-49 | .10296287E+06 | .10000000E-49 | .47075537E+02 |
| ............ | ............. | ............ | ............ | ............ |
| .96666667E+01 | .10000000E-49 | .10482543E+06 | .10000000E-49 | .21429127E+02 |
| .98333333E+01 | .10000000E-49 | .10487014E+06 | .10000000E-49 | .21269789E+02 |
| .10000000E+02 | .10000000E-49 | .10491346E+06 | .10000000E-49 | .21109160E+02 |

***Figure D.2.3**. Part 2: Excerpt from ITOUGH2 plot file* part2i.col. *Time is in the first column, followed by observed and simulated state variables for each data set. This file will be used as the data file in Part 4.*

## D.3 Part 3: Sensitivity Analysis

Part 3 demonstrates how ITOUGH2 is used to perform a sensitivity analysis. By calculating global sensitivity measures for each parameter and each observation type, alternative experimental configurations can be tested against each other, and the design most suitable for estimating parameters in a subsequent inversion can be identified. As in any sensitivity analysis involving non-linear processes, prior knowledge about the parameters has to be available, or the analysis has to be repeated for different parameter combinations. Furthermore, the parameters of interest and their potential variations have to be specified, as well as the type, location, and expected uncertainty of the measurements. Since the experiment has to be optimized with respect to the estimation of permeability, porosity, and initial gas saturation, these three parameters and their respective variations have to be defined in the `> PARAMETER` block of the ITOUGH2 input file (Figure D.3.1).

The permeability of both rock types, SAND and BOUND, are to be varied simultaneously. Note that the initial gas saturation is referred to as primary variable no. 2 of the default initial condition, as specified in TOUGH2 block PARAM.4. Potential parameter variations are specified, scaling the sensitivity coefficients of the Jacobian matrix. For permeability, the variation of the logarithm has to be provided. No parameter annotation is given for porosity, prompting ITOUGH2 to generate an identifier (see output file *part3i.out*).

In the `> OBSERVATION` block, pressures in elements A1112, A1125, and A1138 are examined. The element names correspond to three potential measuring locations within the column, at a distance of 1/4, 1/2, and 3/4 of the total column length from the inlet, respectively. Furthermore, the connections defining the inlet and the outlet are given as the two alternative measurement points for liquid flow rates.

The program option `>>> SENSITIVITY ANALYSIS` is invoked which makes ITOUGH2 run $n + 1 = 4$ TOUGH2 simulations to calculate the Jacobian matrix. Note that option `>>> FORWARD` has to be disabled. This can be done by (i) deleting the corresponding line, (ii) replacing the comand level indicator ">>>" by blanks (or any other character), or (iii) by surrounding the line with "/*" and "*/" comment characters.

The Jacobian matrix and the covariance matrix of the parameters provide the basis for a detailed sensitivity analysis. We notice from the global sensitivity measures shown in Figure D.3.2 that the highest senstivity is realized for the pressure measurements in the center of the column, and that the flow rate data at the inlet contain significantly more information than the flow rate data at the outlet, suggesting implementation of the experimental configuration shown in Figure D.1.1.

```
> PARAMETER

  >> ABSOLUTE permeability
     >>> MATERIAL         : SAND_ BOUND
         >>>> ANNOTATION : log(abs. perm.)
         >>>> LOGARITHM
         >>>> VARIATION  : 0.3
         <<<<
      <<<

  >> POROSITY
     >>> MATERIAL         : SAND_
         >>>> VALUE
         >>>> VARIATION  : 0.10
         <<<<
     <<<

  >> INITIAL condition for primary variable No.: 2
     >>> DEFAULT
         >>>> ANNOTATION : Gas entrapped
         >>>> VALUE
         >>>> VARIATION  : 0.10
         <<<<
     <<<
  <<

> OBSERVATION

  >> select : 60 points in TIME, EQUALLY spaced between
     10.0 600.0 seconds

  >> PRESSURE

     >>> ELEMENT          : A1112
         >>>> ANNOTATION : Pressure 1/4
         >>>> NO DATA available
         >>>> DEVIATION  : 200.0 Pa  (expected measurement error)
         <<<<

     >>> ELEMENT          : A1125
         >>>> ANNOTATION : Pressure 1/2
         >>>> NO DATA available
         >>>> DEVIATION  : 200.0 Pa
         <<<<

     >>> ELEMENT          : A1138
         >>>> ANNOTATION : Pressure 3/4
         >>>> NO DATA available
         >>>> DEVIATION  : 200.0 Pa
         <<<<
     <<<
```

