LBNL-40040
Rev. 2

iTOUGH2
User's Guide

Stefan Finsterle

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

May 2000

This work was supported, in part, by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Geothermal Technologies, of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098, and by a grant from the Swiss National Cooperative for the Disposal of Radioactive Waste (Nagra), Wettingen, Switzerland.

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 About This Manual

iTOUGH2 is a program for parameter estimation, sensitivity analysis, and uncertainty propagation analysis. It is based on the TOUGH2 simulator for nonisothermal multiphase flow in porous and fractured media [*Pruess*, 1987, 1991a].

The key to a successful application of iTOUGH2 is (1) a good understanding of multiphase flow processes, (2) the ability to conceptualize the given flow and transport problem and to develop a corresponding TOUGH2 model, (3) detailed knowledge about the data used for calibration, (4) an understanding of parameter estimation theory and the correct interpretation of inverse modeling results, and (5) proficiency in using iTOUGH2 options. This report primarily addresses Issue (4), through the introduction of inverse modeling concepts for applications in multiphase flow and transport simulations. While inverse modeling can be discussed in the jargon of applied mathematics and mathematical statistics, this manual is tailored to the needs of engineers and scientists who are interested in calibrating TOUGH2 models against observed data. It describes the inverse modeling framework and provides the theoretical background for the methodologies employed by iTOUGH2. Furthermore, it discusses the architecture of iTOUGH2 and contains instructions for code installation and execution. This manual supplements the "iTOUGH2 Command Reference" [*Finsterle*, 1999b], which explains the syntax of all iTOUGH2 commands (Issue 5), and the report "iTOUGH2 Sample Problems" [*Finsterle*, 1999c], which contains a collection of illustrative iTOUGH2 applications. It is assumed that the reader is familiar with the workings of TOUGH2 (Issue 2).

The report is organized as follows. After an introductory discussion of inverse modeling issues (Chapter 1), each element involved in automatic model calibration is described in detail in Chapter 2. These elements include the parameter vector, the vector of observable variables, the stochastic model, the objective function, the minimization algorithm, convergence criteria, the residual and error analyses, and uncertainty propagation analysis. Each element is discussed from a theoretical viewpoint, and reference to the corresponding iTOUGH2 input and output will be made. A line-by-line discussion of a typical iTOUGH2 output file is given in Chapter 3. Chapters 4 and 5 contain information about code architecture as well as instructions for installing and running iTOUGH2.

1.2 Motivation and Scope

Predicting multiphase fluid and heat flow in the subsurface by means of numerical simulation involves the following steps:

1. Developing a conceptual model of the system and creating a numerical model;
2. Assigning values to the numerical model input parameters;
3. Predicting the system state by running the simulator;
4. Interpreting the results and assessing the uncertainty of the predictions.

The first step is the most difficult and also most important task. The conceptual model developed for the system to be studied provides the basis for all subsequent steps. Errors in the conceptual model usually have the largest impact on the model predictions. In multiphase flow modeling, the second step (i.e., assigning parameter values) is likely to be tedious and challenging because of the relatively large number of parameters that need to be specified. Moreover, the physical interpretation of these parameters is often ambiguous, and they are difficult or even impossible to measure directly in the laboratory or the field.

Parameters can be estimated by automatically calibrating the multiphase flow model against measured data of the system response. Inferring model-related parameters from observations by means of a process model is termed *inverse modeling*. As elaborated in Section 1.4, iTOUGH2 supports parameter estimation, sensitivity analysis, and uncertainty propagation analysis. It contributes to conceptual and numerical model development only in the sense that alternative model designs can be tested against one another in their ability to explain observed data. A failure to match certain data may point towards aspects of the model that need to be refined.

1.3 General Remarks About Inverse Modeling

Parameter estimation, history matching, model calibration, and inverse modeling are terms describing essentially the same technique with a slightly different objective in mind. The ultimate goal is to assess the best model and its parameters for predicting the behavior of a dynamic flow system. The reliability of these predictions depends on the appropriateness of the conceptual model and the model parameters. Note that it is the intended use of the model that determines the required degree of model sophistication, as well as the level of accuracy with which the parameters are to be estimated. In this overall scheme, parameter estimation as supported by iTOUGH2 is only one, albeit important step in the process of model development.

Inverse modeling consists of estimating model parameters from measurements of the system response made at discrete points in space and time. Automatic model calibration can be formulated as an optimization problem, which has to be solved in the presence of uncertainty because the available observations are incomplete and exhibit random measurement errors. The parameters to be estimated are selected coefficients in the governing flow equations. They may include hydrogeologic and thermophysical properties, initial and boundary conditions, and parameterized aspects of the conceptual model. The interpretation of these parameters depends on the model structure and the overall purpose of the specific model. In this sense, the parameters are strictly to be seen as *model* parameters (or *model-related* parameters) rather than parameters of the geologic formation (or *aquifer* parameters). Estimating parameter values from measurements relates the real multiphase flow system to its idealized representation.

Inverse modeling involves several interacting steps. Starting from a conceptual model of the physical system, the results of parameter estimation may indicate that the underlying model structure has to be modified. This process of iteratively updating the conceptual model and its parameters is sometimes referred to as *model identification*. iTOUGH2 focuses on the more narrow aspect of inverse modeling, namely parameter estimation by automatic model calibration. Nevertheless, the optimality criteria evaluated by iTOUGH2 make a valuable contribution towards the solution of the model identification problem.

1.4 iTOUGH2 Application Modes

iTOUGH2 is a computer program that provides inverse modeling capabilities for the TOUGH2 code. TOUGH2 [*Pruess*, 1987, 1991a] is a numerical simulator for multidimensional, nonisothermal flows of multiphase, multicomponent fluids in porous and fractured media. While the main purpose of iTOUGH2 is to estimate model-related parameters by automatically calibrating TOUGH2 models to laboratory or field data, the information obtained by evaluating the sensitivity of the calculated system response with respect to certain input parameters can also be used to study the appropriateness of a proposed experimental design and to analyze the uncertainty of model predictions.

iTOUGH2 supports all three application modes, i.e., sensitivity analysis, parameter estimation, and uncertainty propagation analysis.

(1) *Sensitivity Analysis*
The sensitivity of TOUGH2 output variables with respect to TOUGH2 input parameters is numerically evaluated by iTOUGH2. The resulting Jacobian matrix is rescaled to make the sensitivity coefficients comparable with each other. Summary sensitivity measures are calculated to identify the most sensitive parameters as well as the model output most affected by the selected parameters. From an inverse perspective, these values show the information content of individual data points, data sets, and observation types. Furthermore, correlation coefficients between the parameters are calculated, which can be used to detect parameter combinations that lead to a similar or very different system behavior.

(2) *Parameter Estimation*
iTOUGH2 solves the inverse problem for determining TOUGH2 input parameters based on any type of data for which a corresponding TOUGH2 output variable can be calculated. Parameters are estimated by automatically matching the calculated to the observed system response. A number of different objective functions and minimization algorithms are available. An extensive residual and error analysis is performed.

(3) *Uncertainty Propagation Analysis*
The impact of parameter uncertainties on model predictions can be studied by means of linear error propagation analysis or Monte Carlo simulations.

All three application modes are of practical significance. The sensitivity analysis supplies the measures needed for optimizing the design of a laboratory experiment or field test. Parameter estimation by inverse modeling overcomes the labor-intensive tedium of trial-and-error model calibration. More important, the error analysis provides insight into the uncertainty of the estimated parameters, and reveals parameter correlations. Predictability can be improved when relying on effective, model-related parameters estimated by inverse modeling. The quality of these predictions can be assessed, taking into account the uncertainty of the input parameters.

1.5 Inverse Modeling Procedure

iTOUGH2 estimates elements of a parameter vector **p** based on measured data of the system response, which are summarized in vector $\mathbf{z}^*$. The parameters are related to the data by minimizing a measure of misfit, the objective function S, which depends on the residual vector **r** and a weighting matrix $\mathbf{C}_{zz}^{-1}$. The uncertainty of the estimated parameters, i.e., the covariance matrix $\mathbf{C}_{pp}$, is also calculated, along with the uncertainty of the model predictions.

An overview of the inverse modeling concept as implemented in iTOUGH2 is given in this section, followed by a detailed discussion of each element. The major steps are visualized in the flow chart of Figure 1.5.1, and are summarized in Table 1.5.1.

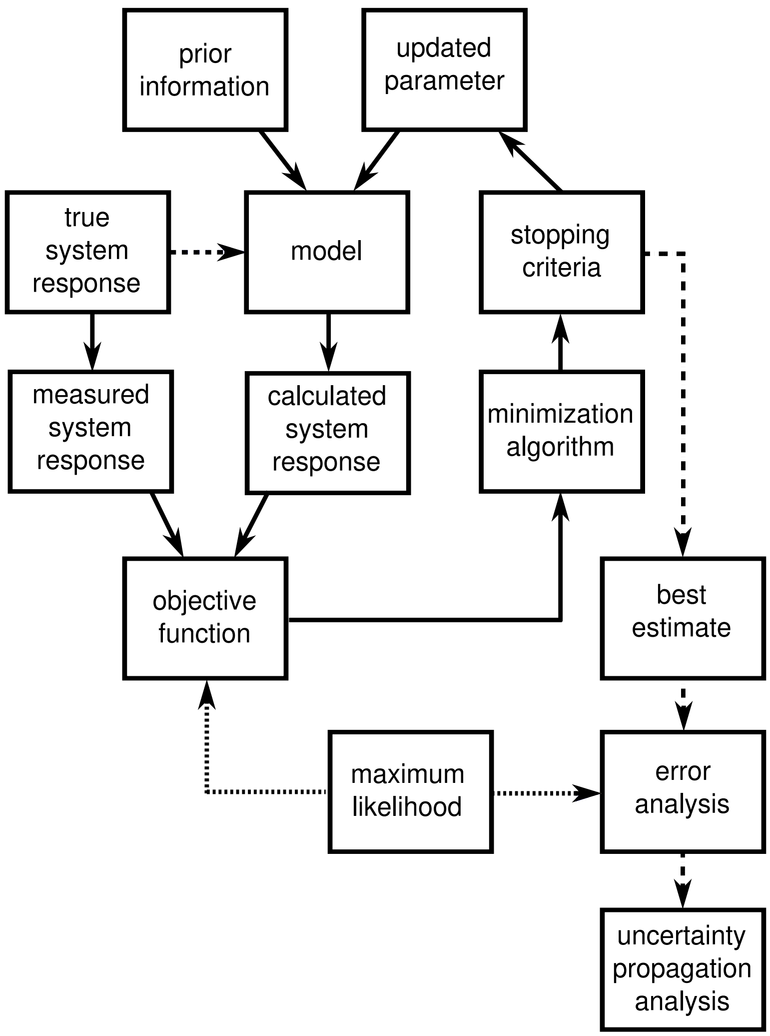


Figure 1.5.1. Inverse modeling flow chart.

Table 1.5.1. Inverse Modeling Procedure: Major Steps

Step	Description	Issue
1	Development of a numerical model, representing the hydrogeological system under test conditions.	Model conceptualization
2	Selection of parameters to be estimated.	Parameter selection
3	Selection of initial parameter values.	Prior information/initial guess
4	Selection of data; identification of points in space and time for calibration.	Calibration points
5	Assignment of weights to each calibration point.	Stochastic model
6	Calculation of system state.	Forward simulation
7	Comparison of calculated and observed system state.	Objective function
8	Updating parameters in order to decrease the objective function.	Minimization algorithm
9	Iteration of Steps 6 through 8 until no further improvement of the fit can be obtained.	Convergence criteria
10	Analysis of residuals and estimation uncertainties.	Residual and error analyses

The key elements as listed in Table 1.5.1 can be described as follows:

(1) Inverse modeling starts with the formulation of the so-called *forward* or *direct problem*. A model must be developed that is capable of simulating the *general* features of the system behavior under measurement conditions. This step involves the mathematical and numerical description of the relevant physical processes, the definition of model geometry, the assignment of initial and boundary conditions, the discretization in space and time, the selection of zones over which the model parameters are believed to be constant, etc. All the parameters that are not subject to the estimation process are then fixed at their best known values. It is important to realize that the fixed parameters are part of the model structure to which the solution of the inverse problem refers. The forward problem is solved by the TOUGH2 simulator.

(2) The next step is to define a vector $\tilde{\mathbf{p}}$ of length n containing the parameters to be estimated by inverse modeling. Since the true parameters cannot be known, we replace them with the corresponding model parameters $\mathbf{p}$. If performing uncertainty propagation analyses, $\mathbf{p}$ holds the parameters considered uncertain. The parameters must be TOUGH2 *input* parameters, and may include hydrogeologic characteristics, thermal properties, initial and boundary conditions, as well as all aspects of the model that can be parameterized. An element of $\mathbf{p}$ may represent a single TOUGH2 input parameter, multiple TOUGH2 input parameters, or a function of TOUGH2 input parameters.

Furthermore, the parameter may be subjected to a transformation (e.g., taking its logarithm) to make the inverse problem more linear or to change the distributional assumption about the parameter.

(3) An initial guess has to be assigned to each element of $\mathbf{p}$. The vector holding the initial guesses is denoted $\mathbf{p}_0$. Note that $\mathbf{p}_0$ affects the initial sensitivity coefficients and the efficiency of the minimization algorithm. Multiple inversions with different initial guesses should be performed to detect potential local minima. While often identical, the vector of initial guesses, $\mathbf{p}_0$, must be distinguished from the prior information vector, $\mathbf{p}^*$, which holds independently measured or estimated parameter values. These measured parameters can be used to constrain or regularize the inverse problem. Prior parameter information must be appropriately weighted (see discussion of matrix $\mathbf{C}_{zz}$ below) to be accounted for in the inversion.

(4) Information about the model parameters is drawn from measurements of the system state. The availability of sufficient, sensitive data of high quality is the key requirement for reliably estimating model parameters. The measured and calculated system response must correspond in terms of character, location, time, and scale. Model output and measured data are compared only at discrete points in space and time, the so-called *calibration points*. Vector $\tilde{\mathbf{z}}$ of length m holds the true, unknown, observable variables at all measurement locations and all calibration times. The vector holding the data measured at or interpolated to the calibration points is denoted by $\mathbf{z}^*$:

$$\mathbf{z}^{*T} = [p_1^*, \ldots, p_n^*, z_{n+1}^*, \ldots, z_m^*] \tag{1.5.1}$$

The corresponding model output, which is a function of space, time, and the model parameters $\mathbf{p}$, are summarized in vector $\mathbf{z}$:

$$\mathbf{z}(\mathbf{p})^T = [p_1, \ldots, p_n, z_{n+1}, \ldots, z_m] \tag{1.5.2}$$

Note that if prior information about the parameters is available, then the first n elements of $\mathbf{z}^*$ are the measured parameter values, i.e., $z_i^* = p_i^*$ $(i = 1, \ldots, n)$, and the estimate $\mathbf{p}$ is considered to be the corresponding model output.

The differences between the measured and calculated system response at the calibration points are summarized in the residual vector $\mathbf{r}$ of length m with elements

$$r_i = z_i^* - z_i \qquad i = 1, \ldots, m \tag{1.5.3}$$

(5) The observation vector includes data that are of different type, magnitude, and accuracy. This requires that each residual be appropriately weighted before an aggregate measure of misfit can be calculated. As will be discussed in Section 2.5.3, it is reasonable to use the inverse of the measurement covariance matrix $\mathbf{C}_{zz}$ as the weighting matrix, i.e., the expected variability of the final residuals has to be assessed based on the size of the measurement errors as well as the random modeling errors.

(6) A TOUGH2 simulation is performed with the current parameter vector $\mathbf{p}$ to obtain the elements of vector $\mathbf{z}(\mathbf{p})$. The simulation will be repeated with updated parameters as proposed by the minimization algorithm (see Steps 8 and 9).

(7) The calculated and measured system response is compared by calculating an aggregate measure of misfit, which is termed *objective function*, S. The objective function is usually some norm of the weighted residuals. If a distributional assumption about the residuals is made, the objective function can be derived from maximum likelihood considerations. The weighted least-squares objective function is the most widely used misfit criterion. iTOUGH2 offers additional objective functions to increase the robustness of an inversion. The objective function is discussed in detail in Section 2.6.

(8) The purpose of the minimization algorithm is to find the minimum of the objective function by iteratively updating the model parameters. Since the model output $\mathbf{z}(\mathbf{p})$ depends on the parameters to be estimated, the fit can be improved by changing the elements of parameter vector $\mathbf{p}$. Consequently, the search for the minimum takes place in the n-dimensional parameter space. A number of strategies exist to find parameter combinations that iteratively yield smaller values of the objective function; they will be discussed in Section 2.7.

(9) Once no further decrease in the objective function can be achieved, the iterative minimization procedure is terminated. Convergence criteria used in iTOUGH2 are discussed in Section 2.7.8. Since the objective function is a global measure of the misfit between the data and the corresponding model output, the parameter vector $\mathbf{p}$ that minimizes S is considered the best-estimate parameter set.

(10) One of the key advantages of a formalized approach to parameter estimation is the possibility to assess the goodness-of-fit, the estimation error, and the uncertainty of the model predictions. Note that if the data are not properly reproduced by the model, i.e., if the final residuals are large or exhibit systematic errors, the resulting parameter set is likely to be inadequate or highly biased. Furthermore, a good match does not imply that the estimates are reasonable. They may be highly uncertain due to high parameter correlations. The residual, error, and uncertainty propagation analyses are discussed in Section 2.8.

1.6 Introductory Example

The process of parameter estimation by automatic model calibration is illustrated in the following example, which is described in detail in *Finsterle and Persoff* [1997]. The example is also part of the collection of iTOUGH2 sample problems [*Finsterle*, 1999c; Problem 2].

A laboratory experiment was designed to estimate hydrogeologic parameters of very tight rock samples. A schematic of the experimental apparatus is shown in Figure 1.6.1. A rock sample is dried and placed in a sample holder, which is attached to two relatively small gas reservoirs. To conduct a test, the upstream reservoir is rapidly pressurized using nitrogen gas to a value about 300 kPa above the initial pressure of the system. Gas starts to flow through the sample, and the change of pressure with time is monitored in both reservoirs.

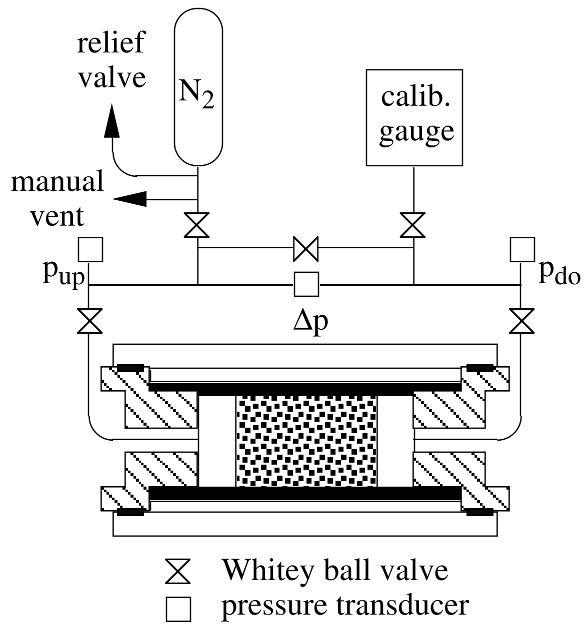


Figure 1.6.1. Schematic of gas-pressure-pulse-decay apparatus.

The steps listed in Table 1.5.1 and discussed in general terms in the previous section are followed here for the specific example.

(1) As part of the model conceptualization, the relevant physical processes have to be identified, mathematically described, and implemented into the numerical simulator. In this example, it is sufficient to consider single-phase gas flow. In porous media with very low permeability and porosity, the mass flux **F** [kg s^{-1} m^{-2}] of gas may be enhanced as a result of slip flow known as the Klinkenberg effect.

$$\mathbf{F} = -k\left(1+\frac{b}{p}\right)\frac{\rho}{\mu}\nabla p \tag{1.6.1}$$

Here, k is the absolute permeability [m^2], ρ is the density [kg m^{-3}], μ is the dynamic viscosity [Pa s], and p is the gas pressure [Pa]. The term in parentheses accounts for enhanced gas slip flow, which occurs when the mean free path of the molecules is large relative to the characteristic dimension of the pores. Slip flow is important at low pressures and in small pores, when a significant fraction of molecular collisions is with the pore wall rather than with other gas molecules. In Equation (1.6.1), b [Pa] is the Klinkenberg slip factor, which is a characteristic of both the geometry of the pore space and the thermophysical properties of the gas. It is directly proportional to the mean free path of the molecules [*Klinkenberg*, 1941]. This flow equation and the appropriate equations-of-state enter the mass- and energy-balance equations solved by TOUGH2. Furthermore, the gas reservoirs and the core are discretized as a one-dimensional flow problem, and the initial pressure in the model is set to the first measured pressure value.

(2) The parameters to be estimated are the porosity ϕ, the absolute permeability k [m^2], and the Klinkenberg factor b [Pa]. Since both k and b are expected to vary over many orders of magnitude, we will estimate the logarithm of these two parameters. Furthermore, logarithmic transformation makes the inverse problem more linear, and prevents the parameters from becoming negative. The parameter vector therefore is of length $n = 3$ and has the elements $\mathbf{p}^T = [\phi, \log(k), \log(b)]$.

(3) The initial parameter values are chosen to be $\phi = 0.015$, $\log(k) = -19.0$, and $\log(b) = 7.0$, defining the elements of vector $\mathbf{p}_0$. These initial values are guesses that are not weighted as prior information. In this example, the inversion does not depend on the initial parameter set.

(4) Observations available for model calibration are the pressure data in the upstream and downstream gas reservoir. It is obvious from Equation (1.6.1) that the absolute permeability and the Klinkenberg factor are strongly correlated if the average pressure in the sample remains constant. Therefore, three experiments performed on the same core but at three different pressure levels are inverted jointly to allow for an independent estimation of k and b (more details can be found in *Finsterle and Persoff* [1997]). We select 30 calibration points in time, logarithmically spaced between 100 and 68,600 seconds. Note that significantly more data points were measured through time, but calibration will occur only against interpolated pressures observed in the two reservoirs during each of the three experiments at 30 selected points in time. The total number of calibration points is therefore $m = m_{experiments} \cdot m_{reservoirs} \cdot m_{times} = 3 \times 2 \times 30 = 180$.

(5) We assume that the measurement errors of the pressure data are uncorrelated and on the order of $\sigma_{z_i} = \sigma_z = 1000$ Pa, $i = n+1, \ldots, m$. The covariance matrix $\mathbf{C}_{zz}$ is therefore a $m \times m$ matrix with σ_z^2 on the diagonal (with the exception of the first three diagonal elements, which are zero to avoid inclusion of prior information about the parameters) and zeroes elsewhere.

(6) The experiment is simulated using the forward model TOUGH2. The initial pressures in the upstream reservoirs of the three experiments are set to about 300 kPa above the respective initial pressures in the core. The dash-dotted lines in Figure 1.6.2 show the pressure in the upstream and downstream reservoirs through time as calculated with the initial parameter set $\mathbf{p}_0$.

(7) The difference between the model calculation and the data at the calibration points is measured by the objective function. The standard least-squares objective function chosen here is the sum of the squared weighted residuals, leading to maximum-likelihood estimates if the residuals are normally distributed:

$$S = \mathbf{r}^T \mathbf{C}_{zz}^{-1} \mathbf{r} = \sum_{i=1}^{m} \frac{(z_i * - z_i)^2}{\sigma_{z_i}^2} \tag{1.6.2}$$

(8) The Levenberg-Marquardt minimization algorithm described in Section 2.7.4 is used to propose new parameter sets $\mathbf{p}_k$ that iteratively reduce the value of the objective function. The Levenberg-Marquardt algorithm requires evaluating the sensitivity of the calculated pressures z_j with respect to the parameters p_i, providing the search direction in the n-dimensional parameter space.

(9) If a certain convergence criterion is met (here, the maximum number of unsuccessful uphill steps was reached), go to Step 10, otherwise repeat Steps 6 through 8 with the updated parameter vector $\mathbf{p}_k$. The fit obtained after 7 iterations is shown in Figure 1.6.2 (solid lines), matching the observed data (symbols) reasonably well.

(10) The error analysis reveals, however, that the standard mean error is larger than the expected mean residual of 1000 Pa. The reason for the unsatisfactory match is a systematic error (a leak in the measuring apparatus and inappropriate initial conditions), which becomes apparent when examining the residual plot shown in Figure 1.6.3. Given this result, the estimated parameters are likely to be biased despite a relatively small estimation uncertainty. These difficulties are resolved by parameterization of the systematic errors as discussed in *Finsterle and Persoff* [1997].

The example illustrates the process and main elements of inverse modeling, which will be discussed in detail in the following chapter. The example also demonstrates the importance of a formalized approach and the error analysis, which identified weaknesses in the data and the conceptual model.

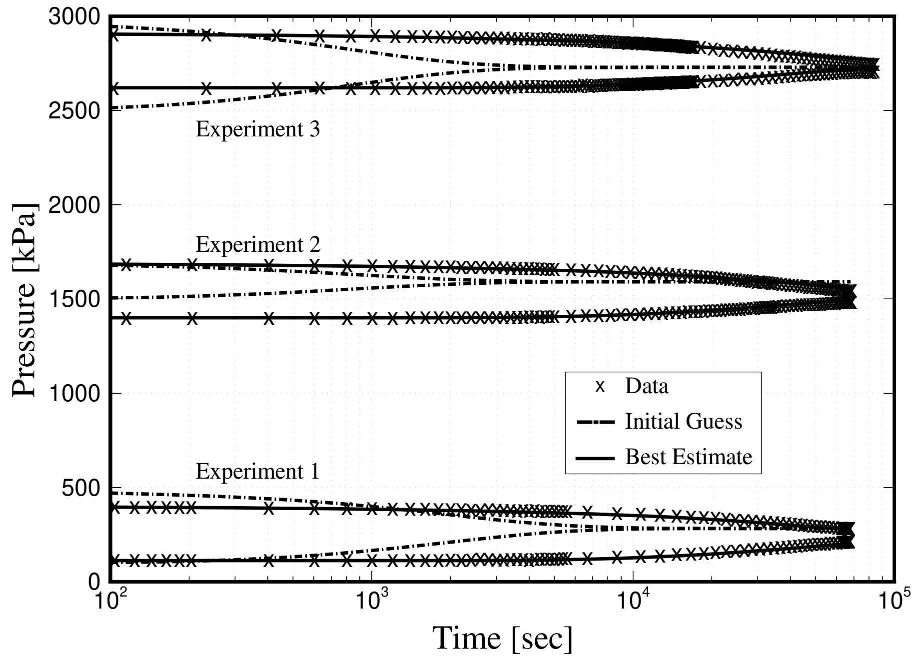


Figure 1.6.2. Comparison between measure and calculated pressure transients with the initial and final parameter sets.

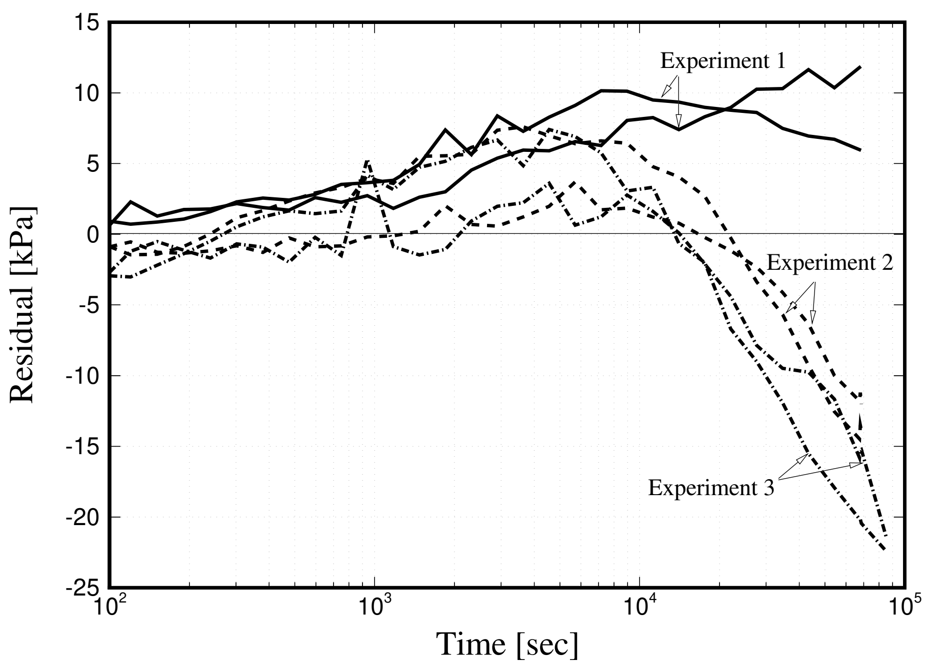


Figure 1.6.3. Residuals as a function of time, showing systematic overprediction of pressures at late times for Experiments 2 and 3.

1.7 Typing Conventions and Variable Definitions

Table 1.7.1 summarizes the main variables. The typing conventions are as follows:

- Scalars are represented by plain characters, e.g., k_{rl}.
- Vectors are lower-case bold characters, e.g., $\mathbf{p}$.
- Matrices are upper-case bold characters, e.g., $\mathbf{J}$, $\mathbf{C}_{zz}$.
- Elements of vectors and matrices are indexed scalars, e.g., J_{ij}, z_i.
- Measured quantities are indicated with an asterisks (*), e.g., $z_i{*}$; the true (usually unknown) values are indicated by a tilde, e.g., $\tilde{z}_i$; the calculated or estimated variables are shown as plain characters, e.g., z_i; best estimates are indicated with a carat, e.g., $\hat{\mathbf{p}}$.

Table 1.7.1. Main Variables and Their Definitions

Variable	Dimension	Definition	Equation
α	scalar	Level of significance; $(1-\alpha)$ is confidence level	-
α	scalar	Perturbation factor for calculating derivatives	-
c_{ij}	scalar	Covariance of estimated parameter (off-diagonal element of $\mathbf{C}_{pp}$)	2.8.4.2
Γ_i	scalar	Ratio of conditional and marginal standard deviation	2.8.4.4
$\mathbf{C}_{pp}$	$n \times n$	Covariance matrix of estimated parameters	2.8.4.2
$\mathbf{C}_{zz}$	$m \times m$	*A priori* covariance matrix of measurement errors	2.5.3.1
$\mathbf{C}_{\hat{z}\hat{z}}$	$m \times m$	*A posteriori* covariance matrix of predictions	2.8.5.7
$\mathbf{H}$	$n \times n$	Hessian matrix	2.7.2.7
$\mathbf{J}$	$m \times n$	Jacobian matrix with sensitivity coefficients	2.7.2.4
K	scalar	Maximum number of iTOUGH2 iterations	-
K	scalar	Number of observation types in an inversion	-
k	scalar	iTOUGH2 iteration index	-
λ	scalar	Levenberg parameter	2.7.4.1
m	scalar	Number of calibration points	2.3.3
ν	scalar	Marquardt parameter	-
n	scalar	Number of parameters	-
$\mathbf{p}$	n	Parameter vector	-
$\mathbf{p}{*}$	n	Prior information	-

$\mathbf{p}_k$	n	Parameter vector at iteration k	2.7.1.1
$\hat{\mathbf{p}}$	n	Best-estimate parameter set	
$\mathbf{r}$	m	Residual	2.4.1
r_{ij}	scalar	Correlation coefficient	2.8.4.3
σ_i^2	scalar	*A priori* error variance (measurement error, diagonal element of $\mathbf{C}_{zz}$)	2.5.3.1
$\sigma_{p_i}^2$	scalar	*A posteriori* error variance of estimated parameter (diagonal element of $\mathbf{C}_{pp}$)	2.8.4.2
σ_0^2	scalar	Dimensionless *a priori* error variance	2.5.3.2
s_0^2	scalar	Dimensionless *posteriori* or estimated error variance	2.8.3.1
S	scalar	Objective function	2.6.4.4
$\mathbf{V}_{zz}^{-1}$	$m \times m$	Weighting matrix	2.5.3.2
$\mathbf{w}$	m	Normalized residual (*a posteriori*)	2.8.5.10
$\mathbf{y}$	m	Normalized residual (*a priori*)	2.6.5.2
y	m	Local reliability	2.8.5.9
$\mathbf{z}$	m	Observable variables	2.3.2
$\mathbf{z}^*$	m	Measurement vector, includes prior information	2.3.1
$\tilde{\mathbf{z}}$	m	True system response	-
$\hat{\mathbf{z}}$	m	Predicted system response	-

2. INVERSE MODELING THEORY

2.1 Introduction

The basic concept of estimating parameters by matching the model to observations dates back to Carl Friedrich Gauss, who introduced the method of least squares for the analysis of astronomical and geodetic data during the last decade of the eighteenth century [*Gauss*, 1821]. Gauss made contributions to all aspects of parameter estimation, providing a detailed discussion of measurement errors, a probabilistic justification of the least-squares objective function, advances in computational methods (Gaussian elimination), and an analysis of estimation uncertainty. While the algorithms for identifying the minimum of the objective function have been continually refined, the basic idea as well as the difficulties associated with solving the inverse problem remain essentially the same.

The theory on inverse modeling is described in a variety of textbooks for applied mathematics and mathematical statistics (see, for example, *Beck and Arnold* [1977], *Bickel and Doksum* [1977], *Gill et al.* [1981], *Scales* [1985], *Larsen and Marx* [1986], *Van Huffel and Vandewalle* [1991], *Stengel* [1994], *Björck* [1996]). Many of these textbooks focus on a discussion of optimality conditions for specific types of functions and constraints. In groundwater and multiphase flow modeling, the model output is usually a highly nonlinear, complex function of the parameters, which are constrained by simple physical bounds. A theoretical analysis of the objective function's convexity is not feasible for numerically calculated model output. Practical aspects of how to formulate the inverse problem, and how to identify the minimum of the objective function, are of primary interest to the hydrogeologist. Good introductions from a general, practical perspective are given by *Beck and Arnold* [1977], *Gill et al.* [1981] and *Scales* [1985]. A concise description of certain aspects of inverse modeling can also be found in *Press et al.* [1992].

A large number of research papers and book articles discuss the concept of inverse modeling in the context of hydrogeology. They are summarized and reviewed by *Neuman* [1973], *Yeh* [1986], *Kool et al.* [1987], *Carrera* [1988], *Ewing and Lin* [1991], *Sun* [1994], and *McLaughlin and Townley* [1996]. The approach implemented in iTOUGH2 is best described in the classic series of papers by *Carrera and Neuman* [1986abc].

This chapter is structured according to the outline given by Figure 1.5.1 and Table 1.5.1, and concludes with a discussion of selected inverse modeling issues. Theoretical considerations are complemented with examples specific to iTOUGH2, and references to related iTOUGH2 commands as documented in the report "iTOUGH2 Command Reference" [*Finsterle*, 1999b] are made where appropriate. A description of the general command syntax is given in Section 3.2 of *Finsterle* [1999b].

2.2 Parameters

The parameter vector **p** of length n contains the TOUGH2 input parameters (or functions thereof) to be estimated by inverse modeling. These parameters may represent hydrogeologic characteristics, thermal properties, initial or boundary conditions, and all aspects of the model that can be parameterized. Note that for a heterogeneous aquifer with properties continuously varying in space, the dimension of the parameter vector is theoretically infinite. In practice, however, the spatial variables as well as the continuous partial differential equations are discretized (e.g., using an integrated finite difference formulation), with constant properties for each gridblock. Furthermore, multiple gridblocks can be assigned to specific subregions of the model domain, which are characterized by constant parameter values, further reducing the number of parameters to be estimated. This process is referred to as *zonation*. Finally, heterogeneity can be described by geostatistical methods, in which the spatial variability is characterized by a relatively small number of geostatistical parameters (e.g., parameters of a variogram, values at pilot points, attractor parameters; for a review of geostatistically based inverse methods, see *Zimmerman et al.* [1998]).

The reduction of the number of parameters from infinity to a finite dimension n is called *parameterization* [*Yeh*, 1986]. This definition includes the description of data points by means of a function and its coefficients. Physical processes such as leaks in an experimental apparatus or time-varying boundary conditions can also be subjected to parameterization by describing them with a coefficient or a function, making these processes accessible for estimation.

It is important to realize that the parameters of vector **p** are only a subset of the parameters specified in the TOUGH2 input file. Vector **p** contains only those parameters that will be subjected to the estimation process. All the other parameters specified in the TOUGH2 input file *are fixed and become part of the conceptual model*. Due to inherent correlations between the fixed and the variable parameters, the best-estimate parameter set depends on the chosen values of the fixed parameters and the conceptual model in general, i.e., the parameters estimated by inverse modeling are always *model-related*.

If one is performing uncertainty propagation analyses, **p** holds the parameters considered uncertain. A probability density function is assigned to these parameters, and the effect on the uncertainty of the model predictions is then evaluated either by means of linear error propagation analysis or Monte Carlo simulations (see Section 2.8.7).

An element of **p** may represent a single TOUGH2 input parameter, multiple TOUGH2 input parameters, or a function of TOUGH2 input parameters (see examples below). Furthermore, the parameter may be transformed to reduce the nonlinearity of the inverse problem, to eliminate constraints, or to reflect a certain distributional assumption about the parameter. The most frequently applied transformation is taking the logarithm of a parameter, which can make the inverse problem more linear (for a discussion, see *Carrera and Neuman* [1986c]), prevents the parameter from becoming negative, and reflects a log-normal distribution of the parameter's uncertainty. Note that all statistical measures calculated by iTOUGH2 refer to the transformed parameter rather than the parameter used in TOUGH2.

While the elements of vector $\mathbf{p}$ are the unknown parameters to be estimated by inverse modeling, there is often some independent *prior information* available about these parameters. Prior information such as measured parameter values can be included in the analysis to regularize the inverse problem and to constrain the estimates [*Carrera and Neuman*, 1986a; *Chavent*, 1991;*Vasco et al.*, 1997; *Neumaier*, 1998]. Differences between measured parameter values and the corresponding estimates are treated in the same manner as the differences between the observed and calculated system state. Consequently, the elements of the prior information vector $\mathbf{p}^*$ are included in the observation vector $\mathbf{z}^*$ (see Section 2.3), i.e., the first n elements of $\mathbf{z}^*$ are identical with the elements of $\mathbf{p}^*$, and the first n elements of vector $\mathbf{z}$ are identical with the estimates $\mathbf{p}$. Prior information about a parameter is only considered if a finite standard deviation is specified for this parameter.

The use of prior information as a means to regularize the inverse problem is convenient. However, it may also be misleading. The data used for calibration contain a finite amount of information about the parameters to be estimated. If an ill-posed inverse problem is formulated using these data, it can be turned into a well-posed problem by regularization. This means, however, that new information must be added, such as a smoothing criterion (see, for example, *Vasco et al* [1997]), an arbitrary conditioning matrix (see, for example, *Kuczera and Mroczkowski* [1998]) or prior parameter information (see, for example, *Carrera and Neuman* [1986a]). While each of these regularization approaches may lead to a solution of the inverse problem, they sometimes mask the fact that the original data do not contain enough information for parameter estimation. Furthermore, the prior information value must be conceptually consistent with the value determined from the observations of the system response to avoid biased estimation. For example, if the permeability measured on a laboratory core is used as prior information in an inversion of a regional flow model, the difference in scale may compromise the solution. The use of prior information is only reasonable if it is an integral, well-understood part of the overall calibration strategy.

An initial guess has to be assigned to each element of $\mathbf{p}$. The vector holding the initial guesses is denoted $\mathbf{p}_0$. Note that the $\mathbf{p}_0$ affects the initial sensitivity coefficients and the efficiency of the minimization algorithm. Multiple inversions with different initial guesses should be performed to detect potential local minima. While often identical, the vector of initial guesses, $\mathbf{p}_0$, can be different from the prior information vector, $\mathbf{p}^*$, which holds independently measured or estimated parameter values.