***Figure D.3.1.*** *Part 3: ITOUGH2 input file* part3i. *Performs sensitivity analysis of 5 observations with respect to 3 parameters.*

```
  >> LIQUID FLOW RATE
     >>> CONNECTION       : IN__0  A11_1
         >>>> ANNOTATION : Flow inlet
         >>>> FACTOR     :-1.666667E-05 (ml/min - kg/sec)
         >>>> NO DATA available
         >>>> DEVIATION  : 5.0 ml/min   (expected measurement error)
         <<<<
     >>> CONNECTION       : A1150  OUT_0
         >>>> ANNOTATION : Flow outlet
         >>>> FACTOR     :-1.666667E-05 (ml/min - kg/sec)
         >>>> NO DATA available
         >>>> DEVIATION  : 5.0 ml/min
         <<<<
     <<<
  <<


> COMPUTATION
  >> OPTION
     ... solve FORWARD problem only
     >>> perform SENSITIVITY analysis
     <<<

  >> OUTPUT
     >>> PLOTFILE contains : COLUMNS
     <<<
  <<
<
```

***Figure D.3.1 (cont.).*** *Part 3: ITOUGH2 input file* part3i. *Performs sensitivity analysis of 5 observations with respect to 3 parameters.*

```
===========================================================================
                               log(abs. perm.)  POROSITY SAND  Gas entrapped    Total
---------------------------------------------------------------------------
Total from data  Pressure 1/4         274.9          113.3          262.4    650.5
Total from data  Pressure 1/2         440.1          180.9          370.1    991.0 *
Total from data  Pressure 3/4         289.3          116.9          255.8    662.1
Total from data  Flow inlet           168.2           41.5          119.6    329.4 *
Total from data  Flow outlet            9.2            2.4           30.5     42.2
---------------------------------------------------------------------------
Total parameter sensitivity          1181.7          455.0         1038.4
===========================================================================
```

***Figure D.3.2.*** *Part 3: Excerpt from ITOUGH2 output file* part3i.out, *showing global sensitivity measures.*

## D.4 Part 4: Parameter Estimation

Part 4 demontrates the main application of ITOUGH2, i.e., the estimation of TOUGH2 input parameters by inverse modeling. From the sensitivity analysis discussed in Section D.3 it was concluded that it is most advantageous to use pressure data at the center of the column and flow rate data at the inlet to estimate the three paramaters of interest. In this synthetic experiment, the data were not actually measured, but obtained by forward simulation (see Part 2). The third and fifth column in file *part2i.col* contain the pressure and flow rate data, respectively, as a function of time which is stored in column 1.

While the true parameters $\mathbf{p}_{true}^T = (-11.7, 0.35, 10.30)$ are given by the TOUGH2 input file *sample* used to generate the data, we pretend not to know their values. An initial guess $\mathbf{p}_o^T = (-12.0, 0.25, 10.25)$ is provided for each parameter through the ITOUGH2 input file *part4i* (see Figure D.4.1). Starting from this initial parameter set, correct identification of the true parameter set may serve as a verification of the ITOUGH2 minimization algorithm.

An admissible range has been specified for the third parameter in order to prevent ITOUGH2 from suggesting parameter values that lead to an error in the TOUGH2 simulation. While the third parameter is initial gas saturation with physical values in the interval $0 \le S_{gi} \le 1$, the admissible range for the corresponding TOUGH2 input parameter is $10.0 < \mathrm{DEP}(2) < 11.0$.

The simulation results are compared at sixty points in time to the data provided on file *part2i.col* (data could also be directly supplied in the ITOUGH2 input file *part4i*). If reading from an external file, keyword `FILE` has to be present, and the file name and time units have to be given on the `>>>> DATA` command line. The data file *part2i.col* (see Figure D.2.3) contains header lines that have to be skipped before actual data can be read (command `>>>> HEADER`). Furthermore, the columns holding the times (default: column no. 1) and observed values (default: column no. 2) are indicated using command `>>>> COLUMN`.

In the `> COMPUTATION` block, the number of ITOUGH2 iterations is being limited to 5. The input file generates 3 warning messages which can be ignored.

```
> PARAMETER
  >> ABSOLUTE permeability
     >>> MATERIAL                : SAND_ BOUND
         >>>> ANNOTATION         : log(abs. perm.)
         >>>> LOGARITHM
         >>>> initial GUESS      : -12.0
         >>>> VARIATION          : 0.3
         <<<<
      <<<
  >> POROSITY
     >>> MATERIAL                : SAND_
         >>>> initial GUESS      : 0.25
         >>>> VARIATION          : 0.10
         <<<<
     <<<
  >> INITIAL condition for primary variable No.:2
     >>> DEFAULT
         >>>> ANNOTATION         : Gas entrapped
         >>>> VALUE
         >>>> initial GUESS      : 10.25
         >>>> admissible RANGE : 10.01 10.99
         >>>> VARIATION          : 0.10
         <<<<
     <<<
  <<

> OBSERVATION

  >> select : 60 points in TIME, EQUALLY spaced between
     10.0 600.0 seconds
  >> PRESSURE
     >>> ELEMENT                    : A1125
         >>>> ANNOTATION            : Pressure 1/2
         >>>> HEADER contains       : 2 lines
         >>>> COLUMNS               : 1 3          (time vs. pressure)
         >>>> Read DATA from FILE : part2i.col    (time is in MINUTES)
         >>>> standard DEVIATION    : 200.0 Pa     (measurement error)
         <<<<
     <<<
  >> LIQUID FLOW RATE
     >>> CONNECTION defining inlet: IN__0  A11_1
         >>>> ANNOTATION            : Flow inlet
         >>>> FACTOR                :-1.666667E-05 (ml/min - kg/sec)
         >>>> HEADER contains       : 2 lines
         >>>> COLUMNS               : 1 5          (time vs. flow rate)
         >>>> Read DATA from FILE : part2i.col    (time is in MINUTES)
         >>>> standard DEVIATION    : 5.0 ml/min   (measurement error)
         <<<<
     <<<
  <<
```

***Figure D.4.1.*** *Part 4: ITOUGH2 input file* part4i. *Performs parameter estimation by inverse modeling.*

```
> COMPUTATION

  >> STOP
     >>> after : 5  ITERATIONS
     >>> ignore WARNINGS
     <<<

  >> ERROR
     >>> use A PRIORI error variance for error analysis
     <<<

/*
  >> OPTION
     ... solve FORWARD problem only
     >>> perform SENSITIVITY analysis
     <<<

  >> OUTPUT
     >>> PLOTFILE contains : COLUMNS
     <<<
*/
  <<
<
```

***Figure D.4.1 (cont.).*** *Part 4: ITOUGH2 input file* part4i. *Performs parameter estimation by inverse modeling.*

By default, the error analysis is based on the *a posteriori* or estimated error variance which is calculated from the final residuals. In our case, however, the estimated error variance would be very close to zero because no random noise representing measurement errors has been added to the synthetic data. The command `>>> A PRIORI` makes ITOUGH2 use the *apriori* defined error variance for the error analysis, i.e., it is assumed that the final residuals exhibit a standard deviation of 200 Pa and 5 ml/min, respectively. Blocks `>> OPTION` and `>> OUPUT` are deactivated by the comment characters "`/*`" and "`*/`", making ITOUGH2 perform the default application, i.e., parameter estimation by means of the Levenberg-Marquardt minimization algorithm. The plot file will be generated using the default time units (seconds) and the default format instead of columns. The default plot file format, which is TECPLOT (plot file extension *.tec*), can be changed by redefining variable `IPLOTFMT` in `BLOCK DATA IT`, file *it2main.f*.