Obtaining meaningful estimates of all the parameters in $\mathbf{p}$ is only possible if enough data of good quality are available, and if the model output at the calibration points is sufficiently sensitive to changes in the parameters. Furthermore, the parameters must not be strongly correlated.

SUMMARY

The parameter vector **p** holds a subset of TOUGH2 input parameters or functions thereof. These are the n unknown or uncertain parameters subjected to parameter estimation, sensitivity, or uncertainty propagation analysis. An initial guess has to be provided, which is the starting point for the optimization, the base-case parameter set for sensitivity analysis, or the mean for uncertainty propagation analysis. Prior information can be included if appropriately weighted against the observations of the system state.

EXAMPLES

The following examples describe individual elements of a hypothetical iTOUGH2 parameter vector **p** of length $n = 10$.

p_1 Porosity of material domain LOAM1.

p_2 Logarithm of the absolute permeability along the third (vertical) principal axis of all elements belonging to material domain SAND1.

p_3 Logarithm of the second parameter of the default capillary pressure function (e.g., representing the van Genuchten parameter $1/\alpha$ if `ICP=11`).

p_4 Initial gas saturation in all elements belonging to material domains LOAM1, LOAM2, and LOAM3.

p_5 A factor multiplying the injection rates specified for sources SOU_1 through SOU10.

p_6 An unknown offset of the pressure sensor at observation Set 3.

p_7 Fracture spacing, which is a parameter of the MINC preprocessor.

p_8 The time at which a spill occurred from element INJ99.

p_9 The pressure in all (boundary) elements belonging to material domain BOUND.

p_{10} A user-specified parameter, which can be any combination of TOUGH2 variables.

RELATED iTOUGH2 COMMANDS

The elements of parameter vector **p** are defined through a combination of iTOUGH2 commands in block `> PARAMETER`.

Simple parameter transformations are performed using commands `>>>> FACTOR`, `>>>> LOGARITHM`, or `>>>> LOG(F)`.

The initial guess and/or prior information values are taken directly from the TOUGH2 input file; they can be overwritten by using commands `>>>> GUESS`, `>>>> PRIOR`, and `>> GUESS`.

In order to consider prior information, a standard deviation must be specified using command `>>>> DEVIATION (p)` or a related command. Potential parameter variability for sensitivity analyses is indicated with command `>>>> VARIATION`.

2.3 Observations

The observation vector $\mathbf{z}$ of length m contains the calibration points. Calibration points consist of observable variables at discrete points in space and time, at which measured data are available. While the true system response is continuous in space and time and would be described by an infinite number of variables, measurements are sparse, limiting the amount of information available for inverse modeling. Because the dimension of vector $\mathbf{z}$ is finite, it is necessary to select a subset of the output variables and pick discrete points in space and time as the points at which the measured and calculated system response will be compared (see Section 2.4). The type of variable used for calibration is obviously given by the type of measurements available, and the location of measurement stations usually determines the points in space at which calibration should occur. The selection of calibration points in time, however, is somewhat subjective and may also depend on the time-stepping algorithm of the model or other factors. Note that calibration points do not have to coincide with the exact observation time of actual data points. In iTOUGH2, linear interpolation is applied to obtain a data point at the selected calibration time. Spatial interpolation from observation points to model gridblocks, however, must be performed as part of data preprocessing outside iTOUGH2.

It is important to realize that the spatial and temporal distribution of calibration points has an impact on the inverse modeling results. For example, selecting logarithmically spaced calibration times to match data from a transient test puts more weight on the early-time data relative to the late-time data, possibly affecting the support scale and thus the nature of the parameter to be estimated. Similarly, a high data density in one area of the model emphasizes the corresponding subsystem, potentially compromising the match to data from an adjacent, less densely sampled region.

Complementary to the observed values at the calibration points (i.e., the actual, transformed, or interpolated data) are the simulation results. Simulation results are represented by TOUGH2 output variables (or functions thereof); they depend on the input parameters to be estimated.

An observable variable qualifies as a calibration point only if it is sufficiently sensitive to changes in the parameters to be estimated. The higher the absolute value of the sensitivity coefficient $|\partial z / \partial p|$, the more information regarding the parameters of interest is contained in the corresponding data point. High sensitivity is a necessary, albeit not sufficient requirement for accurate parameter estimation (for a detailed discussion, see *Finsterle and Persoff* [1997], as well as Section 2.8.2).

The vector of observable variables may also contain parameters. For example, if permeability was 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*.

SUMMARY

The vector holding the data measured at or interpolated to the calibration points is denoted by $\mathbf{z}^*$.

$$\mathbf{z}^{*T} = [p_1^*, \ldots, p_n^*, z_{n+1}^*, \ldots, z_m^*] \tag{2.3.1}$$

$\mathbf{z}^*$ differs from the vector of the true system response $\tilde{\mathbf{z}}$ by the unknown measurement errors (see discussion in Section 2.5.2). The corresponding model output, which is a function of space, time, and the model parameters $\mathbf{p}$, is summarized in vector $\mathbf{z}$:

$$\mathbf{z}(\mathbf{p})^T = [p_1, \ldots, p_n, z_{n+1}, \ldots, z_m] \tag{2.3.2}$$

If prior information about the parameters is available, the first n elements of $\mathbf{z}^*$ are the measured parameter values, i.e., $z_i^* = p_i^*$ $(i = 1, \ldots, n)$, and the current estimate $\mathbf{p}$ is considered to be the corresponding model output.

EXAMPLES

The following examples describe individual elements of a hypothetical iTOUGH2 observation vector $\mathbf{z}$ and the corresponding measurement vector $\mathbf{z}^*$. The example also illustrates the structure of these vectors as stored in iTOUGH2, with the first n elements reserved for prior information, followed by all observations at time T_1, then all observations at T_2, etc. If no time windows are specified, the total length of vector $\mathbf{z}$ is given by

$$m = n + n_{datasets} \cdot n_{times} \leq MAXM \tag{2.3.3}$$

where $n_{datasets} \leq MAXO$ is the number of data sets (i.e., measurement sensors), and $n_{times} \leq MAXTIM$ is the number of calibration times. The parameters $MAXM$, $MAXO$, and $MAXTIM$ are the maximum array dimensions as specified in file *maxsize.inc* (see Section 5.2). If time windows are specified, the number of elements in $\mathbf{z}$ is less than that of Equation (2.3.3).

z_1^*	Independently measured porosity of material domain LOAM1, included as prior information.
z_1	Porosity of material domain LOAM1, currently estimated based on information contained in $\mathbf{z}^*$.
z_2^*	Measured pressure at Sensor X_1 and at time T_1
z_2	Calculated pressure at gridblock SEN 1 (representing Sensor X_1), at time T_1.
z_3^*	Measured TCE concentration in liquid phase at Sensor X_2 and at time T_1.
z_3	Calculated TCE concentration in liquid phase at gridblock SEN 2 (representing Sensor X_2), at time T_1.
z_4^*	Measured cumulative liquid flow rate into opening X_3 at time T_1.

z_4 Calculated change of total liquid mass in all gridblocks associated with material domain DRIFT (representing opening X_3), at time T_1.

z_{3i+n}* Same measurement types as before, at times T_{i+1}.

z_{3i+n} Same model output as before, at times T_{i+1}.

RELATED iTOUGH2 COMMANDS

The elements of observation vector **z** are defined through a combination of iTOUGH2 commands in block `> OBSERVATION`. The second-level commands indicate the observation type; the third-level commands define the location.

The calibration times are specified using command `>> TIMES`. If a specific data set does not cover the entire simulation time, command `>>>> WINDOW` should be used.

The measurement values are submitted using command `>>>> DATA`. One might want to calibrate against the logarithm of the data rather than their values (command `>>>> LOGARITHM`). This data transformation may be useful to make nonsymmetric residual distributions more symmetric, better complying with the distributional assumptions underlying maximum-likelihood estimation.

2.4 Residuals

The residual vector $\mathbf{r}$ of length m contains the differences between the measured and calculated system response with elements

$$r_i = z_i^* - z_i \qquad i = 1,\ldots,m \tag{2.4.1}$$

For example, r_2 is the difference between the measured and calculated pressure at time T_1 (see example in Section 2.3). A special type of residuals consists of the differences between the measured parameters (prior information) and the estimated parameter values. These differences—appropriately weighted (see Section 2.5)—can be used to regularize the inverse problem, making it more stable and well-posed [*Carrera and Neuman*, 1986a; *Neumaier*, 1998]. The residuals obtained with the best-estimate parameter set $\hat{\mathbf{p}}$ at the end of an inversion are termed the final residuals.

In inverse modeling, parameters are estimated by minimizing some measure of misfit, the objective function (see Section 2.6), which is a function of the residuals. Since the residuals determine the misfit criterion, it is crucial that the measurements $\mathbf{z}^*$ and the corresponding model output $\mathbf{z}$ represent the same physical entity. Any conceptual difference between the measured value and its representation in the numerical model necessarily leads to a bias in the estimated parameters. There are many reasons for a potential inconsistency between the measured and calculated values. For example, the support scale or averaging volume of a certain measurement may be significantly smaller than the size of the gridblock used in the numerical model. Drawdown measurements in a pumping well can only be directly compared with gridblock pressures if the well is fully discretized in the model. Downhole pressures or fractional flows at the head of a geothermal well may be different from the vertically averaged values calculated in a two-dimensional model of the reservoir. Relative pressure measurements may be influenced by atmospheric pressure fluctuations. If these effects are not properly accounted for, the parameters are perturbed from their most likely estimates in an attempt to partly compensate for the error.

Consistency between the measured and simulated quantities must be assured. If the two variables are conceptually different, appropriate compensation or correction can be made either in the numerical model (e.g., by discretizing the well or even the measuring device, or by accurate simulation of all factors that affect the measurements) or by preprocessing the data accordingly (e.g., appropriate averaging, interpolation, compensation for temperature effects, removal of shifts and trends in the data). In many cases, both the model output and the measurements must be adjusted to ensure consistency between the two.

SUMMARY

Parameter estimation by inverse modeling is based on a comparison between measured values and the corresponding model output. The residual is defined as the difference between the measured and calculated system response at a given calibration point. Measured and calculated variables must be consistent.

2.5 The Stochastic Model

2.5.1 Introduction

Inverse modeling can be formulated in the framework of mathematical statistics, which may provide a probabilistic justification for using a specific estimator (such as least squares). The maximum likelihood approach will be discussed in Section 2.6. From a more practical point of view, the stochastic model as summarized in the observation covariance matrix $\mathbf{C}_{zz}$ (see Section 2.5.3) simply provides the weighting factors to scale observations of different type, magnitude, and accuracy. Besides these practical considerations, one should realize that inverse modeling makes the implicit statistical assumption that the final residuals $\mathbf{r}(\hat{\mathbf{p}})$ (where $\hat{\mathbf{p}}$ is the best-estimate parameter set) are error terms, i.e., they are random variables following a certain distribution. The stochastic model is therefore defined here as the *a priori* description of the distributional assumption about the residuals.

The sources and nature of the error term will be discussed in Section 2.5.2; the observation covariance matrix, the weighting matrix, and the *a priori* error variance are introduced in Section 2.5.3.

2.5.2 Systematic and random errors

The residuals can be represented by a statistical model of the form

$$r_i = z_i * - z_i(\hat{\mathbf{p}}) = e_{m_i} + e_{d_i} = (b_m + \varepsilon_m)_i + (b_d + \varepsilon_d)_i \tag{2.5.2.1}$$

According to this equation, residual i is the sum of the error in the model, $e_{m_i} = \tilde{z}_i - z_i(\hat{\mathbf{p}})$, and the error in the data, $e_{d_i} = z_i * - \tilde{z}_i$, where $\tilde{z}_i$ is the true value. Both modeling error and data error have a systematic component b_i and a random component ε_i.

Consider a data set that is drawn from a true, but unknown system response (see Figure 2.5.2.1). The individual measurement error is defined as the difference between the measured and the true value. The modeling error is defined as the difference between the true and the calculated value. Since the true system response is unknown, neither the measurement error nor the modeling error is known—only the residual

$$\mathbf{r} = \mathbf{e}_d + \mathbf{e}_m = (\mathbf{z} * - \tilde{\mathbf{z}}) - (\tilde{\mathbf{z}} - \mathbf{z}(\hat{\mathbf{p}})) = \mathbf{z} * - \mathbf{z}(\hat{\mathbf{p}}) \tag{2.5.2.2}$$

can be calculated. However, the errors may be described in statistical terms, implying that they are random following a certain distribution. Recall that estimating parameters by history matching is based on the assumption that the calculated system response is as close to the true system response as possible, the latter being represented by a set of noisy data points. If the true values are identified, the residuals are by definition equal to the measurement errors. In other words, the statistical characteristics of the residuals should be identical or at least similar to those of the measurement errors. This interpretation assumes that only random errors are present, which can be described using statistical concepts. The impact of systematic errors as indicated in Equation (2.5.2.1) will be discussed next.

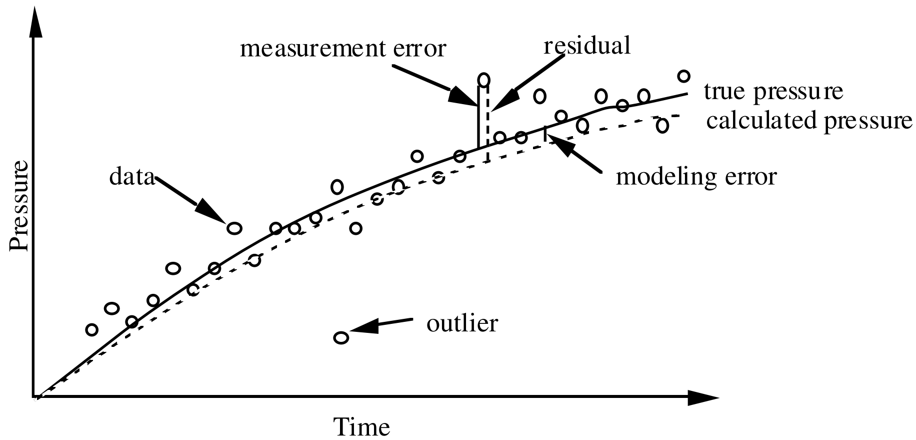


Figure 2.5.2.1. True, measured, and calculated system response, and definition of residual, measurement and modeling error.

It is very important to appreciate the difference between systematic and random components. Because it is usually not possible or relevant to identify whether a deviation between the simulation result and the data is attributable to an error in the data or a modeling error, we disregard the source of an error and only distinguish between its systematic and random components. The systematic error in the residuals is denoted by $b \equiv b_d + b_m$, and the random part is termed $\varepsilon \equiv \varepsilon_d + \varepsilon_m$. Note that in most cases, systematic errors from an incomplete model description outweigh the systematic measurement errors, i.e., $|b_m| >> |b_d|$, whereas random modeling errors such as round-off errors or numerical oscillations are usually small compared with the random errors in the data, i.e., $|\varepsilon_m| << |\varepsilon_d|$.

Systematic and random components and their relation to the functional and stochastic model, respectively, are illustrated for the ideal case (i.e., no systematic errors) in Table 2.5.2.1, and for the nonideal case in Table 2.5.2.2.

Under ideal conditions, systematic errors are absent, i.e., $\mathbf{b} = \mathbf{0}$. As mentioned above, the systematic component of the observed system behavior is identical with the true system behavior, which is presumably identified by accurate modeling of the physics of the flow problem. The systematic component is then represented by the *functional model*, which includes the conceptual model and its application to the conditions under which the data have been collected. In the absence of systematic modeling or measurement errors, the true system response is asymptotically identified, and the distribution of the final residuals is consistent with that of the measurement errors as described by the stochastic model. Note that since only one, finite set of noisy data is available for calibration, the estimated parameters remain uncertain. Nevertheless, this uncertainty can be estimated (see Section 2.7.4).

Table 2.5.2.1. Systematic and Random Components Under Ideal Conditions

data		true response		measurement error
$\mathbf{z}^*$	$=$	$\tilde{\mathbf{z}}$	$+$	$\mathbf{e}_d$
		identified part		unidentified part
		systematic component		random component
		smooth		rough
		conceptual		distributional
		functional model		stochastic model
		TOUGH2 result		$\mathbf{C}_{zz} = \sigma_0^2 \cdot \mathbf{V}_{zz}$
calibration point		calculated response		residual
$\mathbf{z}^*$	$=$	$\mathbf{z}(\hat{\mathbf{p}})$	$+$	$\mathbf{r}(\hat{\mathbf{p}})$

Table 2.5.2.2. Systematic and Random Components Under Non-Ideal Conditions

calibration point		calculated response		residual
$\mathbf{z}^*$	$=$	$\mathbf{z}(\hat{\mathbf{p}})$	$+$	$\mathbf{r}(\hat{\mathbf{p}})$
		$\Vert$		$\Vert$
		$\tilde{\mathbf{z}} + X_b^f \cdot \mathbf{b} + X_\varepsilon^f \cdot \varepsilon$	$+$	$X_b^s \cdot \mathbf{b} + X_\varepsilon^s \cdot \varepsilon$
		true behavior + systematic errors + random errors		systematic error + random error

X_b^f	:	Systematic error explained by functional model
$X_b^s = (1 - X_b^f)$	:	Systematic error to be explained by statistical model
X_ε^s	:	Random error to be explained by statistical model
$X_\varepsilon^f = (1 - X_\varepsilon^s)$	:	Random error explained by functional model

Table 2.5.2.2 shows the nonideal case. Systematic measurement errors, inconsistencies between the observed and modeled variables (see discussion in Section 2.4), and modeling errors almost always lead to a sizable systematic error component. In these cases, our hope is that X_b^f is small, pushing the systematic errors into the residuals, where they can be readily identified. Unfortunately, the size of X_b^f is not known. It depends on the nature of the systematic error as compared to that of the observed system behavior. Large random errors (e.g., outliers) may also bias the inversion. Again, we hope that X_ε^s is small and the outliers can be identified in the residual analysis and eliminated. The Fisher model test (see Section 2.8.3) is a global check to see whether the distribution of the final residuals is consistent with the distributional assumption of the stochastic model, potentially identifying flaws in either the stochastic or the functional model.

The critical point to realize is that the observed system response is not separated into a true component and an error component, but into a systematic part (modeled by the process simulator TOUGH2), and a random part (described by the stochastic model). The functional model will try to explain not only the (true) systematic component of the system behavior, but also any systematic error in the data or the model; the remaining difference—the final residual—is considered a random component to be analyzed statistically. As a result, the estimated parameter set $\hat{\mathbf{p}}$ may not be an unbiased estimate of the true parameter set $\tilde{\mathbf{p}}$, and the model prediction will not closely reproduce the true system behavior $\tilde{\mathbf{z}}$. Furthermore, the assumption of randomness in the final residuals, which underlies the *a posteriori* error analysis, is violated, leading to problematic uncertainty estimates. A careful residual analysis as discussed in Section 2.8.5 may help identify systematic components in the final residuals as well as outliers that do not conform with the distributional assumption.

We conclude that systematic errors must be removed from both the data and the model, so that the final residuals only contain random components that can be described by the stochastic model. The calculated system response, $\mathbf{z}(\hat{\mathbf{p}})$, is close to the true system behavior only if X_b^f, X_ε^s, and $\mathbf{b}$ are sufficiently small, making $\hat{\mathbf{p}}$ a good estimate of the true parameter set.

SUMMARY

The stochastic model describes the random component of the system response, which is not identified by the functional model.

2.5.3 Observation covariance matrix

In the previous section we saw that the unexplained parts of the system response cannot be described individually, but must be treated by means of a stochastic model, assuming that the residuals are random and follow a certain distribution. Furthermore, the distribution of the final residuals is supposed to be consistent with the distribution of the measurement errors, assuming that the true system response is correctly identified by the model.

A reasonable assumption about the measurement errors is that they are uncorrelated, normally distributed random variables with zero mean. The *a posteriori* residual analysis will have to show that this assumption is justified. The *a priori* distributional assumption about the residuals can therefore be summarized in a covariance matrix $\mathbf{C}_{zz}$. $\mathbf{C}_{zz}$ is an $m \times m$ diagonal matrix. The jth diagonal element is the variance that represents the measurement error of observation z_j *:

$$\mathbf{C}_{zz} = \begin{bmatrix} \sigma_{z_1}^2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{z_i}^2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{z_n}^2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sigma_{z_j}^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \sigma_{z_m}^2 \end{bmatrix} \tag{2.5.3.1}$$

The purposes and interpretations of the elements of $\mathbf{C}_{zz}$ are manifold:

- They scale data of different quality, i.e., an accurate measurement obtains a higher weight in the inversion than a poor or highly uncertain measurement.

- They scale observations of different types. For example, flow rates and pressures have different units and their values differ by many orders of magnitude. They need to be scaled appropriately to be comparable in a formalized parameter estimation procedure.

- They weigh the fitting error.

- $\mathbf{C}_{zz}$ is the stochastic model for maximum-likelihood estimation for normally distributed residuals.

One should realize that only the ratios $\sigma_{z_i}^2 / \sigma_{z_j}^2$ are important for parameter estimation, i.e., the estimated parameter set $\hat{\mathbf{p}}$ is not affected by a linear scaling of the covariance matrix. We can therefore introduce a factor σ_0^2 and write

$$\mathbf{C}_{zz} = \sigma_0^2 \cdot \mathbf{V}_{zz} \tag{2.5.3.2}$$

where $\mathbf{V}_{zz}$ is a positive definite matrix. The scalar σ_0^2 is termed the *a priori* error variance. It can be interpreted as the variance of a dimensionless error of size one. In iTOUGH2, a different scalar is internally used for each observation type, which allows iterative updating of their relative weights (see Section 2.8.3). However, since σ_0^2 can assume any positive value, it is convenient to set it to 1.0 and work with the actual covariance matrix $\mathbf{C}_{zz}$ rather than $\mathbf{V}_{zz}$. After an inversion, the *a posteriori* or estimated error variance s_0^2 is calculated. If the assumption about the overall size of the measurement errors was correct, and if the true system behavior is correctly identified, then the ratio s_0^2/σ_0^2 should not significantly deviate from 1.0 (for details about the Fisher model test, see Section 2.8.3).

SUMMARY

The observation covariance matrix $\mathbf{C}_{zz}$ contains the *a priori* assumption about the variances of the final residuals. The elements of $\mathbf{C}_{zz}$ should be based on the assumed size of the measurement errors. Matrix $\mathbf{V}_{zz}^{-1}$ will be used to weigh each residual during the inversion. The estimated error variance s_0^2 should be consistent with the *a priori* error variance σ_0^2.

EXAMPLES

The accuracy of a certain pressure transducer may be given to be 5000 Pa. The square of this value could be directly used in $\mathbf{C}_{zz}$ to reflect the assumed measurement error. However, there are likely to be other random components in the residuals, as a result, for example, of unmodeled variabilities in the formation properties, which affect the measured pressures, but are not represented in the model. The standard deviation of the final residuals is therefore expected to be larger.

The standard deviation assigned to an individual data point reflects its variability, assuming the same observations were repeated many times. In most cases, however, only one measurement is available for each calibration point, i.e., it is usually not possible to estimate the standard deviation from a statistical analysis of a large number of measurements made at that point. Instead of looking at the variability of many realizations at a single calibration point, one is often forced to estimate the standard deviation based on the variability of one realization (the measured data) at many calibration points. In practice, one draws a fitting line through the data points (similar to the one shown in Figure 2.5.2.1) and estimates the standard deviation from the scattering of the data about this line. Keep in mind that the standard deviation may depend on the magnitude of the observed value itself, in which case it should be specified as a percentage of the measured value.

Prior information as well as the weighting of certain data points (e.g., integrated measures or steady-state data points used in combination with transient data points) require special considerations. The relative weight given to prior information or a single steady-state data point depends on the number of calibration points assigned to other observations. It is therefore best to think of the purpose of $\mathbf{C}_{zz}^{-1}$ as a weighting matrix. If a single steady-state data point should receive a weight comparable to that of, for example, one hundred transient data points, then the standard deviation of the steady-state data point should be taken to be 1% of the actual measurement uncertainty (see also *Finsterle* [1999c; Problem 5]).

RELATED iTOUGH2 COMMANDS

The diagonal elements of matrix $\mathbf{C}_{zz}$ are specified using one of the following commands in block `> OBSERVATIONS`:
`>>>> DEVIATION`, `>>>> VARIANCE`, `>>>> WEIGHT`, `>>>> RELATIVE`, and `>>>> AUTO`. The units of the standard deviations specified in iTOUGH2 must be identical to those of the corresponding data set, i.e., the standard deviations are internally multiplied by the factor specified using command `>>>> FACTOR (o)`, which converts the units of the data to standard TOUGH2 units. If calibration occurs against the logarithm of the observed data (see command `>>>> LOGARITHM (o)`), the standard deviations must also refer to the logarithm.

The first n diagonal elements are used to weigh prior information about the parameters, and are specified through one of the following commands in block `> PARAMETERS`:
`>>>> DEVIATION`, `>>>> VARIANCE`, or `>>>> WEIGHT`. If prior information shall not be weighted in the inversion, command `>>>> VARIATION` should be used instead to describe a typical parameter variation.

Command `>>> TAU` allows iterative adjustment of the relative weights between observations of different types by updating the internally generated *a priori* error variances (see Section 2.8.3).

2.6 Objective Function

2.6.1 The norm as a measure of misfit

The purpose of the objective function is to provide an integral measure of misfit between the model and the data, i.e., a parameter set that reduces the value of the objective function is considered superior to those with higher values because it improves the fit *according to the criterion embedded in the objective function*. The best-estimate parameter set is the one that minimizes the objective function. It is the topology of the objective function near the minimum that determines the uncertainty of the estimates and the correlation structure. The objective function may be convex or exhibit multiple local minima; it may be close to quadratic or highly nonlinear in nature; it may be continuous, differentiable, and smooth, or discontinuous, not differentiable, and rough. All these properties affect the choice and efficiency of the minimization algorithm, and—more importantly—the quality of the solution, its stability, and the degree to which the inverse problem is well posed.

The objective function is also termed performance measure, penalty function, energy function, cost function, misfit criterion, etc.

The question of how to identify the minimum of the objective function is addressed in Section 2.7, where various minimization algorithms will be discussed. Note that even if the global minimum of the objective function is identified, this does not necessarily mean that the fit is acceptable. This question is discussed in Section 2.8.

There are many ways to measure the difference between the observed and calculated system response. In the standard procedure of trial and error calibration, the simulation results and data are plotted, and a rather subjective judgment is made as to how well the model output matches the data. A more objective way is to calculate a norm of the residual vector:

$$\|\mathbf{r}\|_p = \left(\sum_{i=1}^{m} |r_i|^p \right)^{1/p} \tag{2.6.1.1}$$

The most commonly used norms are the L_1-norm, the L_2- or Euclidean norm, and the L_∞-norm, corresponding to the L_1-estimator, the Least Squares estimator, and the Minmax estimator, respectively. The choice of an appropriate objective function should be based on the properties of the residuals themselves. The maximum likelihood approach discussed in Section 2.6.3 takes the distributional assumption about the measurement errors as a basis for choosing the objective function. The central limit theorem leads to the assumption that the residuals are normally distributed, making least squares—discussed in Section 2.6.4—a reasonable choice. The normality assumption is also made to facilitate statistical analysis of results. Least squares is used almost exclusively in groundwater inverse modeling [*McLaughlin and Townley*, 1996] and many other fields. However, the distribution of the residuals often deviates from being Gaussian. For example, the presence of outliers in the data or systematic modeling errors lead to non-symmetric distributions, which often exhibit stronger tails than those predicted by the normal distribution. For these cases, alternative objective functions may be more appropriate to avoid biased estimates. These so-called robust estimators are discussed in Section 2.6.5

2.6.2 Properties of the objective function

The objective function is a hypersurface in the n-dimensional parameter space. The global minimum represents the best-estimate parameter set. Figure 2.6.2.1 is a visualization of the objective function for $n = 2$. For a nonlinear model, the topography of the objective function may exhibit a global minimum, multiple local minima, inflection points, stationary points, ridge lines, ledges, etc. However, since the standard objective function is a sum of squares (see Section 2.6.4), the objective function near the global minimum is close to parabolic with elliptical contour lines. The second-order methods for identifying the minimum (see Section 2.7) take advantage of this specific property of the objective function. A linear model yields a parabolic objective function, the minimum of which is easy to identify. In the nonlinear case, the topography away from the minimum becomes intricate, making it difficult for the optimization algorithm to iteratively proceed towards the minimum. Moreover, an ill-posed inverse problem leads to level plains, long narrow valleys, or ridge lines, at which the minimum is poorly defined, if at all. In fact, it is the topology of the objective function that indicates whether an inverse problem is well-posed or ill-posed. For example, multiple parameter combinations with S values close to that obtained at the global minimum indicate nonuniqueness. (The presence of local minima does not constitute nonuniqueness.) Furthermore, the estimation uncertainty is related to the convexity of the objective function near the minimum. (This will be discussed in detail in Section 2.8.) Assuming a continuous objective function that is twice differentiable, the gradient is zero at the minimum and the Hessian matrix is positive definite.

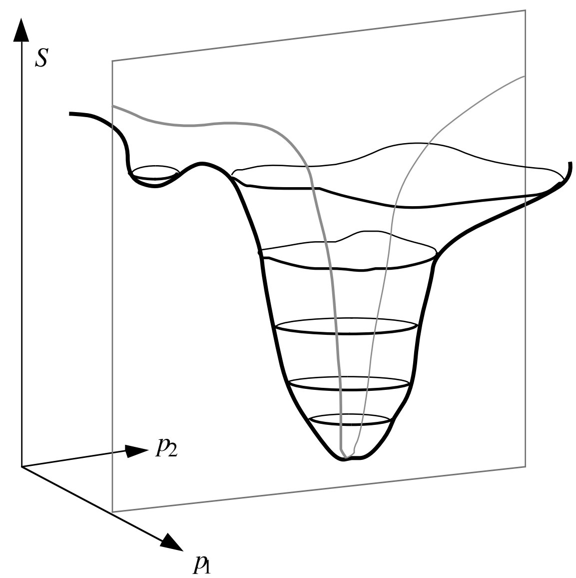


Figure 2.6.2.1. Objective function in two-dimensional parameter space.

2.6.3 Maximum likelihood

Let $\mathbf{p}$ be the parameter vector of length n. More precisely, $\mathbf{p}$ is a hypothesis regarding the values of the deterministic, albeit uncertain model parameters. As before, $\mathbf{z}$ is the observation vector of length m. The probability density function (PDF), $\Phi(\mathbf{z};\mathbf{p})$, is defined as the probability (Pr) of observing the data $\mathbf{z}^*$ if $\mathbf{p}$ were true: $\Phi(\mathbf{z};\mathbf{p}) = \Pr(\mathbf{z}=\mathbf{z}^* | \mathbf{p})$. Note that $\mathbf{p}$ is unknown, nevertheless deterministic. If the observations are independent random variables, the joint PDF is given by the product of the probabilities of the individual observations:

$$\Phi(\mathbf{z};\mathbf{p}) = \prod_{i=1}^{m} \Phi(z_i;\mathbf{p}) \tag{2.6.3.1}$$

From a different perspective, this equation may be seen as describing the likelihood of $\mathbf{p}$ when z_i^* is fixed. This is termed the *likelihood function*:

$$\Phi(\mathbf{z};\mathbf{p}) \Leftrightarrow L(\mathbf{p};\mathbf{z}^*) \tag{2.6.3.2}$$

For each parameter set $\mathbf{p}$, the likelihood function $L(\mathbf{p};\mathbf{z}^*)$ gives the probability of observing $\mathbf{z}^*$. Thus we can think of $L(\mathbf{p};\mathbf{z}^*)$ as a measure of how likely $\mathbf{p}$ is to have produced the observed data $\mathbf{z}^*$. In other words, the likelihood function quantifies the degree to which the data support a given hypothesis regarding the model parameters. The method of maximum likelihood consists of finding the parameter set that is most likely to have produced the data.

The maximum likelihood approach is discussed for the normal distribution in the following section, leading to the least squares estimator. Other estimators, such as the L_1- or Cauchy estimators presented in Section 2.6.5, may also be derived from maximum likelihood considerations (see also *Larsen and Marx* [1986]).

2.6.4 Least squares

Under certain circumstances it is reasonable to assume that the measurement errors $(\mathbf{z}^* - \tilde{\mathbf{z}})$ are normally distributed with mean $E[(\mathbf{z}^* - \tilde{\mathbf{z}})] = 0$ and covariance matrix $E[(\mathbf{z}^* - \tilde{\mathbf{z}})(\mathbf{z}^* - \tilde{\mathbf{z}})^T] = \mathbf{C}_{zz}$. This assumption is only valid if sufficient data points exist, in which case the maximum likelihood estimator becomes consistent and asymptotically efficient, leading to unbiased and normally distributed estimates. Deviations from these assumptions are discussed in Section 2.6.5. The likelihood function for normally distributed errors can be written in its multivariate form as

$$L(\mathbf{p};\mathbf{z}^*) = (2\pi)^{-m/2} |\mathbf{C}_{zz}|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{z}^* - \tilde{\mathbf{z}})^T \mathbf{C}_{zz}^{-1} (\mathbf{z}^* - \tilde{\mathbf{z}})\right] \tag{2.6.4.1}$$

In order to determine the maximum of Equation (2.6.4.1), it is generally easier to minimize the negative *log-likelihood* or *support* criterion:

$$\Gamma = -2\ln[L(\mathbf{p};\mathbf{z}^*)] \tag{2.6.4.2}$$

The log-likelihood criteria from independent data sets can simply be summed up to obtain the log-likelihood of all the data used in an inversion. The negative log-likelihood function for normally distributed errors is given by

$$\Gamma = m \cdot \ln(2\pi) + \ln\left|\mathbf{C}_{zz}\right| + \left[(\mathbf{z}^* - \tilde{\mathbf{z}})^T \mathbf{C}_{zz}^{-1} (\mathbf{z}^* - \tilde{\mathbf{z}})\right] \tag{2.6.4.3}$$

where the second term on the right-hand side could be expanded to include separate terms for the error variance σ_0^2 and matrix $\mathbf{V}_{zz}$, and the last term could be written as a sum of the contributions from different, independent data sets, such as pressures, flow rates, prior information, etc. The first term is a constant.

If the stochastic model—the covariance matrix $\mathbf{C}_{zz}$—is assumed to be known and fixed, minimizing Equation (2.6.4.3) is equivalent to minimizing the Gauss-Markov objective function

$$S = (\mathbf{z}^* - \mathbf{z}(\mathbf{p}))^T \mathbf{C}_{zz}^{-1} (\mathbf{z}^* - \mathbf{z}(\mathbf{p})) \tag{2.6.4.4a}$$

Note that the vector of the true system behavior, $\tilde{\mathbf{z}}$, was replaced with the vector holding the model results, $\mathbf{z}(\mathbf{p})$, which depend on the parameter vector $\mathbf{p}$. Since $(\mathbf{z}^* - \mathbf{z}(\mathbf{p})) = \mathbf{r}$, and $\mathbf{C}_{zz}$ is a diagonal matrix, the objective function S is the sum of the squared residuals, weighted by the inverse of the prior variances σ_i^2:

$$S = \sum_{i=1}^{m} \frac{r_i^2}{\sigma_{z_i}^2} \tag{2.6.4.4b}$$

SUMMARY

If the residuals are normally distributed, minimizing the weighted least squares objective function leads to maximum likelihood estimates.

2.6.5 Robust estimators

Using the definitions introduced in Section 2.5.2, the classical assumption underlying least-squares estimation can be described as follows: (1) the random errors ε are independent, (2) they are normally distributed with mean zero and variance σ_z^2, and (3) there are no systematic errors, i.e., $b = 0$. In many hydrogeologic applications, these assumptions are unlikely to be satisfied. The consequences of making inappropriate assumptions about the residuals are discussed in this section, and the concept of robust estimation is introduced and critically reviewed.

There are two types of violations of the standard assumption. The first considers random errors that do not follow a Gaussian distribution. This might occur if the error distribution is contaminated by a few large outliers. Since the number of data points used in an inversion is finite, even a small number of deviate points causes the least-squares fit to be distorted, leading to parameter estimates with low precision. A similar effect occurs if the error distribution is heavy-tailed, for example, if a Gaussian distribution is contaminated by a large number of relatively small outliers.

The second type of violation occurs in the presence of systematic errors that usually lead to an asymmetric distribution of the residuals. If certain portions of the data exhibit a systematic error, the corresponding residuals are likely to become deviate points. If a certain systematic error affects a single point used for calibration, it cannot be determined whether the large residual stems from a systematic error or is an outlier as a result of a random process; such a distinction is also insignificant. If multiple calibration points are affected by the same systematic error source, the corresponding residuals are strongly correlated and tend to have the same sign over a certain interval in space and time. The ensemble of residuals contaminated with systematic errors, however, can be viewed as one or several outlier points. The interpretation of systematic errors as equivalent, usually large outliers is the main reasoning for subjecting them to robust estimation methods. But it is obvious that if the entire data set or model is flawed, such errors cannot be mitigated by using robust estimators.

Systematic errors may be local both in time and space. For example, inconsistent initial or boundary conditions often result in systematic deviations between the data and the model prediction at early or late times during a transient experiment, leading to errors in a specific time segment. Similarly, a data set from a sensor that is either defective or placed in a unit that is poorly represented in the model leads to erroneous residuals at a specific point in space, again corrupting the inversion. Note that these types of systematic errors may not appear as obvious outliers and are therefore difficult to identify.

Before we introduce the robust estimators, we would like to emphasize that the main effort in estimating parameters by inverse modeling should be placed on avoiding systematic errors and minimizing random errors. The robust estimators presented here do not exempt the experimentalist and/or modeler from a comprehensive test design, careful execution of the experiment, accurate model development, and conscientious analysis of the inverse modeling results. However, systematic errors in the conceptual model and non-Gaussian random errors in the data are inherent in inverse modeling, and the problems associated with systematic errors seem to be accentuated rather than alleviated by the use of the standard least-squares estimator.

An overview of robust statistical procedures with mathematically rigorous definitions of their underlying concepts can be found in *Huber* [1981, 1996]. Here, we follow a more intuitive approach and introduce the robust estimators by discussing their common property of reducing the weight of deviate points.

As a generalization of Equation (2.6.4.4), fitting a model to data for parameter estimation can be formulated as a minimization problem of the form [*Haining*, 1990]

$$\text{minimize} \quad S = \sum_{i=1}^{m} \omega(y_i; \mathbf{p}) \tag{2.6.5.1}$$

Here, ω is an arbitrary loss function, which is a function of the weighted residuals

$$y_i = \frac{z_i * - z_i(\mathbf{p})}{\sigma_{z_i}} \tag{2.6.5.2}$$

where σ_{z_i} is the measurement error. At the minimum of S, the derivatives of the objective function (2.6.5.1) with respect to the parameters p_j vanish

$$\sum_{i=1}^{m} \frac{1}{\sigma_{z_i}} \psi \frac{\partial y_i}{\partial p_j} = 0 \qquad j = 1, \ldots, n \tag{2.6.5.3}$$

where the function ψ is defined as the derivative of the loss function, $\psi \equiv \partial\omega / \partial y$.