The inversion is started by typing

```
itough2 part4i sample 3 &
```

The optimization process can be followed during execution by typing the command `prista` which displays the current status of the inversion, i.e., the number of TOUGH2 runs and ITOUGH2 iterations completed, parameter updates and current parameter values, reduction and current value of objective function, etc. Repeated use of `prista` may suggest termination of the inversion before the specified maximum number of ITOUGH2 iterations has been reached because no significant reduction of the objective function can be achieved. Termination is supported by the `kit` command which ensures that complete output is generated before execution of ITOUGH2 is stopped.

After completion of the inversion, results are written to various output files. The main ITOUGH2 output file is named *part4i.out*, and contains optimization statistics, error and residual analysis, and the best estimate parameter set. File *sample.out* contains the TOUGH2 output from the last simulation which is in most cases the run with the best estimate parameter set. Additional messages can be found in file *part4i.msg*. The values of the best estimate parameter set are also written to file *part4i.par* for convenient restarting of an ITOUGH2 run (see Part 5). The plot file *part4i.tec* contains the interpolated data at the calibration points, the simulated system response using the initial parameter set, and the simulated system response using the best estimate parameter set.

The symbols in Figures D.4.2 and D.4.3 represent the synthetic pressure and flow rate data, respectively. The simulation results obtained with the initial and final parameter set are shown as dashed and solid lines, respectively. The perfect match demonstrates that the minimum of the objective function is accurately identified within 5 iterations. The estimated and true parameter sets are identical, verifying parameter estimation by ITOUGH2 for this well-posed inverse problem.

Part 4 of this sample problem is convenient to explore many ITOUGH2 features. It is suggested to perform a variety of additional inversions to test the capability of ITOUGH2. For example, minimization could be started from different initial parameter guesses, noisy data could be generated and used for inversion, and systematic errors can be introduced to study their impact on the estimates. Additional or different parameters can be determined, such as the pore space compressibility (instead of or in addition to the initial gas saturation), boundary pressure at the inlet, or parameters of the relative permeability and capillary pressure functions can be subjected to the estimation process. Furthermore, the user should experiment with different options for defining parameters, observations, and data. The use of `prista` and `kit` can also be practiced, i.e., runs can be prematurely terminated and restarted, etc.

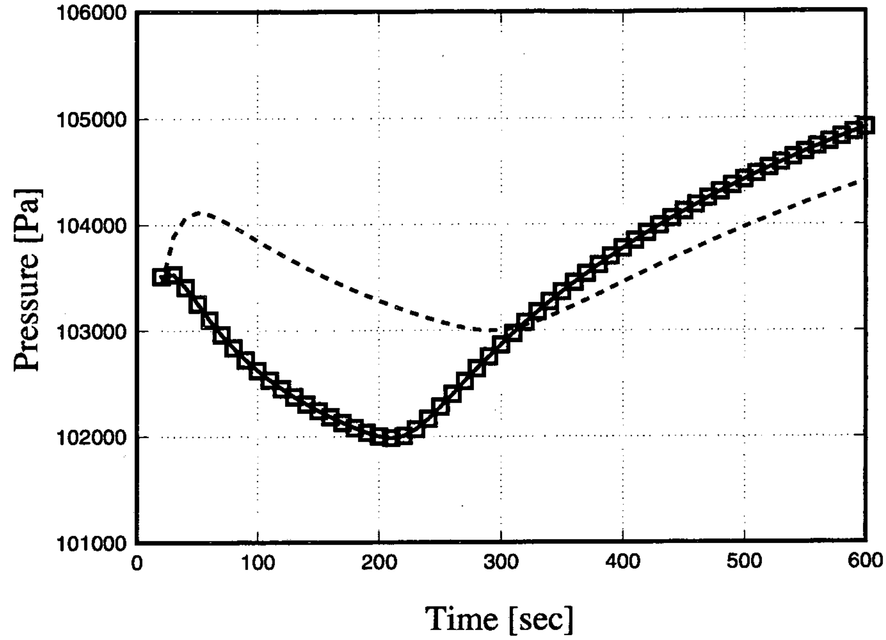


*Figure D.4.2. Part 4: Pressure transient at center of column calculated with initial parameter set (dashed line) and after optimization (solid line). Synthetic data are shown as squares.*

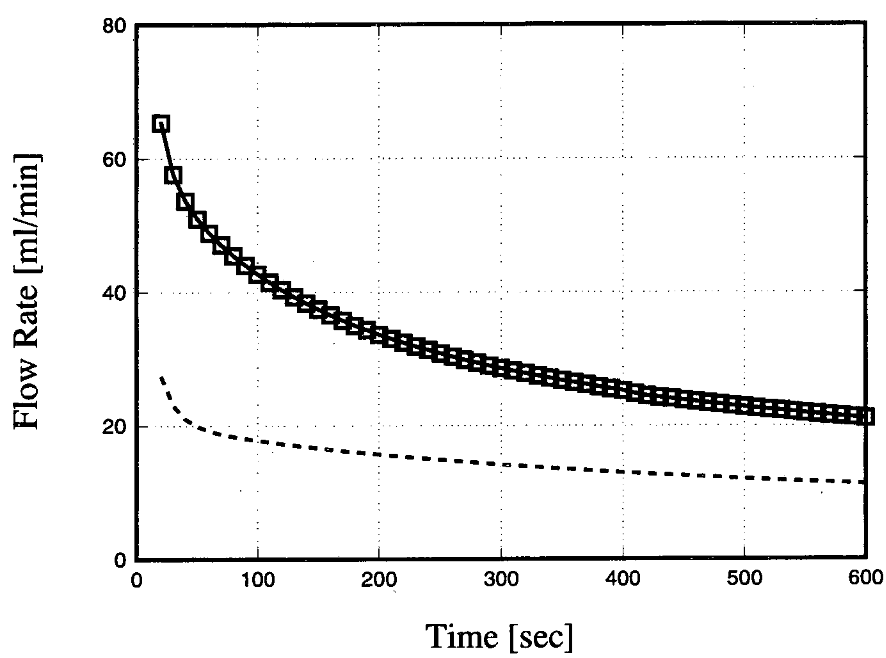


*Figure D.4.3. Part 4: Flow rates at inlet calculated with initial parameter set (dashed line) and after optimization (solid line). Synthetic data are shown as squares.*

## D.5 Part 5: Error Propagation Analysis

ITOUGH2 offers two methods to study the effect of parameter uncertainty on model predictions: (1) first-order-second-moment (FOSM) error propagation analysis, and (ii) Monte Carlo simulations. For small standard deviations of the input parameters, and if the model output can be approximated by a linear function of the parameters within the range of the error band, FOSM is a fast method to calculate a measure of prediction uncertainty which is easy to report. If the model is highly non-linear, and the uncertainties of the input parameters are large, Monte Carlo simulations have to be performed which examine many parameter combinations. The distribution of the output may not be normal or log-normal, i.e., a full distribution analysis can be performed. Monte Carlo simulations are very flexible in handling non-Gaussian distributions of both input parameters and output variables, but they are computationally expensive and results are difficult to report.

In this sample problem we compare both approaches, and at the same time introduce a few additional ITOUGH2 options which are useful for many other ITOUGH2 applications as well.