The choice of the arbitrary loss function ω can be based on probabilistic considerations (as discussed in Section 2.6.3), with ω being the negative logarithm of the probability density function. When adopting this viewpoint, the parameters $\mathbf{p} = \hat{\mathbf{p}}$ of a model $\mathbf{y}(\mathbf{p})$ that minimize Equation (2.6.5.1) are the maximum-likelihood estimates for $\mathbf{p}$. As outlined in Section 2.6.4 for normally distributed errors, the loss function can be directly derived from the Gaussian distribution to be $\omega(y) = (1/2)y^2$ and $\psi(y) = y$. Note that the ψ function serves as a weighting function in Equation (2.6.5.3). For example, least squares assigns greater weights to increasingly deviant points, reflecting the assumption that outliers are very unlikely according to the normal distribution. Consequently, if we suppose that the weighted residuals follow a distribution with a longer tail, that is, with a somewhat larger probability of encountering points removed from the central region, we should choose a ψ function that yields decreasing relative weights for deviant points. It is expected that reducing the weight of outliers makes the estimator more robust.

Many functions with the desired properties have been proposed in the literature (see *Andrews et al.* [1972], *Press et al.* [1992]). Some are maximum-likelihood estimators for known error distributions, whereas others do not correspond to a standard probability density function. Five estimators are implemented into iTOUGH2. They include (1) Least Squares (LS), (2) Least Absolute Residual (LAR) or L_1-estimator, (3) the maximum-likelihood estimator for measurement errors following a Cauchy distribution, (4) one of the robust estimators proposed by Huber, and (5) the Andrews estimator. Their functional forms—c is a user-specified parameter—are summarized in Table 2.6.5.1. The loss function $\omega(y)$ of the five estimators is shown in Figure 2.6.5.1.

Note that for the Andrews estimator, observations with weighted residuals larger than $c\pi$ are considered to be true outliers and are not counted at all in the estimation of the parameters. This property may lead to difficulties when using the Andrews estimator in a nonlinear optimization problem where the initial guess $\mathbf{p}_0$ is far away from the best estimate, in which case the initial residuals are too large. As a consequence, the gradient of the objective function becomes unstable, making it difficult for the minimization algorithm to converge. It is therefore suggested to first perform a standard least-squares fit before switching to the Andrews estimator. The five estimators are compared in a study by *Finsterle and Najita* [1998].

Table 2.6.5.1. Estimator, Underlying Distribution, Loss Function, and ψ Function

Estimator, Distribution	Loss Function ω	ψ Function
Least-squares, Gaussian	$\omega = \frac{1}{2} y^2$	$\psi = y$
L_1-estimator, Double exponential	$\omega = \lvert y \rvert$	$\psi = \begin{cases} 1 \\ -1 \end{cases}$ for $\begin{matrix} y > 0 \\ y < 0 \end{matrix}$
Cauchy, Cauchy	$\omega = \log\left(1 + \frac{1}{2} y^2\right)$	$\psi = \dfrac{y}{1 + \frac{1}{2} y^2}$
Huber (quadratic-linear) @	$\omega = \begin{cases} y^2/2 \\ c\lvert y \rvert - c^2/2 \end{cases}$ for $\begin{matrix} \lvert y \rvert \le c \\ \lvert y \rvert > c \end{matrix}$	$\psi = \begin{cases} -c \\ y \\ c \end{cases}$ for $\begin{matrix} y < -c \\ \lvert y \rvert \le c \\ y > c \end{matrix}$
Andrews, @	$\omega = \begin{cases} 1 - \cos(y / c) \\ 2 \end{cases}$ for $\begin{matrix} \lvert y \rvert \le c\pi \\ \lvert y \rvert > c\pi \end{matrix}$	$\psi = \begin{cases} \sin(y / c) \\ 0 \end{cases}$ for $\begin{matrix} \lvert y \rvert \le c\pi \\ \lvert y \rvert > c\pi \end{matrix}$

@ No standard probability distribution available

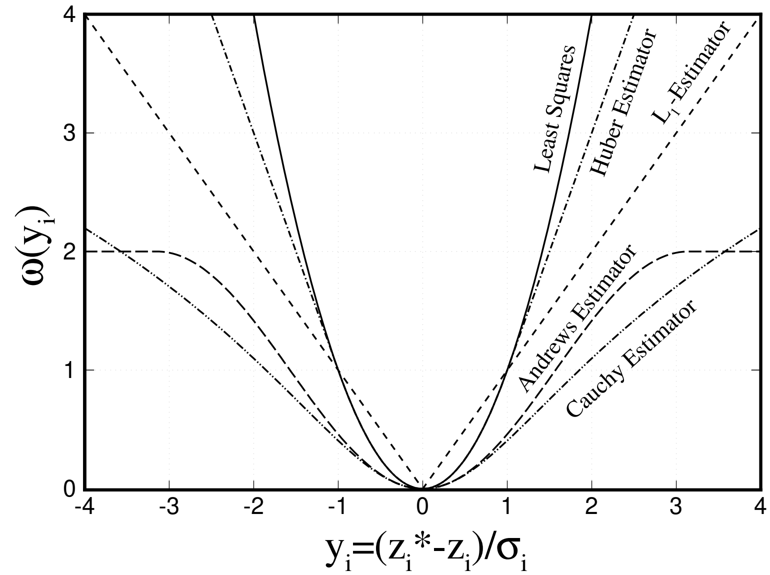


Figure 2.6.5.1. Loss function ω of five estimators as a function of the weighted residual.

SUMMARY

Errors in either the data or the numerical model used for the inversion usually exhibit a non-Gaussian distribution. While the standard error of the residuals is by definition smallest when using least squares, robust estimators are less affected by the presence of random errors following a heavy-tailed distribution or by systematic modeling errors, leading to more consistent and less biased estimates. The robust estimators only perform better for a specific type of systematic errors, and the errors must be contained within a limited portion of the data. Systematic errors and outliers should be eliminated whenever possible.

RELATED iTOUGH2 COMMANDS

The following commands can be used to select the objective function:
`>>> LEAST-SQUARES` (default), `>>> ANDREWS`, `>>> CAUCHY`,
`>>> L1ESTIMATOR`, and `>>> QUADRATIC-LINEAR`.

2.7 Minimization Algorithm

2.7.1 Classification

The purpose of the minimization algorithm is to find the minimum of the objective function by iteratively updating the parameters of the model. The objective function is a global measure of misfit between the data and the corresponding model output. Since the model output $\mathbf{z}(\mathbf{p})$ depends on the parameters, the fit can be improved by changing the elements of the parameter vector $\mathbf{p}$. The search for the minimum occurs in the n-dimensional parameter space. A number of strategies were developed to find parameter combinations that reduce the value of the objective function. The available methods are often classified based on criteria such as the mathematical form of the constraints imposed, the method of regularization, the solution method for obtaining orthogonal matrices, etc. [*Jacoby et al.*, 1972; *Gill et al.*, 1981; *Björck*, 1996]. We classify the methods according to whether they are based on a sequence of forward simulations only, or whether they require calculation of the gradient or second-order derivatives. With the exception of Simulated Annealing, all methods described below identify only a local minimum near the starting point.

Non-Derivative Methods: In these methods, the model is evaluated for different parameter combinations, mapping out the objective function in the n-dimensional parameter space. They are also referred to as *Function Comparison Methods*. Because no derivatives of the objective function with respect to the parameters must be calculated, these methods are not restricted to smooth models. However, they usually require many trial simulations and are therefore inefficient. Examples of such direct search methods include:

- Trial and error
- Grid search
- Downhill Simplex
- Simulated Annealing
- Genetic algorithms

Gradient-Based Methods: These methods require calculating the gradient of the objective function with respect to the parameter vector. Updating the parameter vector in small steps along the search direction determined by the gradient is a robust, albeit inefficient, procedure. Various modifications of this basic scheme have been proposed. Their main difference lies in the choice of an appropriate step length. Efficient ways of calculating the gradient have been described in the literature [*Carrera and Neuman*, 1986b;*Sun and Yeh*, 1990;*Vasco and Datta-Gupta*, 1999]. Examples of gradient-based methods include:

- Steepest descent
- Quasi-Newton methods
- Conjugate gradient methods
  - Method of Fletcher-Reeves
  - Method of Broyden
  - Method of Fletcher-Powell-Davidon

Second Derivative Methods: These methods are based on the Hessian matrix or various approximations thereof (quasi-linearization). They perform well for nearly linear least-squares problems. The computational cost for calculating the second derivatives is usually compensated by an efficient stepping in the parameter space. Examples of second-order methods include:

- Newton method
- Gauss-Newton method
- Levenberg-Marquardt

Each of these methods has its advantages and disadvantages. The choice of an appropriate method largely depends on the presumed properties of the objective function. The following algorithms are implemented into iTOUGH2:

- Gauss-Newton
- Levenberg-Marquardt
- Downhill Simplex
- Simulated Annealing
- Grid Search

The Levenberg-Marquardt minimization algorithm (see Section 2.7.3) was found to perform well for most iTOUGH2 applications. It is a modification of the Gauss-Newton algorithm for nonlinear least-squares optimization, which will be described first in Section 2.7.2. The Downhill Simplex method (Section 2.7.4) does not require the calculation of derivatives; its convergence rate, however, is usually slow. Simulated Annealing, described in Section 2.7.5, has the advantage of being able to escape local minima, but requires many solutions of the forward problem. Simply evaluating parameter combinations over the entire range of possible values (Grid Search, see Section 2.7.6) provides the database for a complete mapping of the objective function in the parameter space. However, this method is computationally prohibitive for most applications, and is used mostly for illustrative purposes (see, for example, *Finsterle and Faybishenko* [1999]). A comparison of all minimization algorithms can be found in Section 2.7.9 as well as *Finsterle* [1999c; Problem 4].

All methods presented here are iterative, i.e., they start with an initial parameter set, and an update vector is calculated at each iteration. A step is successful if the new parameter set at iteration $(k+1)$,

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k \tag{2.7.1.1}$$

leads to a reduction in the objective function, i.e.

$$S(\mathbf{p}_{k+1}) < S(\mathbf{p}_k) \tag{2.7.1.2}$$

The algorithms discussed in Sections 2.7.3 through 2.7.6 differ in the way they calculate $\Delta\mathbf{p}_k$. We first introduce in Section 2.7.2 the gradient, Jacobian, and Hessian matrices as basic elements of Newton's method, before we discuss the Gauss-Newton and Levenberg-

Marquardt algorithms, which are both specific cases of the Newton method. The Downhill Simplex method and Simulated Annealing are described in Sections 2.7.5 and 2.7.6, respectively.

2.7.2 Gradient, Jacobian, and Hessian matrix

The Gauss-Newton and Levenberg-Marquardt algorithms belong to a class of methods that are based on a quadratic approximation of the objective function, in contrast to the linear assumption in steepest-descent methods. If first and second derivatives of S are available, a quadratic model of the objective function can be obtained from the first three terms of the Taylor-series expansion:

$$S(\mathbf{p}_{k+1}) \approx S(\mathbf{p}_k) + \mathbf{g}_k^T \Delta\mathbf{p}_k + \frac{1}{2}\Delta\mathbf{p}_k^T \mathbf{H}_k \Delta\mathbf{p}_k \tag{2.7.2.1}$$

The minimum of the right-hand side of (2.7.2.1) is achieved if $\Delta\mathbf{p}_k$ minimizes the quadratic function

$$\Phi(\Delta\mathbf{p}) = \mathbf{g}_k^T \Delta\mathbf{p} + \frac{1}{2}\Delta\mathbf{p}^T \mathbf{H}_k \Delta\mathbf{p} \tag{2.7.2.2}$$

Here, $\mathbf{g}_k$ is the gradient vector and $\mathbf{H}_k$ is the Hessian matrix. For the least-squares objective function (2.6.4.4) or approximations thereof, the gradient vector has elements

$$g_j = 2\sum_{i=1}^{m} \frac{r_i}{\sigma_{z_i}^2} \frac{\partial r_i}{\partial p_j} \qquad j = 1,\ldots,n \tag{2.7.2.3}$$

Defining the Jacobian matrix as

$$\mathbf{J} = -\frac{\partial \mathbf{r}}{\partial \mathbf{p}} = \frac{\partial \mathbf{z}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial z_1}{\partial p_1} & \cdots & \frac{\partial z_1}{\partial p_n} \\ \vdots & & \vdots \\ \frac{\partial z_m}{\partial p_1} & \cdots & \frac{\partial z_m}{\partial p_n} \end{bmatrix} \tag{2.7.2.4}$$

the gradient vector at iteration k can be written as

$$\mathbf{g}_k = -2\mathbf{J}_k^T \mathbf{C}_{zz}^{-1} \mathbf{r}_k \tag{2.7.2.5}$$

The Hessian $\mathbf{H}$ is an $n \times n$ matrix with the second partial derivatives of the objective function. For least squares, the Hessian can be written as

$$\mathbf{H}_k = 2\left(\mathbf{J}_k^T \mathbf{C}_{zz}^{-1} \mathbf{J}_k + \sum_{i=1}^{m} r_i \mathbf{G}_i \right) \tag{2.7.2.6}$$

where $\mathbf{G}_i = \nabla^2 r_i / \sigma_{z_i}$ is the Hessian of the weighted residuals. Denoting the sum in Equation (2.7.2.6) with $\mathbf{B}$, it becomes obvious that the Hessian of a least-squares objective function consists of a special combination of first- and second-order information:

$$\mathbf{H}_k = 2\left(\mathbf{J}_k^T \mathbf{C}_{zz}^{-1} \mathbf{J}_k + \mathbf{B}\right) \tag{2.7.2.7}$$

Note that $\mathbf{B}$ is zero if the model is linear, and becomes significant only for highly nonlinear models and if the residuals are large, as is the case far away from the minimum and with noisy data. Also note that the positive and negative residuals in $\mathbf{B}$ do not cancel one another, i.e., the Hessian is not necessarily a positive-definite matrix.

At the minimum of Equation (2.7.2.2), $\Delta\mathbf{p}_k$ satisfies the linear system

$$\mathbf{H}_k \Delta\mathbf{p}_\mathbf{k} = -\mathbf{g}_k \tag{2.7.2.8}$$

Combining Equations (2.7.2.5), (2.7.2.7) and (2.7.2.8) yields Newton's method:

$$\Delta\mathbf{p}_k = \left(\mathbf{J}_k^T \mathbf{C}_{zz}^{-1} \mathbf{J}_k + \mathbf{B}\right)^{-1} \mathbf{J}_k^T \mathbf{C}_{zz}^{-1} \mathbf{r}_k \tag{2.7.2.9}$$

The various iterative solutions to the nonlinear least-squares problem are based on different approximations to the Hessian, as discussed in the following two sections.

There are several methods for calculating the elements of the Jacobian matrix, i.e., the sensitivity coefficients, including (1) the *Influence Coefficient* or *Perturbation Method*, (2) the *Sensitivity Equation* or *Direct Derivative Method*, and (3) the *Variational Method* [*Yeh*, 1986; *Carrera*, 1988]. In iTOUGH2, the Jacobian matrix (2.7.2.4) is evaluated numerically using the Perturbation Method with either forward or centered finite differences:

forward
$$J_{ij} = \frac{\partial z_i}{\partial p_j} \approx \frac{z_i(\mathbf{p}; p_j + \delta p_j) - z_i(\mathbf{p})}{\delta p_j} \tag{2.7.2.10a}$$

centered
$$J_{ij} = \frac{\partial z_i}{\partial p_j} \approx \frac{z_i(\mathbf{p}; p_j + \delta p_j) - z_i(\mathbf{p}; p_j - \delta p_j)}{2\delta p_j} \tag{2.7.2.10b}$$

Here, δp_j is a small perturbation of parameter j, usually given as a fraction α of the parameter value, $\delta p_j = \alpha \cdot p_j$. Note that calculating a forward finite-difference approximation of the Jacobian requires $(n+1)$ TOUGH2 simulations. Centered finite differences are more accurate, but require $(2n+1)$ forward runs. High accuracy is usually not required by the minimization algorithm, but may be desirable for the error analysis.

RELATED iTOUGH2 COMMANDS

The evaluation of the Jacobian matrix is governed by the subcommands of command `>> JACOBIAN`.

2.7.3 The Gauss-Newton method

The Gauss-Newton method is based on the premise that the first-order term $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$ of the Hessian dominates relative to the second-order term $\mathbf{B}$. This assumption is justified for linear and mildly nonlinear problems and for nonlinear problems near the solution, where the residuals are expected to be small, i.e., when the objective function is sufficiently smaller than the eigenvalues of $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$.

In the Gauss-Newton method, the Hessian is approximated by setting $\mathbf{B}$ to zero, which ensures that matrix $\mathbf{H}$ in Equation (2.7.2.8) is positive definite. This is equivalent to approximating the actual objective function by a quadratic function as illustrated for one and two parameters in Figure 2.7.3.1. The Gauss-Newton direction is given by

$$\Delta\mathbf{p}_k = \left(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k\right)^{-1}\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{r}_k \qquad (2.7.3.1)$$

which is the solution of the linear least-squares problem. For nonlinear models, the parameter vector is updated, and a new Gauss-Newton direction is calculated. The procedure is summarized in Table 2.7.3.1.

Table 2.7.3.1. Gauss-Newton Minimization Algorithm

Step 1:	Initialization: - Set iteration index $k = 0$. - Define initial parameter set $\mathbf{p}_{k=0} = \mathbf{p}_0$ (usually $\mathbf{p}_0 = \mathbf{p}^*$).
Step 2:	Run simulation model with parameter vector $\mathbf{p}_k$.
Step 3:	Evaluate $\mathbf{r}(\mathbf{p}_k)$, $S(\mathbf{p}_k)$, and $\mathbf{J}(\mathbf{p}_k)$.
Step 4:	Calculate parameter update: $\Delta\mathbf{p}_k = \left(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k\right)^{-1}\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{r}_k$.
Step 5:	Update parameter vector: $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k$.
Step 6:	Perform simulation and evaluate $S(\mathbf{p}_{k+1})$.
Step 7:	Evaluate convergence criteria (see Section 2.7.8). If converged, go to Step 8, else set $k = k + 1$ and go to Step 2.
Step 8:	Minimization terminated. Proceed with residual and uncertainty analysis.

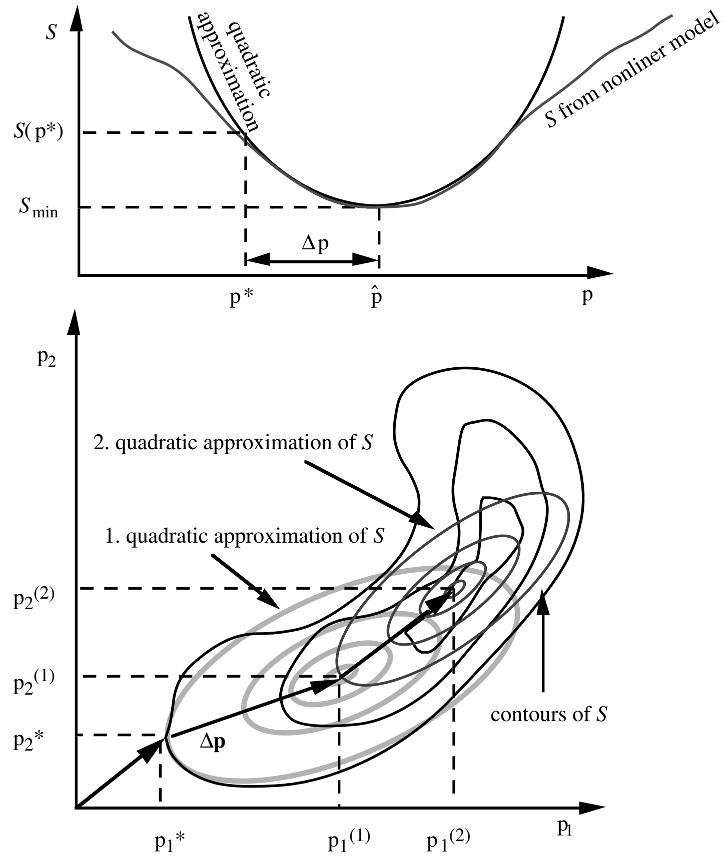


Figure 2.7.3.1. Objective function of a linearized least-squares problem as a function of one parameter (top), and two parameters (bottom).

The Gauss-Newton method is efficient if the initial guess is close to the minimum and/or the model is nearly linear, i.e., if $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$ is a good approximation of the Hessian. However, if the model is highly nonlinear, the parameter update calculated by Equation (2.7.3.1) can be too large, leading to an inefficient or even unsuccessful step in which the value of the objective function is increased rather than decreased.

RELATED iTOUGH2 COMMANDS

The Gauss-Newton algorithm is invoked by command `>>> GAUSS-NEWTON`. This sets the Levenberg parameter (see Section 2.7.4) to zero and skips the control runs and updating of the Levenberg parameter.

2.7.4 The Levenberg-Marquardt method

For strongly nonlinear models, if the parameter vector $\mathbf{p}_k$ is far away from the optimum parameter set, the Hessian is not necessarily a positive-definite matrix, and the approximation $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$ used by the Gauss-Newton method may not lead to an efficient or successful step. In the Levenberg-Marquardt method, the approximation to the Hessian is made positive definite by replacing $\mathbf{B}$ in Equation (2.7.2.7) with an $n \times n$ diagonal matrix $\lambda_k\mathbf{D}_k$:

$$\Delta\mathbf{p}_k = \left(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k + \lambda_k\mathbf{D}_k\right)^{-1}\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{r}_k \qquad (2.7.4.1)$$

The scalar λ is the so-called *Levenberg parameter* [*Levenberg*, 1944], and the elements of matrix $\mathbf{D}_k$ are given by $D_{jj} = (\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k)_{jj}$, $j = 1,\ldots,n$. If λ_k is zero, $\Delta\mathbf{p}_k$ is identical with the Gauss-Newton step; as $\lambda_k \to \infty$, the approximation of the Hessian becomes diagonally dominant. Consequently, $\Delta\mathbf{p}_k$ becomes parallel to the steepest-descent direction, and the step length approaches zero. After each iteration, the Levenberg parameter is either increased or decreased following a scheme proposed by *Marquardt* [1963] (see Table 2.7.4.1). Far away from the minimum, i.e., during the first few iterations, a relatively large value of λ_k is chosen, leading to small steps along the gradient of the objective function. Stepping along the steepest-descent direction is a robust strategy, ensuring that $S(\mathbf{p}_{k+1}) < S(\mathbf{p}_k)$ for sufficiently large λ_k. However, the step length $\|\Delta\mathbf{p}_k\|_2$ may be very small and minimization becomes inefficient. Therefore, λ_k is decreased by a factor of $1/\nu$ after each successful step, where $\nu > 1$ is the so-called Marquardt parameter. With decreasing λ_k, $\Delta\mathbf{p}_k$ approaches the Gauss-Newton step with its quadratic convergence rate. If an unsuccessful step is proposed, i.e., the objective function is increased, λ_k is increased by ν.

Figure 2.7.4.1 shows the contours of $(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k)$, which is the quadratic approximation of S at iteration k. The curved line connecting $\mathbf{p}^*$ with the center of the ellipse indicates the possible end points of Levenberg-Marquardt steps $\Delta\mathbf{p}_k(\lambda)$ as a function of λ.

The Levenberg-Marquardt method can be viewed as a flexible combination of the robustness of a steepest-descent method and the efficiency of a second-order Gauss-Newton method. While standard first-derivative methods require a strategy to estimate the length of the step to be taken along the gradient, the Levenberg-Marquardt algorithm takes this information from the curvature matrix $(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k)$, and uses it to increase the efficiency as the minimum is approached. The Marquardt scheme of adapting the search direction and step length provides the flexibility needed for the minimization of objective functions with a complex topography.

RELATED iTOUGH2 COMMANDS

The Levenberg-Marquardt method is the default minimization algorithm in iTOUGH2; it can be explicitly selected using command `>>> LEVENBERG-MARQUARDT`. The initial value of the Levenberg parameter (default: $\lambda_0 = 10^{-3}$) and the Marquardt parameter (default: $\nu = 10$) are selected using commands `>>> LEVENBERG` and `>>> MARQUARDT`.

Table 2.7.4.1. Levenberg-Marquardt Minimization Algorithm

Step 1:	Initialization: - Set iteration index $k = 0$. - Define initial Levenberg parameter (default: $\lambda_0 = 10^{-3}$). - Define Marquardt parameter (default: $\nu = 10$). - Define initial parameter set $\mathbf{p}_{k=0} = \mathbf{p}_0$.
Step 2:	Run simulation model with parameter vector $\mathbf{p}_k$.
Step 3:	Evaluate $\mathbf{r}(\mathbf{p}_k)$, $S(\mathbf{p}_k)$, and $\mathbf{J}(\mathbf{p}_k)$.
Step 4:	Calculate parameter update: $\Delta\mathbf{p}_k = \left(\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k + \lambda_k\mathbf{D}_k\right)^{-1}\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{r}_k$ where $\mathbf{D}_k$ is an $n \times n$ matrix with elements $D_{jj} = (\mathbf{J}_k^T\mathbf{C}_{zz}^{-1}\mathbf{J}_k)_{jj}$, $j = 1,\ldots,n$.
Step 5:	Update parameter vector: $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta\mathbf{p}_k$.
Step 6:	Perform simulation and evaluate $S(\mathbf{p}_{k+1})$.
Step 7:	If $S(\mathbf{p}_{k+1}) < S(\mathbf{p}_k)$, multiply λ by factor $1/\nu$ and go to Step 8. If $S(\mathbf{p}_{k+1}) > S(\mathbf{p}_k)$, multiply λ by factor ν and go to Step 4.
Step 8:	Evaluate convergence criteria (see Section 2.7.8). If converged, go to Step 9, else set $k = k+1$ and go to Step 2.
Step 9:	Minimization terminated. Proceed with residual and uncertainty analysis.

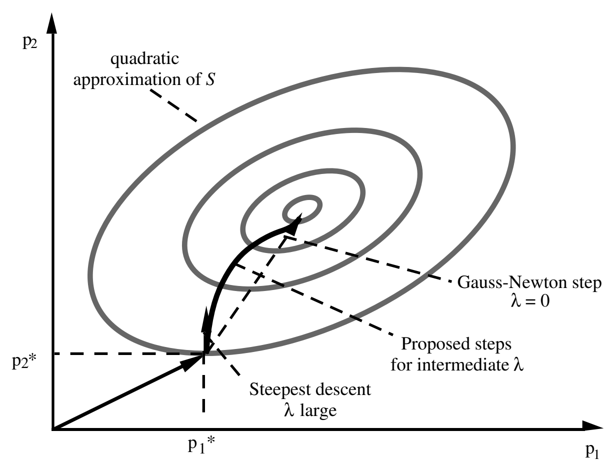


Figure 2.7.4.1. Steps proposed by the Levenberg-Marquardt method as a function of the Levenberg parameter λ.

2.7.5 The Downhill Simplex method

The Downhill Simplex method [*Press et al.*, 1992] does not make any assumptions about the topography of the objective function and does not require derivatives. It is a robust, albeit rather inefficient method.

A *simplex* is a geometrical figure defined by $n+1$ vertices in the n-dimensional parameter space. In iTOUGH2, the initial simplex consists of the initial guess $\mathbf{p}_0$ defining the origin, and n additional points, each of which lies on one of the parameter axes at a distance σ_{p_j} from the origin, where σ_{p_j} is the prior standard deviation or expected variation of parameter j. The Downhill Simplex method takes one of the following steps (see Figure 2.7.5.1):

Reflection The point of the simplex with the largest objective function is moved through the opposite face of the simplex.

Reflection and Expansion If the reflected point is lower than the lowest point of the original simplex, the simplex is expanded in that direction.

One-dimensional Contraction If the reflected point is higher than the second-highest point of the original simplex, the simplex is contracted in that direction to find an intermediate point.

Overall Contraction If one-dimensional contraction is unsuccessful, an overall contraction around the lowest point is performed.

After convergence (see Section 2.7.8), iTOUGH2 evaluates the Jacobian matrix for the subsequent error analysis.

RELATED iTOUGH2 COMMANDS

The Downhill Simplex algorithm is invoked by using command `>>> SIMPLEX`. The size of the initial simplex is defined by commands `>>>> DEVIATION (p)` or `>>>> VARIATION`.

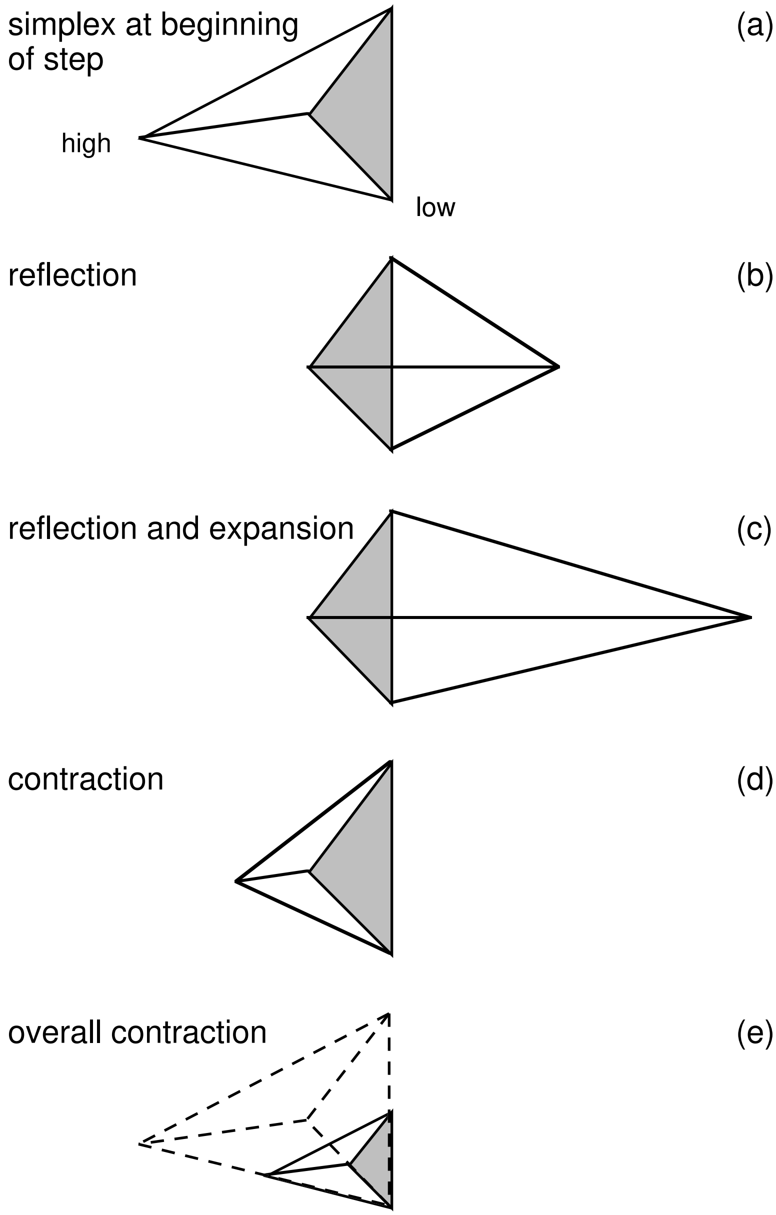


Figure 2.7.5.1. Possible outcomes for a step in the Downhill Simplex method (after *Press et al*. [1992]). The simplex for $n = 3$ is a tetrahedron; (a) shows it at the beginning of a step. The simplex after a step can be (b) a reflection away from the high point, (c) a reflection and expansion away from the high point, (d) a contraction along one dimension from the high point, or (e) a contraction along all dimensions towards the low point.

2.7.6 Simulated Annealing

Simulated Annealing [*Metropolis et al.*, 1953] is a technique suitable for solving large optimization problems, where the objective function is likely to exhibit many local minima. Moreover, no derivatives are required, and the objective function may even be discontinuous.

The basic idea behind the algorithm is an analogy with thermodynamics, specifically with the way metals cool and anneal (see, for example, *Press et al*. [1992]). Simulated Annealing takes random steps $\Delta\mathbf{p}_{k,i}$, which are based on the expected variability of the parameter. In iTOUGH2, the variability decreases during the optimization process. A new parameter set $\mathbf{p}_{k,i} = \mathbf{p}_{k,j} + \Delta\mathbf{p}_{k,i}$ is accepted with probability

$$\phi_k = e^{-\Delta S/\tau_k} \tag{2.7.6.1}$$

where $\Delta S = S(\mathbf{p}_{k,i}) - S(\mathbf{p}_{k,j})$ and τ_k is a controlling parameter analog to the current temperature during the cooling and annealing process; index j counts the number of successful steps at a given temperature level k. The initial temperature τ_0 should be a fraction of the initial objective function $S(\mathbf{p}_0)$. If ΔS is negative, i.e., the objective function is decreased, Equation (2.7.6.1) yields a probability greater than one and the corresponding step $\Delta\mathbf{p}$ is always accepted as a successful downhill move. An uphill move (ΔS is positive) may also be accepted, albeit only with probability ϕ_k. This scheme of always taking a downhill step and sometimes accepting an uphill step with a certain probability, which depends on the temperature τ_k, has come to be known as the *Metropolis* algorithm. The so-called *annealing schedule* describes the reduction of the control parameter τ_k. There are two different annealing schedules available in iTOUGH2:

$$\tau_k = \alpha^k \tau_0 \tag{2.7.6.2a}$$

$$\tau_k = \tau_0 (1 - k / K)^\beta \tag{2.7.6.2b}$$

where $0 < \alpha < 1$ and $\beta > 1$ are constants, and K is the total number of iterations. Furthermore, the standard deviation of the random steps decreases as follows:

$$\left(\sigma_{\Delta p_i}\right)_k = \left(\frac{K+10}{K+11}\right)^k \left(\sigma_{\Delta p_i}\right)_0 \tag{2.7.6.3}$$

Here, $\left(\sigma_{\Delta p_i}\right)_0$ is the standard deviation of the random parameter steps during the first iteration. Both the temperature τ_k and the average step size $\Delta\mathbf{p}$ start large and decrease during the course of the optimization. Thus, strongly varying parameter sets are tested early in the inversion, and uphill steps are more likely to be accepted, allowing the minimization algorithm to escape local minima. Later in the inversion, subtle changes in the parameter set are tested, and only successful downhill steps are likely to be accepted. The method of Simulated Annealing is summarized in Table 2.7.6.1.

Table 2.7.6.1. Minimization by Simulated Annealing

Step 1:	Initialization: - Set iteration index $k = 0$. - Set trial index $i = 1$. - Set counter of successful steps $j = 0$. - Define total number of iterations K. - Define maximum number of trials $i_{\max}$ at each temperature (default: $i_{\max} = 10n$). - Define initial parameter set $\mathbf{p}_{k=0,j=0} = \mathbf{p}_0$. - Define initial control parameter (temperature) τ_0 (default: $\tau_0 = 0.05 \cdot S(\mathbf{p}_0)$).
Step 2:	Generate random perturbation $\Delta\mathbf{p}_{k,i}$. The probability density function of $\Delta\mathbf{p}_{k,i}$ is either uniform or Gaussian. The average step size decreases during the optimization (see Step 6).
Step 3:	Perform simulation with $\mathbf{p}_{k,i} = \mathbf{p}_{k,j} + \Delta\mathbf{p}_{k,i}$ and evaluate $\Delta S = S(\mathbf{p}_{k,i}) - S(\mathbf{p}_{k,j})$.
Step 4:	If $\Delta S < 0$, accept step, i.e., set $\mathbf{p}_{k,j+1} = \mathbf{p}_{k,j} + \Delta\mathbf{p}_{k,i}$; If $\Delta S > 0$, accept step with probability $\phi_k = e^{-\Delta S/\tau_k}$. If step accepted, set $j = j + 1$.
Step 5:	If $i < i_{\max}$ and $j < j_{\max} = i_{\max}/5$, set $i = i + 1$ and go to Step 2
Step 6:	Reduce control parameter τ_k according to the annealing schedule (2.7.6.2); Reduce average size of random steps (Equation 2.7.6.3); Set $i = 0$; set $j = 0$; set $k = k + 1$.
Step 7:	If $k < K$, go to Step 2.
Step 8:	Minimization terminated. Proceed with residual analysis.

An advantage of Simulated Annealing is its ability to escape local minima. However, the method is inefficient because of the randomness of the trials $\Delta\mathbf{p}_{k,i}$, which almost always propose an uphill step, especially near the minimum or in narrow valleys of the objective function. It is therefore suggested to use Simulated Annealing in combination with the other minimization algorithms discussed above.

RELATED iTOUGH2 COMMANDS

See subcommands of command `>>> ANNEAL`.

2.7.7 Grid Search

Grid search refers to the systematic evaluation of the objective function in the n-dimensional parameter space. Parameter sets are either generated on a regular grid in the parameter space, or can be supplied by the user to examine certain regions using any search pattern deemed reasonable.

While inefficient and often prohibitive for large numbers of parameters, grid search provides the complete topography of the objective function, revealing the presence of local minima, nonuniqueness, instabilities, etc.

RELATED iTOUGH2 COMMANDS

See command `>>> GRID SEARCH`.

2.7.8 Stopping criteria

All minimization algorithms presented in Sections 2.7.3 through 2.7.6 are iterative methods, in which the minimum is approached by proposing new parameter sets that lead to reduced values of the objective function. Convergence or stopping criteria must be specified to decide whether the minimum is identified. Theoretically, the minimum is detected if all elements of the gradient vector $\partial S / \partial \mathbf{p}_k$ are zero. In practice, however, one of the following convergence criteria is used to stop optimization:

- The number of iterations (steps), k, exceeds a specified number, K;
- The number of forward runs exceeds a specified number;
- The number of unsuccessful uphill steps exceeds a specified number;
- The normalized step size is smaller than a specified tolerance;
- The norm of the gradient vector is smaller than a specified tolerance;
- The objective function is smaller than a specified tolerance.

Only the Gauss-Newton and Levenberg-Marquardt algorithms can make use of the criteria related to the gradient. The objective function is usually substantially reduced during the first few steps. Limiting the number of iterations based on experience is thus a reasonable stopping strategy.

An iTOUGH2 simulation may also stop due to an error or convergence failure in the forward run, interrupting the optimization process.

RELATED iTOUGH2 COMMANDS

See subcommands of command `>> CONVERGE/STOP`.

2.7.9 Step size limitation and parameter selection

It is sometimes useful to limit the size of the step $\Delta\mathbf{p}_k$ taken during any one iteration. Large steps are usually proposed by the second-order methods whenever the sensitivity of a parameter is small and the parameter is strongly correlated to a parameter with high sensitivity. Step-size limitation may also prevent the algorithm from moving too far beyond the region in which the linearity assumption is justified.

Several strategies for limiting the step size are implemented into iTOUGH2. They include:

- Limitation of the step size of an individual parameter;
- Limitation of the total step length;
- Limitation of the scaled total step length;
- Automatic parameter selection.

Limitation of the step size of an individual parameter. A maximum step length can be specified individually for each parameter. Figure 2.7.9.1 illustrates that reducing the step size of parameter i from $\Delta p'_i$ to $\Delta p_{i,\max}$ also changes the direction of the step taken. Nevertheless, this approach was found to be superior to one that maintains the original search direction.

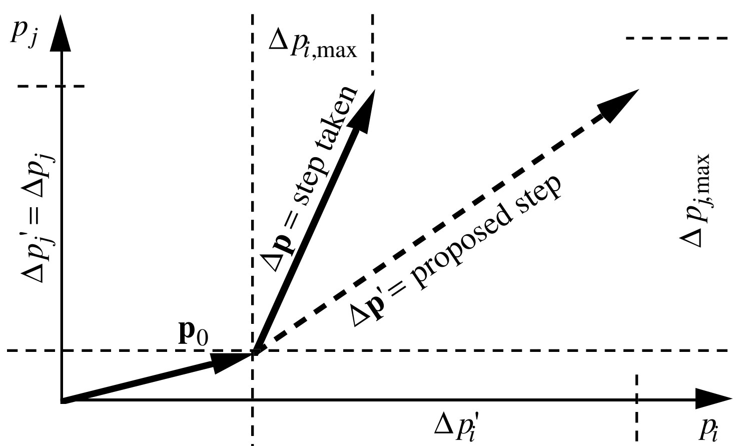


Figure 2.7.9.1. Step-size limitation of a single parameter.