The standard deviations of three uncorrelated TOUGH2 input parameters, $\log(k)$, $\phi$, and $S_{gi}$, are assumed to be 0.1, 0.05, and 0.05, respectively. The best estimates of the three parameters are taken from the previous inversion, and are directly read from file *part4i.par*. This ITOUGH2 option (see Figure D.5.1) allows the execution of a sequence of problems in series as a batch job. Performing a simulation of a synthetic laboratory experiment, we are interested in the uncertainty of the model prediction, e.g., the uncertainty of the pressure in the center of the column. The laboratory experiment consists of three parts: (1) injection of water into a partially saturated sand column for 5 minutes under constant pressure, (2) injection of gas for 2.5 minutes, followed by (3) a 2.5 minute shut-in recovery period. In standard TOUGH2, the three test events would have to be run separately in sequence where the simulation is stopped after 5 and 7.5 minutes, and restarted after adjustment of the boundary condition at the inlet. In ITOUGH2, however, it is necessary to handle all three test events in a single TOUGH2 simulation. This requires automatic adjustment of boundary conditions at $t = 5.0$ and $t = 7.5$ minutes. While general, time-dependent boundary conditions can be supplied through subroutine USERBC, simple changes of primary variables and element volumes can be conveniently specified directly in the ITOUGH2 input file using the `>>>> RESTART` option (see Figure D.5.1).

```
> PARAMETER

  >> Take first GUESS (=mean) from FILE: part4i.par

  >> ABSOLUTE permeability
     >>> MATERIAL                  : SAND_ BOUND
         >>>> ANNOTATION           : log(abs. perm.)
         >>>> LOGARITHM
         >>>> standard DEVIATION : 0.10
         <<<<
      <<<
  >> POROSITY
     >>> MATERIAL                  : SAND_
         >>>> VALUE
         >>>> admissible RANGE   : 0.01 0.99
         >>>> standard DEVIATION : 0.05
         <<<<
     <<<
  >> INITIAL condition for primary variable No.:2
     >>> DEFAULT
         >>>> ANNOTATION           : Gas entrapped
         >>>> VALUE
         >>>> admissible RANGE   : 10.01 10.99
         >>>> standard DEVIATION : 0.05
         <<<<
     <<<
  <<

> OBSERVATION

  >> select : 60 points in TIME, EQUALLY spaced between
     10.0 600.0 seconds

  >> RESTART TIME: 1  in [MINUTE]
     5.0
     IN__0  2  10.99   (replace water by air in injection grid block)

  >> RESTART TIME: 1  in [MINUTE]
     7.5
     IN__0  0  1.0E-06 (reduce volume for shut-in recovery)

  >> PRESSURE
     >>> ELEMENT                   : A1125
         >>>> ANNOTATION           : Pressure 1/2
         >>>> NO DATA (this is a prediction)
         <<<<
     <<<
  <<
```

***Figure D.5.1.*** *Part 5: ITOUGH2 input file* part5i. *Examines prediction uncertainty using first-order-second-moment (FOSM) error propagation analysis.*

```
> COMPUTATION

  >> JACOBIAN
     >>> use CENTERED finite difference quotient and
     >>> PERTURB by as much as: 5 %
     <<<

  >> ERROR
     >>> draw error band on (1-ALPHA)=: 95 % confidence level
     >>> First-Order-Second-Moment (FOSM) error propagation analysis
     <<<
  <<
<
```

***Figure D.5.1 (cont.).*** *Part 5: ITOUGH2 input file* part5i. *Examines prediction uncertainty using first-order-second-moment (FOSM) error propagation analysis.*

For FOSM analysis it is suggested to use a relatively large perturbation factor of 5 % in combination with a centered finite difference quotient. The plotfile *part5i.tec* contains the predicted pressure for the mean parameter values as well as the upper and lower bound of the error band on the confidence level specified by the `>>> ALPHA` command.

In order to invoke Monte Carlo simulations, only the `> COMPUTATION` block has to be adjusted as shown in Figure D.5.2. Various seed numbers should be tried in combination with keyword `GENERATE only`, until a satisfactory distribution of the input parameters has been achieved as shown in Figure D.5.3. Then, keyword `GENERATE` can be deleted to invoke the actual Monte Carlo simulations. Make sure that the parameter range is specified in the `> PARAMETER` block.

```
> COMPUTATION

  >> STOP
     >>> after :100 Monte Carlo SIMULATIONS
     <<<

  >> ERROR
     >>> MONTE CARLO (SEED: 777, G_ENERATE only)
     <<<
  <<
<
```