Limitation of the total step size. The size of the scaled or unscaled update vector $\Delta\mathbf{p}_k$ can be limited. The step length of the scaled ($f_i = p_i$, default) or unscaled ($f_i = 1$) parameter update vector is defined as follows:

$$|\Delta\mathbf{p}| = \left[\sum_{i=1}^{n}\left(\frac{\Delta p_i}{f_i}\right)^2\right]^{1/2} \tag{2.7.9.1}$$

This is a global step-size limitation as opposed to the one specified for individual parameters. The scaling is necessary if the concurrently estimated parameters vary considerably in size. Figure 2.7.9.2 illustrates that limiting the step size maintains the direction of the step taken in the parameter space.

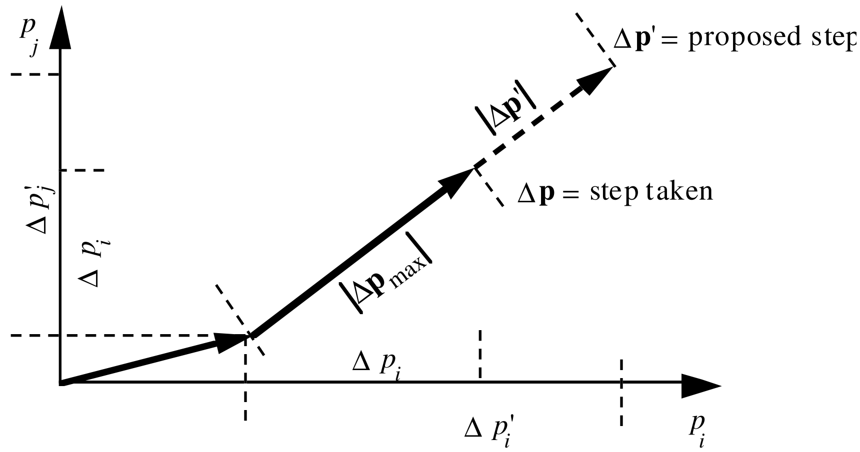


Figure 2.7.9.2. Global step-size limitation.

Automatic parameter selection. If a parameter is not sensitive enough to be estimated from the available data at a given iteration, it should be removed from the set of parameters being updated. Parameters that are (temporarily) removed from the parameter set remain at their current value, which is equivalent to setting the maximum step size for these parameters to zero.

The parameter set is screened according to two selection criteria. Only the most sensitive and/or most independent parameters are subjected to the optimization process. The sensitivity criterion examines the potential of parameter j to reduce the objective function. It is defined as follows:

$$\delta_j = |\Delta S| \qquad (2.7.9.2)$$

Here, ΔS is the change of the objective function if the parameter is perturbed by a small value. Normalizing to the maximum value $\delta_{max} = \max(\delta_j)$ yields the selection criterion ω:

$$\omega_j = \frac{\delta_j}{\delta_{max}} \qquad 0 < \omega_j \leq 1 \qquad (2.7.9.3)$$

Those parameters with ω larger than a predefined value ω_{min}, i.e., the most sensitive parameters, are selected. Parameters that are unable to significantly reduce the objective function are (temporarily) excluded from the optimization process. As an option, the selection criterion ω_{min} can be relaxed with each iteration k:

$$\omega_{\min,k} = \omega_{\min} \cdot \left(1 - \frac{k}{K}\right) \tag{2.7.9.4}$$

Equation (2.7.9.4) reaches zero for the last iteration K, i.e., all parameters are selected for the final step.

The second selection criterion examines the ratio between the apparent conditional standard deviation σ'_p and the marginal standard deviation σ_p as a measure of overall parameter correlation:

$$\Gamma_j = \frac{\sigma'_{p_j}}{\sigma_{p_j}} \qquad 0 < \Gamma_j \leq 1 \tag{2.7.9.5}$$

The calculation of σ_p and σ'_p is described in Section 2.8.4. Those parameters with a ratio larger than $\Gamma_{\min}$, i.e., the most independent parameters, are selected. Strongly correlated parameters are (temporarily) excluded from the optimization process. As an option, the selection criterion $\Gamma_{\min}$ can be relaxed with each iteration k:

$$\Gamma_{\min,k} = \Gamma_{\min} \cdot \left(1 - \frac{k}{K}\right) \tag{2.7.9.6}$$

Equation (2.7.9.6) reaches zero for the last iteration K, i.e., all parameters are selected for the final step. The standard deviations used in Equation (2.7.9.5) cannot be interpreted as actual estimation uncertainties because they are not evaluated at the minimum.

Due to the nonlinearity of the inverse problem at hand, sensitivity coefficients and parameter correlations change during the optimization. Therefore, the selection criteria must be reevaluated from time to time, i.e., parameters may be deactivated and reactivated during the course of an inversion.

Automatic parameter selection makes the inversion faster because fewer parameters must be perturbed for calculating the Jacobian matrix (the full Jacobian is only calculated every few iterations when the selection criteria are reevaluated). The inversion is usually also more stable. Parameters that are not sensitive or highly correlated tend to be changed drastically during an iTOUGH2 iteration, causing unnecessary numerical difficulties.

RELATED iTOUGH2 COMMANDS

Steps of individual parameters are limited using command `>>>> STEP` in the parameter definition block. Global step limitation is invoked using command `>>> STEP` in block `> COMPUTATION`, with the keyword `UNSCALED` indicating that scaling should be omitted. Command `>>> SELECT` implements automatic parameter selection, and its subcommands define the selection criteria.

2.7.10 Example

Figure 2.7.10.1 shows the contours of the objective function—the byproduct of a grid search—and the solution paths for the four minimization algorithms described in Sections 2.7.3 through 2.7.6. The inverse problem solved in the example is described in *Finsterle* [1999c; Problem 1]. The search area was confined to the region shown in the figure. With the exception of the Gauss-Newton algorithm, which is misguided by its linearity assumption, the global minimum is accurately identified by all algorithms. The Levenberg-Marquardt algorithm is the most efficient method for this problem. Notice that the strategy underlying each method is clearly revealed by the solution path taken.

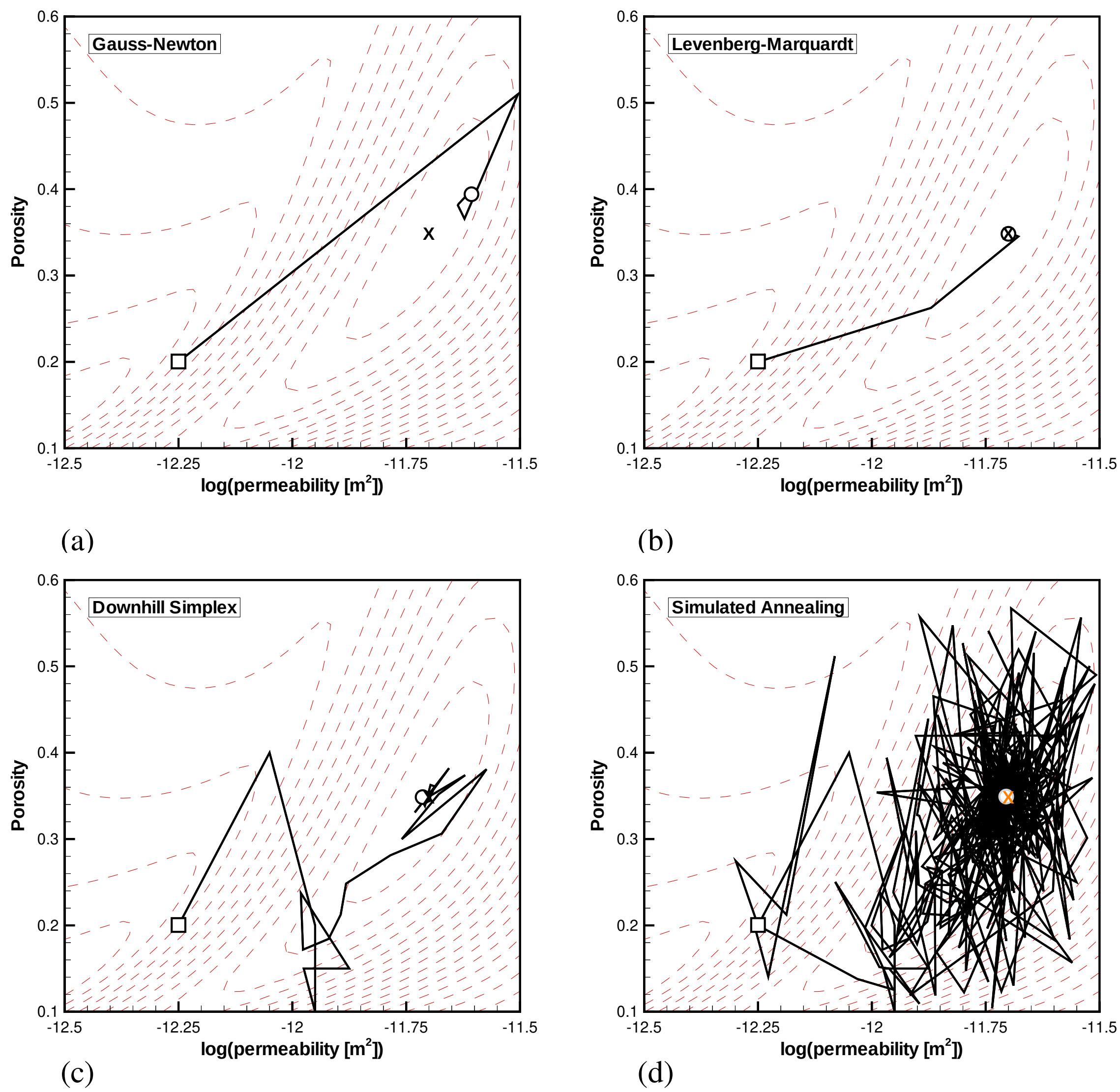


Figure 2.7.10.1. Solution paths of (a) Gauss-Newton, (b) Levenberg-Marquardt, (c) Downhill Simplex, and (d) Simulated Annealing minimization algorithms in the two-dimensional parameter space porosity–log(permeability). The square, circle, and cross indicate, respectively, the starting point, endpoint, and global minimum.

2.8 Sensitivity and Error Analysis

2.8.1 Introduction

One of the key advantages of a formalized approach to parameter estimation is the possibility to perform an *a posteriori* error analysis. The sensitivity matrix evaluated at the minimum of the objective function contains much information regarding the impact of the parameters on the system behavior, and how valuable certain data were for the solution of the inverse problem at hand. The residual analysis provides some measure of the overall goodness-of-fit, and identifies systematic errors, trends in the model, or outliers in the data. 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 high parameter correlation, which is usually an indication of overparameterization. The covariance matrix of the estimated parameters can be further analyzed to obtain correlation coefficients and parameter combinations that lead to similar matches. Model identification criteria provide a measure to compare the performance of alternative models with a different model structure. Finally, the uncertainty of model predictions can be calculated using either linear error propagation analysis or Monte Carlo simulations.

2.8.2 Sensitivity analysis

The sensitivity coefficients Equation (2.7.2.4) show the impact of a small parameter change on the calculated system behavior at the calibration point. They can also be interpreted as a measure of the relative contribution of the corresponding data point to the solution of the inverse problem. As Equation (2.8.4.2) below reveals, the higher the absolute value of the sensitivity coefficient, the lower the estimation uncertainty of the corresponding parameter. High sensitivity is, however, a necessary but not sufficient condition for meaningful parameter estimation (see *Finsterle and Persoff* [1997]).

In order to make sensitivity coefficients comparable with one another, it is suggested to scale them by the *a priori* standard deviation of the observation, σ_z, and the expected parameter variation, σ_p:

$$\tilde{J}_{ij} = J_{ij}\frac{\sigma_{p_j}}{\sigma_{z_i}} = \frac{\partial z_i}{\partial p_j}\cdot\frac{\sigma_{p_j}}{\sigma_{z_i}} \qquad (2.8.2.1)$$

where J_{ij} is an element of the Jacobian matrix, Equation (2.7.2.4). The scaling is necessary because the parameters concurrently estimated by inverse modeling may have different units and vary by orders of magnitude. The same is true for the different observation types used for calibration. Unlike J_{ij}, the scaled sensitivity coefficients $\tilde{J}_{ij}$ are dimensionless.

The scaling of the sensitivity coefficients allows one to directly compare the contribution of each data point to the estimation of each parameter, and to evaluate a number of composite sensitivity measures. Equation (2.8.2.1) indicates that the more accurate a measurement is, the higher its contribution to the solution of the inverse problem. A data point is also considered more valuable if the parameter to be estimated is expected or allowed to be uncertain.

Four sums of absolute, scaled sensitivity coefficients are calculated in iTOUGH2. The first sum consists of the absolute values of the elements within each row of the scaled Jacobian matrix:

$$a_i = \sum_{j=1}^{n} \left| \tilde{J}_{ij} \right| \tag{2.8.2.2}$$

The quantity a_i is a relative measure of how important data point i is for the estimation of all parameters of interest. Comparing the a_i values enables identification of those data points in space and time that are most valuable.

By adding all the coefficients belonging to the same data set (e.g., a time series of pressure measurements at a certain location), the contribution of this data set $k = 1,\ldots,K \le m$ to the estimation of parameter j can be quantified as follows:

$$b_{kj} = \sum_{i=1}^{m} \left| \tilde{J}_{ij} \right| \Big|_{i \in k} \tag{2.8.2.3}$$

The overall contribution of a certain data set to the solution of the inverse problem at hand is given by

$$c_k = \sum_{j=1}^{n} \sum_{i=1}^{m} \left| \tilde{J}_{ij} \right| \Big|_{i \in k} = \sum_{i=1}^{m} a_i \Big|_{i \in k} = \sum_{j=1}^{n} b_{kj} \tag{2.8.2.4}$$

A comparison of c_k values shows whether a certain data set was worth collecting, or whether the position of measurement points should be changed (in space and/or time), or whether the accuracy of the corresponding sensor must be improved to make it a valuable source of information, comparable with the contribution from other observations.

Finally, building the sum of each column provides a relative measure of parameter sensitivity, taking all available observations into account:

$$d_j = \sum_{i=1}^{m} \left| \tilde{J}_{ij} \right| \tag{2.8.2.5}$$

A parameter with a high d_j value is more likely to reach an estimation uncertainty of σ_{p_j} than a parameter with a lower d_j value. Again, this neglects the impact of parameter correlations on the estimation uncertainty, which cannot be assessed by a simple sensitivity analysis, but must be evaluated by actually inverting the data and calculating the estimation covariance matrix, Equation (2.8.4.2). If the sensitivity analysis is performed to assess the impact of a parameter on the model predictions, the parameter with the highest d_j value is the most important one, i.e., one should try to determine it as accurately as possible.

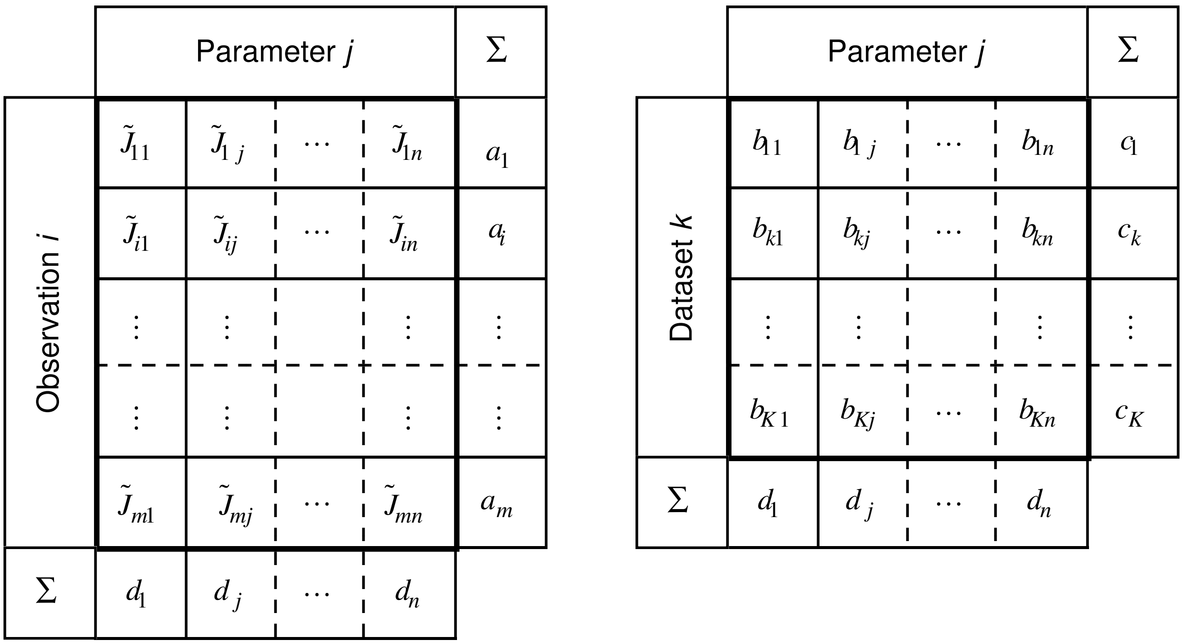
The sensitivity measure d_j must be distinguished from δ_j (Equation 2.7.9.2), which examines the sensitivity of the objective function (not the model output) to a change of the parameter:

$$\delta_j = |\Delta S| \tag{2.8.2.6}$$

Here, ΔS is the change of the objective function if the parameter is perturbed by a small value. The objective function can be best reduced by updating parameters with a large δ_j value.

The sensitivity coefficients, the scaled sensitivity coefficients, and the composite sensitivity measures—Equations (2.8.2.2) through (2.8.2.5)—are useful to design an experiment, to analyze inverse modeling results, and to study the impact of parameters on selected model predictions. The interpretation of these measures changes depending on the purpose of the sensitivity analysis. If evaluated prior to data collection, they help optimize the design of an experiment by identifying the most appropriate observation types and the necessary measurement accuracy. Furthermore, measurement locations with large sensitivities data can be selected as well as time windows that contain valuable data. Since the sensitivity coefficients depend on the (unknown) parameter set itself, the analysis must be repeated with different assumptions about the system properties. An example is described in *Finsterle and Faybishenko* [1999].

The scaled sensitivity matrix and the various composite sensitivity measures—rows and columns labeled with "$\sum$"—are shown in Figure 2.8.2.1.



(a) (b)

Figure 2.8.2.1. (a) Scaled sensitivity matrix and (b) composite sensitivity measures.

RELATED iTOUGH2 COMMANDS

By default, iTOUGH2 prints the dimensionless, scaled sensitivity matrix with elements given by Equation (2.8.2.1) along with the composite sensitivity measures (2.8.2.2) through (2.8.2.6). In order to also print the (unscaled) sensitivity matrix, command `>>> SENSITIVITY (o)` must be used.

2.8.3 Estimated error variance and Fisher Model Test

The *a posteriori* or estimated error variance represents the variance of the mean weighted residual and is thus a measure of goodness-of-fit [*Larsen and Marx*, 1986]:

$$s_0^2 = \frac{\mathbf{r}^T \mathbf{C}_{zz}^{-1} \mathbf{r}}{m-n} \tag{2.8.3.1}$$

If the model does not match the data sufficiently well, i.e., s_0^2 is too large, then the estimated parameters are meaningless, because the underlying model is erroneous. On the other hand, obtaining a good match does not guarantee that the inverse problem is solved in a reasonable way; the results must be subjected to a critical residual analysis. The model identification criteria discussed in Section 2.8.6 therefore not only contain a goodness-of-fit measure such as the estimated error variance, but also additional terms to prevent overparameterization. A more detailed discussion of these points can be found in *Finsterle and Persoff* [1997].

If the residuals are consistent with the distributional assumption about the measurement error, which is expressed through covariance matrix $\mathbf{C}_{zz}$, then the estimated error variance assumes a value close to one. The *a posteriori* error variance s_0^2 can be considered an estimate of the *a priori* error variance σ_0^2 (see Equation 2.5.3.2) with a degree of freedom of $(m-n)$. It can be shown [*Larsen and Marx*, 1986] that the ratio s_0^2/σ_0^2 follows an F-distribution with the two degrees of freedom $f_1 = m-n$ and $f_2 = \infty$. We can therefore statistically test whether the average match deviates significantly from the modeler's expectations.

Table 2.8.3.1 shows the *Fisher Model Test*, in which the significance of a deviation from σ_0^2 is tested. If the estimated error variance is significantly larger than σ_0^2, there is likely to be an error in the functional model, or the assumption about the measurement errors were too optimistic. The Fisher Model Test is only indicative of modeling errors if the stochastic model is well defined. Otherwise, s_0^2 can only be considered a relative measure of goodness-of-fit. The second column in Table 2.8.3.1 is the value to be used for scaling the covariance matrix of the estimated parameters (see Section 2.8.4).

Table 2.8.3.1. Fisher Model Test

Fisher Model Test	Error Variance	Comment
$F_{m-n,\infty,1-\alpha} < s_0^2/\sigma_0^2$	s_0^2	Error in functional and/or stochastic model
$1 \le s_0^2/\sigma_0^2 \le F_{m-n,\infty,1-\alpha}$	s_0^2	Model test passed
$s_0^2/\sigma_0^2 < 1$	σ_0^2	Error in stochastic model

Equation (2.8.3.1) can also be used to iteratively adjust the relative contributions of different data types (pressures, temperatures, saturations, flow rates, prior information, etc.) to the objective function. The relative weight assigned to each observation type is given by the ratio $\lambda_{kl} = \tau_k / \tau_l$, where τ is a scalar analog to σ_0^2 such that $\mathbf{C}_{zz,k} = \tau_k \mathbf{V}_{zz,k}$ (see also Equation 2.5.3.2). Here, $\mathbf{C}_{zz,k}$—a submatrix of $\mathbf{C}_{zz}$—is the covariance matrix of all observations of type k, and $\mathbf{V}_{zz,k}$ is a positive-definite matrix. By default, τ_k is fixed at 1. If relative weights are not well known, λ_{kl} can be updated in an iterative process, where τ_k is recalculated every few iterations according to

$$\tau_k = \frac{\mathbf{r}_k^T \mathbf{C}_{zz,k}^{-1} \mathbf{r}_k}{m_k - n} \tag{2.8.3.2}$$

This procedure is similar to that referred to as *iterated re-weighted least squares* [*Haining*, 1990]. The process assigns weights such that the relative contribution of each observation type to the objective function approaches $1/K$, where K is the number of observation types used in the inversion. The Fisher Model Test becomes meaningless if λ is updated during the inversion because the test will be fulfilled by definition.

RELATED iTOUGH2 COMMANDS

The multiplication factor shown in the second column of Table 2.8.3.1 can either be specified using commands `>>> A PRIORI` and `>>> A POSTERIORI`, or automatically selected by using command `>>> FISHER`. The confidence level is specified using command `>>> ALPHA`. Command `>>> TAU` invokes the iterative reweighting of different observation types according to Equation (2.8.3.2).

2.8.4 Covariance matrix of estimated parameters

The best-estimate parameter set $\hat{\mathbf{p}}$ was determined by matching the model to a specific data set. Because the data have a random component—the measurement errors (see Section 2.5.2)—the actually observed data set can be considered to be one realization of a universe of potentially observed data sets. Consequently, the estimated parameter vector $\hat{\mathbf{p}}$ represents only one point of a probability distribution in the n-dimensional parameter space of all possible parameter vectors determined by matching the hypothetical data sets. The goal is to find an approximation of this probability distribution despite the fact that the true parameter vector $\tilde{\mathbf{p}}$ is unknown, and that only one data set is available for inversion.

The covariance matrix of the estimated parameters, $\mathbf{C}_{pp}$, contains the standard errors or uncertainties of the estimates $\hat{\mathbf{p}}$, as well as the covariances, which describe the statistical correlations between pairs of parameters. $\mathbf{C}_{pp}$ is an approximation of the probability distribution of $\hat{\mathbf{p}} - \tilde{\mathbf{p}}$. Given a best-estimate parameter set $\hat{\mathbf{p}}$ and the corresponding covariance matrix $\mathbf{C}_{pp}$, one can determine the confidence region around $\hat{\mathbf{p}}$, which should contain the true parameter set with a certain level of confidence.

We first present the statistical concept that provides the basis of the interpretation of estimation covariance matrices. This discussion uses a thought experiment with multiple realizations of the data to arrive at the distribution of the parameter estimates, for which arbitrary confidence regions can be derived using simple statistics. In the second part of this section, we look at the more realistic situation, where there is only one data set available for parameter estimation, yielding only one best-estimate parameter set. Consequently, the covariance matrix is in itself an estimate, which needs to be based on a linearity and normality assumption, leading to ellipsoidal confidence regions. Note that the calculation of the confidence region is fundamentally different in the two cases, the first providing a theoretical justification for using the second. The interpretation of the covariance matrix $\mathbf{C}_{pp}$ and its relation to a confidence region will be discussed in detail, followed by some thoughts about the underlying linearity assumption.

Figure 2.8.4.1 illustrates the situation for a case described in *Finsterle and Najita* [1998]. 200 synthetic data sets were generated with different random measurement errors. These data sets were then inverted individually, yielding 200 estimates of porosity ϕ and initial gas saturation S_{gi}. The individual estimates are shown as triangles in the two-dimensional parameter space. The mean of the estimates $\overline{\mathbf{p}}$, shown as a square, is very close to the true parameter set of $\phi = 0.35$ and $S_{gi} = 0.30$. The covariance matrix of the cloud of triangles is visualized as a dash-dotted ellipse with the center point at $\overline{\mathbf{p}}$. In practice, only one data set is available, yielding a single estimate $\hat{\mathbf{p}}$, shown as a circle. The solid ellipse with center at $\hat{\mathbf{p}}$ represents the 95% confidence region around $\hat{\mathbf{p}}$, as inferred from $\mathbf{C}_{pp}$ (see Equation (2.8.4.2) below). It is similar in size and orientation to that representing the probability distribution of $\hat{\mathbf{p}} - \tilde{\mathbf{p}}$, and it contains the true parameter set on the 95% confidence level.

In Figure 2.8.4.1, the full probability distribution—the cloud of triangles—was assumed to be accurately represented by an elliptical confidence region, and an ellipse was drawn around the best estimate $\hat{\mathbf{p}}$. The choice of an elliptical confidence region is based on a normality and linearity assumption, which will be critically reviewed later in this section. A quantitative relationship among the value of the objective function at the minimum, $S_{\min}$, the covariance matrix $\mathbf{C}_{pp}$, and the confidence level can only be accurate if (1) the measurement errors are normally distributed, and (2) the estimation uncertainty is small enough so that the nonlinear model can be replaced by a suitable linearized model within the confidence region.

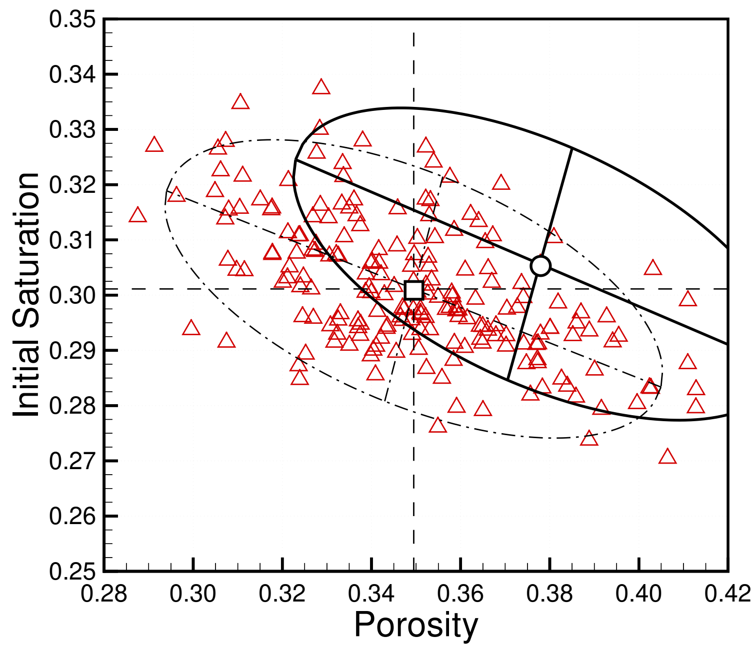


Figure 2.8.4.1. Probability distribution of estimates in a two-dimensional parameter space. Triangles represent solutions from 200 least-squares fits to hypothetical data sets. The square indicates the mean of all solutions. The solid ellipse is an approximation of the estimation uncertainty of a single best-estimate parameter set, shown as a circle.

We now discuss the case of estimating $\mathbf{C}_{pp}$ if only one data set is available for inversion. Under the assumption of normality and linearity, the $100(1-\alpha)\%$ confidence region contains those values $\mathbf{p}$ for which [*Donaldson and Schnabel*, 1987]

$$S(\mathbf{p}) - S(\hat{\mathbf{p}}) \leq s_0^2 \cdot n \cdot F_{n,m-n,1-\alpha} \tag{2.8.4.1}$$

where $\hat{\mathbf{p}}$ is the vector holding the optimum parameter set, s_0^2 is the estimated error variance, Equation (2.8.3.1), and $F_{n,m-n,1-\alpha}$ is a quantile of the F-distribution. Here, α is the probability that the hypothesis stated above is rejected even though it is true. In the general case, this confidence region is of arbitrary shape; it is reasonable, however, to bound it by the points of constant likelihood, i.e., a contour of the objective function for maximum likelihood estimates. Near the minimum, where the linearity assumption holds, the confidence region is ellipsoidal, which makes it inexpensive to construct and easy to report. For a maximum likelihood estimator, the covariance matrix of the estimated parameters is asymptotically given by the inverse of the curvature or Fisher information matrix $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$, multiplied by the estimated error variance s_0^2:

$$\mathbf{C}_{pp} = s_0^2\left(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J}\right)^{-1} \tag{2.8.4.2}$$

Equation (2.8.4.2) can be derived by inserting Equation (2.7.3.1) into the definition of a covariance matrix, $\mathbf{C}_{pp} = \mathrm{E}[(\mathbf{p} - \mathrm{E}[\mathbf{p}])(\mathbf{p} - \mathrm{E}[\mathbf{p}])^T]$, where $\mathrm{E}[]$ denotes the expected value.

The interpretation of the covariance matrix $\mathbf{C}_{pp}$ provides the key criteria for evaluating inverse modeling results. The diagonal elements of $\mathbf{C}_{pp}$ contain the variances of the estimated parameters, σ_p^2. They are a measure of how uncertain the estimate is given the uncertainty of all the other, concurrently estimated parameters. Note that they are directly proportional to the overall goodness-of-fit expressed by s_0^2. The higher the quality of the data and the better the fit, the more accurate the estimates. Furthermore, estimation uncertainty is inversely proportional to the absolute size of the sensitivity coefficients. The more sensitive the calculated system response at the calibration point, the more information contained in the data regarding the parameters of interest. Since the sensitivity coefficients can be evaluated without actually collecting data, the design of an experiment can be optimized *a priori* by looking for observation types as well as measurement points in space and time that yield large sensitivity coefficients.

The off-diagonal elements of $\mathbf{C}_{pp}$ are the covariances c_{ij}. Correlations among the parameters are a result of a conjoint impact of parameter changes on the system behavior. The correlation coefficient is given by

$$r_{ij} = \frac{c_{ij}}{\left(\sigma_{p_i}^2 \cdot \sigma_{p_j}^2\right)^{1/2}} \qquad -1 \le r_{ij} \le 1 \tag{2.8.4.3}$$

Correlation coefficients assume values between -1 and 1; a value of zero indicates no statistical correlation between parameter i and j; a value close to -1 or 1 indicates a strong correlation, i.e., the two parameters cannot be determined independently. For example, if two parameters are negatively correlated, a similar system response is obtained by concurrently increasing one and decreasing the other parameter. In an inversion involving three or more parameters, the correlation coefficients are usually difficult to interpret from a physical point of view because of indirect parameter dependencies. Two parameters may exhibit a statistical correlation even though they are not physically related. The non-zero correlation coefficient is a result of the fact that both parameters are correlated to a third parameter. Such a case is described in *Finsterle* [1999c; Problem 2].

iTOUGH2 also prints a matrix of "direct" correlations between pairs of parameters. Direct correlation coefficients are calculated by taking the $n \times n$ curvature matrix $(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$, copying the intersections of the two rows and columns corresponding to the two parameters of interest into a 2×2 matrix, inverting this matrix, and applying Equation (2.8.4.3). This procedure is repeated for all $n(n-1)/2$ parameter pairs. The resulting direct correlation coefficient matrix is easier to interpret on the basis of a physical understanding of the system, i.e., it indicates to which degree a change in one parameter can be compensated by a change in the other parameter.

Even though certain pairs of parameters may exhibit preferential correlation structures, correlations are not intrinsic features of parameter combinations. They obviously depend on the data available and also on the number of simultaneously estimated parameters, because indirect correlations may overwhelm the direct correlations.

If correlations exist, the uncertainty of one parameter does affect the uncertainty of the other parameter. The diagonal elements of $\mathbf{C}_{pp}$ account for this fact. The standard deviation from the joint probability density function, σ_p, is also termed *marginal* standard deviation as it measures the uncertainty of a parameter without regard to the value of the other parameters. It must be distinguished from the *conditional* standard deviation, σ_p', which measures the uncertainty of a parameter assuming that all the other parameters are either precisely known or uncorrelated. The conditional standard deviation of parameter i is the reciprocal of the ith diagonal element of the scaled curvature matrix $\mathbf{F} = s_0^{-2}(\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J})$. The conditional standard deviation is always smaller than the marginal standard deviation (see Figure 2.8.4.2 below). Note that each joint confidence region can be interpreted as a conditional confidence region of a higher-order parameter set, i.e., the uncertainty estimates are always optimistic because they neglect the influence of all the parameters that are fixed despite being uncertain. The ratio

$$\Gamma_i = \frac{\sigma'_{p_i}}{\sigma_{p_i}} \qquad 0 < \Gamma_i \le 1 \tag{2.8.4.4}$$

can be interpreted as an overall measure of how independently parameter i can be estimated. Small values of Γ usually indicate that the uncertainty σ_p of a parameter could be reduced by lowering its correlation to other parameters. Test design should aim at obtaining high Γ values.

Figure 2.8.4.2 illustrates the elliptical (or hyperellipsoidal if $n > 2$) region that represents the covariance matrix. It can be constructed from the eigenvalues and eigenvectors of $\mathbf{C}_{pp}$. The length of the semiaxis are the square roots of the eigenvalues, and their orientations are given by the corresponding eigenvectors. Parameter combinations along the eigenvector with the largest eigenvalue lead to a similar system response and are thus difficult to identify. Additional measures of the overall size of the ellipsoid are given in Section 2.8.6.

The linearization approach assumes that the nonlinear model can be adequately approximated by linear functions at the solution. Thus, the actual, nonellipsoidal confidence region (see Equation 2.8.4.1) can be approximated by a region consisting of those values $\mathbf{p}$ for which

$$(\mathbf{p}-\hat{\mathbf{p}})^T\mathbf{C}_{pp}^{-1}(\mathbf{p}-\hat{\mathbf{p}}) \le n \cdot F_{n,m-n,1-\alpha} \tag{2.8.4.5}$$

The approximation of the actual confidence region, which is the contour of the objective function on level $S(\hat{\mathbf{p}}) + s_0^2\, n\, F_{n,m-n,1-\alpha}$, by an ellipsoidal confidence region is visualized in Figure 2.8.4.3a. In this case, the linearization leads to a slight overprediction of the size of the confidence region along its longest axis. The probability that all parameters are within the standard ellipsoid (i.e., the joint probability) is much less than 68.3% (the probability for a single parameter); it gets smaller with increasing n and decreasing m.

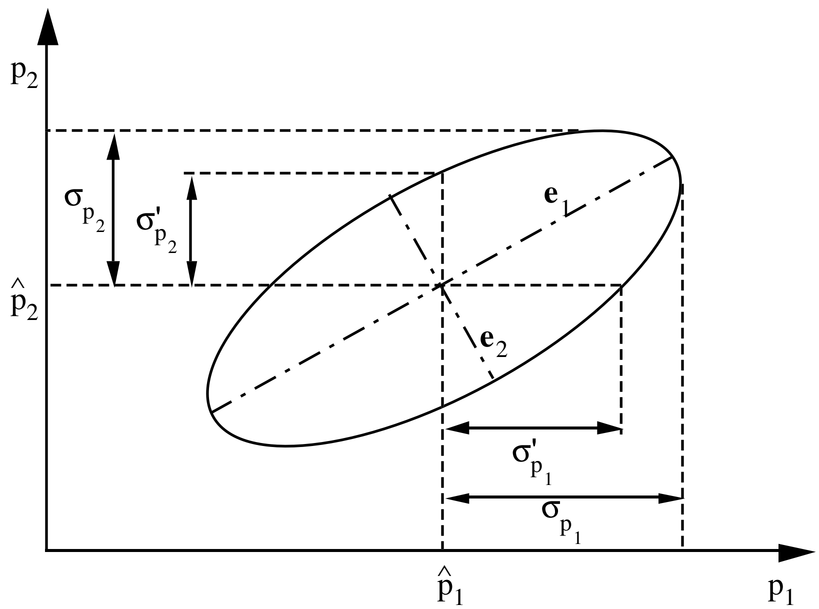


Figure 2.8.4.2. Visualization of estimation covariance matrix as an elliptical confidence region, indicating marginal and conditional standard deviations.

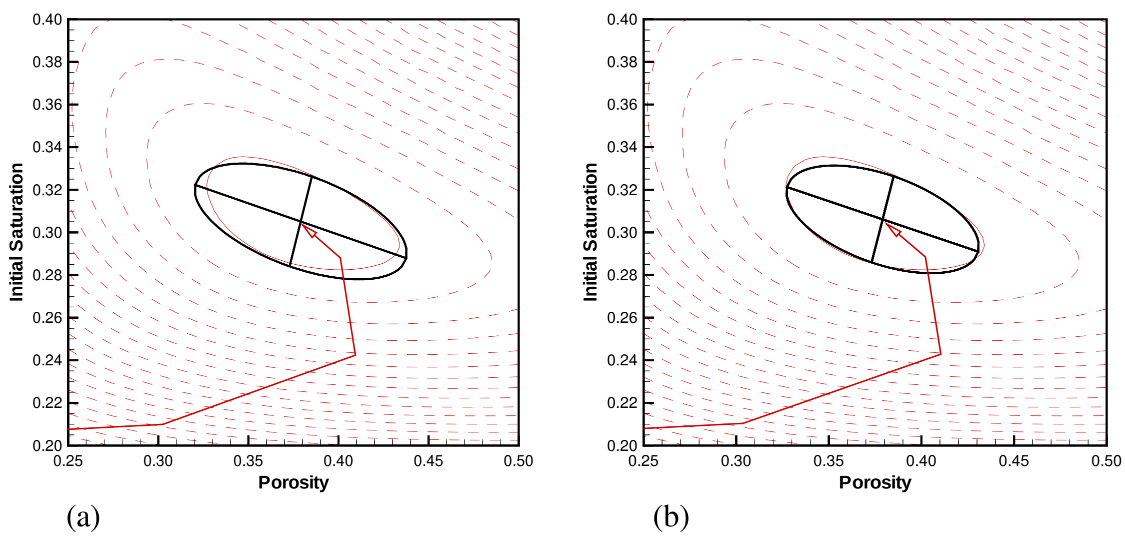


Figure 2.8.4.3. Original (a) and corrected (b) approximation of the confidence region. The ellipses approximate the contours of the objective function (dashed) at the minimum. The solid contour represents the actual confidence region. The arrow indicates the solution path taken by the minimization algorithm.