***Figure D.5.2.*** *Part 6: Excerpt of ITOUGH2 input file* part6i. *Examines prediction uncertainty using Monte Carlo simulations.*

```
================================================================================
Parameter No.  1 : log(abs. perm.)  Distribution    :          Normal
Mean             :    -.117208E+02  Std. Deviation :    .100016E+00
Lower bound      :    -.120347E+02  Upper bound     :   -.114310E+02

 -.120045E+02     1   *
 -.119441E+02     1   *
 -.118837E+02     3   ***
 -.118234E+02    19   *******************
 -.117630E+02    25   *************************
 -.117026E+02    21   *********************
 -.116423E+02    18   ******************
 -.115819E+02     8   ********
 -.115215E+02     2   **
 -.114612E+02     2   **

================================================================================
Parameter No.  2 :   POROSITY SAND  Distribution    :          Normal
Mean             :     .345833E+00  Std. Deviation :    .496882E-01
Lower bound      :     .226377E+00  Upper bound     :    .458681E+00

  .237992E+00     3   ***
  .261222E+00     5   *****
  .284453E+00     9   *********
  .307683E+00    15   ***************
  .330914E+00    13   *************
  .354144E+00    19   *******************
  .377374E+00    16   ****************
  .400605E+00     9   *********
  .423835E+00     8   ********
  .447065E+00     3   ***

================================================================================
Parameter No.  3 :   Gas entrapped  Distribution    :          Normal
Mean             :     .102915E+02  Std. Deviation :    .498949E-01
Lower bound      :     .101773E+02  Upper bound     :    .104157E+02

  .101892E+02     2   **
  .102130E+02     5   *****
  .102369E+02    14   **************
  .102607E+02    20   ********************
  .102845E+02    16   ****************
  .103084E+02    11   ***********
  .103322E+02    14   **************
  .103561E+02     9   *********
  .103799E+02     7   *******
  .104037E+02     2   **

================================================================================
```

***Figure D.5.3.*** *Part 6: Excerpt of ITOUGH2 output file* part6i.out, *showing the distribution of the uncertain input parameters.*

The results from both the FOSM and Monte Carlo uncertainty analysis are visualized in Figure D.5.4. While the linear FOSM analysis gives a reasonable estimate of prediction uncertainty for most parts of the experiment, the Monte Carlo simulations reveal an asymmetry of the output distribution in the period where non-linearities prevail. Note that FOSM analysis assigns a certain probability to pressure responses that are below 1 bar which is physically not possible. The Monte Carlo simulations stay away from that lower bound, yielding the largest differences between the two methods. The outliers leading to significantly higher pressures result from a parameter combination of low permeability, high porosity, and low initial gas saturation.

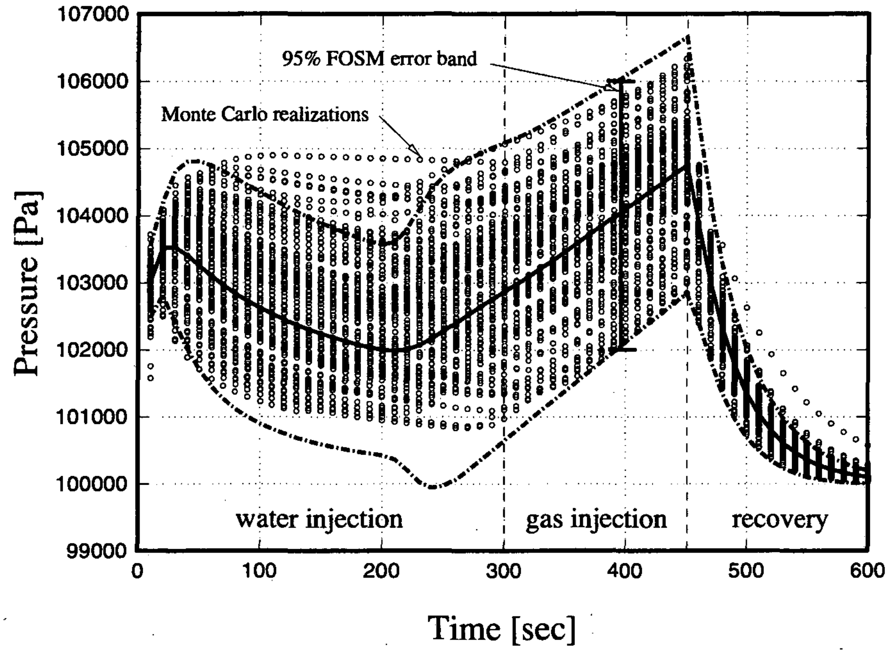


***Figure D.5.4.*** *Part 5/6: Comparison between FOSM and Monte Carlo error propagation analysis*

Ernest Orlando Lawrence Berkeley National Laboratory
One Cyclotron Road | Berkeley, California 94720

Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098