A $100(1-\alpha)\%$ confidence interval for $\tilde{p}_i$ contains those values p_i for which

$$\left|p_i - \hat{p}_i\right| \leq \sigma_{p_i} \cdot t_{m-n,1-\alpha/2} \tag{2.8.4.6}$$

where $t_{m-n,1-\alpha/2}$ is the quantile of the Student t-distribution. The confidence regions and confidence intervals introduced here are only exact for normally distributed errors and linear models. If the errors are not normally distributed, the covariance matrix has no clear quantitative interpretation. Furthermore, if the model is nonlinear, the coverage of the confidence region by the ellipsoidal approximation may be poor. Reparameterization, such as logarithmic transformation of some of the parameters, is a possibility to reduce nonlinearity effects.

Carrera [1984] proposed a procedure that adjusts the size of the hyperellipsoid to account for nonlinearities, assuming that the orientation is accurately obtained from the linear error analysis. The result of the correction procedure is still a covariance matrix, i.e., it can be interpreted in the usual manner and is easy to report. The method is based on a comparison of the actual objective function with the results from the linear approximation at discrete points in the parameter space. These test points $\breve{\mathbf{p}}$ are located at the end of the main axis of the hyperellipsoid, i.e. [*Finsterle and Pruess*, 1995]:

$$\breve{\mathbf{p}}_{i\pm} = \hat{\mathbf{p}} \pm \left(n \cdot F_{n,m-n,1-\alpha}\right)^{1/2} a_i \mathbf{u}_i \quad i = 1,\cdots,n \tag{2.8.4.7}$$

Here, $\breve{\mathbf{p}}_{i\pm}$ are two test parameter sets on the ith axis, the direction of which is given by the eigenvector $\mathbf{u}_i$ of covariance matrix $\mathbf{C}_{pp}$. The distance from the optimal parameter set $\hat{\mathbf{p}}$ is selected as a multiple of the corresponding eigenvalue a_i^2 and the quantile of the F-distribution. This means that the correction is tailored to approximate the confidence region on a certain confidence level $1-\alpha$. The eigenvalues a_i^2, which determine the length of the semi-axes, are corrected as follows:

$$a'^2_i = a_i^2 s_0^2 \left(\frac{A_+ + A_-}{2}\right)_i \tag{2.8.4.8}$$

with

$$A_{\pm i} = \frac{n \cdot F_{n,m-n,1-\alpha}}{S(\breve{\mathbf{p}}_{i\pm}) - S(\hat{\mathbf{p}})} \tag{2.8.4.9}$$

Finally, the new covariance matrix, $\mathbf{C}'_{pp}$, is calculated from the original eigenvectors $\mathbf{u}_i$ and the updated eigenvalues a'^2_i:

$$\mathbf{C}'_{pp} = \mathbf{U}^T \mathbf{D}' \mathbf{U} \tag{2.8.4.10}$$

where $\mathbf{D}'$ is a diagonal matrix consisting of the corrected eigenvalues, and $\mathbf{U}$ is the modal matrix, formed by the n eigenvectors as its columns. This correction procedure requires $2n$ additional solutions of the direct problem and is thus relatively inexpensive. While the resulting confidence region is ellipsoidal by definition, the differences between $S(\breve{\mathbf{p}}_{i+})$ and $S(\breve{\mathbf{p}}_{i-})$

provide—as a byproduct of the correction procedure—some insight into the asymmetry of the true confidence region. The corrected covariance matrix is rendered in Figure 2.8.4.3b. The endpoints of the semiaxes match the actual confidence region on the 95% confidence level reasonably well .

RELATED iTOUGH2 COMMANDS

The covariance matrix of the estimated parameters, Equation (2.8.4.2), is evaluated by default. Command `>>> LINEARITY` checks the linearity assumption of the error analysis using Equations (2.8.4.7) through (2.8.4.10).

2.8.5 Residual analysis

Minimizing the objective function leads to the best-estimate parameter set for a given functional and stochastic model. However, this does not imply that the real system is properly represented by the model. If the conceptual model fails to reproduce the salient features of the system, the calibrated model may not be able to match the observed data as expected, where the expectation regarding the attainable fit is reflected in the *a priori* covariance matrix $\mathbf{C}_{zz}$. A first and rather crude assessment of the match is the Fisher Model Test described in Section 2.8.3. However, a more detailed analysis is required to reveal potential trends in the residuals, indicating that there is a systematic error in the model or the data. In general, an inspection and detailed analysis of the residuals are used in pointing towards aspects of the model that need to be modified. In addition, large residuals (outliers) may be detected by visual inspection, or by use of a more rigorous approach based on mathematical statistics. Note that if the statistics of the residuals significantly deviate from normal, the estimates are likely to be biased, and the formal error analysis (which establishes quantitative relationships among the objective function, the covariance matrix, and the confidence level) is not valid.

A trend in the residuals is usually immediately identified by visual inspection of a scatter plot of the residuals, which is printed to the iTOUGH2 output file. A moment analysis is performed on the residuals of each data set, each observation type, and all scaled residuals. First, the *mean* and *variance* are calculated:

$$\bar{r} = \frac{1}{M}\sum_{i=1}^{M} r_i \qquad (2.8.5.1)$$

$$VAR = \frac{1}{M-1}\sum_{i=1}^{M}\left(r_i - \bar{r}\right)^2 \qquad (2.8.5.2)$$

Here, r_i is the residual from a certain data set or certain observation type, respectively, and $M \leq m$ is the number of such residuals. The mean of the residuals is expected to be close to zero, and the variance should be consistent with that specified by the prior covariance matrix. A large positive (negative) mean indicates that data are systematically underpredicted (overpredicted) by the model. Notice that the variance (2.8.5.2) is not the variance of the residuals themselves, but the variance of the residuals about the mean residual $\bar{r}$. A large variance either indicates that the data were noisier than expected, or that there is a trend in the residuals. Taking the ratio of the mean (bias) and the standard deviation $SDEV = \sqrt{VAR}$ of the residuals provides a measure of whether the bias is acceptable; the ratio $\bar{r}/SDEV$ should be close to zero.

The third moment or *skewness* characterizes the degree of asymmetry of the distribution:

$$SKEW = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{r_i - \bar{r}}{SDEV}\right)^3 \qquad (2.8.5.3)$$

A positive (negative) skewness signifies an asymmetric tail extending to more positive (negative) residuals.

The fourth moment or *kurtosis* measures the peakedness or flatness of the distribution relative to the Gaussian distribution:

$$KURT = \left[\frac{1}{M} \sum_{i=1}^{M} \left(\frac{r_i - \bar{r}}{SDEV} \right)^4 \right] - 3 \tag{2.8.5.4}$$

where the -3 term makes the value zero for a normal distribution. A distribution with positive (negative) kurtosis is relatively peaked (flat).

The mean and especially the higher moments are statistics that are not robust in the sense discussed in Section 2.6.5. A more robust estimator of the center of the distribution is the *median*, which is defined as the quantity for which larger and smaller values are equally probable.

$$\hat{r} = \begin{cases} r_{(M+1)/2} & \text{for} \quad M \text{ odd} \\ 0.5 \cdot r_{M/2} + r_{M/2+1} & \phantom{\text{for}} \quad M \text{ even} \end{cases} \tag{2.8.5.5}$$

An estimator of the width of the distribution around the median is given by the *mean absolute deviation*:

$$ADEV = \frac{1}{M} \sum_{i=1}^{M} \left| r_i - \hat{r} \right| \tag{2.8.5.6}$$

A large difference between the mean and the median or the standard deviation and the mean absolute deviation is indicative of a robustness problem, i.e., the distribution is likely to be heavy-tailed and asymmetric, or the residuals contain outliers. The mean used in Equation (2.8.5.2) minimizes the variance, whereas the median used in Equation (2.8.5.6) minimizes the mean absolute deviation.

Note that the moment analysis is performed on the residuals themselves, i.e., not the residuals weighted by the measurement error. This fact may affect the conclusions if largely different standard deviations are assigned to data belonging to the same data set or same observation type. iTOUGH2 also performs a moment analysis on all weighted residuals. This analysis should yield a mean close to zero and a variance close to s_0^2 (see Equation 2.8.3.1).

A plot of the calculated versus the observed system response should show points distributed closely around the diagonal line. In iTOUGH2, a linear regression analysis is conducted on the scatter plots, individually for each data set. An intercept of zero and a slope of one are expected. Note that the linear regression analysis does not properly account for differences in measurement quality within a data set. Furthermore, the smaller values have a higher influence on the intercept estimate, and small and large values determine the slope more strongly than intermediate values. Consequently, the results of this analysis should be interpreted with care. An example of the linear regression analysis is discussed in *Finsterle* [1999c; Problem 6].

In order to further analyze the residuals, it is necessary to estimate the uncertainty of the calculated system response. As will be discussed in Section 2.8.7, the covariance matrix of the model prediction is given by

$$\mathbf{C}_{\hat{z}\hat{z}} = \mathbf{J}\mathbf{C}_{pp}\mathbf{J}^T \tag{2.8.5.7}$$

The square-root of the diagonal element of $\mathbf{C}_{\hat{z}\hat{z}}$ is the standard deviation of the model prediction. Note that the standard deviation of the calculated system response is always smaller than that of the corresponding measurement (i.e., $\sigma_{\hat{z}_i} < \sigma_{z_i}$). This is because the model prediction at a given point is inferred not only from the corresponding data point, but also from all the other observations.

The covariance matrix of the residuals is given by [*Weisberg*, 1980]

$$\mathbf{C}_{rr} = \mathbf{C}_{zz} - \mathbf{C}_{\hat{z}\hat{z}} \tag{2.8.5.8}$$

The elements of $\mathbf{C}_{rr}$ depend on the number and location (i.e., correlation) of the observation points and their sensitivities to the model parameters; they do not depend on the actually measured value.

Next, we calculate a measure termed *local reliability* or *partial redundancy* [*IGP*, 1990]:

$$y_i = \frac{\sigma_{r_i}^2}{\sigma_{z_i}^2} = \frac{\sigma_{z_i}^2 - \sigma_{\hat{z}_i}^2}{\sigma_{z_i}^2} = 1 - \left(\frac{\sigma_{\hat{z}_i}}{\sigma_{z_i}}\right)^2 \tag{2.8.5.9}$$

The local reliability realizes values between zero and one. It is a measure of how much a data point is controlled by redundant observations. If y_i is close to zero, even a large error in the corresponding data point z_i* cannot be detected. A y_i value close to one indicates a well-controlled observation. Adding more observation points in the vicinity of this measurement may improve the accuracy, but does not improve the reliability of the inverse modeling system. In other words, y_i can also be considered a measure of the degree of redundancy.

Observations with y_i values smaller than about 0.25 are considered poorly controlled; values greater than 0.75 indicate a high degree of redundancy. For a given configuration, a relatively uncertain measurement is better controlled than an accurate measurement. Note that

y_i can be evaluated *a priori* and can therefore be used to improve the design of an experiment.

The *normalized* or *Studentized residual* [*Weisberg*, 1980]

$$w_i = \frac{r_i}{\sigma_{r_i}} \tag{2.8.5.10}$$

is a normally distributed random variable with zero mean and a variance of one. Hence, the size of a residual can be statistically tested to see whether it is acceptable or a potential outlier. If $|w_i| > u_{1-\alpha}$, where $u_{1-\alpha}$ is the quantile of the standard normal distribution on the $1-\alpha$ confidence level, then the corresponding residual is likely to be an outlier and should be discarded; the risk of discarding a correct data point is α. The assumption that the normalized residuals follow a normal distribution is an approximation considered acceptable. This test is based on the assumption that multiple large errors do not cancel each another. Note that the w_i-test checks each observation individually, whereas the Fisher Model Test described in Section 2.8.3 is based on the ensemble of all measurements.

The probable size of the error depends on the local reliability and is given by

$$g_i = -\frac{r_i}{y_i} \tag{2.8.5.11}$$

For a poorly controlled observation, the size of the actual error can be significantly larger than the residual.

The following two equations can be used to check the numerical accuracy of the residual analysis [*Weisberg*, 1980]:

$$\sum_{i=1}^{m}\left(\frac{\sigma_{\hat{z}_i}}{\sigma_{z_i}}\right)^2 = n \tag{2.8.5.12}$$

$$\sum_{i=1}^{m} y_i = m - n \tag{2.8.5.13}$$

Finally, the relative contribution of each data point, each data set, and each observation type to the objective function may be used to identify errors in portions of the data or the model. Since the objective function is built using the weighted residuals, an imbalance in these contributions may also signify an error in the stochastic model.

2.8.6 Optimality and model identification criteria

The previous sections dealt with statistical measures that assess the results of a single inversion. In this section, a number of additional criteria are given, which allow comparison of different inversions—for example, calibrating against different data sets, estimating different parameters, or using a different conceptual model. If competing models have been developed and matched to the data, a criterion is needed to decide which of the alternatives is preferable. A number of tests for model discrimination have been developed as described by *Steinberg and Hunter* [1984], *Carrera and Neuman* [1986a], and *Russo* [1988], *Russo et al.* [1991].

One of the most widely used criteria is the estimated error variance as a measure of goodness-of-fit (see Section 2.8.3). The model that best matches the data is considered to be the best. However, since the match can always be improved by adding more fitting parameters, the goodness-of-fit is an inappropriate basis for model selection because it almost always leads to overparameterization. Overparameterization means that an improvement of the fit comes at the expense of a reduction in model reliability. Increasing the number of parameters also increases the correlations among the parameters, which results in higher estimation uncertainties if the match is not significantly improved. Consequently, model identification and optimality criteria should include some aggregate measure of overall estimation uncertainty to guard against overparameterization.

The first group of criteria includes quantities that measure the overall size of the estimation covariance matrix $\mathbf{C}_{pp}$. A key objective of parameter estimation by inverse modeling is to reduce the uncertainty of a set of parameters considered important for the subsequent model prediction. Therefore, a measure of overall parameter uncertainty can serve as a criterion to compare the performance of competing test designs or alternative inversions. The inversion realizing the smallest value is considered superior. There are three scalar measures of $\mathbf{C}_{pp}$ one might use as *design evaluation* or *optimality* criteria [*Steinberg and Hunter*, 1984]:

$$A - optimality = \mathrm{trace}\left(\mathbf{C}_{pp}\right) \tag{2.8.6.1}$$

$$E - optimality = \text{ max eigenvalue}\left(\mathbf{C}_{pp}\right) \tag{2.8.6.2}$$

$$D - optimality = \det\left(\mathbf{C}_{pp}\right) \tag{2.8.6.3}$$

A-optimality consists of minimizing the trace of $\mathbf{C}_{pp}$, i.e., it minimizes the sum of all parameter uncertainties. *E-optimality* seeks minimization of the maximum eigenvalue of $\mathbf{C}_{pp}$. As discussed in Section 2.8.4, the maximum eigenvalue represents the largest axis of the hyper-ellipsoid, i.e., the length of the vector in the parameter space that is associated with the largest estimation uncertainty. Taking the determinant of $\mathbf{C}_{pp}$ yields the so-called *D-optimality* objective for design evaluation. The design with the smallest value minimizes the area of the joint confidence region around the parameter estimates.

If vector **p** contains parameters of different types and orders of magnitude, matrix $\mathbf{C}_{pp}$ should be appropriately scaled before evaluating the optimality criteria. The elements of the scaled covariance matrix are defined as follows:

$$c'_{ij} = \frac{c_{ij}}{p_i \cdot p_j} \tag{2.8.6.4}$$

Recall that $\mathbf{C}_{pp}$ is directly proportional to the overall goodness-of-fit expressed by s_0^2 or—in the case of design calculations—the expectation thereof.

Carrera and Neuman [1986a] have introduced the Akaike Information Criterion (AIC) and the Kashyap criterion [*Kashyap*, 1982]. For normally distributed residuals, AIC can be written as:

$$AIC = (m-n)s_0^2 + \ln\left|\mathbf{C}_{zz}\right| + m \cdot \ln(2\pi) + 2n \tag{2.8.6.5}$$

where $|\bullet|$ indicates the determinant of the corresponding matrix. The AIC takes into account both goodness-of-fit and parsimony of the model. The first term is the objective function, the second and third terms measure the uncertainty of the data, and the last term penalizes overparameterization. The Kashyap criterion is given by [*Carrera*, 1984]:

$$d_k{}^* = (m-n)s_0^2 + \ln\left|\mathbf{C}_{zz}\right| + m \cdot \ln(2\pi) + n \cdot \ln\left(\frac{m}{2\pi}\right) + \ln\left|\mathbf{J}^T\mathbf{C}_{zz}^{-1}\mathbf{J}\right| \tag{2.8.6.6}$$

with $d_k{}^*$ proportional to the negative logarithm of the posterior probability that model k is correct, given the available data [*Kashyap*, 1982].

All criteria discussed in this section are based on a linearity and normality assumption. If this assumption is violated, the criteria should be applied with care. As a safeguard against misinterpretation, it is suggested to consider a model superior to its competitors only if that model realizes significantly lower values for most or all criteria.

2.8.7 Uncertainty propagation analysis

Model predictions are inherently uncertain and may significantly deviate from the true system behavior. There are many reasons for the inconsistency between model predictions and the actual or observed behavior. The main sources for modeling errors include:

Inconsistencies and Errors in the Conceptual Model: As repeatedly mentioned in the previous sections, the conceptual model is by far the most important element in numerical modeling. Considerable effort should be spent on carefully developing the conceptual model because errors in the model structure are difficult to identify and to correct, and they usually have the greatest impact on the model predictions.

Uncertainty in the Input Parameters: Another source of prediction errors is insufficient knowledge about the model parameters. Errors or uncertainties in the input parameters lead to errors or uncertainties in the model predictions. The purpose of inverse modeling is to estimate the best parameter set for a given model structure, and to reduce parameter uncertainty. There is a need to quantify the uncertainty in the model predictions as a result of parameter uncertainty, which will be discussed in this section.

Discretization Errors: The numerical solution of the governing equations has only finite precision and may suffer from discretization errors such as numerical dispersion or oscillations. While care must be taken when choosing the numerical scheme to solve a specific problem, errors from the numerical implementation of the model are usually smaller than those made by using wrong parameter values, which in turn are small compared with the errors from using an inappropriate conceptual model.

When assessing modeling errors, one should consider the fact that modeling involves an abstraction process, i.e., no exact solution is sought, but an approximation that is reasonable and capable of reproducing the salient features of the system to be studied.

This section presents two methods to assess prediction uncertainties as a result of parameter uncertainty:

- Linear uncertainty propagation analysis;
- Monte Carlo simulations.

Linear or *First-Order-Second-Moment (FOSM)* uncertainty propagation analysis quantifies the uncertainty in model predictions by linearization. As the name indicates, FOSM is the analysis of the mean and covariance of a random function (the model prediction) based on its first-order Taylor series expansion. The covariance of the input parameters is translated into the covariance of the system response. This presumes that the mean and covariance are sufficient to characterize the distribution of the dependent variables, i.e., the model results are assumed to be normally or log-normally distributed. This assumption is valid whenever parameter uncertainties are sufficiently small, or when the model is linear and the distribution of the input parameters is normal. The validity of the normality and linearity assumption must be tested before applying FOSM.

We first develop expressions for the mean and covariance matrix of the model prediction. Let $\hat{\mathbf{p}}$ be the vector of length n holding the best-estimate values of the input parameters that are considered uncertain. The uncertainty is described by the covariance matrix $\mathbf{C}_{pp}$. Furthermore, $\mathbf{z}$ is a vector of length m containing the simulation results at certain points in space and time. These model predictions are a nonlinear function of the parameter vector $\mathbf{p}$. Finally, let $\mathbf{J}$ be the $m \times n$ Jacobian matrix holding sensitivity coefficients $J_{ij} = \partial z_i(\hat{\mathbf{p}}) / \partial p_j$. The model prediction $\mathbf{z}(\mathbf{p})$ can be approximated using first-order Taylor series expansion about $\hat{\mathbf{p}}$ as follows:

$$\mathbf{z}(\mathbf{p}) \stackrel{1}{=} \mathbf{z}(\hat{\mathbf{p}}) + \mathbf{J}(\mathbf{p} - \hat{\mathbf{p}}) \qquad (2.8.7.1)$$

The mean is given by:

$$\mathrm{E}[\mathbf{z}(\mathbf{p})] \stackrel{1}{=} \mathrm{E}[\mathbf{z}(\hat{\mathbf{p}})] + \mathrm{E}[\mathbf{J}(\mathbf{p} - \hat{\mathbf{p}})]$$

$$\stackrel{1}{=} \underbrace{\mathrm{E}[\mathbf{z}(\hat{\mathbf{p}})]}_{\mathbf{z}(\hat{\mathbf{p}})} + \mathrm{E}[\mathbf{J}] \cdot \underbrace{\mathrm{E}[(\mathbf{p} - \hat{\mathbf{p}})]}_{\mathbf{0}}$$

$$\mathrm{E}[\mathbf{z}(\mathbf{p})] \stackrel{1}{=} \mathbf{z}(\hat{\mathbf{p}}) \qquad (2.8.7.2)$$

The first-order approximation of the expected values of the dependent variables is the vector of the model prediction obtained using the mean parameters. The mean parameters are approximated by the estimates $\hat{\mathbf{p}}$.

The covariance matrix of the simulated system response is derived using the following definition, with $\hat{\mathbf{z}} = \mathbf{z}(\hat{\mathbf{p}})$:

$$\mathrm{Cov}[\mathbf{z}(\mathbf{p})] \stackrel{1}{=} \mathrm{E}[\{\mathbf{z} - \hat{\mathbf{z}}\}\{\mathbf{z} - \hat{\mathbf{z}}\}^T] \qquad (2.8.7.3)$$

$$\mathrm{Cov}[\mathbf{z}(\mathbf{p})] \stackrel{1}{=} \mathrm{E}[\{\underbrace{\mathbf{z}(\mathbf{p})}_{\mathbf{z}(\hat{\mathbf{p}}) + \mathbf{J}(\mathbf{p} - \hat{\mathbf{p}})} - \underbrace{\mathrm{E}[\mathbf{z}(\mathbf{p})]}_{\mathbf{z}(\hat{\mathbf{p}})}\} \cdot \{\underbrace{\mathbf{z}(\mathbf{p})}_{\mathbf{z}(\hat{\mathbf{p}}) + \mathbf{J}(\mathbf{p} - \hat{\mathbf{p}})} - \underbrace{\mathrm{E}[\mathbf{z}(\mathbf{p})]}_{\mathbf{z}(\hat{\mathbf{p}})}\}^T]$$

$$\stackrel{1}{=} \mathrm{E}[\{\mathbf{J}(\mathbf{p} - \hat{\mathbf{p}})\}\{\mathbf{J}(\mathbf{p} - \hat{\mathbf{p}})\}^T]$$

$$\stackrel{1}{=} \mathbf{J}\{\underbrace{\mathrm{E}[\{\mathbf{p} - \hat{\mathbf{p}}\}\{\mathbf{p} - \hat{\mathbf{p}}\}^T]}_{\mathrm{Cov}(\hat{\mathbf{p}})}\mathbf{J}^{\mathbf{T}}$$

$$\mathrm{Cov}[\mathbf{z}(\mathbf{p})] \stackrel{1}{=} \mathbf{J}\mathbf{C}_{pp}\mathbf{J}^T \equiv \mathbf{C}_{\hat{z}\hat{z}} \qquad (2.8.7.4)$$

Equation (2.8.7.4) was also used in Section 2.8.5. Here, it describes the prediction uncertainty for any set of uncertain input parameters, whereas in Section 2.8.5 it was used to evaluate the uncertainty of the calculated system response during an inversion, as a result of the uncertainty of the best-estimate parameter set.

Linear uncertainty analysis has the following advantages and disadvantages:

Advantages:

- The uncertainties in the model predictions can be described in a compact way by means of the covariance matrix $\mathbf{C}_{\hat{z}\hat{z}}$, i.e., results are easy to understand and convenient to report;
- Correlations among the parameters are taken into account;
- The output covariance matrix $\mathbf{C}_{\hat{z}\hat{z}}$ contains correlations among model predictions;
- FOSM is computationally inexpensive, requiring $n+1$ forward simulations.

Disadvantages:

- The uncertainty in the input parameters are expected to be accurately described by a covariance matrix $\mathbf{C}_{pp}$;
- If parameters are highly uncertain, the linearity assumption may be violated;
- FOSM assigns probabilities to physically unreasonable system responses, i.e., FOSM should not be used to analyze extreme events in the tail of the distribution.

An alternative to FOSM uncertainty propagation analysis is to perform *Monte Carlo* simulations. Monte Carlo (MC) requires repetitive solution of the simulation model, with the parameters randomly sampled from their suspected probability distributions. The output from MC runs is then used to analyze the statistical properties of the resulting distribution, which represents the uncertainty of the model predictions. The procedure is summarized in Table 2.8.7.1.

Table 2.8.7.1. Monte Carlo Simulations

Step 1:	Define probability distribution for all uncertain input parameters.
Step 2:	Randomly sample parameter values from the defined distributions.
Step 3:	Combine sampled parameter values randomly to obtain a parameter vector.
Step 4:	Run simulation and store the results.
Step 5:	Repeat Steps (2) through (4) n_{MC} times.
Step 6:	Perform statistical analyses (histogram, moments) of ensemble of model output.

To consider correlations among the parameters, the random combination of parameter values in Step 3 must be modified such the covariance function is correctly repoduced (see, for example, *Kitterød and Gottschalk* [1997]).

How many Monte Carlo runs should be performed? The number of Monte Carlo simulations n_{MC} can be considered sufficient if:

(1) The selected probability density function of the input parameters is reasonably well approximated by the histogram of the randomly generated parameter values.

(2) The histogram of the model predictions allows for a statistical analysis. That means that a sufficient number of realizations (simulation results) should fall within each interval used to calculate probabilities. For example: The probability that the model prediction z_i falls within the interval $[a,b]$ is approximated by:

$$\Gamma_{[a,b]} = \Pr(a \le z_i \le b) \approx \frac{\text{number of realizations in interval } [a,b]}{\text{total number of Monte Carlo simulations}} = \frac{n_{[a,b]}}{n_{MC}}$$

Therefore, the minimum number of Monte Carlo simulations, $n_{MC,\min}$, should be large enough so that $\Gamma_{[a,b]}$ remains constant, i.e., independent of n_{MC}. This condition is fulfilled for relatively small values of n_{MC} in the case of intervals around the mean, where $n_{[a,b]}$ is usually large due to the high probability density. However, if one is interested in the tail of the distribution, then the number of Monte Carlo simulations required is much higher.

(3) The minimum number of Monte Carlo simulations must be increased if the number of uncertain parameter increases because more parameter combinations are possible.

(4) From experience, the number of Monte Carlo simulations can be as low as 50 and as high as 2000 or greater.

Uncertainty propagation analysis by means of Monte Carlo simulations has the following advantages and disadvantages:

Advantages:

- Any distribution (uniform, normal, log-normal, exponential, any arbitrary histogram) can be chosen to describe parameter uncertainty;
- No assumption is made about the distributional form of the model output, i.e., a full distribution of the predictions is obtained; Monte Carlo is termed a *full distribution analysis*;
- Nonlinearities are inherently taken into account;
- Results from Monte Carlo simulations are always in the physically feasible range.

Disadvantages:

- Results from Monte Carlo simulations are difficult to report because they usually don't follow a normal distribution;
- Since the combination of parameter values in Step 3 (see Table 2.8.7.1) is random, no correlations between parameters are included;
- Monte Carlo simulations are computationally expensive.

EXAMPLE

A short example illustrates the differences between the linear FOSM uncertainty propagation analysis and Monte Carlo simulations. As indicated above, for small standard deviations of the input parameters, and for model output that can be well 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 that is easy to report. If the model is highly nonlinear, and the uncertainties of the input parameters are large, Monte Carlo simulations have to be performed to examine many parameter combinations according to their probabilities. Monte Carlo simulations provide the full distribution of the model output at the selected points in space and time. The Monte Carlo method is very flexible in handling non-Gaussian distributions of both input parameters and output variables, but it is computationally expensive and the results are difficult to report.

Both approaches are compared using a synthetic laboratory experiment consisting of three parts: (1) injection of water into a partially saturated sand column under constant pressure for 300 seconds; (2) injection of gas for 150 seconds, followed by (3) a 150-second shut-in recovery period. The experiment is described in *Finsterle* [1999c; Problem 1].

The standard deviations of three uncorrelated input parameters—the logarithm of the absolute permeability, porosity, and the initial gas saturation in the soil column—are assumed to be 0.1, 0.05, and 0.05, respectively. As a result of parameter uncertainty, the prediction of the pressure at the center of the column will also be uncertain.

The results from both the FOSM and Monte Carlo uncertainty analyses are visualized in Figure 2.8.7.1. 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 nonlinear effects prevail. Note that FOSM analysis assigns a certain probability to pressure responses that are below 1 bar, which is physically not possible in this experiment. The Monte Carlo simulations naturally stay away from this lower bound. The highest pressures were achieved with a parameter combination of low permeability, high porosity, and low initial gas saturation.

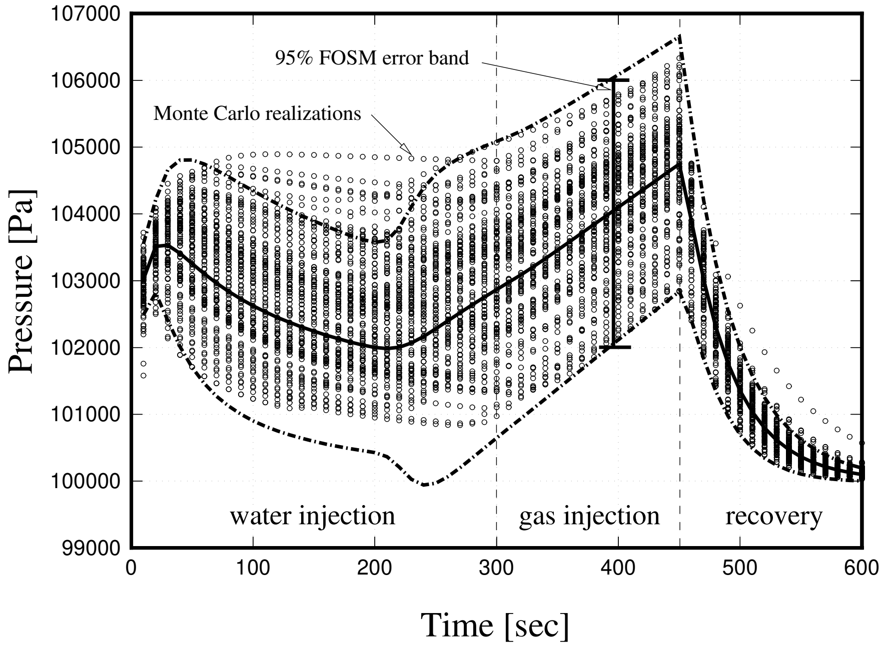


Figure 2.8.7.1. Comparison between FOSM and Monte Carlo uncertainty propagation analyses.

RELATED iTOUGH2 COMMANDS

Linear error propagation analysis is invoked by command `>>> FOSM`, where the standard deviations of the input parameters are provided either through the appropriate fourth-level commands in block `> PARAMETER`, or—if correlations among the parameters are to be specified—as a full covariance/correlation matrix (see keyword `MATRIX`). The Jacobian matrix can be calculated using either `>>> FORWARD` or `>>> CENTERED` finite differences.

Monte Carlo simulations are invoked using command `>>> MONTE CARLO` and its keywords. By default, the input parameters are sampled from a normal distribution in the range indicated by command `>>> RANGE`, with the initial parameter guess as the mean, and with the standard deviation provided through the appropriate fourth-level commands in block `> PARAMETER`. Adding command `>>>> LOGARITHM` makes iTOUGH2 sample from a log-normal distribution, and command `>>>> UNIFORM` chooses a uniform distribution for the corresponding input parameter. The total number of Monte Carlo simulations, n_{MC}, is given through command `>>> SIMULATIONS`. A special plot file is created for convenient plotting of Monte Carlo runs; the corresponding file name contains the label "_mc".

3. iTOUGH2 OUTPUT

3.1 Introduction

The inverse modeling capabilities outlined in Section 2, which include sensitivity analysis, parameter estimation, and uncertainty propagation analysis, are implemented into iTOUGH2. iTOUGH2 requires the user to supply a standard TOUGH2 input file, which describes the forward problem, and an iTOUGH2 input file, which identifies the parameters to be estimated, the observations against which to calibrate the model, as well as various program options. The report "iTOUGH2 Command Reference" [*Finsterle*, 1999b] as well as the web site `http://www-esd.lbl.gov/iTOUGH2` contain descriptions of all iTOUGH2 commands. Full iTOUGH2 applications are discussed in *Finsterle* [1999c].

An iTOUGH2 run produces a standard TOUGH2 output file and a number of iTOUGH2 output files. In this section, the main iTOUGH2 output file is described, mainly by making reference to equations discussed in Section 2. The contents of an iTOUGH2 output file change depending on the selected options. The one shown in this section is a typical output file produced by running an inversion for parameter estimation. The related inverse problem is described in *Finsterle* [1999c; Problem 3]. Line numbers have been added to the output file to facilitate easy referencing in the text. While an iTOUGH2 output file should be viewed and printed with a width of 132 columns, it was reformatted here to fit the width of the page. Omissions are indicated with "*(...)*". Information not to be found in the output file but added for clarification (e.g., column numbers, comments) is printed in *italics*. Also note that the contents of an iTOUGH2 output file are frequently modified as the code is improved.

3.2 Header

Figure 3.2.1 shows the iTOUGH2 header information. It identifies the version of the code with version number, date, and computer architecture for which the program was compiled (Line 11). Line 15 indicates the time when the run was started, followed by the iTOUGH2 and TOUGH2 input file names and the working directory. The temporary directory is usually deleted at the end of a run, but may be saved using command option `-no_delete` (see Section 6.2). Note that additional input files may have been provided (e.g., TOUGH2 input files MESH, INCON, and GENER, or iTOUGH2 data files). The names of these files must be provided on the Unix command line (see Section 6.2), which is reproduced later in the output (see Figure 3.3.4, Line 136). The number of the equation-of-state (EOS) module is given along with a list of valid component and phase names. These are the names to be used in conjunction with commands and keywords `COMPONENT` and `PHASE`, respectively. Lines 27 and 31 indicate that portions of the iTOUGH2 input file were commented out using "`/*`" and "`*/`". Line 29 indicates that Line 717 of the iTOUGH2 input file was ignored because of a "#" sign in the first column.

```
               @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
               @                                             @
               @   @  @@@@@   @@   @  @   @@@  @  @   @@@@   @
               @        @    @  @  @  @  @     @  @  @    @  @
               @   @    @    @  @  @  @  @ @@  @@@@     @@   @
               @   @    @    @  @  @  @  @  @  @  @    @@    @
               @   @    @     @@    @@    @@@  @  @  @@@@@@  @
               @                                             @
               @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

       iTOUGH2 V4.0 (JANUARY 9, 1999) FOR SUN WORKSTATION, S. FINSTERLE
              COPYRIGHT 1999, REGENTS, UNIVERSITY OF CALIFORNIA

================================================================================
                 >>>>>>>>>>>>> 3-Nov-98 10:54 <<<<<<<<<<<<<
================================================================================
iTOUGH2 INPUT FILE            :  sam3p1i
TOUGH2 INPUT FILE             :  sam3
WORKING DIRECTORY             :  /m/presto/u/finster/itough2/samples/sample3
TEMPORARY DIRECTORY           :  /m/presto/u/finster/it2_24135
EQUATION OF STATE MODULE NO.  :  1 - Pure and traced water in liquid, (...)
COMPONENTS                    :  WATER     TRACER    HEAT
PHASES                        :  VAPOR     LIQUID
TOUGH2 TITLE                  :  sam3: Five spot injection/production in (...)
================================================================================

--- Lines  700 to  703 skipped.

--- Line   717 ignored.

--- Lines  722 to  731 skipped.
```

Figure 3.2.1. iTOUGH2 header information.

3.3 Printout of iTOUGH2 Input

This section of the output file reiterates the information specified in the iTOUGH2 input file. The output follows the main structure of the iTOUGH2 input file, with three sections describing (1) the parameters to be estimated, (2) the observations used for calibration, and (3) the program options. In addition, some information is given regarding the computer system used.

Figure 3.3.1 shows the list of parameters to be estimated. The first column indicates the order of the parameters as given in the iTOUGH2 input file. The second column contains an internal parameter identifier. The annotation given in the third column is either specified by the user (see command `>>>> ANNOTATION`) or internally generated. The parameter type is shown in the fourth column, followed by a flag indicating whether the parameter value, its logarithm, or a multiplication factor is estimated. Most parameters refer to a rock type. If more than one rock type is given, the material name is followed by a "+" sign and the number of additional rock types specified. The actual list of rock types is provided below (see Lines 55–60). The prior parameter value is given in the seventh column. A "`@`" sign indicates that the initial guess is different from the prior information value. The initial guess is not reiterated here, but can be seen later as the first parameter set starting minimization (see Figure 3.4.1, Line 160). The eighth column contains the standard deviations that weigh prior information, i.e., the square roots of the first n diagonal elements of covariance matrix $\mathbf{C}_{zz}$, Equation (2.5.3.1). In this example, prior information is not weighted. Lower and upper bounds are indicated in Columns 9 and 10. The step size taken at each iteration during the minimization process can be limited for each parameter individually (see Section 2.7.9). No such maximum step size is given for parameters No. 2 and 5. This does not mean, however, that their step sizes are not restricted by the global step-size criterion, Equation (2.7.9.1), if such a criterion were specified. Finally, certain parameters are further categorized by one or more indices, as indicated in the last column. For example, the parameter representing the logarithm of the absolute permeability refers to one or multiple flow directions, indicated by one or multiple numbers between 1 and 3 in Column 12.

```
================================================================================
                                      INPUT
================================================================================
PARAMETERS
==========
1  2     3              4        5    6        7      8      9       10      11 12
--------------------------------------------------------------------------------
# ID ANNOTATION     PARAMETER   VLF ROCKS    PRIOR  SDEV L-BOUND U-BOUND M-STEP P
--------------------------------------------------------------------------------
1  2 PERMEABILITY   ABS. PERM.   L  FRACT  -.130E+2  N/W -.18E+2 -.12E+2 .10E+1 3
2  6 POROSITY FRACT POROSITY     V  FRACT   .500E+0@ N/W  .50E-1  .90E+0  UNLIM 1
3 16 SPECIFIC HEAT  SPEC. HEAT   V  FRACT+1 .800E+3  N/W  .10E+3  .90E+4 .10E+3 1
4 15 HEAT COND.     HEAT COND.   V  FRACT+1 .250E+1  N/W  .10E+1  .50E+1 .50E+0 1
5  9 FRACT. SPACING MINC PAR.    V  -----   .200E+2  N/W  .10E+2  .50E+3  UNLIM 2
6  7 RESERVOIR TEMP INIT. COND.  V  DEFAU   .250E+3  N/W  .20E+3  .40E+3 .40E+2 1
--------------------------------------------------------------------------------
@ indicates that initial guess is different from prior information
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
 #  DEFINITION OF MULTIPLE MATERIALS/SOURCES
--------------------------------------------------------------------------------
 3  FRACT+1   =   FRACT + MATRX
 4  FRACT+1   =   FRACT + MATRX
--------------------------------------------------------------------------------
```

Figure 3.3.1. Parameters selected for estimation; N/W means taht prior information is not weighted.

Figure 3.3.2 shows the output that provides information about the calibration points at which the observed and calculated system response is compared. First comes a list of calibration times as specified in block `>> TIMES`. These points are usually not identical with the actual times at which data were collected; these are the times at which model output is provided for comparison against measured or interpolated data.

```
 62
 63 OBSERVATIONS
 64 ============
 65
 66 TIMES  [a]
 67 -----------
 68  0.1000000E+00  0.2000000E+00  0.3000000E+00  0.4000000E+00  0.5000000E+00
 69  0.6000000E+00  0.7000000E+00  0.8000000E+00  0.9000000E+00  0.1000000E+01
 70  0.1100000E+01  0.1200000E+01  0.1300000E+01  0.1400000E+01  0.1500000E+01
 71  0.1600000E+01  0.1700000E+01  0.1800000E+01  0.1900000E+01  0.2000000E+01
 72  0.2100000E+01  0.2200000E+01  0.2300000E+01  0.2400000E+01  0.2500000E+01
 73  0.2600000E+01  0.2700000E+01  0.2800000E+01  0.2900000E+01  0.3000000E+01
 74  0.3100000E+01  0.3200000E+01  0.3300000E+01  0.3400000E+01  0.3500000E+01
 75  0.3600000E+01  0.3700000E+01  0.3800000E+01  0.3900000E+01  0.4000000E+01
 76  0.4100000E+01  0.4200000E+01  0.4300000E+01  0.4400000E+01  0.4500000E+01
 77  0.4600000E+01  0.4700000E+01  0.4800000E+01  0.4900000E+01  0.5000000E+01
 78
 79   1       2       3         4       5      6      7    8      9        10    11
 80 --------------------------------------------------------------------------------
 81 SET ANNOTATION  TYPE   ELEM/CONN   SDEV  MIN/MAX TIME LVMS DATAPOINTS FACTOR IOBS
 82 --------------------------------------------------------------------------------
 83  1  P. INJECT.  PRES.  AA 1      .20E+6 .10E+0 .60E+1  V   DATA: 60  .10E+6  1
 84  2  P. PRODUCT. PRES.  KA 1      .20E+6 .10E+0 .60E+1  V   DATA: 60  .10E+6  1
 85  3  P. OBS. 1   PRES.  DB 1      .20E+6 .10E+0 .60E+1  V   DATA: 60  .10E+6  1
 86  4  P. OBS. 2   PRES.  GC 1      .20E+6 .10E+0 .60E+1  V   DATA: 60  .10E+6  1
 87  5  T. PRODUCT. TEMP.  KA 1      .50E+1 .10E+0 .60E+1  V   DATA: 60  .10E+1
 88  6  T. OBS. 1   TEMP.  DB 1      .50E+1 .10E+0 .60E+1  V   DATA: 60  .10E+1
 89  7  T. OBS. 2   TEMP.  GC 1      .50E+1 .10E+0 .60E+1  V   DATA: 60  .10E+1
 90  8  WATER PROD. Q-LIQ. JA 1 KA 1 .20E+0 .10E+0 .60E+1  V   DATA: 60 -.10E+1
 91  9  VAPOR PROD. Q-VAP. JA 1 KA 1 .10E-1 .10E+0 .60E+1  V   DATA: 60 -.10E+1
 92 --------------------------------------------------------------------------------
 93
 94
 95 ------------------------------------------------
 96 Number of datasets                     :   9
 97 Number of calibration times            :  50
 98 Number of parameters                   :   6
 99 Number of parameters with prior info. :        0
100 Number of PRESSURE                     :      200
101 Number of FLOW RATE                    :      100
102 Number of TEMPERATURE                  :      150
103                                               ----
104 Total number of observations           :      450
105                                               ====
106 Degree of freedom                      : 444
107 ------------------------------------------------
108
```

Figure 3.3.2. Observations available for calibration.

Lines 83–91 list the data sets, with their user-specified annotations (Column 2), and their data type. The fourth column contains one or multiple gridblock names, identifying the element or connection representing the location to which the data refer. The standard deviation shown in the fifth column is the square root of a constant variance that is assigned to all diagonal elements of matrix $\mathbf{C}_{zz}$, referring to the corresponding data set. Note that constant measurement errors are specified using commands `>>>> VARIANCE`, `>>>> DEVIATION`, `>>>> WEIGHT` and `>>>> AUTO`. If a measurement error is specified as a fraction of the observed value, this will also be indicated in Column 5. Any diagonal element of $\mathbf{C}_{zz}$ can also be provided individually either through the data file (see command `>>>> COLUMNS`) or using command `>> COVARIANCE`. These latter two cases are, however, not reflected in Column 5. Columns 6 and 7 contain the minimum and maximum time, respectively, at which data are available. These time limits are given either by the data themselves, or when a time window is specified. They are affected by the time units specified for each data set, and by command `>>>> SHIFT TIME`. Column 8 indicates whether the calculated value itself or its logarithm will be used for comparison to the data (see command `>>>> LOGARITHM (o)`). Furthermore, if multiple elements or connections are provided, either the sum (see command `>>>> SUM`) or the mean (see command `>>>> AVERAGE`) of the variables calculated at these locations will be used for calibration. Column 9 shows the data definition. In most cases, the data are given as a time series, i.e., a list of times versus observed values. The number of data points read is indicated in Column 9. It is affected by command `>>>> PICK`. Column 10 shows the factor specified to convert the observed data to the standard units used in TOUGH2. The last column is used for further specification of an observation, identifying, for example, whether the observed pressure refers to the gas or capillary pressure.

Lines 96–104 summarize the calibration points available. The total number of data sets given on Line 96 is usually identical with the number of points in space, i.e., each data set provides a time series of observations at a certain location. As mentioned above, the total number of calibration times solely refers to the number of times specified in the iTOUGH2 input file, and may be different from the number of times at which data were collected. The total number of parameters is reported here since they may serve as additional data points (prior information). Because no standard deviations are provided for the parameters, they are not included as prior information. Here, the model will be calibrated simultaneously against pressure, flow rate, and temperature measurements. The 150 temperature measurements, for example, consist of data at 3 locations, each calibrated at 50 points in time. The degree of freedom (Line 106) is the difference between the total number of observations and the number of parameters to be estimated.

Figure 3.3.3 shows a summary of the computational parameters and the selected program options. Line 136 contains the arguments submitted to the Unix script file named on Line 135. It would indicate all the additional file names provided for a specific run; no such files were specified in this case. The information about the computer system (see Figure 3.3.4) varies depending on the system calls made from subroutines in file *mdep$COMP.f*, where *$COMP* is the name of a computer platform.

```

COMPUTATIONAL PARAMETERS
========================

Application                                      :      Levenberg-Marquardt
Maximum number of iTOUGH2 iterations             :                       10
Maximum number of TOUGH2 simulations             :                     9999
Maximum number of uphill steps                   :                       10
Maximum size of scaled parameter step            :              0.10000E+01
Initial Levenberg parameter                      :              0.10000E-01
Marquardt parameter                              :              0.10000E+02
Finite difference quotient for Jacobian          :   6  forward -> centered
Increment factor for computing derivatives       :              0.10000E-01
Variance for error analysis                      :        Fisher Model Test
Format of plotfile                               :                  Tecplot
Objective Function                               :            Least-Squares
Automatically select parameter if:
- relative sensitivity is greater than           :                 -0.05000
- independence measure is greater than           :                  0.00000
Revisit selection criteria every  3 iterations

```

Figure 3.3.3. Summary of computational parameters and selected program options.

```

COMPUTER SYSTEM
===============

Machine type                                     : SUN Workstation
UNIX script file name                            : /m/presto/u/finster/bin/itough2
UNIX command line arguments                      : sam3p1i sam3 1
Host name                                        : presto.lbl.gov
User name                                        : not available
Executable                                       : (...)/itough2/itough2_1.(...)
Computer is as fast as a SUN ULTRA 1


--- End of iTOUGH2 input job:  736 lines read,    0.66 CPU-seconds used

```

Figure 3.3.4. Information about the computer system used.

3.4 Printout From Minimization Algorithm

The next section of the output file contains printout from the minimization algorithm. The messages depend on the minimization method chosen, the solution path taken, and the constraints imposed. Figure 3.4.1 shows an example from the Levenberg-Marquardt minimization algorithm, with the parameters automatically being chosen using the sensitivity criterion (see Equations (2.7.9.2) through (2.7.9.4), and Figure 3.3.3, Lines 125–128).

The value of the objective function obtained with the initial parameter set is shown on Lines 160–161. The 394th element of the residual vector yielded the maximum squared weighted residual. This element is related to the injection pressure after 4.4 years, as can be seen from the list of residuals discussed below (see Figure 3.7.1). The parameters shown on Lines 160–161 are identical to the prior information values previously reported (see Figure 3.3.1, Lines 44–49). An exception is the second parameter, porosity, which was marked on Line 45 (see Figure 3.3.1) as a parameter with an initial guess different from its prior information value.

Next, the gradient of the objective function, Equation (2.7.2.3), is calculated using forward finite differences, Equation (2.7.2.10a). Automatic parameter selection was invoked (Lines 164–174). According to the sensitivity criterion (2.7.9.3), the initial reservoir temperature is the only sensitive parameter worth updating; all the other parameters are temporarily deactivated. The Levenberg-Marquardt algorithm proposes to increase the initial reservoir temperature by 62 ˚C. However, the user has limited the step to a maximum of 40 ˚C, as previously noted (see Figure 3.3.1, Line 49). As a consequence of automatic parameter selection and step size limitation, only the sixth element of the parameter vector is updated by 40, as indicated on Line 179. The new parameter set (Lines 180–181) leads to a rather significant reduction of the objective function from the initial value shown on Line 160 to the value given on Line 180. This concludes the first iteration. Similar output is generated for each additional iteration (not shown).

Lines 256–273 show a sequence of unsuccessful attempts to reduce the objective function. After each unsuccessful step, the Levenberg parameter is increased by a factor of 10, which is the Marquardt parameter given on Line 119 of Figure 3.3.3. Increasing the Levenberg parameter effectively reduces the length of the step and changes its orientation (see Table 2.7.4.1). After four unsuccessful steps, the objective function was finally reduced, completing the fifth iteration.

Since the parameter selection criterion is relaxed with each iteration according to Equation (2.7.9.4), all parameters are updated for the last iteration as shown on Lines 370–380. Centered finite differences (Line 368) were used to more accurately calculate the Jacobian matrix for the last few iterations and to provide a better basis for the subsequent error analysis (see also Figure 3.3.3, Line 120). A final step in the parameter space is performed, leading to the best-estimate parameter set. This inversion was terminated by reaching the maximum number of iterations (Line 388), as specified by the user (see Figure 3.3.3, Line 114).

```

================================================================================
                                     OUTPUT
================================================================================


LEVENBERG-MARQUARDT ALGORITHM

I  = NEW ITERATION       J  = JACOBIAN          S  = STEP    U  = UNSUCC. STEP
PS = PARAMETER SELECTION  PU = PARAMETER UPDATE   B  = BOUNDS  M  = MESSAGE

--------------------------------------------------------------------------------
ITER TOUGH2 OBJ FUNC. MAX. RES. EQU. PERMEABILITY  POROSITY FRACT   SPECIFIC HEAT
                                       HEAT COND.  FRACT. SPACING RESERVOIR TEMP.
--------------------------------------------------------------------------------
I  0    1 0.14300E+06 0.104E+4  394 -0.130000E+02    0.250000E+00    0.800000E+03
                                     0.250000E+01    0.200000E+02    0.250000E+03
J  1 Gradient     =   0.92100E+04 (forward)
-------------------------------------------------------------------
                  Automatic Parameter Selection
-------------------------------------------------------------------
    Parameter      Rel. Sensitivity     Independence        Status
    Critical Value          0.0500          0.0000
-------------------------------------------------------------------
  1 PERMEABILITY            0.0212 -        0.7439      deactivated
  2 POROSITY FRACT          0.0000 -        0.9340      deactivated
  3 SPECIFIC HEAT           0.0048 -        0.4608      deactivated
  4 HEAT COND.              0.0001 -        0.0848      deactivated
  5 FRACT. SPACING          0.0002 -        0.0785      deactivated
  6 RESERVOIR TEMP.         1.0000 +        0.9653           active
-------------------------------------------------------------------
MS   Param. No. 6: RESER. TEMP. Step = 0.616E+2 exceeds max. step size = 0.400E+2
S    Step size = 0.40E+2 Scaled step size = 0.16E+0 Levenberg parameter = 0.10E-1
PU   Parameter update:               0.000000E+00    0.000000E+00    0.000000E+00
                                     0.000000E+00    0.000000E+00    0.400000E+02
I  1    9 0.38033E+05 0.764E+3   15 -0.130000E+02    0.250000E+00    0.800000E+03
                                     0.250000E+01    0.200000E+02    0.290580E+03
... (...)
MS   Param. No. 4: HEAT COND. Step = -0.119E+1 exceeds max. step size = -0.500E+0
S    Step size = 0.64E+2 Scaled step size = 0.23E+0 Levenberg parameter = 0.10E-5
U       1. unsuccessful step!            F(k+1)/F(k) =  0.101937E+01
MS   Param. No. 4: HEAT COND. Step = -0.119E+1 exceeds max. step size = -0.500E+0
S    Step size = 0.64E+2 Scaled step size = 0.23E+0 Levenberg parameter = 0.10E-4
U       2. unsuccessful step!            F(k+1)/F(k) =  0.101923E+01
MS   Param. No. 4: HEAT COND. Step = -0.118E+1 exceeds max. step size = -0.500E+0
S    Step size = 0.63E+2 Scaled step size = 0.23E+0 Levenberg parameter = 0.10E-3
U       3. unsuccessful step!            F(k+1)/F(k) =  0.101762E+01
MS   Param. No. 4: HEAT COND. Step = -0.113E+1 exceeds max. step size = -0.500E+0
S    Step size = 0.61E+2 Scaled step size = 0.23E+0 Levenberg parameter = 0.10E-2
U       4. unsuccessful step!            F(k+1)/F(k) =  0.100364E+01
MS   Param. No. 4: HEAT COND. Step = -0.784E+0 exceeds max. step size = -0.500E+0
S    Step size = 0.50E+2 Scaled step size = 0.21E+0 Levenberg parameter = 0.10E-1
PU   Parameter update:               0.804699E-02    0.133708E-01    0.503350E+02
                                    -0.500000E+00   -0.418822E-01   -0.347546E+00
I  5   31 0.12689E+04 0.105E+3   60 -0.142207E+02    0.263371E+00    0.101722E+04
                                     0.240071E+01    0.396282E+02    0.299815E+03
... (...)
```

Figure 3.4.1. Output from Levenberg-Marquardt minimization algorithm.

```
J 10 Gradient     =   0.33632E+03 (centered)
-------------------------------------------------------------------
                   Automatic Parameter Selection
-------------------------------------------------------------------
    Parameter       Rel. Sensitivity     Independence         Status
    Critical Value          0.0000            0.0000
-------------------------------------------------------------------
  1 PERMEABILITY            0.1437 +          0.9189           active
  2 POROSITY FRACT          0.0067 +          0.6146        activated
  3 SPECIFIC HEAT           0.2968 +          0.1120           active
  4 HEAT COND.              0.1393 +          0.0816           active
  5 FRACT. SPACING          0.2299 +          0.1406           active
  6 RESERVOIR TEMP.         1.0000 +          0.8275           active
-------------------------------------------------------------------
S     Step size = 0.88E+2 Scaled step size = 0.76E+0 Levenberg parameter = 0.10E-2
PU    Parameter update:              0.345031E-03    0.183997E+00   -0.880571E+02
                                     0.968430E-01   -0.332190E+01    0.294084E-01
>I    Best fit parameter set:       -0.142203E+02    0.447367E+00    0.948594E+03
                                     0.280089E+01    0.547089E+02    0.300218E+03

C   Maximum number of iterations reached. MITER = 10  --> Terminate!

```

Figure 3.4.1. (cont.) Output from Levenberg-Marquardt minimization algorithm.

3.5 Printout From Sensitivity Analysis

The next section of the output file contains the scaled sensitivity matrix and the aggregate sensitivity measures discussed in Section 2.8.2. Figure 3.5.1 shows an excerpt from the scaled sensitivity matrix, with elements given by Equation (2.8.2.1). Each column refers to a parameter, and each row represents an observation, with prior information occupying the first n rows. Since prior information is not weighted, the corresponding sensitivity coefficients are zero. The elements of the Jacobian matrix (Equation 2.7.2.4) are obtained by multiplying the scaled sensitivity coefficient with the standard deviation of the corresponding observation (see Figure 3.3.2, Lines 83–91, Column 5), and by dividing it by the expected parameter variation (see Figure 3.5.3, Column 3). Alternatively, command `>>> SENSITIVITY` can be used to print the unscaled sensitivity matrix. The last column in Figure 3.5.1 contains the sum of the absolute values of the scaled sensitivity coefficients over all columns of a row, i.e., it is the aggregate sensitivity measure a_i (Equation 2.8.2.2), indicating the potential contribution of each calibration point to the inverse problem.

```

Sensitivity Analysis
--------------------

Element Sij of the scaled sensitivity matrix is the partial derivative of the
calculated system response zi with respect to parameter pj,
scaled by inverses of the respective standard deviations:

      dz  * sigma(p )
        i         j
S   = ---------------
 ij   dp  * sigma(z )
        j         i


    obervation     time            1             2    3 - 6            7
                              PERMEABILITY      POROSITY    (...)           Total
  1 PERMEABILITY           0.00000E+00   0.00000E+00    (...)    0.00000E+00
  2 POROSITY FRACT         0.00000E+00   0.00000E+00    (...)    0.00000E+00
  3 SPECIFIC HEAT          0.00000E+00   0.00000E+00    (...)    0.00000E+00
  4 HEAT COND.             0.00000E+00   0.00000E+00    (...)    0.00000E+00
  5 FRACT. SPACING         0.00000E+00   0.00000E+00    (...)    0.00000E+00
  6 RESERVOIR TEMP.        0.00000E+00   0.00000E+00    (...)    0.00000E+00
  7 P. INJECT       0.1   -0.86315E+01  -0.71697E-01    (...)    0.14688E+02
  8 P. INJECT       0.2   -0.88301E+01  -0.50854E-01    (...)    0.14818E+02
  9 P. INJECT       0.3   -0.89879E+01  -0.39032E-01    (...)    0.14870E+02
 10 P. INJECT       0.4   -0.88552E+01  -0.26367E-01    (...)    0.14947E+02
 11 P. INJECT       0.5   -0.89262E+01  -0.29528E-01    (...)    0.15129E+02
 12 P. INJECT       0.6   -0.90259E+01  -0.28070E-01    (...)    0.15303E+02
 13 P. INJECT       0.7   -0.91199E+01  -0.25602E-01    (...)    0.15401E+02
 14 P. INJECT       0.8   -0.91994E+01  -0.24324E-01    (...)    0.15478E+02
 15 P. INJECT       0.9   -0.92716E+01  -0.23412E-01    (...)    0.15548E+02
(...)
 57 P. PRODUC       0.1    0.20721E+02  -0.10753E+00    (...)    0.26841E+02
 58 P. PRODUC       0.2    0.20751E+02  -0.52436E-02    (...)    0.26758E+02
 59 P. PRODUC       0.3    0.21173E+02   0.22315E-01    (...)    0.27240E+02
 60 P. PRODUC       0.4    0.26601E+02   0.20814E-01    (...)    0.38486E+02
 61 P. PRODUC       0.5    0.21822E+02  -0.28815E+00    (...)    0.34494E+02
 62 P. PRODUC       0.6    0.18871E+02   0.24173E-01    (...)    0.25370E+02
 63 P. PRODUC       0.7    0.19345E+02   0.11348E-01    (...)    0.25633E+02
 64 P. PRODUC       0.8    0.19473E+02  -0.21395E-02    (...)    0.25777E+02
 65 P. PRODUC       0.9    0.19510E+02  -0.51889E-02    (...)    0.25834E+02
(...)
446 VAPOR PRO       4.0    0.22503E+02   0.47209E-01    (...)    0.27378E+02
447 VAPOR PRO       4.1    0.22478E+02   0.50432E-01    (...)    0.27728E+02
448 VAPOR PRO       4.2    0.22430E+02   0.55152E-01    (...)    0.28109E+02
449 VAPOR PRO       4.3    0.22372E+02   0.60786E-01    (...)    0.28517E+02
450 VAPOR PRO       4.4    0.22313E+02   0.66910E-01    (...)    0.28949E+02
451 VAPOR PRO       4.5    0.22258E+02   0.73230E-01    (...)    0.29405E+02
452 VAPOR PRO       4.6    0.22210E+02   0.79779E-01    (...)    0.29884E+02
453 VAPOR PRO       4.7    0.22171E+02   0.86491E-01    (...)    0.30385E+02
454 VAPOR PRO       4.8    0.22143E+02   0.93112E-01    (...)    0.30908E+02
455 VAPOR PRO       4.9    0.22127E+02   0.99762E-01    (...)    0.31451E+02
456 VAPOR PRO       5.0    0.22124E+02   0.10646E+00    (...)    0.27346E+02

```

Figure 3.5.1. Scaled sensitivity matrix.

```
1321
1322 Contributions of data to sets parameter sensitivity
1323 ---------------------------------------------------
1324
1325                          1           2           3           4           5
1326                    P. INJECT.  P. PRODUC.   P. OBS. 1   P. OBS. 2  T. PRODUC.
1327 1 PERMEABILITY     0.48563E+3  0.11085E+4  0.12314E+3  0.54652E+2  0.50155E+3
1328 2 POROSITY FRACT   0.94417E+0  0.17972E+1  0.66888E+0  0.50871E+0  0.81130E+0
1329 3 SPECIFIC HEAT    0.96928E+0  0.98518E+1  0.70927E+0  0.69971E+0  0.45125E+1
1330 4 HEAT COND.       0.54330E+0  0.83641E+1  0.52197E+0  0.51636E+0  0.38202E+1
1331 5 FRACT. SPACING   0.32530E+1  0.41695E+2  0.28610E+1  0.28355E+1  0.19021E+2
1332 6 RESERVOIR TEMP.  0.29654E+3  0.30371E+3  0.29800E+3  0.29815E+3  0.13708E+3
1333
1334                          6           7           8           9
1335                     T. OBS. 1   T. OBS. 2  WATER PROD  VAPOR PROD
1336 1 PERMEABILITY     0.69771E-1  0.22367E-1  0.60649E+2  0.12120E+4
1337 2 POROSITY FRACT   0.22391E+0  0.18904E-1  0.35593E+0  0.82275E+1
1338 3 SPECIFIC HEAT    0.13286E+1  0.12533E-2  0.16705E+1  0.33329E+2
1339 4 HEAT COND.       0.10026E+1  0.11835E-2  0.15066E+1  0.30073E+2
1340 5 FRACT. SPACING   0.50529E+1  0.64648E-2  0.75294E+1  0.14998E+3
1341 6 RESERVOIR TEMP   0.98569E+2  0.99992E+2  0.14440E+2  0.28651E+3
1342
1343 Sum of Sensitivity Coefficients
1344 -------------------------------
1345       1                   2              3              4              5
1346 ----------------------------------------------------------------------------
1347 PARAMETER/OBSERVATION     TOTAL       VARIATION   SENS. OUTPUT  SENS. OBJ. F.
1348 ----------------------------------------------------------------------------
1349 PERMEABILITY   :    0.70926E+04     0.50000E+00    0.35463E+04    0.40542E+01
1350 POROSITY FRACT :    0.67783E+02     0.20000E+00    0.13557E+02    0.18943E+00
1351 SPECIFIC HEAT  :    0.10614E+01     0.50000E+02    0.53072E+02    0.83725E+01
1352 HEAT COND.     :    0.23175E+03     0.20000E+00    0.46350E+02    0.39291E+01
1353 FRACT. SPACING :    0.23224E+02     0.10000E+02    0.23224E+03    0.64866E+01
1354 RESERVOIR TEMP.:    0.18330E+03     0.10000E+02    0.18330E+04    0.28209E+02
1355 ----------------------------------------------------------------------------
1356 P. INJECTION   :                                   0.78790E+03
1357 P. PRODUCTION  :                                   0.14740E+04
1358 P. OBS. 1      :                                   0.42591E+03
1359 P. OBS. 2      :                                   0.35737E+03
1360 T. PRODUCTION  :                                   0.66680E+03
1361 T. OBS. 1      :                                   0.10625E+03
1362 T. OBS. 2      :                                   0.10004E+03
1363 WATER PROD.    :                                   0.86152E+02
1364 VAPOR PROD.    :                                   0.17201E+04
1365 ----------------------------------------------------------------------------
1366
```

Figure 3.5.2. Summary of contributions of data sets to parameter sensitivity.

Figure 3.5.2 contains the table that summarizes the relative contribution of each data set to each of the parameters, given by Equation (2.8.2.3). For example, vapor flow rates and pressures measured at the production well contain the most information for the determination of absolute permeability, whereas temperature measurements in the two remote observation wells do not make a contribution to the estimation of this specific parameter.

The second table in Figure 3.5.2 contains the remaining aggregate sensitivity measures. The overall parameter sensitivity d_j, Equation (2.8.2.5), is given in Column 4, Lines 1349–1354. It is the product of the sum of the absolute sensitivity coefficients scaled by the measurement errors (Column 2) and the expected parameter variation σ_{p_j}, which is reproduced in Column 3. The expected parameter variation σ_{p_j} is specified either using command `>>>> DEVIATION (p)`, which also weighs prior information, or command `>>>> VARIATION`. If neither of these commands is given, the value given in Column 3 is set to 10% of the initial parameter guess. Note that the only purpose of command `>>>> VARIATION` is to provide σ_{p_j} for the scaling of the sensitivity matrix (Figure 3.5.1), and its aggregate measures (Figure 3.5.2); σ_{p_j} does not affect the outcome of the inversion, unless it is also used to weigh prior information. Column 5 shows the sensitivity of the objective function with respect to each parameter, as given by Equation (2.8.2.6). The lower part of Figure 3.5.2., Lines 1356–1364, lists measure c_k, Equation (2.8.2.4). It is the overall sensitivity of each data set. It shows, for example, that the vapor flow data are crucial for the solution of the inverse problem, whereas the amount of water produced doesn't seem to be sufficiently sensitive to the parameters of interest. If this sensitivity analysis were performed for test design, one could also conclude that the accuracy of the water production measurements would have to be improved by an order of magnitude to be able to make a significant contribution to the inverse problem at hand.

If fewer than 6 parameters are estimated, the information shown in Figure 3.5.2 is directly appended to the output of the scaled sensitivity matrix in the format schematically shown in Figure 2.8.2.1b.

3.6 Printout From Error Analysis

This section of the iTOUGH2 output deals with the covariance matrix of the estimated parameter set discussed in Section 2.8.4. Depending on the outcome of the Fisher Model Test (see Table 2.8.3.1) or the user's choice, the inverse of the curvature matrix is multiplied with either the *a priori* error variance $\sigma_0^2 = 1$ or the *a posteriori* error variance s_0^2, Equation (2.8.3.1). The factor used for the subsequent error analysis is first reported (see Figure 3.6.1, Line 1372). Lines 1377–1383 constitute the covariance matrix $\mathbf{C}_{pp}$, Equation (2.8.4.2). The diagonal elements are the variances σ_p^2; the covariances are reproduced as the lower triangular matrix, whereas the upper triangular matrix shows the corresponding correlation coefficients, Equation (2.8.4.3). These correlation coefficients contain contributions from indirect dependencies, which are difficult to interpret physically. The matrix of direct correlations, revealing the dependence of pairs of parameters, is shown below, Lines 1388–1394. The calculation of the matrix of direct correlations is described in Section 2.8.4.

```
1367
1368 ===========================================================================
1369                                 ERROR ANALYSIS
1370 ===========================================================================
1371
1372 Error analysis is based on >>> a posteriori <<< variance:  0.1009964E+01
1373
1374
1375 Covariance(L+D)/Correlation(U) Matrix of Estimated Parameters
1376 ------------------------------------------------------------
1377                      PERM.  POROSITY SPEC HEAT HEAT COND   SPACING     TEMP.
1378 PERMEABILITY     0.413E-5    -0.207     0.202    -0.272    -0.266    -0.175
1379 POROSITY FRACT  -0.596E-4  0.200E-1    -0.362     0.576     0.657     0.221
1380 SPECIFIC HEAT    0.201E-1 -0.251E+1  0.241E+4    -0.855    -0.275    -0.060
1381 HEAT COND.      -0.177E-3  0.261E-1 -0.135E+2  0.103E+0     0.724     0.125
1382 FRACT. SPACING  -0.989E-3  0.170E+0 -0.247E+2  0.426E+0  0.336E+1     0.087
1383 RESERVOIR TEMP. -0.398E-4  0.350E-2 -0.328E+0  0.449E-2  0.179E-1  0.125E-1
1384
1385
1386 Matrix of Direct Correlations
1387 -----------------------------
1388                      PERM.  POROSITY SPEC HEAT HEAT COND   SPACING     TEMP.
1389 PERMEABILITY        1.000     0.133     0.240     0.225    -0.255    -0.245
1390 POROSITY FRACT      0.133     1.000    -0.529    -0.507     0.598     0.418
1391 SPECIFIC HEAT       0.240    -0.529     1.000    -0.993     0.974     0.508
1392 HEAT COND.          0.225    -0.507    -0.993     1.000     0.982     0.508
1393 FRACT. SPACING     -0.255     0.598     0.974     0.982     1.000    -0.515
1394 RESERVOIR TEMP.    -0.245     0.418     0.508     0.508    -0.515     1.000
1395
```

Figure 3.6.1. Covariance matrix of estimated parameter set and direct correlation coefficients.

Lines 1400–1405 of Figure 3.6.2 contain a list with the parameter estimates and their *a priori*, conditional, and marginal standard deviations. A value is given in Column 3 only if prior information of a parameter is weighted using its *a priori* standard deviation. The conditional standard deviation indicates the estimation uncertainty assuming that all the other parameters are perfectly known. The marginal standard deviation given in Column 5 is the square-root of the diagonal element of $\mathbf{C}_{pp}$. Column 6 holds Γ, the measure of overall parameter correlation given by Equation (2.8.4.4); the higher its value, the more independent the estimate. The parameters in this table and the following correlation chart are sorted according to this criterion. In this example, permeability is the most independent parameter, whereas heat conductivity is strongly correlated to all the other parameters. Column 7 can be interpreted as a measure of how much has been learned about a specific parameter by performing the inversion. Since no prior information was given, i.e., nothing was assumed to be known about any of the parameters, Column 7 contains only ones in this example. If the uncertainty of a parameter after the inversion is only slightly lower than the *a priori* standard deviation, the information gain would have been minimal, resulting in a value close to zero.

In the correlation chart, all parameters are connected to each other, where the vertical lines linking two parameters indicate the correlation coefficient (Line 1411). Since the parameters are sorted according to their overall correlation Γ, the correlation chart displays a pyramid-like structure, with long horizontal lines extending from the most strongly correlated parameter at the bottom to shorter lines connecting the most independent parameter at the top.

```
1396
1397 Standard Deviations
1398 -------------------
          1                2       3          4           5          6       7
1399 PARAMETER          ESTIMATE PRIOR CONDITIONAL     MARGINAL     C/M    1-J/P
1400 PERMEABILITY     -0.1422E+2  N/A     0.1867E-2   0.2032E-2    0.919   1.000
1401 RESERVOIR TEMP.   0.3001E+3  N/A     0.9280E-1   0.1121E+0    0.827   1.000
1402 POROSITY FRACT    0.2633E+0  N/A     0.8702E-1   0.1415E+0    0.615   1.000
1403 FRACT. SPACING    0.5745E+2  N/A     0.2577E+0   0.1833E+1    0.141   1.000
1404 SPECIFIC HEAT     0.1036E+4  N/A     0.5503E+1   0.4914E+2    0.112   1.000
1405 HEAT COND.        0.2704E+1  N/A     0.2623E-1   0.3213E+0    0.082   1.000
1406
1407
1408 Correlation Chart
1409 -----------------
1410
1411  -0.8    -0.6    -0.4    -0.2     0.0         0.0     0.2     0.4     0.6     0.8
1412    |-------|-------|-------|-------| PARAMETER |-------|-------|-------|-------|
1413
1414                          |--||-----|    PERM.   |------|
1415                          |  ||                         |
1416                          |  ||----||    TEMP.   |-|-|--||
1417                          |  |     |               | |  ||
1418                       |--|--|-----||  POROSITY  |-|-|--||-------------|--|
1419                       |  |        |               | |  |              |  |
1420                       |  |--------||  SPACING   |-|-|--|--------------|--|--|
1421                       |  |        |                 |  |              |     |
1422   |-------------------|--|--------||  SP. HEAT  |---|--|              |     |
1423   |                      |                          |                 |     |
1424   |----------------------|---------|  HEAT COND |---|-----------------|-----|
1425
```
Figure 3.6.2. Standard deviations and correlation chart.

iTOUGH2 performs an eigenanalysis of the estimation covariance matrix. A performance index (Line 1429) of less than 1 indicates an accurate eigenanalysis. The condition number (Line 1430) is defined as the ratio of the largest and smallest eigenvalue. It determines the dominance of one eigenvalue over the others, also affecting the accuracy of the eigenanalysis. A large scaled condition number (Line 1431), which is the condition number based on the eigenvalues divided by the corresponding parameter estimates, indicates the presence of a long valley in the objective function. The eigenvalues given on Line 1439 represent the actual lengths of the semiaxes of the hyper-ellipsoid. We prefer to analyze the scaled eigenvalues (Line 1447). The parameter associated with the largest scaled eigenvalue is usually the most uncertain, and—if also strongly correlated—responsible for poor identifiability of all parameter combinations along the corresponding eigenvector, i.e., the respective column of the modal matrix shown on Lines 1455–1460. Note that the eigenvectors are normalized.

```
1426
1427 Eigenanalysis of Covariance Matrix
1428 ----------------------------------
1429 Performance index      :  0.29851783E-01
1430 Condition number       :  0.69291990E+09
1431 Scaled condition number:  0.69291990E+05
1432
1433
1434 Eigenvalues
1435 -----------
1436
1437                            1          2          3          4          5          6
1438                        PERM.   POROSITY  SPEC HEAT  HEAT COND    SPACING      TEMP.
1439   1 Eigenvalue:  0.3486E-5  0.9328E-2  0.2415E+4  0.6878E-3  0.3141E+1  0.1420E-1
1440
1441
1442 Scaled Eigenvalues
1443 ------------------
1444
1445                            1          2          3          4          5          6
1446                        PERM.   POROSITY  SPEC HEAT  HEAT COND    SPACING      TEMP.
1447   1 Eigenvalue:  0.1394E-4  0.2332E+0  0.9662E+0  0.1719E-1  0.3141E-1  0.1420E-3
1448
1449
1450 Eigenvectors
1451 ------------
1452
1453                             1          2          3          4          5          6
1454                         PERM.   POROSITY  SPEC HEAT  HEAT COND    SPACING      TEMP.
1455 1 PERMEABILITY    0.9998E+0  0.1608E-2  0.8340E-5  0.1700E-1 -0.2504E-3 -0.2003E-2
1456 2 POROSITY FRACT -0.2874E-2  0.8071E+0 -0.1042E-2  0.1600E+0  0.4643E-1  0.5663E+0
1457 3 SPECIFIC HEAT  -0.8171E-4 -0.5880E-3  0.9999E+0  0.4587E-2  0.1078E-1  0.4976E-3
1458 4 HEAT COND.     -0.1611E-1 -0.2143E+0 -0.5591E-2  0.9719E+0  0.9213E-1  0.2316E-1
1459 5 FRACT. SPACING  0.1856E-2 -0.1518E-1 -0.1027E-1 -0.9688E-1  0.9945E+0 -0.3253E-1
1460 6 RESERVOIR TEMP  0.4937E-2 -0.5498E+0 -0.1359E-3 -0.1412E+0  0.4763E-2  0.8232E+0
1461
1462
```

Figure 3.6.3. Eigenanalysis of estimation covariance matrix.

3.7 Printout From Residual Analysis

The residual analysis has been discussed in Section 2.8.5. We first describe the columns of the main table shown in Figure 3.7.1. The first column contains the index i, $i = 1, \ldots, m$, frequently used to refer to a certain calibration point (see, for example, Figures 3.4.1 and 3.5.1 as well as specially requested output). The first n observations represent prior information (Lines 1477–1482). In this section of the iTOUGH2 output, the observations are grouped by data set (i.e., not by time as in the actual residual vector), as can be seen in Column 2, which labels the data set of the corresponding residual.

The calibration time in user-specified units (default: seconds) is shown in Column 3. Columns 4 through 6 contain, respectively, the measured and calculated value and their difference, the residual, Equation (2.4.1). For the first n rows, Column 4 holds the prior information value, and Column 5 the best estimate. Note that the measured value (Column 4) may not be an actually recorded measurement, but a value interpolated between two data points. It may also be multiplied by the conversion factor shown in Figure 3.3.2, Column 10. The computed value (Column 5) is the result from the TOUGH2 simulator; a shift and/or linear trend may have been applied if such parameters were estimated.

Column 7 holds the weight $1/\sigma_z$, i.e., the square-root of the reciprocal diagonal element of matrix $\mathbf{C}_{zz}$. For measurement errors that are constant for each data set, the value is the reciprocal of the standard deviation given in Figure 3.3.2, Column 5. Column 8 contains the loss function ω (see Table 2.6.5.1), i.e., the contribution of the calibration point to the final objective function. For least squares, this is the square of the weighted residual, r^2/σ_z^2 .

The standard deviation reported in Column 9 is the uncertainty of the simulation result, calculated using Equation (2.8.5.7). It is also the uncertainty of the model prediction when performing linear error propagation analysis (FOSM, see Section 2.8.7), i.e., the square-root of the diagonal element of matrix $\mathbf{C}_{\hat{z}\hat{z}}$, Equation (2.8.7.4). Column 10 holds the reliability measure y_i, Equation (2.8.5.9). Local reliabilities less than 0.25 are marked with "*" (see, for example, Line 1887). Finally, the normalized residual, Equation (2.8.5.10), is given in Column 11. Normalized residuals that exceed the quantile of the normal distribution on the chosen confidence level are marked with "*" (see, for example, Lines 1573 and 1887); they should be checked as potential outliers.

The values in all columns (except time in Column 3) are given in standard TOUGH2 units.

```
1463
1464 ===============================================================================
1465                                RESIDUAL ANALYSIS
1466 ===============================================================================
1467
1468 RESIDUAL : Measured - computed
1469 OMEGA    : Loss function (= squared weighted residual for least squares)
1470 STD. DEV.: A posteriori standard deviation of computed system response
1471 Yi       : Local reliability. Observations with Yi<0.25 are poorly controlled.
1472 Wi       : Normalized residual. If |Wi|>u(0.99) = 2.58 obs. is potential outlier.
1473
       1     2        3        4        5        6       7       8       9     10    11
1474 -------------------------------------------------------------------------------
1475    # OBSERV.  TIME [a] MEASURED COMPUTED RESIDUAL  WEIGHT   OMEGA    SDEV  Yi    Wi
1476 -------------------------------------------------------------------------------
1477    1 PERMEABILITY      -.130E+2 -.142E+2  .122E+1 .10E-49 .000E+0 .203E-2
1478    2 POROSITY FRACT     .500E+0  .263E+0  .236E+0 .10E-49 .000E+0 .141E+0
1479    3 SPECIFIC HEAT      .800E+3  .103E+4 -.236E+3 .10E-49 .000E+0 .491E+2
1480    4 HEAT COND.         .250E+1  .270E+1 -.204E+0 .10E-49 .000E+0 .321E+0
1481    5 FRACT. SPACING     .200E+2  .574E+2 -.374E+2 .10E-49 .000E+0 .183E+1
1482    6 RESERVOIR TEMP.    .250E+3  .300E+3 -.501E+2 .10E-49 .000E+0 .112E+0
1483    7 P. INJECT. .10E+0  .100E+8  .100E+8 -.769E+5 .500E-5 .147E+0 .165E+5 .99 -0.39
1484   16 P. INJECT. .20E+0  .102E+8  .101E+8  .127E+6 .500E-5 .407E+0 .157E+5 .99  0.64
1485   25 P. INJECT. .30E+0  .101E+8  .101E+8  .809E+4 .500E-5 .163E-2 .154E+5 .99  0.04
1486   34 P. INJECT. .40E+0  .100E+8  .101E+8 -.105E+6 .500E-5 .280E+0 .154E+5 .99 -0.53
1487   43 P. INJECT. .50E+0  .101E+8  .101E+8 -.483E+5 .500E-5 .585E-1 .155E+5 .99 -0.24
(...)
1533    8 P. PRODUC. .10E+0  .599E+7  .583E+7  .155E+6 .500E-5 .604E+0 .252E+5 .98  0.78
1534   17 P. PRODUC. .20E+0  .582E+7  .579E+7  .311E+5 .500E-5 .242E-1 .197E+5 .99  0.16
1535   26 P. PRODUC. .30E+0  .598E+7  .570E+7  .281E+6 .500E-5 .198E+1 .208E+5 .99  1.42
1536   35 P. PRODUC. .40E+0  .582E+7  .571E+7  .103E+6 .500E-5 .267E+0 .333E+5 .97  0.52
1537   44 P. PRODUC. .50E+0  .561E+7  .598E+7 -.369E+6 .500E-5 .341E+1 .432E+5 .95 -1.89
(...)
1573 368 P. PRODUC. .41E+1  .502E+7  .560E+7 -.579E+6 .500E-5 .840E+1 .226E+5 .99 -2.92*
(...)
1887  51 VAP. PROD. .50E+0 -.214E+0 -.189E+0 -.255E-1 .100E+3 .649E+1 .945E-2 .11*-7.53*
(...)
1928 420 VAP. PROD. .46E+1 -.210E+0 -.207E+0 -.264E-2 .100E+3 .701E-1 .134E-2 .98 -0.27
1929 429 VAP. PROD. .47E+1 -.200E+0 -.210E+0  .972E-2 .100E+3 .946E+0 .150E-2 .98  0.98
1930 438 VAP. PROD. .48E+1 -.223E+0 -.213E+0 -.102E-1 .100E+3 .105E+1 .170E-2 .97 -1.04
1931 447 VAP. PROD. .49E+1 -.215E+0 -.216E+0  .529E-3 .100E+3 .280E-2 .193E-2 .96  0.05
1932 456 VAP. PROD. .50E+1 -.224E+0 -.219E+0 -.483E-2 .100E+3 .233E+0 .217E-2 .95 -0.50
1933
```

Figure 3.7.1. Residual analysis.

For each observation type, the residuals are visualized in a scatter plot of the residual versus the calculated value. In the example shown in Figure 3.7.2, pressure residuals are depicted as digits, where the number refers to the number of the data set it belongs to (see Figure 3.3.2, Column 2). The residuals are expected to be randomly distributed around the center line (Line 1649). Unwanted trends in the residuals indicating a systematic error are usually easy to detect. Similar residual plots are generated for the temperature and flow rate measurements (not shown).

Notice that the residual plot does not reflect the weight assigned to each data point. Thus, the residual plot may be misleading if the data sets or individual data points have been assigned significantly different measurement errors.

```
1934
1935 Residual Plots
1936 --------------
1937
1938    RESIDUAL     --------------------------------------------------------------
1939   0.504E+06     |                                        3                   |
1940   0.452E+06     |                                                            |
1941   0.400E+06     |                                    4    3                  |
1942   0.349E+06     |    2                                                       |
1943   0.297E+06     |  2                                     33                  |
1944   0.246E+06     |    2                               44  33               1  |
1945   0.194E+06     |   22  2                            444 3                  1|
1946   0.142E+06     |  2222  2                             4 33                 1|
1947   0.909E+05     |   2222  2                           44   3             1 11|
1948   0.393E+05     |  222 2  2                          444 3                11|
1949  -0.123E+05     |-222-2- 2 - - - - -  - - - - - - - - - 44-3-3- - - - - - - 11|
1950  -0.639E+05     | 222  22                             44 33             11 1|
1951  -0.115E+06     | 22 2                               444  3             11 1|
1952  -0.167E+06     | 2    2 2                             4  3              111|
1953  -0.219E+06     |                                     4  33               11|
1954  -0.270E+06     | 2      22                           4  33                1|
1955  -0.322E+06     |                                     44  3               1 |
1956  -0.373E+06     |     2                              44   3                1|
1957  -0.425E+06     |        2                                               1 |
1958  -0.477E+06     | 2                                    4 3                  |
1959  -0.528E+06     |                                     4  3                  |
1960  -0.580E+06     |   2                                4                      |
1961                 --------------------------------------------------------------
1962               0.537E+07                                                 0.104E+08
1963                                            PRESSURE [Pa]
(...)
```

Figure 3.7.2. Scatter plot of residuals.

The residual analysis is summarized in a table reproduced in Figure 3.7.3. The maximum weighted residual (Lines 2026–2028, see also Figure 3.7.1, Line 1573) is the one with the largest contribution to the objective function. However, it may not be a potential outlier, i.e., its w_i value may be acceptably low as a result of high prediction uncertainty. In this example, the largest normalized residual (Lines 2030–2032) is associated with the vapor flow measurement shown in Figure 3.7.1, Line 1887, which is also the only poorly controlled measurement (Line 2029). Line 2033 shows the probable size of the maximum error (Equation 2.8.5.11) for residual No. 51.

A total of 10 iterations, i.e., minimization steps, were performed in this example (Line 2038), requiring 89 solutions of the forward problem (Line 2039). The two measures controlling numerical accuracy of the residual analysis, Equations (2.8.5.12) and (2.8.5.13), are shown on Lines 2044 and 2045.

During the 10 iterations, the objective function was reduced from the value obtained with the initial parameter set (Line 2050) to its final value shown on Line 2051; the relative contribution of residuals from individual data sets and observation types (see Figure 3.7.4) is reported as a percentage of the final objective function.

```
2023
2024 Summary of Residual Analysis
2025 ----------------------------
2026 Max weighted residual at observation    :          368
2027 Max weighted residual                   :    -0.2899E+01
2028 Max residual                            :    -0.5798E+06
2029 Number of poorly controlled observations:            1
2030 Number of large normalized residuals    :            5
2031 Max normalized residual at observation  :           51
2032 Max normalized residual                 :         7.53
2033 Probable size of maximum error          :     0.2221E+00
2034
2035
2036 Iteration Statistics
2037 --------------------
2038 Number of iterations                    :           10
2039 Number of TOUGH2 calls                  :           89
2040
2041
2042 Control Measures
2043 ----------------
2044 Trace (P*QLL) : n   =   6               :     0.6000E+01
2045 Sum   (Yi)    : m-n = 444               :     0.4440E+03
2046
2047
2048 Objective Function                                                  C.O.F.
2049 ------------------
2050 Initial value of objective function     :     0.1430E+06            31888.6 %
2051 Minimum value of objective function     :     0.4484E+03              100.0 %
2052
```

Figure 3.7.3. Summary of residual analysis and iteration statistics.

Figure 3.7.4 shows the moment analysis of the residuals, Equations (2.8.5.1) through (2.8.5.6). The first block (Lines 2068–2077) presents the analyses for each data set, wheras Line 2079 contains the moments of all weighted residuals. The second block, Lines 2086–2089 shows the same information for each observation type. Of special interest is Column 9, which shows the ratio of the bias—the mean of the residuals—and the standard deviation. If this ratio significantly deviates from zero, the corresponding data set or observation type is systematically over- or underpredicted by the model, i.e., there is likely to be a systematic error in either the data or the model. Column 10 shows the contribution of the data set or observation type to the final objective function. Ideally, the contributions should reflect the number of points (see Column 2) in the data set in proportion to the total number of calibration points, m.

Figure 3.7.5 shows the linear regression analysis of a scatter plot with the calculated versus observed system response. The intercept and slope are expected to be close to zero and one, respectively. For more details see the discussion in *Finsterle* [1999c; Problem 6].

```

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

MEAN     : Mean of residuals = bias
MEDIAN   : Median of residuals
STD. DEV.: Root mean squared deviation of residuals from bias
AVE. DEV.: Mean absolute deviation of residuals from bias
SKEWNESS : Degree of asymmetry of residuals around bias
KURTOSIS : Relative peakedness of distribution
B/S      : Ratio of bias and standard deviation
C.O.F.   : Relative contribution to final objective function
    1                  2       3       4         5        6      7      8     9    10
=========================================================================================
DATASET       DATAPOINTS     MEAN   MEDIAN     SDEV     ADEV   SKEW   KURT   B/S C.O.F.
-----------------------------------------------------------------------------------------
PRIOR INFORMATION      6                                                         0.0 %
P. INJECT.   [Pa]      50 -.651E+4  .904E+4  .149E+6  .120E+6  -.370  -.589  .044  6.0 %
P. PRODUCT.  [Pa]      50  .282E+5  .633E+5  .193E+6  .144E+6  -.908   .717  .146  1.3 %
P. OBS. 1    [Pa]      50 -.149E+5 -.268E+5  .217E+6  .175E+6   .320  -.516  .069 12.9 %
P. OBS. 2    [Pa]      50 -.175E+4  .234E+5  .209E+6  .162E+6  -.510  -.213  .008 11.9 %
T. PRODUCT.  [C]       50 -.107E+1 -.821E+0  .451E+1  .348E+1   .263  -.017  .237  9.4 %
T. OBS. 1    [C]       50 -.419E+0 -.404E+0  .536E+1  .430E+1  -.344  -.509  .078 12.6 %
T. OBS. 2    [C]       50  .134E+1  .172E+1  .497E+1  .403E+1  -.003  -.690  .270 11.6 %
WATER PROD.  [kg/sec] 50  .125E-1  .677E-2  .200E+0  .161E+0   .306  -.458  .063  1.9 %
VAPOR PROD.  [kg/sec] 50 -.480E-3  .685E-3  .113E-1  .896E-2  -.235  -.426  .042 14.1 %
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ALL RESIDUALS  [-]   456  .119E-2  .859E-2  .993E+0  .789E+0  -.157  -.120  .001  100 %
=========================================================================================


=========================================================================================
DATATYPE      DATAPOINTS     MEAN   MEDIAN     SDEV     ADEV   SKEW   KURT   B/S C.O.F.
-----------------------------------------------------------------------------------------
PRIOR INFORMATION      6                                                         0.0 %
PRESSURE    [Pa]      200  .127E+4  .169E+5  .193E+6  .153E+6  -.324  -.010  .007 41.3 %
FLOW RATE   [kg/sec] 100  .601E-2  .103E-2  .141E+0  .849E-1   .571  2.226  .043 25.0 %
TEMPERATURE [C]      150 -.483E-1 -.325E+0  .503E+1  .403E+1  -.049  -.340  .010 33.6 %
=========================================================================================

```

Figure 3.7.4. Statistical moment analysis of residuals.

```

Linear Regression Analysis Calculated Vs. Observed
--------------------------------------------------

======================================================================
DATASET                DATAPOINTS     INTERCEPT      SLOPE           R
----------------------------------------------------------------------
P. INJECTION      [Pa]        50     0.794E+06   0.922E+00    0.375414
P. PRODUCTION     [Pa]        50    -0.437E+06   0.108E+01    0.741147
P. OBS. 1         [Pa]        50    -0.358E+07   0.139E+01    0.266365
P. OBS. 2         [Pa]        50    -0.260E+07   0.130E+01    0.257011
T. PRODUCTION     [C]         50     0.207E+02   0.920E+00    0.410527
T. OBS. 1         [C]         50     0.217E+03   0.273E+00    0.066226
T. OBS. 2         [C]         50    -0.180E+05   0.610E+02    0.031780
WATER PROD.       [kg/sec]    50    -0.203E+01   0.425E+00    0.063968
VAPOR PROD.       [kg/sec]    50     0.353E-02   0.102E+01    0.939064
======================================================================

```

Figure 3.7.5. Linear regression analysis of plot calculated versus observed.

3.8 Model Test and Optimality Criteria

The outcome of the Fisher Model Test (see Table 2.8.3.1) is summarized in Lines 2113–2120. The root mean square error and estimated error variance, Equation (2.8.3.1), measure the overall goodness-of-fit. If both the stochastic and functional models are correct, the estimated error variance should not deviate significantly from one, i.e., it should be smaller than the critical value of the F-distribution shown on Line 2115. The quantile depends on the degree of freedom and the confidence level. Depending on the outcome of the Fisher Model Test or following the user's choice (see discussion of Figure 3.3.3, Line 122), the error analysis is based on either the *a priori* or *a posteriori* error variance. In this example, the *a posteriori* error variance was used as a result of the Fisher model test (see Line 2119). The quantile of the t-distribution given on Line 2120 can be used to construct confidence intervals according to Equation (2.8.4.6).

The optimality criteria, Equations (2.8.6.1) through (2.8.6.3), are given on Lines 2125–2127. The criteria are evaluated using either the actual, unscaled covariance matrix $\mathbf{C}_{pp}$, or the one scaled according to Equation (2.8.6.4). Line 2128 shows the log-likelihood criterion, Equation (2.6.4.2), followed by the model identification criteria after Akaike, Equation (2.8.6.5), and Kashyap, Equation (2.8.6.6).

```
2110
2111 Fisher Model Test
2112 -----------------
2113 Root mean square error              :      0.1005E+01
2114 Estimated error variance            :      0.1010E+01
2115 Critical value of F-distribution    :      0.1203E+01
2116 Degree of freedom                   :            444     (No prior information)
2117 Confidence level (1-alpha)          :           99.0 [%]
2118 Lucky you                           : Model test successful!
2119 Error analysis based on             : a posteriori variance = 0.1009964E+01
2120 Quantile of t-distribution          :      0.2587E+01
2121
2122
2123 Optimality Criteria                           unscaled         scaled
2124 -------------------
2125 D-optimality = det(Cpp)             :      0.2412E-08     0.7358E-25
2126 A-optimality = trace(Cpp)           :      0.2419E+04     0.3064E+00
2127 E-optimality = max eigenvalue       :      0.2416E+04     0.9663E+00
2128 Log-likelihood ln(L)                :     -0.3010E+04
2129 Akaike  =-2ln(L)+2n                 :      0.6031E+04
2130 Kashyap =-2ln(L)+n*ln(m/2Pi)+ln|F|  :      0.6059E+04
2131
```
Figure 3.8.1. Model test and optimality criteria.

3.9 Summary Output

The final block in the output file, reproduced in Figure 3.9.1, shows a summary of the major inverse modeling results. The "+" in the first column of Line 2133 indicates that the Fisher Model Test was successfully passed. In Lines 2136–2141, the parameters are identified in Columns 1 through 4, followed by their initial guesses and best estimates. The number of significant digits printed in Column 6 depends on the parameter's estimation uncertainty, which is reported in Column 8. The ratio of the conditional and marginal standard deviation given in Column 9 is the measure of overall parameter correlation, Equation (2.8.4.4). Two aggregate sensitivities can be found in Columns 10 and 11. The first is the total parameter sensitivity, Equation (2.8.2.5), and the second is the sensitivity of the objective function with respect to the parameter, Equation (2.8.2.6).

Each iTOUGH2 run is logged in file *itough2.log* with the date of its completion and the CPU time used (Line 2144). The number of error and warning messages is given on Line 2146.

```
2132      1          2       3     4      5         6        7       8         9      10    11
2133 +!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2134 PARAMETER    V/L/F ROCKS PAR  INITIAL  ESTIMATE   STANDARD DEVIATIONS  SENSITIVITY
2135                                                PRIOR  MARGINAL    C/M  OUTPUT   OF
2136 PERMEABILITY LOG10 FRACT   1 -.13E+2 -.14221E+2  N/A  0.203E-2  0.919  3546.3 4.05
2137 POROSITY     VALUE FRACT   1  .25E+0      .26E+0  N/A  0.142E+0  0.615    13.6 0.18
2138 SPEC. HEAT   VALUE FRACT+1 1  .80E+3     .104E+4  N/A  0.491E+2  0.112    53.1 8.37
2139 HEAT COND.   VALUE FRACT+1 1  .25E+1      .27E+1  N/A  0.321E+0  0.082    46.4 3.92
2140 SPACING      VALUE -----   1  .20E+2     .575E+2  N/A  0.183E+1  0.141   232.2 6.48
2141 TEMPERATURE  VALUE DEFAU   1  .25E+3  .30019E+3  N/A  0.112E+0  0.827  2133.0 8.21
2142 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2143
2144 -- 12124th iTOUGH2 simulation job completed: 3-Nov-98 11:13 - CPU time = 972.2 sec
2145
2146 --     0 error(s) and  0 warning(s) detected
```

Figure 3.9.1. Summary of inverse modeling results.

3.10 Version Control

Version control information is always written to the iTOUGH2 message and TOUGH2 output file (unless option `NOVER` is set), and is appended to the iTOUGH2 output file if command `>>> VERSION` is used. The dimensions of major arrays as specified in the FORTRAN include file *maxsize.inc* is reproduced first (Figure 3.10.1), followed by the list of version control statements shown in Figure 3.10.2. The list of subroutines touched during an iTOUGH2 run varies depending on the application.

```
=====================================================================================
ARRAY DIMENSIONS (SEE FILE maxsize.inc)
-------------------------------------------------------------------------------------
MAXEL    = 4000  Maximum number of elements
MAXCON   = 8000  Maximum number of connections
MAXK     =    2  Maximum number of components
MAXEQ    =    3  Maximum number of equations
MAXPH    =    2  Maximum number of phases
MAXB     =    8  Maximum number of phase-dependent secondary variables
MAXSS    =   50  Maximum number of sinks/sources
MAVTAB   =   20  Maximum average number of table entries per sink/source
MAXROC   =   50  Maximum number of rock types
MAXTSP   =    5  Maximum number of specified time steps, divided by eight
MAXLAY   =   10  Maximum number of reservoir layers for wells on deliverability
MXRPCP   =    7  Maximum number of parameters for rel. perm. and cap. pres. functions
MXPCTB   =   30  Maximum number of points in table for ECM capillary pressure
MXTBC    =   10  Maximum number of elements with time vs. boundary condition
MXTBCT   =   10  Maximum number of time vs. pressure data
MAXTIM   =  500  Maximum number of calibration times
MAXN     =   20  Maximum number of parameters to be estimated
MAXO     =  100  Maximum number of datasets
MAXM     = 2000  Maximum number of calibration points
MAXPD    = 1000  Maximum number of paired data
MAXR     =   25  Maximum number of elements or indices of each parameter or observation
MAXBRK   =   20  Maximum number of points in time at which SAVE file is written
MAXEBRK  =   20  Maximum number of elements with new initial conditions after restart
MAXCOEFF =    5  Maximum number of coefficients for data modeling functions
MAXMCS   =  100  Maximum number of Monte Carlo simulations
MAXCURVE =  100  Maximum number of curves to be plotted
MAXXGR   =    3  Dimension of third index of array XGUESSR
MTYPE    =   17  Number of observation types
MPFMT    =    6  Number of plot file formats
MAXPV    =    4  Maximum number of primary variables
-------------------------------------------------------------------------------------
```

Figure 3.10.1. Printout of array dimensions.

```
------------------------------------------------------------------------------------
PROGRAM  VERSON DATE              COMMENT
------------------------------------------------------------------------------------
iTOUGH2         Current version    iTOUGH2 V4.0 (January 19, 1999)
------------------------------------------------------------------------------------
iTOUGH   1.0     1 AUGUST    1992   iTOUGH User's Guide, Version 1.0, Report NIB 92-99
iTOUGH2  2.2     1 FEBRUARY  1994   iTOUGH2 User's Guide, Version 2.2, Report LBL-34581
iTOUGH2  3.0    12 JULY      1996   YMP Software qualification, Report LBNL-39489
iTOUGH2  3.1     1 APRIL     1997   iTOUGH2 Command Reference, V3.1, Report LBNL-40041
iTOUGH2  3.2    30 JUNE      1998   YMP Software qualification, Report LBNL-42002
iTOUGH2  3.3     1 OCTOBER   1998   Parallelization using PVM, Report LBNL-42261
iTOUGH2  4.0    19 JANUARY   1999   Report LBNL-40040
------------------------------------------------------------------------------------
WHATCOM  1.0    10 AUGUST    1993   #35: Q: WHAT COMPUTER IS USED? A: SUN
CALLSIG  1.0     5 DECEMBER  1995   #112: SIGNAL HANDLER
CPUSEC   1.0    10 AUGUST    1993   #--: RETURNS CPU-TIME (VERSION SUN)
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 3.2    27 MAY       1998   #29: PRINTS iTOUGH2 HEADER
DAYTIM   1.0    10 AUGUST    1993   #32: RETURNS DATE AND TIME (VERSION SUN)
THEADER  3.2    27 MAY       1998   #30: PRINTS TOUGH2 HEADER
INPUT    4.0    19 JANUARY   1999   READ ALL DATA PROVIDED THROUGH FILE *INPUT*,  + IFS
MESHM    1.0    24 MAY       1990   EXECUTIVE ROUTINE FOR INTERNAL MESH GENERATION
MINC     1.0    22 JANUARY   1990   "SECONDARY" FRACTURED-POROUS MEDIUM MESH
PART     1.0    22 JANUARY   1990   READ SPECIFICATIONS OF MINC-PARTITIONING
GEOM     1.0     1 MAY       1991   CALCULATE GEOMETRY PARAMETERS OF SECONDARY MESH
PROX     1.0    22 JANUARY   1990   CALCULATE PROXIMITY FUNCTIONS
INVER    1.0    22 JANUARY   1990   INVERT A MONOTONIC FUNCTION
MINCME   1.0    14 FEBRUARY  1990   PROCESS PRIMARY MESH, WRITE SECONDARY MESH ON *MINC*
CHECKMAX 4.0    19 JANUARY   1999   #41: CHECK KEY DIMENSIONS
FLOPP    1.0    11 APRIL     1991   CALCULATE NUMBER OF SIGNIFICANT DIGITS
RFILE    4.0    19 JANUARY   1999   INITIALIZE DATA FROM FILES *MESH*, *GENER*, *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 3.2    20 JUNE      1998   # 3: READS PARAMETERS TO BE ESTIMATED
INPAR    3.1    17 MARCH     1997   # 4: READS PARAMETER VALUES, WEIGHTS, ETC.
INELEM   3.1     3 APRIL     1997   #23: READS GRID BLOCK NAME AFTER A COLON
NEXTWORD 2.5     9 FEBRUARY  1996   #27: EXTRACTS NEXT WORD ON A LINE
INWBP    3.1    17 MARCH     1997   #11: READS WEIGHT, BOUNDS, ANNOTATION, AND INDICES
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 3.3    17 JULY      1998   #12: READS TYPE OF OBSERVATION
INTIMES  3.1    29 APRIL     1997   #13: READS TIMES AT WHICH OBSERVATIONS ARE AVAILABLE
INOBS    3.3    17 JULY      1998   #15: READS OBSERVATION INFOS
INOBSDAT 2.5    13 JANUARY   1996   #17: READS ALL OBSERVED DATA
INPAIRED 3.1     2 APRIL     1997   #19: READS PAIRED DATA SET
INWEIGHT 3.2     7 OCTOBER   1997   #20: READS WEIGHTS
INCOMPUT 1.0     1 AUGUST    1992   #16: READS VARIOUS COMPUTATIONAL PARAMETERS
INPRINT  4.0    19 JANUARY   1999   #80: READS OUTPUT OPTIONS
INTOLER  3.3     14 AUGUST   1998   #83: READS TOLERANCE/STOPPING CRITERIA
INJACOB  1.0     1 AUGUST    1992   #84: READS PARAMETERS FOR COMPUTING JACOBIAN
INOPTION 3.3    30 SEPTEMBER 1998   #85: READS PROGRAM OPTIONS
INSELECT 2.5     5 JANUARY   1996   #14: READ OPTIONS FOR PARAMETER SELECTION
GETINDEX 3.3    17 JULY      1998   #45: GETS INDEX OF ELEMENTS, CONNECTIONS, SOURCES
INIGUESS 4.0    19 JANUARY   1999   #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 4.0    19 JANUARY   1999   #39: INITIALIZES ARRAY WSCALE
```

Figure 3.10.2. Version control statements.

```
OBSMEAN  1.0     1 AUGUST    1992   #40: CALCULATES MEAN OF OBSERVATIONS
SETXSCAL 1.0     1 AUGUST    1992   #42: INITIALIZES ARRAY XSCALE
IN_OUT   3.2    20 JUNE      1998   #35: PRINTS A SUMMARY OF INPUT DATA
TIMEWIND 2.5    30 NOVEMBER  1995   #53: SETS TIME WINDOW
PRSTATUS 3.1    20 FEBRUARY  1997   #91: PRINTS STATUS MESSAGES
ERRORMSG 2.5    21 MARCH     1996   #34: PRINTS ERROR MESSAGES
LEVMAR   2.5    26 MARCH     1996   #99: LEVENBERG-MARQUARDT OPTIMIZATION ALGORITHM
FCNLEV   3.3    30 JULY   1  1998   #50: RETURNS WEIGHTED RESIDUAL VECTOR
UPDATE   3.3     7 JULY      1998   #37: UPDATES PARAMETERS
PRIORINF 2.1    21 SEPTEMBER 1993   #48: PRIOR INFORMATION
OBSERVAT 3.2    17 JULY      1998   #62: COMPARES MEASURED AND CALCULATED QUANTITIES
GETMESH  3.2    20 JUNE      1998   #47: READS FILE MESH, MINC, GENER, AND INCON
GETINCON 4.0    19 JANUARY   1999   #46: READS FILE INCON
INITTOUG 2.5    18 APRIL     1996   #54: INITIALIZES TOUGH2 RUN (REPLACES CYCIT)
EOS      1.0    15 AUGUST    1990   *EOS1* THERMOPHYSICAL PROPERTIES MODULE FOR WATER
SAT      1.0    22 JANUARY   1990   STEAM TABLE EQUATION: SAT. PRES. = F(TEMPERATURE)
RELP     4.0    19 JANUARY   1999   RELATIVE PERMEABILITIES
TSAT     1.0    14 MARCH     1991   SATURATION TEMPERATURE AS FUNCTION OF PRESSURE
PCAP     3.2     1 JUNE      1998   CAPILLARY PRESSURE
COWAT    1.0    22 JANUARY   1990   LIQUID WATER DENSITY AND INT. ENERGY
SUPST    1.0    29 JANUARY   1990   VAPOR DENSITY AND INTERNAL ENERGY
VIS      1.0    22 JANUARY   1990   VISCOSITY OF LIQUID WATER AND VAPOR
INDATA   4.0    19 JANUARY   1999   PROVIDE PRINTOUT OF MOST INPUT DATA
BALLA    3.3    17 JULY      1998   SUMMARY BALANCES FOR VOLUME, MASS, AND ENERGY
CALLTOUG 3.1     2 APRIL     1997   #55: CALLS TOUGH2 FOR ONE TIME STEP
TSTEP    3.1    27 MARCH     1997   ADJUST TIME STEPS TO COINCIDE WITH TARGET TIMES
MULTI    4.0    19 JANUARY   1999   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    0.91CG31 JANUARY    1994   INTERFACE FOR LINEAR EQUATION SOLVERS
                                    CAN CALL MA28 OR A PACKAGE OF CONJ. GRADIENT SOLVERS
MC19A                               HARWELL SUBROUTINE FOR SCALING MATRIX
VISW     1.0    22 JANUARY   1990   VISCOSITY OF LIQUID WATER
CONVER   2.5    13 JUNE      1996   UPDATE PRIMARY VARIABLES AFTER CONVERGENCE
OUT      1.0    15 AUGUST    1990   PRINT RESULTS FOR ELEMENTS, CONNECTIONS, SINKS
OBSERVED 2.4     4 AUGUST    1996   #78: RETURNS OBSERVED DATA AS A FUNCTION OF TIME
OBJFUN   4.0    29 JANUARY   1999   #49: COMPUTE OBJECTIVE FUNCTION
WRITEPAR 1.0    17 JUNE      1996   #56: WRITE BEST FIT PARAMETER SET AND BLOCK ROCKS
PLOTFILE 4.0    19 JANUARY   1999   #58: WRITES PLOTFILE IN PLOPO-FORMAT
RANDOM   1.0     1 AUGUST    1992   #71: RANDOM NUMBER GENERATOR
JAC      4.0    19 JANUARY   1999   #51: CALCULATES FINITE DIFFERENCE JACOBIAN
MLLAMBDA 2.2    14 FEBRUARY  1994   #67: ESTIMATES NEW LAMBDAS
VISS     1.0    22 JANUARY   1990   VISCOSITY OF VAPOR AS FUNCTION OF TEMPERATURE
TERMINAT 3.3    28 OCTOBER   1998   61: PERFORM ERROR ANALYSIS AND TERMINATE iTOUGH2
WRIFI    4.0    19 JANUARY   1999   WRITE PRIMARY VARIABLES ON FILE *SAVE*
QFISHER  3.3    22 OCTOBER   1998   QUANTILE OF F-DISTRIBUTION
FISHER   1.0    22 OCTOBER   1998   F-DISTRIBUTION
BETAI    1.0    22 OCTOBER   1998   INCOMPLETE BETA FUNCTION
GAMMLN   1.0    22 OCTOBER   1998   GAMMA FUNCTION
BETACF   1.0    22 OCTOBER   1998   CONTINUED FRACTION EVALUATION
EIGEN    3.2    14 AUGUST    1997   #59: PERFORMS EIGENANALYSIS
LOGLIKE  2.1    29 SEPTEMBER 1993   #68: COMPUTE LOG-LIKELIHOOD
QNORMAL  2.5    13 JANUARY   1996   #87: RETURNS QUANTILE OF NORMAL DISTRIBUTION
MOMENT   3.3    28 OCTOBER   1998   #90: MOMENTS OF DISTRIBUTION
SORT     3.1    17 APRIL     1997   #113: SORTS ARRAY
MOMENT   3.1    17 APRIL     1997   #75: LINEAR REGRESSION ANALYSIS
QSTUDENT 1.0    22 OCTOBER   1998   QUANTILE OF STUDENT T-DISTRIBUTION
STUDENT  1.0    22 OCTOBER   1998   STUDENT T-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 3.10.2. (cont.) Version control statements.

4. PROGRAM ARCHITECTURE

4.1 Program Structure

iTOUGH2 is written in a modular form. Figure 4.1.1 shows the program architecture in a simplified flow chart. The program first reads the TOUGH2 input deck, which defines the forward problem. The required TOUGH2 input may vary depending on the module used. In the iTOUGH2 input file, the user defines the parameters to be estimated, provides the data and the associated measurement errors, and selects program options such as the objective function, the minimization algorithm, the output format, and convergence criteria. Optimization is then initiated, iteratively updating parameter vector **p** and calling TOUGH2 for the calculation of the system response, **z**. Some of the forward calculations may be performed in parallel using PVM [*Finsterle*, 1998b]. The optimization routines of iTOUGH2 communicate with the TOUGH2 simulator through selected COMMON blocks shared by both modules. This program architecture allows one to update both the forward and the inverse part of the code more or less independently.

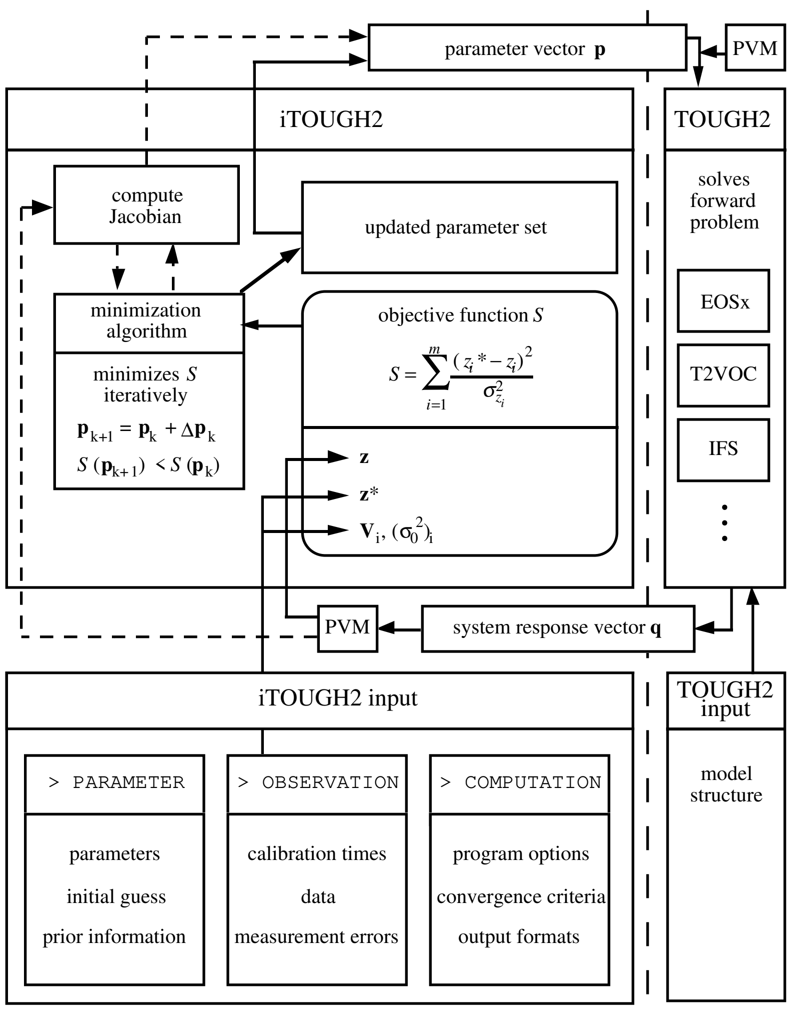


Figure 4.1.1. Simplified iTOUGH2 flow chart.

The architecture of iTOUGH2 allows for safe and convenient maintenance of the program. The code was developed based on the following principles:

- Each subroutine contains an `IMPLICIT NONE` statement, i.e., each constant and variable used in iTOUGH2 is explicitly declared.

- All `COMMON` blocks holding major arrays are defined in FOTRAN include files, ensuring that modifications are made consistently throughout the code.

- Array dimensions are given by constants, which are defined using `PARAMETER` statements in FORTRAN include file *maxsize.inc* (see Section 5.1). This allows for convenient redimensioning of arrays and ensures consistency of array sizes.

- Variables of different types are stored in separate `COMMON` blocks for efficient alignment.

- Compilation is supported by a *Makefile*. The dependencies specified in the makefile ensure that all files affected by a change are recompiled and properly linked.

- Checks are made within iTOUGH2 to ensure that a given array is sufficiently large to accommodate the problem to be solved. If an array index is greater than the size of the array, an error message is issued indicating the constant that must be increased.

- For traceability, array dimensions used for a specific run as well as version control statements are reported in output files (see Section 3.10).

- Subroutines and functions containing machine-dependent system calls are isolated (see files *mdep$COMP.f*, where *$COMP* is the name of a computer platform).

- iTOUGH2 was tested on different platforms to enhance portability.

- The use of non-ANSI FORTRAN extensions is minimized.

5.2 Directory Structure

It is recommended that the installation of iTOUGH2 on Unix machines be performed according to the instructions in this section. However, different file structures can be chosen, requiring minor modifications of the utility script files (see Section 6).

The iTOUGH2 source files and executables are expected to be stored in the iTOUGH2 home directory *$HOME/itough2* or *$HOME/itough2v$VERS*, where *$HOME* is the user's home directory name. If iTOUGH2 is installed in *$HOME/itough2v$VERS*, where *$VERS* is a version identifier, the `-v` option (see Section 6.2) must be used at run time to indicate which version should be selected. If option `-v` is not given on the command line, the executable installed in directory *$HOME/itough2* is used.

The source code is comprised of files with extensions *.f* and *.inc*. The FORTRAN include files contain the COMMON blocks and constants for dimensioning of major arrays (see Section 5.2). The executable is named *itough2_$EOS.$HOST*, where *$EOS* identifies the equation-of-state module, and *$HOST* is the name of the Unix host.

iTOUGH2 is executed using a Unix shell script file *itough2* (see Section 6.2), expected to be stored in a subdirectory *$HOME/bin*, which should be added to the Unix command search path. iTOUGH2 input files can be stored in any directory. For each iTOUGH2 run, the shell script creates a temporary directory with the unique name *$HOME/it2_$PID*, where *$PID* is the process identifier. All input files are copied from the arbitrary working directory to this temporary directory, before iTOUGH2 is started. After termination of the run, the most important output files are copied back to the working directory. The temporary directory is then removed, unless command option `-no_delete` is set (see Section 6.2). During execution, command `prista` can be invoked to check the status file in the temporary directory (see Section 6.4). Furthermore, command `kit` allows a user to send signals to iTOUGH2 applications to gracefully terminate the run (see Section 6.5). Finally, command `it2help` searches file *$HOME/itough2/it2help.txt* and displays manual pages of iTOUGH2 commands (see Section 6.6). The recommended directory structure is visualized in Figure 4.2.1. Directory names are <u>underlined</u>, file names are printed in **bold**, and environment and shell variables to be set by the user are in *italics*.

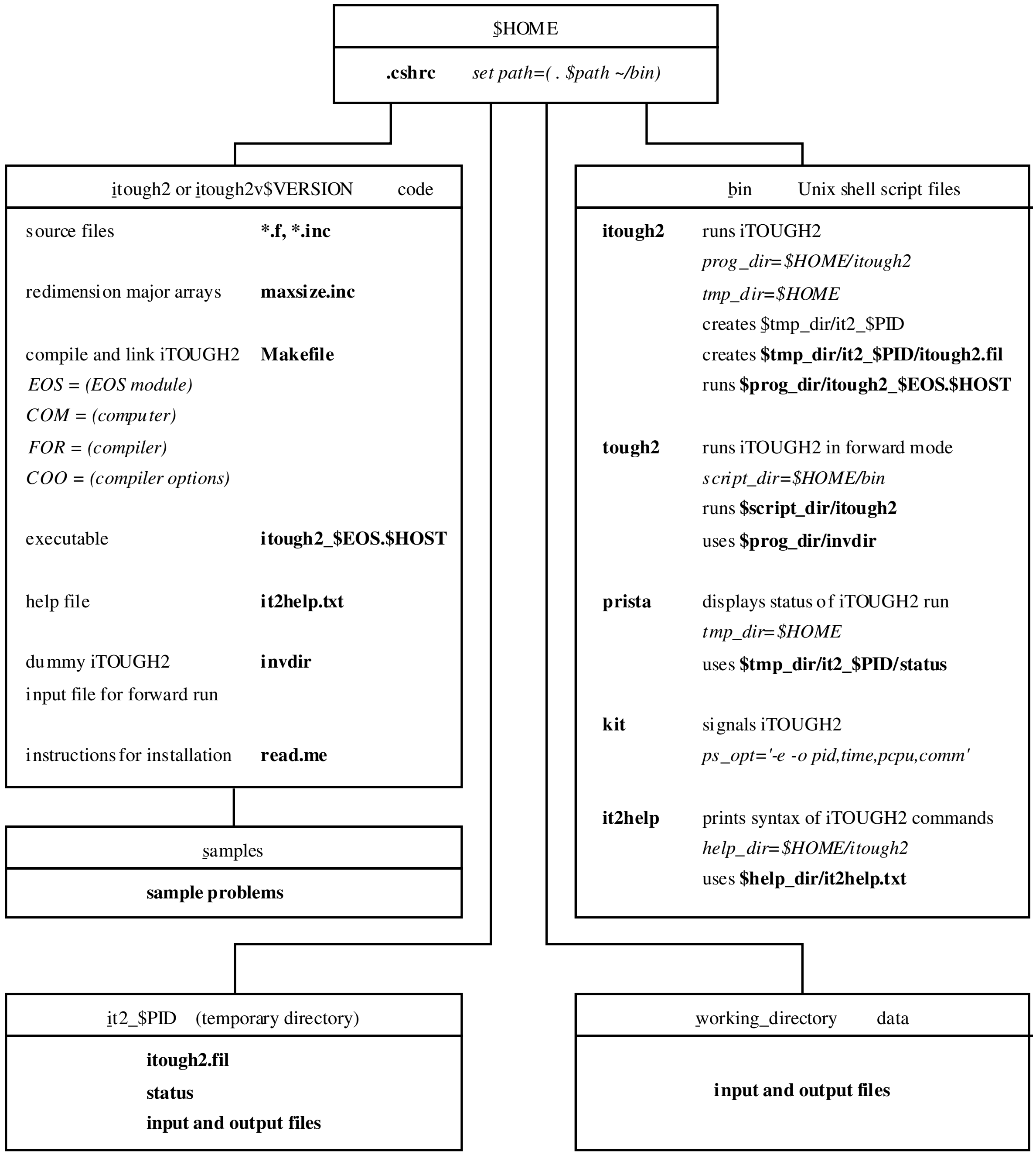


Figure 4.2.1. iTOUGH2 directory structure.

4.3 iTOUGH2 Input and Output Files

Table 4.3.1 shows the names and contents of iTOUGH2 input and output files. Input file names are arbitrary, and are specified at the time an iTOUGH2 job is submitted (see Section 6.2). The output file names are by default a combination of the TOUGH2 and iTOUGH2 input file names and a predefined three-character extension. Some of the output file names can be changed upon job submission.

The Unix shell script *itough2* (see Section 6.2) writes a file *itough2.fil* with the names of the working directory, temporary directory, TOUGH2 input file, iTOUGH2 input file, as well as the arguments submitted to the shell script. This information allows iTOUGH2 to access the appropriate input files.

The user must provide at least two input files to run iTOUGH2. The first one is a TOUGH2 input file in standard TOUGH2 format as described in *Pruess* [1987, 1991a, 1991b], *Finsterle et al.* [1994], *Falta et al.* [1995], *Moridis and Pruess* [1995], *Battistelli et al.* [1997], *Finsterle* [1998a], and *Finsterle* [1999b; Appendix A], as well as other publications pertaining to particular TOUGH2 modules and code enhancements. This input file defines the conceptual model, i.e., the forward problem, which must run successfully not only for the initial parameter set, but also for a wider range of parameter combinations that may potentially arise during the iTOUGH2 run.

The second input file is the iTOUGH2 input file, in which the user specifies the parameters to be estimated, the observations used for calibration, and various program options. The basic concepts of the iTOUGH2 input language and a detailed description of each iTOUGH2 command are given in *Finsterle* [1998b, 1999bc].

Additional TOUGH2 input files may be given with information about the mesh, sinks and sources, and initial conditions. Furthermore, initial parameter guesses and calibration data can be provided either in the iTOUGH2 input file or on separate data files, with their names specified in the iTOUGH2 input file. All these additional input files are optional and depend on the TOUGH2 and iTOUGH2 options invoked.

iTOUGH2 creates a number of output files. In addition to the standard TOUGH2 output files, iTOUGH2 generates a main output file with information about the minimization process, the sensitivity coefficients, the residuals, the estimated parameters and their uncertainties; an example is described in Section 3. Furthermore, separate output files are generated upon request with the best estimate parameter set and corresponding TOUGH2 ROCKS block, the covariance matrix of the calculated system response, the observed data and the modeling result for plotting purposes, and a plot file with the relative permeability and capillary pressure curves. The message file, which contains Unix standard error and standard output information along with version control statements, should be consulted whenever execution problems persist.

Table 4.3.1. List of iTOUGH2 Input and Output Files

Filenames Generic	Example	File Content
		Input Files
-	*itough2.fil*	Names of working directory and input files
dir_file	*test.inp**	Standard# TOUGH2 input file (forward problem)
inv_file	*testi.inp**	iTOUGH2 input file (inverse problem)
dat_file	*pressure.dat*	Measured data
par_file	*testi.par*	Initial parameter guesses
$dir_file.mes	*coarse.mes*	Elements and connections (TOUGH2 file *MESH*)
$dir_file.ini	*equi.ini*	Initial condition information (TOUGH2 file *INCON*)
$dir_file.gen	*sinks.gen*	Sinks and sources (TOUGH2 file *GENER*)
		Output Files
-	*itough2.ver*&	Version control statements
-	*status*&	Current status (updated after each forward run)
-	*fort.99*&	File used for debugging
*$dir_file.**out***	*test.**out***@	Standard# TOUGH2 output file
*$dir_file.**sav***	*test.**sav***	Primary variables for restarting (TOUGH2 file *SAVE*)
*$dir_file.**mes***	*test.**mes***	Mesh information (TOUGH2 file *MESH*)
*$dir_file.**min***	*test.**min***	Mesh information after MINC preprocessing
*$dir_file.**ini***	*test.**ini***	Initial conditions written from block INCON in *dir_file*
*$dir_file.**lin***	*test.**lin***	Messages on linear equation solution
*$dir_file.**tab***	*test.**tab***	Data from semi-analytical heat exchange calculation
*$inv_file.**out***	*testi.**out***@	Main iTOUGH2 output file
*$inv_file.**tec***	*testi.**tec***@%	Plot file showing match
*$inv_file.**err***	*testi.**err***&	Summary error mesages
*$inv_file.**msg***	*testi.**msg***@+	iTOUGH2 message file
*$inv_file.**par***	*testi.**par***@	Best estimate parameter set and ROCKS block
*$inv_file**_ch.tec***	*testi**_ch.tec***%	Relative permeability and capillary pressure curves
*$inv_file.**cov***	*testi.**cov***	Covariance matrix of calculated system response

* These files are mandatory; all other input files are optional and depend on various program options.
\# See pertaining user's guide, source code, and Appendix A of report "iTOUGH2 Command Reference."
@ These files are automatically returned to the working directory.
% The extension depends on the chosen plotformat (see command `>>> FORMAT`).
& Contents appended to file *inv_file.out* and/or *inv_file.msg* after completion of run.
\+ If using script file *tough2*, the message file name is *t2.msg*.

5. CODE INSTALLATION

5.1 Getting Started

This section describes the installation of iTOUGH2 on a Unix workstation, assuming that the source code, shell script files, and sample problem input files are distributed as a compressed archive file *it2_tar.Z*. The installation procedure may vary if iTOUGH2 is distributed differently, or if it is not installed in the directory structure shown in Figure 4.2.1. Additional instructions can be found on file *read.me* and in the header of the *Makefile* and the Unix shell script files *itough2*, *tough2*, *prista*, *kit*, and *it2help*. Installation and execution of iTOUGH2 on a PC is different; see instructions in file *read.me*.

Table 5.1.1. Code Installation Procedure

Step 1:	Create the iTOUGH2 home directory:
	`cd; mkdir itough2`
	
	If multiple iTOUGH2 versions must be accessible, they should be installed in separate directories `itough2v$VERS`, where `$VERS` is a version identifier to be used with the `-v` option at run time:
	`cd; mkdir itough2v$VERS`

Step 2:	Move the iTOUGH2 distribution file(s) to the iTOUGH2 home directory.
Step 3:	Go to the iTOUGH2 home directory and extract all files from the distribution. If iTOUGH2 is distributed as a compressed tar file *it2_tar.Z,* the following command sequence establishes the directory structure shown in Figure 4.2.1: `uncompress it2_tar.Z; tar xvf it2_tar`
Step 4:	Customize iTOUGH2, as necessary: • Set the maximum problem size in file *maxsize.inc* (see Section 5.2). • Set default format of plot file (see file *it2main.f*, `BLOCK DATA IT`, variable `IPLOTFMT`). • Add code to file *it2user.f* to provide user-specified parameters, observations, boundary conditions, and data definitions. • Provide machine-dependent system calls if ported to a new platform not supported by iTOUGH2 (see files *mdep$COMP.f*, where *$COMP* is the name of a computer platform).
Step 5:	Edit *Makefile* and compile iTOUGH2 (see Section 5.3).
Step 6:	Install and customize Unix shell script files (see Section 6).
Step 7:	Test installation by running sample problems [*Finsterle*, 1999c].

5.2 Dimensioning of Major Arrays

Problems solved by iTOUGH2 vary considerably in size, depending on the number of gridblocks and connections, the number of equations solved per gridblock, the number of parameters estimated, the number of observations available, etc. It is important to be able to adjust the dimensions of major arrays to make the code fit on a specific computer with limited memory. Because iTOUGH2 is written in FORTRAN77, no dynamic memory allocation is possible, i.e., arrays are redimensioned by changing their size in the source code, followed by recompilation.

The user must set the appropriate constants in file *maxsize.inc*. The constants of greatest impact on memory requirement are MAXEL, MAXCON, MAXN, and MAXM. The number of mass components, phases, and balance equations per gridblock is given by MAXK, MAXPH, and MAXEQ, respectively. Note that they must be set to the maximum number required by the chosen equation-of-state module, regardless of the values given in TOUGH2 block MULTI. For example, if EOS7 is used [*Pruess*, 1991b], MAXK must be set to 3 and MAXEQ to 4, even though the model may be run in isothermal mode with no air involved, i.e., with NK=2 and NEQ=2. If an array is improperly dimensioned, iTOUGH2 issues a corresponding error message.

5.3 Compiling and Linking

A *Makefile* is provided for convenient compilation of iTOUGH2 on Unix workstations. File *Makefile* must be edited to indicate the desired equation-of-state (EOS) module and to provide the name of the FORTRAN compiler as well as various compiler options, which are specific to the computer platform. Table 5.3.1 shows some of the *Makefile* variables that must be set by the user. Compiler options are provided for most Unix platforms and the PC compilers by *Lahey*® and *Compaq*® *Visual Fortran* (formerly *DIGITAL Visual Fortran*). It is expected that iTOUGH2 can be compiled using other compilers with minor modifications (see file *read.me* for additional information). If the *Makefile* is used, the appropriate options can be selected by deleting the pound sign (#) in the first column, and by commenting out the portions of the *Makefile* that do not apply.

Depending on the compiler used, a linking error may occur if a subroutine is specified more than once, as is the case for *eos9.f* and *eos10.f*. In these instances, the user must ensure that the subroutine encountered first in the list of source files (see variables OBJxxx in Table 5.3.1) is linked to iTOUGH2, i.e., the subroutine in the second file must be renamed. For example, there are two versions of subroutine MULTI, one in file *eos9.f*, another in file *t2f.f*. If setting EOS=9, the subroutine has to be renamed (e.g., to MULTIx) in file *t2f.f* .

Table 5.3.2 shows the different targets defined in the *Makefile*. For example, in order to make a standard iTOUGH2 executable for the equation-of-state module defined through variable EOS, one must simply type `make`. For example, iTOUGH2-PVM [*Finsterle*, 1998b] is created by typing `make pvm`.

Table 5.3.1. Variable Definitions in Makefile

Variable	Description	Examples
EOS	Equation-of-state module identifier	`5` to compile and link *eos5.f* (see files *eos$EOS.f*)
COM	Name of computer platform	`ibm`, `sun`, `dec`, `sgi`, `hp`, `cray`, etc. (see files *mdep$COM.f*)
FOR	Compiler name	`f77, f90`
COO	Compiler options	`-c -O3` (see manual pages of compiler $FOR)
LIN	Linking options	`+U77` (see manual pages of compiler $FOR)
EXS	Extension of source files	`f`, `FOR`
EXO	Extension of object files	`o`, `OBJ`
EXE	Extension of executable	`$hostname`, `EXE`
LPVM	Location of PVM library	see *Finsterle* [1998b]
OBJSTD	List of files for standard iTOUGH2	`it2main`, `it2input`, `it2xxxx`, `it2user`, `mdep$COM`, `eos$EOS`, `t2cg1`, `t2f`, `meshm`, `ma28`, `ifsdummy`#, `gslibdum`, `pvmdummy`#

In order to invoke IFS [*Doughty*, 1995] and PVM [*Finsterle*, 1998b] capabilities, *ifsdummy* and/or *pvmdummy*, which contain subroutine stubs, must be replaced by *ifs* or *pvm*, respectively (see variables `OBJIFS` and `OBJPVM` in *Makefile*).

Table 5.3.2. Targets in *Makefile*

Target	Executable/Action	Comment
`make`	Creates standard iTOUGH2 *tough2_$EOS.$EXE*	Rename# subroutine MULTI in file *t2f.f* if EOS=9. Rename# subroutines INPUT, MULTI, RELP, and PCAP in file *t2f.f* if EOS=10.
`make pvm`	Uses OBJPVM and links iTOUGH2 to PVM library	Parallel execution of forward runs on workstation cluster [*Finsterle*, 1998b].
`make ifs`	Uses OBJIFS and links IFS subroutines to iTOUGH2	Includes subroutines for generating heterogeneous fields based on Iterated Function Systems [*Doughty*, 1995].
`make pcf77` `make pcf90`	Creates iTOUGH2 on PC	Supports Lahey® compiler f77l3. Supports Lahey® compiler lf90.
`make tar` `make zip`	Creates archive file *itough2.tar* Creates archive file *itough2.zip*	Creates tar or zip file with iTOUGH2 source code and sample problems.

Renaming is only necessary if compiler/linker does not accept multiply defined subroutines.

6. UTILITIES

6.1 Installation of Unix Shell Script Files

The iTOUGH2 distribution includes five Unix shell script files: *itough2*, *tough2*, *prista*, *kit*, and *it2help*. While iTOUGH2 can be run by typing the name of the executable, use of the Unix script files discussed in this section adds convenience and increases safety. A Unix script file is a command file that contains a shell program, and—if properly installed—can be invoked like any other Unix command, i.e., simply by typing the file name (e.g., the script file *kit* is invoked by typing `kit`). The five script files discussed here make use of the Bourne shell (`/bin/sh`); they can be executed from any shell.

The five Unix shell script files are assumed to be installed in a directory *$HOME/bin*, where *$HOME* is the user's login directory. The directory must be part of the search path where the shell looks for commands. It is suggested to add the following line to file *.cshrc*:

```
set path=($path ~/bin)
```

All script files must be executable; if not, type:

```
cd ~/bin; chmod a+x itough2 tough2 prista kit it2help
```

If the directory structure is different from that shown in Figure 4.2.1, the user must redefine some of the shell variables according to Table 6.1.1. The default values assume that iTOUGH2 is installed as described in Section 4.2. Script file *kit* requires identifying a suitable option for Unix command `ps`, which varies with the Unix flavor. If typing "`ps $ps_opt`" during the execution of an iTOUGH2 run, the command output must contain the process ID and the string "`itough2_`". Depending on the selected option `ps_opt`, the process ID appears in either the first or the second column (`ipid` = 1 or 2). Command line `` awk `{print $ipid}` `` near the end of file *kit* must be adjusted accordingly (for more details see header of file *kit*).

Table 6.1.1. Customizing Shell Variables in iTOUGH2 Script Files

Script File	Shell Variable	Default	Description
itough2	`prog_dir`	`$HOME/itough2`	iTOUGH2 home directory
	`tmp_dir`	`$HOME`	Main temporary directory
tough2	`script_dir`	`$HOME/bin`	Directory of *itough2* script file
prista	`tmp_dir`	`$HOME`	Main temporary directory
kit	`ps_opt`	`ux`	Option for UNIX command `ps`
it2help	`help_dir`	`$HOME/itough2`	Directory of file *it2help.txt*

6.2 Submitting an iTOUGH2 Job (Command `itough2`)

Shell script file *itough2* should be used to submit an iTOUGH2 job on a Unix workstation. The command usage, reproduced in Figure 6.2.1, can be displayed by typing `itough2` without any arguments.

```
===============================================================================
iTOUGH2 --- iTOUGH2 --- iTOUGH2 --- iTOUGH2 --- iTOUGH2 --- iTOUGH2 --- iTOUGH2
===============================================================================

Script file   : /m/presto/u/finster/bin/itough2

Syntax
------

itough2 [Options] InverseFile ForwardFile EOS &

InverseFile : iTOUGH2 input file
ForwardFile : TOUGH2 input file
EOS         : EOS module identifier

Options
-------
-no_delete  : temporary directory /m/presto/u/finster/it2_29321 is not deleted
-m file     : copies $file to temporary directory as input file MESH
-i file     : copies $file to temporary directory as input file INCON
-g file     : copies $file to temporary directory as input file GENER
-fi file    : copies $file to temporary directory
-ito file   : names iTOUGH2 output $file instead of $InverseFile.out
-to file    : names TOUGH2 output $file instead of $ForwardFile.out
-save file  : names output file SAVE $file instead of $ForwardFile.sav
-mesh       : returns output file MESH to working directory as $ForwardFile.mes
-lin        : returns output file LINEQ to working directory as $ForwardFile.lin
-plo        : returns output file PLOPO to working directory as $InverseFile.plo
-cov        : returns covariance file to working directory as $InverseFile.cov
-fo file    : returns $file from temporary directory to working directory
-v vers     : uses version in directory ~/itough2v$vers
-pvm        : runs iTOUGH2 in parallel under PVM

Examples
--------
itough2 sample1i sample1 3 &
itough2 -mesh dummi meshm.inp 1 &
itough2 -i equil.inc -m coarse.mes inverse.inp t2voc.inp 10 &
itough2 -pvm -no_delete pvmtesti test 9ecm &
```

Figure 6.2.1. Usage and options of command `itough2`.

The general command syntax is:

```
itough2 [options] InverseFile ForwardFile EOS &
```

Command `itough2` is followed by at least three arguments in the following order: (1) the name of the iTOUGH2 input file, (2) the name of the TOUGH2 input file, and (3) an indicator of the EOS module being used (usually a number). Additional options discussed below may precede the three mandatory arguments. iTOUGH2 should always be run in the background (i.e., add "`&`" at the end of the command line) to allow usage of commands `prista` and `kit`. Example:

```
command                 TOUGH2 input file   background
  ↓                            ↓                ↓
itough2 samplei.inp sample.inp 3 &
            ↑                        ↑
        iTOUGH2 input file    Number of EOS module
```

Submitting this command line has the following effect. A temporary directory is created named *$(tmp_dir)/it2_$$*, where *$(tmp_dir)* is the directory specified by shell variable `tmp_dir` (see Table 6.1.1), and *$$* is a unique process ID number. The specified input files are copied from the working directory to the temporary directory. iTOUGH2 is started where the name of the executable depends on the EOS module requested. In this example, EOS3 is used. After completion of the run, the main output files are copied from the temporary to the working directory, and the temporary directory is deleted unless flag `-no_delete` is specified or the program was terminated with an error signal.

Some iTOUGH2 runs may require additional input files containing mesh information, initial conditions, sinks and sources, or other special data. These files must be explicitly specified using flags `-m`, `-i`, `-g`, and `-fi`, respectively, followed by the appropriate file name. (All data and input files explicitely specified in the iTOUGH2 input file do not need to be given on the command line.) For example, if the TOUGH2 blocks ELEME and CONNE are provided on a separate file named *coarse.mes*, the command reads:

```
itough2 -m coarse.mes -v 3.2 samplei.inp sample.inp 3 &
```

By default, only the main output files are returned to the working directory, before the temporary directory is deleted. If additional files should be preserved, such as the mesh files *MESH* and *MINC*, the linear equation file *LINEQ*, the PLOPO plot file, or the covariance file of the calculated system response, the flags `-mesh`, `-lin`, `-plo`, and `-cov` must be set, respectively. Finally, the default names of the iTOUGH2 output file, the TOUGH2 output file, and the TOUGH2 *SAVE* file can be changed using options `-ito`, `-to`, and `-save`, respectively, followed by the desired file name. Example:

```
itough2 -i test.sav -save test.sav2 -mesh testi test 10 &
```

In this example, initial conditions are not provided through the TOUGH2 input file *test*, but are read from file *test.sav*, which apparently is the *SAVE* file from a previous run. In order not to overwrite this file, the default file name for the current *SAVE* file is changed to *test.sav2*. The *MESH* file generated by this run is returned to the working directory with name *test.mes*. This is an inversion using the T2VOC simulator [*Falta et al.*, 1995]; its EOS number is 10.

6.3 Submitting a TOUGH2 Job (Command `tough2`)

iTOUGH2 can also be used to run standard TOUGH2 simulations. There are several advantages of using iTOUGH2 for forward runs: (1) only one version of the simulation program has to be installed and maintained; (2) the exact same code is used for solving both the forward and inverse problem, reducing the risk of introducing errors when modifying the programs; (3) additional program features are available (see *Finsterle* [1999b; Appendix A]); and (4) a forward run can be observed and terminated using commands `prista` and `kit`, respectively.

The general command syntax is:

```
tough2 [options] ForwardFile EOS &
```

Command `tough2` is followed by two arguments in the following order: (1) the name of the TOUGH2 input file, and (2) an indicator of the EOS module being used (usually a whole number). Additional options may precede the mandatory arguments. They are the same as those for command `itough2` (see Section 6.2).

Command `tough2` calls shell script file `itough2` with a dummy iTOUGH2 input file *invdir*, which is expected to be present in directory *$(prog_dir)* (see Table 6.1.1). There is only minimal overhead associated with reading the dummy input file and writing additional output. The name of the message file created by command `tough2` is always *t2.msg*.

6.4 Status Checking of an iTOUGH2 Job (Command `prista`)

The progress of an iTOUGH2 can be observed using command `prista`, which accesses file *status* in the temporary directory. File *status* is updated after each forward simulation, and contains iteration statistics, information about the current parameter set, and the development of the objective function.

If more than one iTOUGH2 simulations are running at the same time, the user is first prompted to select which run shall be checked (see Figure 6.4.1). Then, the contents of file *status* are displayed. The current parameter set is printed along with the latest and total update. The sensitivity measure $\delta = |\partial S|$ (Equation 2.8.2.6) shows the change of the objective function if the parameter is perturbed by a small amount. Lastly, information about the objective function is given, with its current value, the update due to the perturbation of the parameter indicated by the arrow `<---`, and the total improvement with respect to the initial value printed in the last column.

```
your_prompt> prista

 ID      STIME   DIRECTORY   ARGUMENTS
  1 -->  09:29   it2_87631   -tough2 test 9
  2 -->  09:31   it2_87633   sam2p1i sam2 3
  3 -->  09:32   it2_87639   -fi atmos.dat sam5i sam5 3

 Enter ID: 2

 ==================================================================
      S T A T U S   O F   i T O U G H 2   S I M U L A T I O N
 ==================================================================

 Date                        : 23-Nov-98 09:33
 iTOUGH2 file                : sam2p1i
 TOUGH2 file                 : sam2
 Working directory           : /m/presto/u/finster/itough2/samples/sample2
 Temporary directory         : /m/presto/u/finster/it2_29625
 UNIX command line arguments : sam2p1i sam2 3
 Comment                     : No fit improvement. Jacobian (forward).

 Iterations completed        :              1    Iterations to go      7
 TOUGH2 runs completed  (+/-):         8/   0    Unsuccessful steps    0
 TOUGH2 runs to go (approx.) :             43    Log(Levenberg)       -3
 CPU time used     [sec]     :          41.63    for last step      5.48

     Parameter           Current  Last Update Total Update       Sens. Active
  1. ABS. K GEYS1+8 :  -.1939E+02   -.3917E+00   -.3917E+00      937.787
  2. KLINK GEYS1+8  :   .6543E+01   -.4569E+00   -.4569E+00      871.680
  3. POROSITY GEYS1+:   .1080E-01   -.4311E-02   -.4311E-02       24.210 <---

     TOUGH2 run No. :           8       8 -  5       5 -  1      Initial
     Obj. Function  :   .6016E+05    .2421E+02   -.2572E+06    .3174E+06

Do you want to read an output file?

 1 --> sam2.out
 2 --> sam2p1i.out
 * --> another file

Enter file No. plus v(i) or t(ail): 2
```

Figure 6.4.1. Screen dump from command `prista`.

The user has then the opportunity to look at some of the output files using either the `vi`-editor or Unix command `tail`. If the file number is given, command `tail` is invoked, displaying the last 10 lines of the chosen output file. The `tail` command continues to display lines as they are added to the output file until stopped by pressing the `Ctrl-C` key sequence. In order to user the `vi`-editor, the file number has to be followed by the character `v`. Note that the file should not be modified since it is continuously updated by iTOUGH2.

Command `prista` allows a user also to inspect a TOUGH2 simulation. If deemed necessary, the run can be gracefully terminated using command `kit` (see Section 6.5).

6.5 Terminating an iTOUGH2 Job (Command `kit`)

An iTOUGH2 run can be interrupted and in particular gracefully terminated using command `kit`, which sends a signal to the process running iTOUGH2. A signal handler is installed (see file *mdep$COMP.f*) to trigger the desired action. Command `kit` is usually used after checking the status of an iTOUGH2 run using command `prista` (see Section 6.4). The signal numbers and their effects are described in Table 6.5.1.

Table 6.5.1. Signals Sent by Command `kit`

Signal	Effect
0	Exit shell script (no action taken).
1	Terminates iTOUGH2 immediately, but gracefully.
2	Terminates iTOUGH2 after completion of the current forward run. The Jacobian matrix used for the error analysis may contain columns that are from the previous evaluation. The corresponding parameter sensitivities are printed in brackets `[]` in the iTOUGH2 output file.
3	Terminates iTOUGH2 after completion of the next time step. The system state is written to the TOUGH2 output file (even if a negative value is specified for variable *KDATA*; see *Finsterle* [1999b; Appendix A2]). This option is useful especially for long TOUGH2 runs. A run can be terminated at any time and later restarted using the returned *SAVE* file *$dir_file.sav*.
4	Terminates iTOUGH2 after completion of next iTOUGH2 iteration.
5	TOUGH2 output is generated after completion of the next time step; the run continues.
6	The output buffers are flushed[%] so the full content of the output file can be seen during the run using command `prista`.
7	An iTOUGH2 run is stopped without terminating it (sends signal `STOP`).[#]
8	An iTOUGH2 run previously stopped using Signal 7 is continued (sends signal `CONT`).[#]
9	Kills run immediately and abruptly (sends signal `KILL`).
10	A previously submitted Signal 2, 3, or 4 is canceled.

% Not necessary if output is not buffered.
# Does not work on Sun workstations using Solaris.

Figure 6.5.1 shows a screen dump from command `kit`. If multiple iTOUGH2 inversions are running at the same time, they are listed and numbered. The information displayed depends on the option of Unix command `ps` (see `ps_opt` in Table 6.1.1).

```
your_prompt> kit

 #     USER       PID %CPU %MEM   SZ  RSS    TTY STAT    STIME  TIME  COMMAND
 1 --> finster  13181 49.8  3.0  876 1668  pts/0 R    15:45:56 31:56  itough2_1.itelos
 2 --> finster  13997 76.8  3.0  872 1664  pts/0 R    14:25:55 110:47 itough2_3.itelos

Select iTOUGH2 run by number : 2

  0: exit
  1: terminate immediately
  2: terminate after completion of TOUGH2 run
  3: terminate after completion of TOUGH2 time step
  4: terminate after completion of iTOUGH2 iteration
  5: print output now
  6: flush output buffers
  7: sleep
  8: wake up
  9: kill run
 10: cancel previous signal

Choose signal: 6

Signal 6 sent to process 13997
```

Figure 6.5.1. Screen dump from command `kit`.

The command `kit` sometimes fails to recognize a Unix process as an iTOUGH2 run. In these cases, iTOUGH2 can be killed manually using command:

```
kill -SIGNAL PID
```

where `SIGNAL` is the signal number of Table 6.5.1, and `PID` is the process ID of the iTOUGH2 executable with the command name `$prog_dir/itough2_$EOS.$HOST`. Do not kill the shell script `itough2` or `tough2`. If either of these script commands is killed, the actual iTOUGH2 simulation in fact keep running but fails to return the output files from the temporary to the working directory after its completion. Moreover, the temporary directory is not removed, making it reappear whenever command `prista` is used. In order to remove left behind temporary directories, type:

```
rm -r ~/it2_*
```

6.6 Obtaining On-line Help (Command `it2help`)

The content of *Finsterle* [1999b; Section 4] can be accessed on-line at the web site `http://www-esd.lbl.gov/iTOUGH2`, by clicking on "Command Index". Alternatively, command `it2help` can be used, followed by the command name and, optionally, the command level. Type `it2help` to display the command usage (see Figure 6.6.1). There are three special commands. In order to display the complete iTOUGH2 command index (see *Finsterle* [1999b; Appendix B]), type:

```
it2help index
```

The following command displays iTOUGH2 update information:

```
it2help update
```

The following command displays the iTOUGH2 log book *$prog_dir/itough2.log*:

```
it2help logbook
```

The manual pages are stored on file *it2help.txt*. The content of this file is continually updated to include added features of later versions of iTOUGH2.

```
===================================================================
IT2HELP --- IT2HELP --- IT2HELP --- IT2HELP --- IT2HELP --- IT2HELP
===================================================================

Purpose:

Displays iTOUGH2 manual pages on-line.

Syntax:

it2help command [command_level]

  command       = iTOUGH2 command
  command_level = integer indicating command level (1, 2, 3, or 4)

Examples:

1. it2help iteration
   Prints manual pages of command ITERATION on all command levels.

2. it2help iteration 3
   Prints manual page of third-level command >>> ITERATION.

3. it2help index
   Prints the iTOUGH2 command index.

4. it2help update
   Prints version update information.
===================================================================
```

Figure 6.6.1. Screen dump from command `it2help` showing command usage.

ACKNOWLEDGMENT

iTOUGH2 (up to Version 1.1) was developed at the Laboratory of Hydraulics, Hydrology, and Glaciology (VAW) of the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, in collaboration with the Swiss National Cooperative for the Disposal of Radioactive Waste (Nagra), Wettingen, Switzerland. Subsequent versions were supported, in part, by a grant from the Swiss National Cooperative for the Disposal of Radioactive Waste (Nagra), Wettingen, Switzerland, and by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Geothermal Technologies, of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098.

Many thanks are due to Chris Doughty, Curt Oldenburg, and Grimur Björnsson, who carefully reviewed the manuscript and made many valuable suggestions for improvement. Daniel Hawkes' services as a technical editor are gratefully acknowledged.

REFERENCES

Andrews, D. F., P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey, *Robust Estimates of Location: Survey and Advances*, Princeton University Press, Princeton, NJ, 1972.

Battistelli, A., C. Calore, and K. Pruess, The simulator TOUGH2/EWASG for modeling geothermal reservoirs with brines and a non-condensible gas, *Geothermics*, *26*(4), 437–464, 1997.

Beck, J. V., and K. J. Arnold, *Parameter Estimation in Engineering and Science*, John Wiley, New York, New York, 1977.

Bickel, P. J., and K. A. Doksum, *Matchematical Statistics*, Holden-Day, Inc., Oakland, California, 1977.

Björck, A., *Numerical Methods for Least Squares Problems*, Society of Industrial and Applied Mathematics, Philadelphia, PA, 1996.

Carrera, J., Estimation of aquifer parameters under transient and steady-state conditions, Ph.D. dissertation, Dep. of Hydrol. and Water Resour., Univ. of Ariz., Tucson, 1984.

Carrera, J., State of the art of the inverse problem applied to the flow and solute transport equations, in: Custodio, E., A. Gurgui, and J. P. Lobo Ferreira (eds.), *Groundwater Flow and Quality Modeling*, NATO ASI Series C: Mathematical and Physical Sciences, Vol. 224, 549–583, Kluwer, Norwell, Mass., 1988.

Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions: 1. Maximum likelihood method incorporating prior information, *Water Resour. Res.*, *22*(2), 199–210, 1986a.

Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions: 2. Uniqueness, stability, and solution algorithms, *Water Resour. Res.*, *22*(2), 211–227, 1986b.

Carrera, J., and S. P. Neuman, Estimation of aquifer parameters under transient and steady state conditions: 3. Application to synthetic and field data, *Water Resour. Res.*, *22*(2), 228–242, 1986c.

Chavent, G., On the theory and practice of non-linear least-squares, *Adv. Water Resources*, *14*(2), 55–63, 1991.

Datta-Gupta, A., S. Yoon, I. Barman, and D. W. Vasco, Streamline-based production-data integration into high-resolution reservoir models, SPE 52981, *J. of Petr. Tech., 12*, 72–76, 1998

Donaldson, J. R., and R. B. Schnabel, Computational experiences with confidence regions and confidence intervals for nonlinear least squares, *Technometrics*, *29*(1), 67–82, 1987.

Doughty, C., *Estimation of Hydrologic Properties of Heterogeneous Geologic Media with an Inverse Method Based on Iterated Function Systems*, Ph.D. Thesis, Dept. of Material Sci. and Mineral Eng., University of California, Report LBL-38136, Lawrence Berkeley Laboratory, Berkeley, Calif., 1995.

Ewing, R. E., and T. Lin, A class of parameter estimation techniques for fluid flow in porous media, *Adv. Water Resources, 14*(2), 89–97, 1991.

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

Finsterle, S., *iTOUGH2 V3.2 Verification and Validation Report*, Report LBNL-42002, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1998a.

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

Finsterle, S., *iTOUGH2 User's Guide,* Report LBNL-40040, Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999a.

Finsterle, S., *iTOUGH2 Command Reference,* Report LBNL-40041 (Updated reprint), Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999b.

Finsterle, S., *iTOUGH2 Sample Problems*, Report LBNL-40042 (Updated reprint), Lawrence Berkeley National Laboratory, Berkeley, Calif., 1999c.

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

Finsterle, S., and K. Pruess, Solving the estimation-identification problem in two-phase flow modeling, *Water Resour. Res.*, *31* (4), 913–924, 1995.

Finsterle, S., and P. Persoff, Determining permeability of tight rock samples using inverse modeling, *Water Resour. Res.*, *33*(8), 1803–1811, 1997.

Finsterle, S., and J. Najita, Robust estimation of hydrogeologic model parameters, *Water Resour. Res.*, *34*(11), 2939–2947, 1998.

Finsterle, S., and B. Faybishenko, Inverse modeling of a radial multistep outflow experiment for determining unsaturated hydraulic properties, *Adv. Water Resour.*, *22*(5), 431–444, 1999.

Gauss, C. F., *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae, pars prior*, 1821, translated by G. W. Stewart, *Theory of the Combination of Observations Least Subject to Errors*, Society of Industrial and Applied Mathematics, Philadelphia, PA, 1995.

Gill, P. E., W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, Inc., London, 1981.

Haining, R., *Spatial Data Analysis in the Social and Environmental Sciences*, Cambridge University Press, Cambridge, 1990.

Huber, P. J., *Robust Statistical Procedures*, Society of Industrial and Applied Mathematics, Second Edition, Philadelphia, 1996.

Huber, P. J., *Robust Statistics*, Wiley, New York, New York, 1981.

IGP, *Zuverlässigkeit in der Vermessung*, Bericht Nr. 169, Institut für Geodäsie und Photogrammetrie, Eidgenössische Technische Hochschule, Zürich, Switzerland, 1990.

Jacoby, S. L. S., J. S. Kowalik, and J. T. Pizzo, *Iterative Methods for Nonlinear Optimization Problems*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1972.

Kashyap, R. L., Optimal choice of AR and MA parts in autoregressive moving average models, *IEEE Trans Pattern Anal. Mach. Intel.*, *PAMI-4*(2), 99–104, 1982.

Kitterød, N.-O., and L. Gottschalk, Simulation of normal distributed smooth fields by Karhunen-Loéve expansion in combination with kriging, *Stochastic Hydrology and Hydraulics*, *11*, 459–482, 1997.

Klinkenberg, L. J., The permeability of porous media to liquids and gases, *API Drill. and Prod. Prac.*, 200–213, 1941.

Kool, J. B., J. C. Parker, and M. T. van Genuchten, Parameter estimation for unsaturated flow and transport models—a review, *Journal of Hydrol., 91,* 255–293, 1987.

Kuczera, G., and M. Mroczkowski, Assessment of hydrologic parameter uncertainty and the worth of multiresponse data, *Water Resour. Res.*, *34*(6), 1481–1489, 1998.

Larsen, R. J., and M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1986.

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.

McLaughlin, D., and L. R. Townley, A reassessment of the groundwater inverse problem, *Water Resour. Res., 32*(5), 1131–1161, 1996.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *J. Chem. Phys.*, *21*, 1087–1092, 1953.

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

Neumaier, A., Solving ill-conditioned and singular linear systems: A tutorial on regularization, *SIAM Rev.*, *40*(3), 636-666, 1998.

Neuman, S. P., Calibration of distributed parameter groundwater flow models viewed as a multiple objective decision process under uncertainty, *Water Resour. Res., 9*(4), 1006–1021, 1973.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN, the Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, 1992.

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

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

Pruess, K., *EOS7, An Equation-of-State Module for the TOUGH2 Simulator for Two-Phase Flow of Saline Water and Air*, Report LBL-31114, Lawrence Berkeley Laboratory, Berkeley, Calif., 1991b.

Russo, D., Determining soil hydraulic properties by parameter estimation: On the selection of a model for the hydraulic properties, *Water Resour. Res.*, *24*(3), 453-459, 1988.

Russo, D., E. Bresler, U. Shani, and J. C. Parker, Analyses of infiltration events in relation to determining soil hydraulic properties by inverse problem methodology, *Water Resour. Res.*, *27*(6), 1361-1373, 1991.

Scales, L. E., *Introduction to Non-Linear Optimization*, Springer, New York, 1985.

Steinberg, D. M., and W. G. Hunter, Experimental design: Review and comment, *Technometrics*, *28*, 71-97, 1984.

Stengel, R. F., *Optimal Control and Estimation*, Dover Publications, Inc., Mineola, New York, 1994.

Sun, N.-Z., *Inverse Problems in Groundwater Modeling*, Kluwer Acad., Norwell, Mass., 1994.

Sun, N.-Z., and W. W.-G. Yeh, Coupled inverse problems in groundwater modeling, 1, Sensitivity analysis and parameter identification, *Water Resour. Res.*, *26*, 2507–2525, 1990.

Van Huffel, S., and J. Vandewalle, *The Total Least Squares Problem, Computational Aspects and Analysis*, Society of Industrial and Applied Mathematics, Philadelphia, PA, 1991.

Vasco, D. W., A. Datta-Gupta, and J. C. S. Long, Resolution and uncertainty in hydrologic characterization, *Water Resour. Res.*, *33*(3), 379–397, 1997.

Vasco, D. W., and A. Datta-Gupta, Asymptotic solutions for solute transport: A formalism for tracer tomography, *Water Resour. Res.*, *35*(1), 1–16, 1999.

Weisberg, S., *Applied Linear Regression*, John Wiley & Sons, New York, 1980.

Yeh, W. W-G., Review of parameter identification procedures in groundwater hydrology: The inverse problem, *Water Resour. Res.*, *22*(2), 95–108, 1986.

Zimmerman, D. A., G. de Marsily, C. A. Gotway, M. G. Marietta, C. L. Axness, R. L. Beauheim, R. L. Bras, J. Carrera, G. Dagan, P. B. Davies, D. P. Gallegos, A. Galli, G. Gómez-Hernández, P. Grindrod, A. L. Gutjahr, P. K. Kitanidis, A. M. Lavenue, D. McLaughlin, S. P. Neuman, B. S. RamaRao, C. Ravenne, and Y. Rubin, A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow, *Water Resour. Res., 34*(6), 1373–1413, 1998.

INDEX